DESIGN OPTIMIZATION OF A CONTINUOUS DETECTOR FOR PEM IMAGING WITH HIGH RESOLUTION AND DOI CAPABILITY

by

SERKAN BERK

A.S., Electronics, Boğaziçi University, 1999
B.S., Civil Engineering, Boğaziçi University, 2003

Submitted to the Institute of Biomedical Engineering
in partial fulfillment of the requirements
for the degree of
Master of Science
in
Biomedical Engineering

Boğaziçi University
October 2006

ACKNOWLEDGMENTS

I wish to thank to my thesis supervisor, Assoc. Prof. Dr. Albert Güveniş for his constant help and motivation to make me complete this study. I also would like to thank Gözde Göncü for drawing the 3D illustrations, Didar Talat and Hakan Berk for their help in writing the additional modules for the simulation programs. Finally, I thank Emile Hoskinson for providing me the GRIT simulation software.

ABSTRACT

DESIGN OPTIMIZATION OF A CONTINUOUS DETECTOR FOR PEM IMAGING WITH HIGH RESOLUTION AND DOI CAPABILITY

The objective of this thesis is to improve the resolution of a continuous detector by using an algorithm other than Anger algorithm. Our aim is to obtain a reasonable resolution necessary for Positron Emission Mammography (PEM) imaging. Many research groups have been involved in developing different types of high resolution Positron emission tomography (PET) systems. Among those, most designs have consisted of detectors assembled using tiny discrete crystal elements identified by position sensitive or multi-channel photomultiplier tubes. However using narrow, pixellated crystals for higher resolution causes several problems including inter crystal scatter, light collection difficulty, practical difficulties of crystal size, and high cost.

In this work, the feasibility of using a continuous crystal detector for PEM imaging with high resolution has been investigated through simulations. We aim to reduce the system cost and to improve system performance. Simulations confirmed that Anger algorithm is not a feasible algorithm to use in a small size detector since it needs further processing to correct the linearity distortion problem and it does not provide any information about the depth of interaction. Simulations also showed that, when calculating the location of interaction with Anger algorithm, there can be a shift about 1 mm depending on the depth of interaction in a 10 mm thick NaI crystal.

The nearest neighbor algorithm by using a lookup table gave better results than Anger algorithm and also provided information about depth of interaction. We adapted a second threshold, called Proximity value, to the algorithm to eliminate possible Compton scatterings. This improves the resolution while trading off the number of interactions used. An optimum proximity value has been suggested depending on the simulation results.

Keywords: Positron emission tomography (PET), continuous scintillator, depth of interaction (DOI), Anger logic.

ÖZET
PEM GÖRÜNTÜLEME İÇİN YÜKSEK ÇÖZÜNÜLÜRLÜKLÜ VE ETKİLEŞİM DERİNLİĞİ YETENEĞİNE SAHİP SÜREKLİ DETEKTÖR DİZAYNI OPTİMİZASYONU

Bu tezin amacı sürekli bir detektörün çözünülürlüğünü Anger algoritmasından farklı bir algoritma kullanarak iyileştirmektir. Hedefimiz Pozitron Emisyon Mamografi (PEM) görüntüleme yöntemi için kayda değer bir çözünülürlük elde etmektir. Birçok araştırma grubu farklı türlerde yüksek çözünülürlüklü Pozitron Emisyon Tomografi (PET) sistemleri geliştirme uğraşında bulunmaktadır. Bunların içerisinde, çoğunluğunu küçük birbirinden bağımsız kristal elemanlar ve pozisyon hassasiyetli yada çok kanallı foton çoklayıcı tüpler içeren dizaynlar oluşturmaktadır. Bununla birlikte, yüksek çözünülürlük elde etmek için kullanılan dar, pikselleştirilmiş kristallerin, kristaller arası saçılma, ışık toplamada zorluk, kristal boyutlarının pratik kullanıma çok elverişli olmayışı ve yüksek fiyatlı olmaları çeşitli sorunlar yaratmaktadır.

Bu çalışmada, simülasyonlar kullanarak yüksek çözünülürlüklü PEM görüntüleme için sürekli kristal detektör kullanmanın uygunluğu araştırılmıştır. Bir yandan sistemin maliyetini azaltıp diğer yandan sistemin performansı iyileştirmeyi istemekteyiz. Simülasyonlar, Anger algoritmasının doğrusal bozulma problemini düzeltmek için daha fazla işlem gerektiği için küçük boyutlu detektörler için elverişli olmadığını ve etkileşim derinliği hakkında bir bilgi sağlayamadığını göstermiştir. Simülasyonlar ayrıca göstermiştir ki, 10 mm kalınlığındaki bir NaI kristali için Anger algoritması ile bir etkileşimin konumu hesaplandığında, etkileşimin meydana geldiği derinliğe göre konumlandırmada 1 mm civarı bir kayma meydana gelebilmektedir.

Kayıt tablosu ile birlikte kullanılan en yakın komşu algoritması, Anger algoritmasından daha iyi sonuçlar vermiş olup, ayrıca etkileşim derinliği hakkında da bilgi sağlamıştır. Olası Compton sıçramalarını elemek için algoritmaya, Yakınlık değeri adını verdiğimiz, ikinci bir eşik adapte ettik. Bu, kullanılabilecek etkileşim sayısından feragat

ederek çözünülürlüğün iyileşmesini sağladı. En uygun yakınlık değeri simülasyon sonuçlarına dayanarak önerilmiştir.

Anahtar Sözcükler: Pozitron emisyon tomografi (PET), sürekli sintilatör, etkileşim derinliği, Anger algoritması.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS

$\beta+$	Positron emission
E	Energy
E_{max}	Maximum energy
m	Mass
m_e	Mass of electron
m_p	Mass of proton
m_0	Mass of electron at rest
c	Speed of light
d	Distance
Δt	Time difference
σ	Interaction cross-section
Z	Atomic number
h	Planks Constant
hv	Photon energy
hv'	Final photon energy
θ	Angle
r_e	Classical electron radius
μ	Linear attenuation coefficient
$d\Omega$	Solid angle
V	Signal amplitude
ω	Weight coefficient
δp	Parallax shift
Δz	Difference in depth
m_i	PMT output
μ	Mean
N	Average number of light photons

LIST OF ABBREVIATIONS

PEM	Positron emission mammography
PET	Positron emission tomography
DOI	Depth of interaction
LCC	Linear correlation coefficient
CSE	Chi square error
LRF	Light response function
RBF	Radial basis function
TRIUMF	Tri-University Meson Facility
PMT	Photomultiplier tube
TOFPET	Time-of-flight positron emission tomography
APD	Avalanche photodiode
PSPMT	Position sensitive photomultiplier tube
LSF	Line spread function
FWHM	Full width at half maximum

1. INTRODUCTION

Breast cancer is the most frequent malignant disease in women and the second leading cause of cancer death in western countries [1]. Early diagnosis is very important for a successful treatment since the disease is curable when detected in the early stages. Positron emission mammography (PEM) has been used successfully in many oncological applications. It is an excellent clinical method to detect breast cancer since it enables visualization of increased glucose metabolism of malignant tissue.

Positron emission tomography (PET) system with a good image quality will increase the reliability in clinical diagnosis. For this purpose, an ideal PET system should have high spatial resolution to increase efficiency of detection [2, 3].

1.1 Problems in PET Imaging

One approach in PET imaging is to use a continuous detector. This is a cost effective way to build a PET system, on the other hand it brings some problems. In gamma ray imaging for high energies, a thick scintillation crystal is needed to efficiently stop the incident gamma particle. As the crystal gets thicker, it becomes more difficult to estimate the position of the source. If neither the depth of interaction (DOI) nor angle of incidence is measured, it is impossible to distinguish between parallel gamma rays that have the same two dimensional interaction positions, even though the spatial components of the impact position parallel to the entrance window is obtained with high precision. This effect is called the parallax error, which degrades significantly the image quality of current PET systems [4].

The majority of existing detection algorithms is based on centroid arithmetic. Their application appears, however to be problematic in the case of thick crystals due to significant parallax observed at large radiation incidence angles [5]. With a 10 mm thick crystal and a 30 slant angle, the obliquity factor can be responsible for a 5.5 mm variation in event position [6]. Another problem caused by centroid arithmetic is edge effects.

1.2 State of the Art

The image quality could be improved potentially if sufficiently good estimates of DOIs are provided. There are various methods to solve the DOI problem. One possible strategy to partially overcome the effect of parallax errors is to restrict the transaxial field of view of PET system. Unfortunately this causes other unwanted effects such as loss of efficiency and an increasing relevance of photon non-co-linearity, besides the fact that it makes compact PET scanners impossible.

For PET detectors based on pixellated scintillators, a number of techniques for designing DOI capable detectors for gamma rays in the fields of nuclear medical imaging have been proposed. One of them consists of measuring the ratio of scintillation light detected at opposed crystal surfaces using two photodetectors [7]. To avoid additional expensive photodetectors and their associated electronics, other techniques can be used; a staggered double layer array of crystal needles [8], the phoswich technique [9], or the effect of light absorption within the detector crystal [10, 11]. Also combinations of different techniques are possible. However, all these approaches imply costly detector modifications to a greater or lesser extent. They need additional photosensors, smaller crystals, or crystals of a different type.

There are also studies about possibility of measuring DOI without using any additional detectors or modifications that do not substantially increase the cost. Recent study published by Lerche et al. [12] uses the correlation between the width of the undisturbed light distribution in continuous crystals and the DOI. Their analytic model derived for the light distribution is verified by Monte Carlo simulation. The model fits satisfactorily for DOI values $\geq$ 3 mm and scintillation crystals with large transversal extension. In two simulations performed, the method of DOI determination works satisfactorily with an intrinsic detector resolution of 2.5 $\pm$ 0.5 mm and 2.3 $\pm$ 0.5 mm, respectively. Together with the error from Compton scattering, DOI resolution is obtained to be $\leq$ 5 mm.

Another study performed by Joung et al. [13] investigates the use of linear correlation coefficient (LCC) and Chi square error (CSE) method for bias free positioning estimators. The LCC method uses correlation information between the true light response function (LRF) and measured data in mapping an event characterization vector to the associated position. The CSE method estimates the position where the CSE between two functions becomes minimized. In order to determine true statistics as a function of position, LRF was estimated based on sample measurements by using a cubic spline interpolation technique which provides smooth first order and continuous second order derivative of the LRF. Both methods have superior linearity properties compared to the weighted centroid. Each method can be considered as a bias free positioning estimator within the effective field of view of the detector. The spatial resolution performance of the CSE method is 7% and 16% better than the weighted centroid method for a 16 and 25 mm thick crystal, respectively. The spatial resolution performance of the LCC method is comparable to that of the weighted centroid method.

Tomitani et al. [14] proposed an iterative maximum likelihood algorithm for position estimation and depth encoding in thick scintillation crystals, in order to compensate for the parallax effect. Delorme et al. [15] and Clément et al. [16] have implemented artificial neural networks in a depth encoding scintillation detection.

A recent study presented by Bronstein et al. [17] compares Anger algorithm with feed-forward neural networks, radial basis function (RBF) networks, and neuro-fuzzy systems. They trained sets of neural networks independently on simulated photomultiplier tube (PMT) responses resulting from scintillation events in appropriate regions and angles. They also used the simulation platform developed by Tri-University Meson Facility (TRIUMF). Their method is capable of estimating directly the photon line of flight, given PMT responses from a pair of detectors. Incidence angle is estimated using additional information on the coincident event in the opposite detector. The proposed algorithm allows compensation for the parallax effect, it reduces the resolution degradation due to multiple Compton scattering and increased effective detection area [17].

1.3 Our Approach

In this research, we used a nearest neighbor position estimation based on a lookup table. There are two methods to implement lookup table based position estimation: (1) experimental method, and (2) computer simulation method. The experimental method is a direct sampling from the detector system. The sampling process is prone to measurement error in experimental situations and it needs laborious works. For our case, sampling directly from the system to construct the lookup table is very hard to achieve and would be costly with today's technology. Second method, computer simulation, uses simulations that should contain all stages from scintillation event. It also needs the experimental verification. In this paper, we have proposed a computer simulation method to collect the data to form the lookup table.

Our work based on the assumption that the results from Monte Carlo Simulations are very close to real experimental results since the data is acquired from Monte Carlo Simulations. The basis for our assumption is that these simulation programs are validated through many studies experimentally. In our work, we formed a lookup table composed of average PMT outputs for definite points inside a crystal. These average values determined via the Monte Carlo simulations.

The algorithm we applied works as follows: If an interaction above the energy threshold is recorded, PMT outputs for that event are compared with the average PMT output data stored in the lookup table. The point with the most similar PMT output among the table is chosen for the point of interaction. The main difference in our method with previous studies is that it can eliminate Compton scattering by using a second threshold value, named as Proximity value. By utilizing a matching system through a lookup table, we also get information about the DOI.

The simulations showed that DOI effects the location estimation and therefore the resolution when using Anger algorithm. Even if the incoming ray is perpendicular to the surface, the problem persisted. Using lookup table with Nearest Neighbor algorithm

provided better resolution than Anger algorithm. The optimum spatial resolution achieved is around 2.4 mm with our algorithm while with Anger algorithm it is about 3 mm.

1.4 Thesis Outline

This thesis work is composed of eight chapters. The first chapter presents an introduction to problems in PET imaging and various approaches introduced. The second chapter gives information about the history of PET and the first device built by Hal Anger. The third chapter is about the basic physics concepts used in PET imaging. In the fourth chapter, materials and equipments used in PET imaging are introduced. The fifth chapter explains the algorithms used in this work. Sixth chapter gives explanation about the simulation softwares used in the experiments. The results of the performed experiments and statistical computations are in the seventh chapter. The last chapter presents the conclusions and recommended future work.

2. POSITRON EMISSION TOMOGRAPHY

Medical imaging is a powerful clinical diagnostic tool. With the discovery of X-rays in 1985 by Röntgen, the use of different techniques has been developed and today the field of medical imaging is growing very rapidly. Radionuclide imaging is one of the tools used today in this field.

Radionuclide imaging yields detailed information about the physiological or metabolismic functioning of the tissues by making use of radioactive isotopes. PET is a nuclear imaging technique that uses the unique decay characteristics of radionuclides which decay by positron emission. These radionuclides are produced in a cyclotron and then used to label compounds of biological interest. The labeled compound is introduced into the body generally by intravenous injection. It is distributed in tissues in a manner determined by its biochemical properties. When the radioactive atom on a particular molecule decays, a positron is ejected from the nucleus, ultimately leading to the emission of high energy photons that have a good probability of escaping from the body. A PET scanner consists of a set of detectors that surround the object to be imaged and are designed to convert these high energy photons into an electrical signal. Later on these signals are fed to subsequent electronic circuits.

In PET scan, detected events are corrected for a number of factors and then reconstructed into a tomographic image using mathematical algorithms. The output of the reconstruction process is a two or three dimensional image, where the signal intensity in any particular image pixel or voxel is proportional to the amount of the radionuclide. Thus, PET images allow the spatial distribution of radio-labeled tracers to be mapped quantitatively. By taking a time sequence of images, the tissue concentration of the radio-labeled molecules as a function of time is measured, and with appropriate mathematical modeling, the rate of specific biological processes can be determined.

The use of radionuclides for diagnostic purposes in medicine was started by Hevesy in 1935 [18]. In 1951, positron emitters were suggested as potential substances to be used

in medical imaging. The advance in instrumentation was the introduction of the scintillation camera, known as the Anger Camera by Anger in 1957 [19]. In 1963, Kuhl and Edwards showed that radiographic tomography methods could be applied to radionuclide imaging and presented the first tomographic images produced using the Anger Camera [20].

2.1 History of Anger Camera

The first gamma camera was developed by Hal Anger in 1957. It is frequently called the Anger camera, and today it is still widely used. The Anger camera uses sets of vacuum tube photomultipliers. Each tube has an exposed face of about 3 inches in diameter and the tubes are arranged in hexagonal configuration, behind the absorbing crystal. The electronic circuit connecting the photodetectors is wired to reflect the relative coincidence of light fluorescence sensed by the members of the hexagon detector array; all the PMTs which simultaneously detect the same flash of light. Thus the spatial location of each single flash of fluorescence is reflected as a pattern of voltages within the interconnecting circuit array.

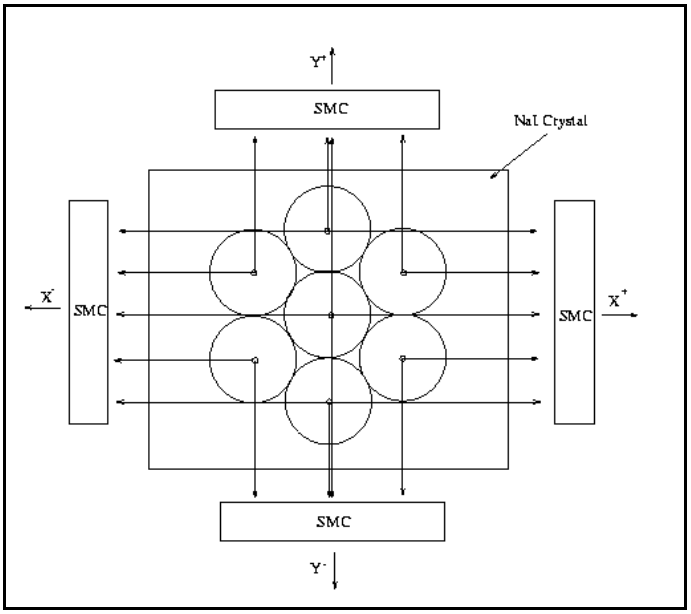

Figure 2.1 Positioning circuit of Anger Camera [21].

The system accumulates counts of gamma photons that are absorbed by a crystal in the camera, a large flat crystal of sodium iodide with thallium doping in a light-sealed housing. The crystal scintillates in response to incident gamma radiation: when the energy of an absorbed gamma photon is released, a faint flash of light is produced. This phenomenon is similar to the photoelectric effect. PMTs behind the crystal detect the fluorescent flashes and a computer sums the fluorescent counts. The computer in turn constructs and displays a two dimensional image of the relative spatial count density on a monitor. This image then reflects the distribution and relative concentration of radioactive tracer elements present in the organs and tissues imaged [22].

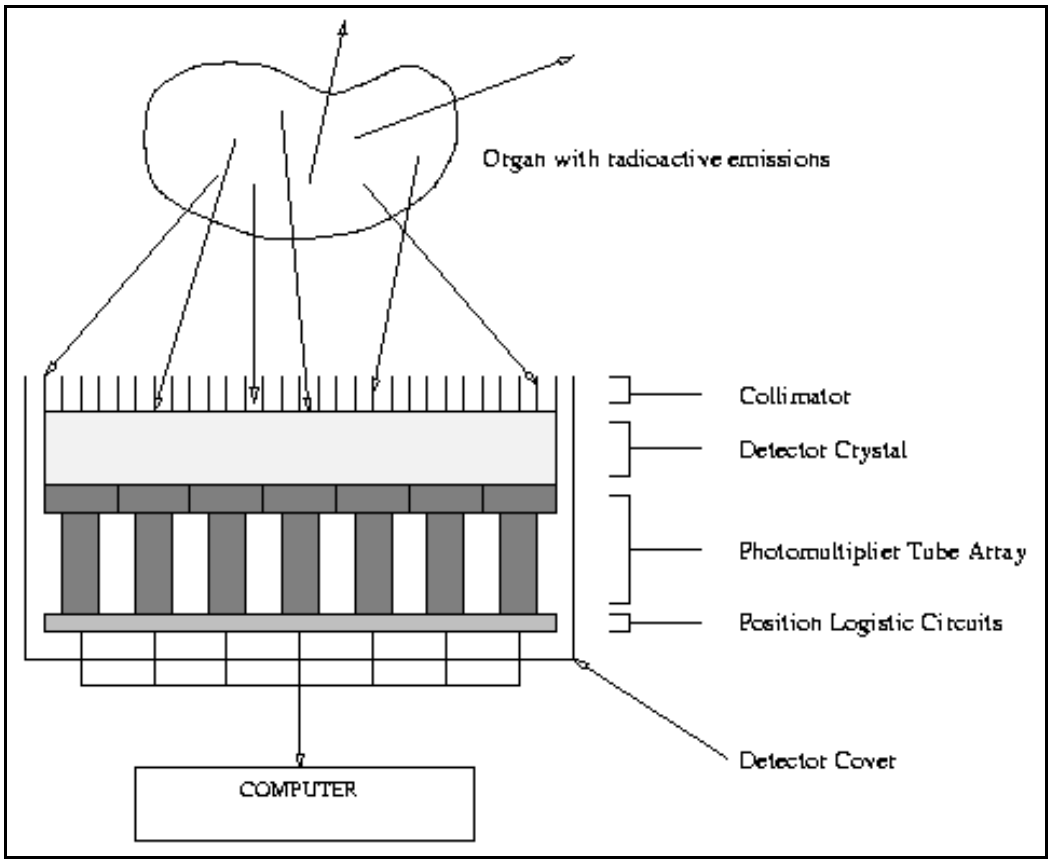

Figure 2.2 The components making up the gamma camera are the collimator, detector crystal, PMT array, position logic circuits, and the data analysis computer [21].

In order to obtain spatial information about the gamma emissions from an imaging subject, a method of correlating the detected photons with their point of origin is required. The conventional method is to place a collimator over the detection crystal/PMT array. The

collimator essentially consists of a thick sheet of lead, typically 1-3 inches thick, with thousands of adjacent holes through it. The individual holes limit photons which can be detected by the crystal to a cone; the point of the cone is at the midline center of any given hole and extends from the collimator surface outward. However, the collimator is also one of the sources of blurring within the image; lead does not totally attenuate incident gamma photons, there can be some crosstalk between holes [23].

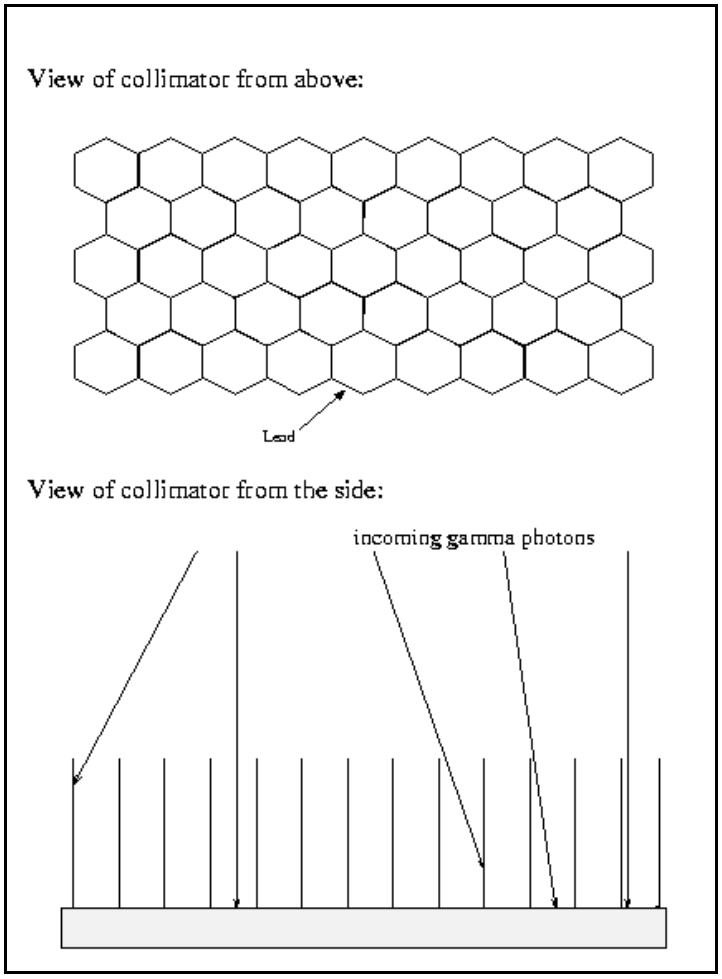

Figure 2.3 Collimator top view and side view [21].

Unlike a lens, as used in visible light cameras, the collimator attenuates most (>99%) of incident photons and thus greatly limits the sensitivity of the camera system. Large amounts of radiation must be present so as to provide enough exposure for the camera system to detect sufficient scintillation dots to form a picture [24]. Spatial

resolution decreases rapidly at increasing distances from the camera face. This limits the spatial accuracy of the computer image, it is a fuzzy image made up of many dots of detected but not precisely located scintillation.

3. BASIC NUCLEAR AND PET PHYSICS

The nucleus of an atom is composed of protons and neutrons. These particles have similar masses but different charges. Protons have positive charge and neutrons are uncharged. Negatively charged electrons surround the nucleus. The number of protons in an atom is known as atomic number. The total number of protons and neutrons is known as mass number. Atoms with the same atomic number but with different mass numbers are called isotopes. In an uncharged atom, the number of electrons is equal to the number of protons. If a nucleus has either an excess number of protons or neutrons, it is unstable and prone to radioactive decay. Nuclei that decay in this manner are called radionuclides.

One method by which nuclei with excess protons may decay is Positron emission (β+ decay). In positron emission, a proton in the nucleus is converted into a neutron and a positron. The positron is the antiparticle to the electron; it has the same mass but opposite electrical charge. The positron is ejected from the nucleus, along with a neutrino. The net energy released during positron emission is shared between the daughter nucleus, the positron and the neutrino. Therefore, the emitted positrons have a range of energies from zero up to maximum endpoint energy E_{max}. This endpoint energy is determined by the difference in atomic masses between the parent atom and the daughter atom. Decay by positron emission is the basis of PET imaging.

Another way of decaying for nuclei with excess protons is electron capture. The nucleus captures an orbital electron and converts a proton into a neutron, thus decreasing the atomic number by one. Once again a neutrino is released. Decay by positron emission occurs usually in nuclei with low atomic number while decay by electron capture occurs more likely in nuclei with higher atomic number.

3.1 Annihilation

The positron that is ejected after β+ decay has a very short lifetime in an electron rich material such as tissue. Since it is an antiparticle, it ultimately loses its kinetic energy

in inelastic interactions with electrons in the medium. Once the most of its energy is dissipated, it will combine with an electron and form a hydrogen-like state called positronium. This state lasts only 10^{-10} seconds, and the process called annihilation occurs. The mass of the electron and the positron is converted into electromagnetic energy. When this occurs the positron and the electron are almost at rest, the energy released comes mainly from the mass of the particles, and it can be computed from Einstein's formula:

$$E = mc^2 = m_e c^2 + m_p c^2 \tag{3.1}$$

where m_e is the mass of the electron (9.1×10^{-31} kg), m_p is the mass of the positron (9.1×10^{-31} kg) and c is the speed of light (3×10^8 m/s). The energy released is found to be 1.022 MeV by inserting the values into the Eq. 3.1.

The energy released is in the form of high energy photons. The net momentum is close to zero because the positron and electron are assumed to be almost at rest when annihilation happens. Since both momentum and energy must be conserved, it is not in general possible for annihilation to result in the emission of a single photon; otherwise, the net momentum would occur in the direction of that photon. Instead, two photons are emitted simultaneously in opposite directions (~180º apart) with an energy equals to 1.022 MeV/2 or 511 keV. Higher order annihilation, where more than two photons are emitted, is also possible. But this happens in 0.003% of the annihilations.

The annihilation process has very important properties that are advantageous for imaging and lead directly to the concept of PET. First of all, the annihilation photons are very energetic which means they have a good chance of escaping the body. Second, two photons are emitted with a geometric relationship. Third, independent of the element involved or the energy of emitted positrons, a PET scanner can be developed for imaging all positron emitting nuclides at this single energy.

When both annihilation photons are detected and localized, the line joining the detected locations pass directly through the point of annihilation. This point gives a good indication of where the radioactive was in the body because the point of annihilation is very close to the point of positron emission. This process forms the basis of PET imaging.

Two approaches can be used to form an image that reflects the actual locations of the radioactive atom. The first approach is rarely used because it is very difficult and expensive to build. It involves measuring the difference in arrival time of the two photons to the detectors. The relationship between the difference in arrival time of two annihilation photons and the location of annihilation with respect to half of the distance between two detectors is:

$$d = \frac{(\Delta t \times c)}{2} \tag{3.2}$$

This method is known as time-of-flight. Due to the very small time differences, it is very hard to implement. With currently available detector technology the best timing resolution can be achieved is on the order of few hundred picoseconds, this only yields a positional resolution of ~1 cm. One of the time-of-flight positron emission tomography (TOFPET) systems was built at the Royal Institute of Technology in Sweden. It gives a time resolution of 363 picoseconds and the spatial resolution is 10 mm. In order to get a better time and spatial resolution in TOFPET systems the distance between the detectors is increased but this decreased the efficiency [25].

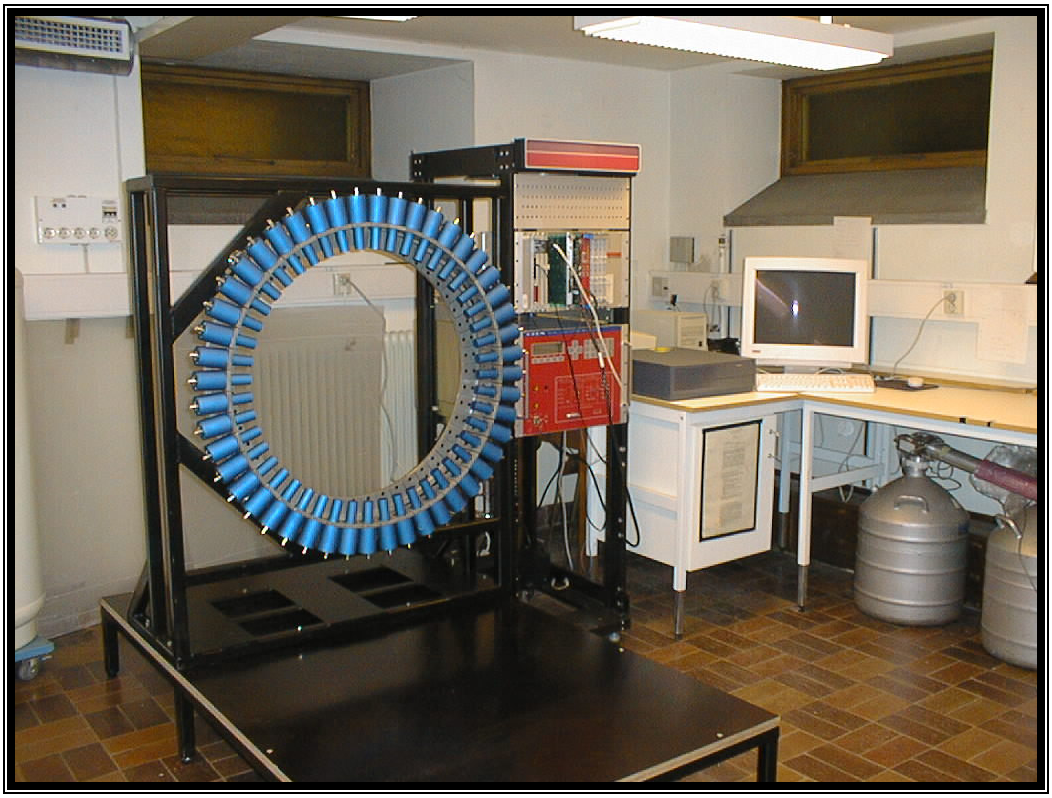

Figure 3.1 TOFPET developed by Royal Institute of Technology in Sweden [25].

Second approach is widely used. By measuring the total radioactivity along lines that pass at many different angles through the object, mathematical algorithms are used to compute the cross sectional images that give the concentration of the positron emitting radionuclide in tissues.

The aim of image reconstruction is to give quantitatively true cross sectional images of the distribution of radiopharmaceuticals in the body. Reconstruction step is needed because raw PET data only defines the location of emitting atom. We will not go into detail of image reconstruction within this work.

3.2 Photon Interactions in Matter

Among the particles interacting electromagnetically, photons are the most difficult particles to detect. The reason for this is the fact that they are non-ionizing particles, in

other words, they do not lose energy by ionization. We can only detect their secondary products created during an interaction in detector material. Therefore it is crucial to know which interactions they undergo.

When electromagnetic radiation passes through the matter, some photons will be absorbed totally by depositing all its energy, some will penetrate the matter without any interaction and some photons will interact then scatter into a different direction while loosing some of its energy. The probability of how the interaction will be depends on the photon energy and atomic number of the material.

3.2.1 Photoelectric Interaction

When a photoelectric interaction occurs, the energy of the photon is completely transferred to an atomic electron. The electron may gain sufficient kinetic energy to be ejected from the electron shell of the atom and pass thorough the surrounding material. The electron rapidly loses its energy and moves a relatively short distance from its original location. The photon's energy is deposited in the matter close to the site of photoelectric interaction. The energy transfer is a two step process. Transfer of energy from photon to electron is the first step and deposition of energy to the surrounding matter by electron is the second step. The energy of incoming photon must be higher than the binding energy for the electron otherwise photoelectric interaction cannot occur.

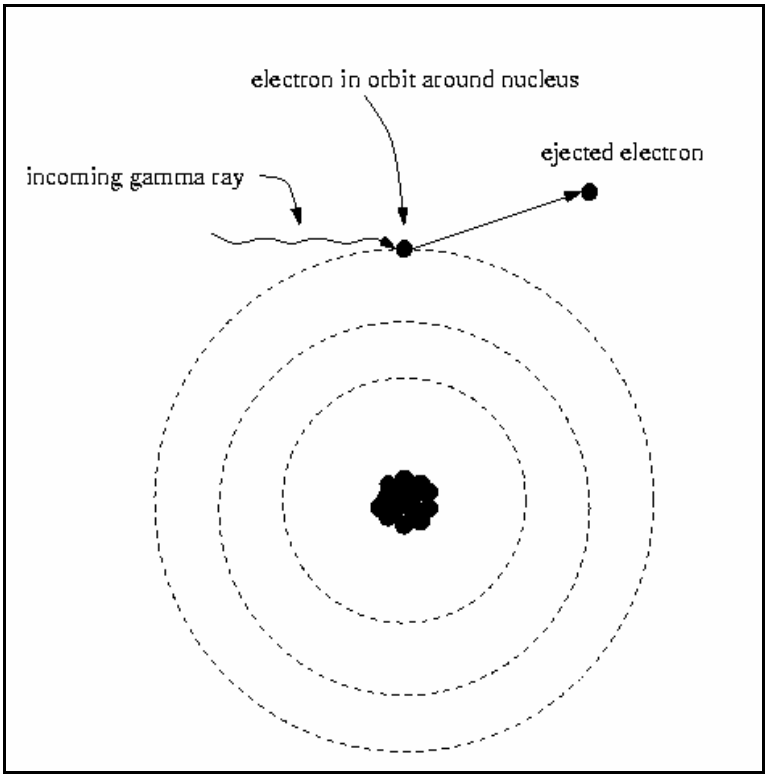

Figure 3.2 The incoming gamma ray is completely absorbed by the atom and the energy absorbed is used to eject an electron from the atom [21].

The most probable location for photoelectric interaction is with the most tightly bounded electrons in the K-shell. A photoelectric interaction results in a vacancy in the atomic electron shell which will be filled by an electron from one of the outer shells. The difference in biding energy between two electron shells can be emitted as characteristic X-ray photon or Auger electrons. The characteristic X-rays may have an energy range of 1-100 keV, while typical Auger electrons have energy of the order of few keV.

The cross section for photoelectric interaction depends on photon energy and the atomic number, Z, of the absorber material. A rough approximation for photoelectric absorption per atom is given by the below equation [26].

$$\sigma_{photoelectric} \cong \frac{Z^5}{(h\nu)^{3.5}} \tag{3.3}$$

3.2.2 Compton Scattering

Compton scattering also called incoherent scattering is a process in which the incident gamma ray interacts with an electron in a material and scatters with an angle of θ relative to its incoming direction. Only a fraction of the photon energy is transferred to the electron. The scattered photon energy *hv'* is given by:

$$hv' = \frac{hv}{1 + \frac{hv}{m_0 c^2}(1 - \cos\theta)} \tag{3.4}$$

where m_0 is the mass of electron at rest, and c is the speed of light. Equation above assumes that the collision between photon and electron is elastic and electron is initially at rest. The maximum energy transfer takes place when the photon is back scattered 180º relative to its incoming direction. Relative energy transfer from photon to electron also increases for increasing photon energies.

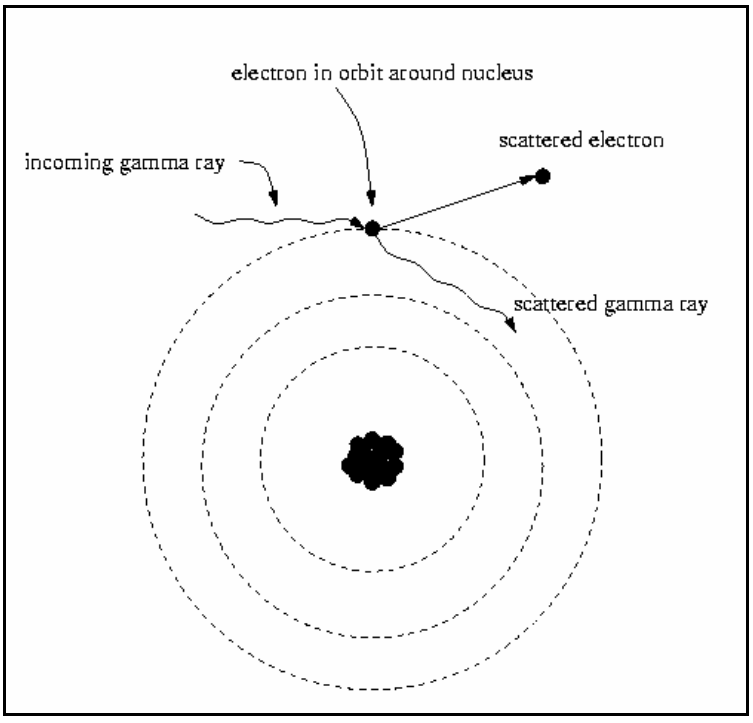

Figure 3.3 The energy of an incoming gamma ray is partially absorbed by the atom, ejecting an electron from the atom and scattering photon [21].

The differential cross section per electron for an incoherent scattering of unpolarized electrons can be calculated from Klein-Nishina Formula:

$$\frac{d\sigma}{d\Omega} = \frac{r_e^2}{2}\left[\frac{hv'}{hv}\right]^2\left(\frac{hv}{hv'} + \frac{hv'}{hv} - \sin^2\theta\right) \tag{3.5}$$

where r_e is the classical electron radius. The distribution yields the result that there is a strong tendency for forward scattering of gamma rays at very high incident energies.

The differential cross section per atom can be calculated by multiplying the Klein-Nishina cross section by incoherent scatter function *S(x, Z)*:

$$\frac{d\sigma_{incoherent}}{d\Omega} = \frac{d\sigma}{d\Omega}S(x,Z) \tag{3.6}$$

In coherent scattering the photon interacts with the whole atom in contrast to incoherent scattering where photons interacts only with atomic electron. The transfer of energy to the atom is neglected due to large rest mass of the nucleus. Coherent scattering results only in a small change in the direction of the photon since momentum change is very small. The differential cross section per atom is given by the classical Thompson cross section per electron multiplied by the square of atomic number form factor *F(x, Z)*:

$$\frac{d\sigma_{coherent}}{d\Omega} = \frac{r_e^2}{2}\left[1+\cos^2\theta\right]F^2(x,Z) \tag{3.7}$$

3.2.3 Pair Production

A process which is possible to occur at photon energies higher than twice the rest mass energy of the electron and the positron is called pair production. Such an energetic gamma ray encountering a material may decay into an electron and a positron due to the coulomb field of the nucleus. Thus, a photon is absorbed and an electron-positron pair is created. The excess energy above the rest mass threshold is shared by the created pair as kinetic energies. The positron produced in the process eventually annihilates and leads to two new photons as secondary products of the interaction. The energies of the photons and

relative angle between photons may deviate a little if the annihilation occurs in flight. The cross section κ_{pair} is proportional to the square of the atomic number.

3.2.4 Linear Attenuation Coefficient

Interactions of 511 keV photons by matters can be described with a simple exponential relationship:

$$I(x) = I(0)\exp(-\mu x) \tag{3.8}$$

where *I(0)* is 511 keV photon flux on the medium, *x* is the thickness of medium, and *I(x)* is the flux 511 keV photons that passes through the medium without interaction. μ is the linear attenuation coefficient and the probability of any interaction process to occur is given by:

$$\mu = \sigma_{photoelectric} + \sigma_{incoherent} + \sigma_{coherent} + \kappa_{pair} \tag{3.9}$$

4. PET INSTRUMENTATION

A PET scanner is made of detectors which must have high efficiency for detecting 511 keV photons that hit on their surface and must give precise information on the spatial location of the interaction. Detectors consist of dense crystalline scintillator material which forms the medium for interaction of gamma rays and high energy photons. It emits visible light when energy is deposited inside. Then the light is detected by light photon sensors, these sensors convert the light detected into electrical current.

Spatial resolution is one of the most important parameters for determining the quality of a system. It is the minimum resolvable distance between two scintillation events, and it is usually expressed in terms of full width at half maximum (FWHM). When perfectly collimated, infinitesimal narrow gamma beam hits upon a given point of the scintillation crystal surface, the corresponding responses will not form an equally precise spot, instead there will be a spread. One of the main reasons is that the light yield of the scintillator is not constant, another reason is the statistical fluctuations in the number of photons reaching the PMTs, also the quantum efficiency of PMTs are not constant and the randomness in the direction of scattered photons due to the Compton effect will lead to this spread. Such fluctuations due to the randomness of the entire process limit the minimal distance between two scintillation events, which the camera is able to resolve.

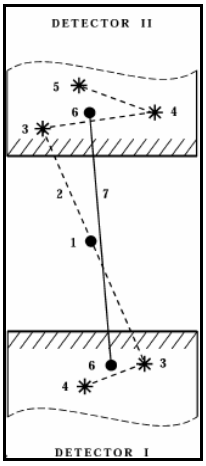

Figure 4.1 Above figure shows the effect of Compton scattering [5].

Spatial resolution varies as a function of scintillation position and photon incidence angle and strongly depends on the camera geometry. It is influenced by the limitations of the detection algorithm as well.

It is also important to determine when a photon hit the detectors. In all detector systems there is a fraction of time when, for some reasons, the system is not active, namely not able to detect events. For instance, a minimum dead time must separate two events in order that they are recorded as two separate pulses. The ability of a pair detector to determine the time difference in arrival of the annihilation photons is called time resolution and it is typically on the order of 2-6 ns. Timing window, which is 2 or 3 times the time resolution, is used to accept annihilation photon pairs. Detectors also should indicate the energy of incoming annihilation photons so that scattered photons can be rejected. This ability of the detector is called energy resolution.

Another parameter for evaluating the performance of a PET detector is the sensitivity. The sensitivity of a PET system is defined as the counting efficiency of the system for a known amount and distribution of activity. NEMA NU-2 2001 defines absolute sensitivity as the number of counts for a 70 cm line source with specified activity [27].

4.1 Scintillators

Scintillator is the most important part of the PET detector system. Scintillators emit visible light when high energy photons deposited inside them. The light is emitted isotropically and the amount of light emitted is proportional to the energy deposited. Scintillator materials can be organic or inorganic. Stopping power, light output, wavelength of the light emitted, time over which light is produced are some of the properties that should be considered when comparing scintillator materials. Table 4.1 lists some of the properties of scintillator materials suitable for detecting gamma rays in the 100 to 1000 keV range.

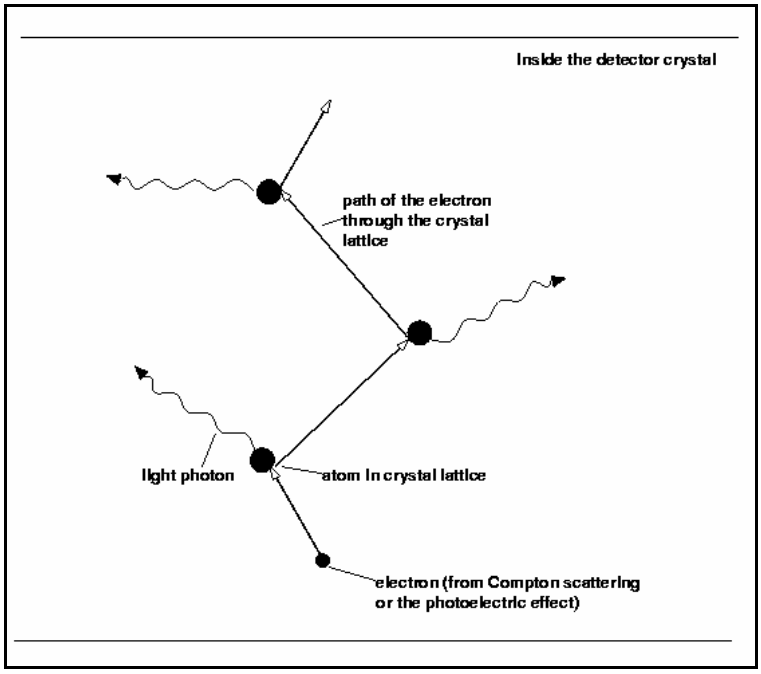

Figure 4.2 An electron released from an atom in the crystal by means of Compton scattering or photoelectric effect travels through the crystal lattice of the detector, causing atoms it hits to release visible light photons [21].

In PET imaging, high stopping capability is preferred. Since a dense material can stop large amount of incident gamma rays, it is better to choose a scintillator material with high density. The thickness of the material required to stop a large percentage of incident rays is related to the attenuation coefficient of gamma rays in that material. Another factor which is important is the amount of light produced per interaction. The light signal from the detector is used for several purposes. The relative amplitudes of signals are used to determine the location of the interaction. And the total amount of signal is used as a measure of energy deposited in the scintillator, therefore by placing a lower threshold on the output; it is possible to reject low energy photons. In both cases, major source of noise in measurements are statistical fluctuations in the number of scintillation photons detected. These fluctuations are governed by Poisson counting statistics and reduce as $1/\sqrt{N}$, where N is the number of photons detected.

Decay time is the major factor for estimating when the photon interacts in detector. Because PET imaging involves the coincident of the two annihilation photons, it is important to have less timing variation in signals produced. Index of refraction determines

how efficiently the photons will be transmitted to photodetector. Large mismatches in index result in significant internal reflection at the boundary and reduce light transmission to photodetector.

Table 4.1
Comparison table for scintillator materials [28].

	CsI (undoped)	CsI(Na)	CsI(Tl)	NaI	BGO	YAP(Ce)
Density (g/cm^3)	4.53	4.53	4.53	3.67	7.13	5.37
Decay time (ns)	16	630	1000	250	300	27
Light output (phot/MeV)	2000	41000	54000	38000	8500	18000
Conversion Efficiency (%)	4-6	85	45	100	15-20	35-40
Emission Wavelength (nm)	315	420	550	415	480	350
Refraction Index	1.95	1.84	1.79	1.85	2.15	1.94
Hygroscopic	Slightly	Yes	Slightly	Yes	No	No

4.2 Photosensors

4.2.1 Photomultiplier Tubes

PMTs are photon detectors; they convert scintillation light into electrical current. It consists of a series of dynodes each of which is held at a greater voltage with a resistor chain. Each dynode is coated with an emissive material in an evacuated glass tube. The inner surface of entrance window (photocathode) is also coated with emissive material. Light photons striking the photocathode release electrons into the tube. The probability of light photons to release an electron on the photocathode is called the quantum efficiency of the PMT. The quantum efficiency of a PMT is usually between 15% and 25% depending on the wavelength of the light photon. The released electrons are accelerated by a potential difference to the first dynode. Each electron has sufficient energy to release further electrons when they strike on the first dynode and accelerated to the second dynode. After 10 dynode stages each electron produced at the photocathode will be amplified into approximately 10^6 electrons and this will produce a measurable current at the PMT output (anode).

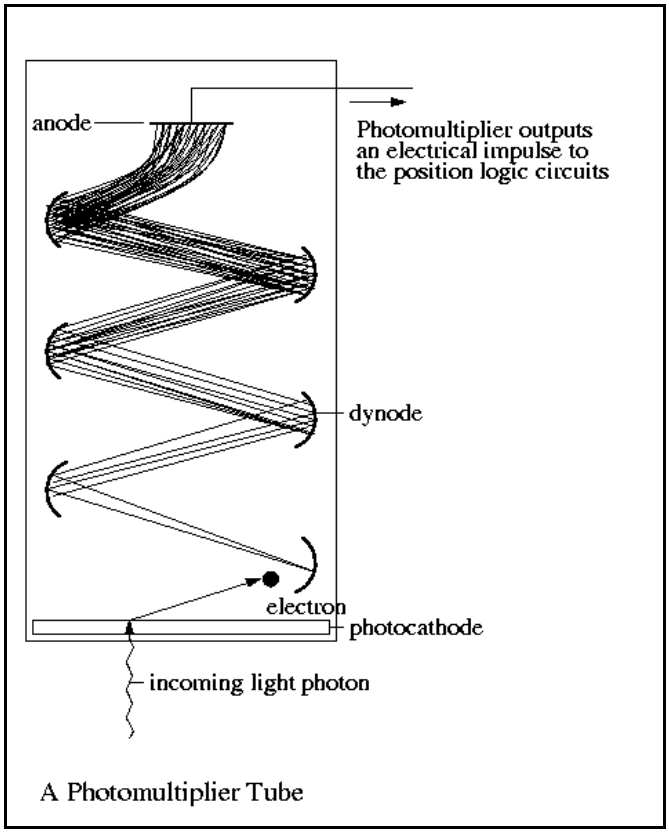

Figure 4.3 PMT [21].

The advantages of PMTs are their high gain (amplification), which leads to high signal-to-noise pulses. They are also fast and stable. Their disadvantages are that they are quite bulky.

4.2.2 Solid State Photodetectors

Silicon photodiodes are also used as an alternative to PMTs. They are small and relatively cheaper but they have a drawback since the photodiodes have no internal gain. A modification is done by applying voltage across the photodiode to achieve internal gain. These modified photodiodes are called Avalanche photodiodes (APDs) and they have an internal gain on the order of 10^2 to 10^3. The quantum efficiency of photodiodes is approximately 60% to 80%. When these two factors combine, it leads to roughly equivalent performance in terms of energy and timing performance. The stability and long term

reliability of APDs and the need for large numbers of channels of electronics have so far limited their widespread use.

4.3 Detector Configurations

4.3.1 Continuous Detectors

The major concept in designing a PET detector is based on using a continuous NaI(Tl) plate coupled to a matrix of PMTs. The location of interaction is determined by the distribution of scintillation light among the PMTs. The performance of continuous detectors depends on the number of scintillator photons detected, because this impacts the spatial resolution of the detector by determining the signal-to-noise of the PMT signals. This is the reason why a scintillator material with relatively high light output such as NaI(Tl) is chosen. The spatial resolution degrades with increasing crystal thickness, therefore crystal thickness is limited. This is due to the fact that the scintillation light will spread over a larger area before reaching the PMTs, producing a lower amplitude signal across a more number PMTs. With a 10 mm thick NaI(Tl) crystal, it is possible to achieve an intrinsic spatial resolution as high as 3 mm, at 30 mm thickness this degrades to 4 to 5 mm [29]. The size of the detector is also important. If it is large, special electronics must to be used to handle multiple events occurring in different parts of the detector at the same time. Otherwise, detector dead time becomes a limiting factor in performance. Furthermore, the edge of the scintillator plate yields poor spatial resolution since the shape of the light distribution that interacts with the edges of the crystal is altered.

4.3.2 Block Detectors

Many of the PET scanners designed today have detectors based on the block design proposed originally by Casey and Nutt [30]. A block of scintillator material is segmented into an array of smaller elements. The saw cuts are filled with a white reflective material which helps to optically isolate individual elements in the block. The scintillator block is coupled to a four single channel PMTs. The depth of the saw cut is determined to share

scintillation light in a linear fashion between four PMTs as a function of the position of the annihilation photon interaction within the block. The spatial resolution of the block detector is mainly determined by the width of the detector elements, which is commonly 3 to 5 mm.

4.3.3 Detectors with Multi-Channel PMTs and Position Sensitive PMTs

Several research PET systems have been developed with multi-channel PMTs and position sensitive PMTs to replace single-channel PMTs in a block detector or in a continuous detector. Because of their compact size and the ability to provide positional information these devices are often used to decode arrays with large numbers of scintillator elements. Both multi-channel PMTs and position sensitive PMTs have significant amount of dead space around their periphery and their cost are quite high when compared to single-channel PMTs, therefore their use has been largely limited to more specialized applications.

4.3.4 Detectors with Avalanche Photodiodes

Both single-channel APDs and array of smaller APDs encourages new design possibilities, for example, multiple concentric rings of detectors [31], or the use of APD arrays on both front and back surface of scintillator arrays [32]. Both these designs also provide DOI information. APDs are also used on small scale animal PET scanners [33, 34].

4.3.5 Depth Encoding Detectors

Detectors described so far have all focused on determining the X and Y location of an interaction. This is fine for thin detectors; however for detectors with 2 to 3 cm thickness, the uncertainty in DOI leads to a loss of spatial resolution. If the depth coordinate of interaction can be determined, the resolution degradation would be removed. This is an active area of research and many possible approaches have been proposed.

One of the approaches is to use two layers of scintillator materials, which is called phoswich. This provides a two level depth encoding capability [35, 36]. The scintillator materials are differentiated by their different decay times. The layer in which interaction occurs can be simply identified by looking at the decay time of the pulses.

Another approach is to use photodetectors at both ends of a scintillator and identify the DOI by using the ratio of signals from the photodetectors. The photodetector at far end must be thin due to minimize the attenuation of the annihilation photons that must pass through in order to reach the scintillator. This approach has been studied by Moses and colleagues [37] using a PIN photodiode array at far end.

4.4 Current Trends in Designs

4.4.1 Scintillator Choice

NaI detectors operated in Anger logic mode have served well for PET imaging, although the hygroscopic nature of the detector material has not favored the development of such desirable features as crystal segmentation. Such a feature is indeed valuable when the detector is operated without a collimator and thus requires high count rate and good linearity performances. On the other hand, the current BGO detectors for PET have insufficient light output to provide adequate energy resolution. Both BGO and NaI have relatively long light decay time (300 ns and 230 ns, respectively), which may limit their count rate capability. The technology breakthrough in positron imaging is the discovery of LSO crystal which has a number of advantages compared to the other scintillators used for nuclear medicine instrumentation.

For 511 keV photons, LSO has a mean free path almost equal to that of BGO, thus it is an efficient scintillator with the same potential for high resolution imaging. The light output of LSO is approximately three-fourths that of NaI, which is much larger than that of BGO or BaF_2, resulting in a much better energy resolution. The LSO crystal is very rugged and is non-hygroscopic.

Table 4.2
Shows a comparison of NaI, BGO, BaF_2 and LSO scintillators [38].

Properties of NaI(TI), BGO, BaF_2 and LSO for 511 keV photons.

Parameter (* at 511 keV)	NaI	BGO	BaF_2	LSO
Density (gm/cm3)	3.7	7.1	4.9	7.4
Mean free path (cm)*	2.9	1.1	2.2	1.2
Index of refraction	1.85	2.15	1.50-1.54	1.82
Hygroscopic?	yes	No	No	No
Decay time (ns)	230	300	0,8-630	40
Light output [NaI(TI)=100]	100	15	5-21	75

LSO can be thinner for the same stopping power as NaI and the light distribution is narrower in a sheet configuration, which will yield better intrinsic spatial resolution. The scintillation light decay time for LSO is 40 ns compared to 230 ns for NaI, 300 ns for BGO, and 0.8 to 630 ns for the fast and slow scintillation process of BaF_2. It provides rapid time response and may give a time resolution of 750 ps FWHM. Thus a coincidence time window of about 4 to 6 ns is feasible, which should significantly reduces random coincidences relative to current PET cameras, and improves the signal-to-noise ratio. Moreover the event processing time for LSO can be almost six times lower than that achievable with BGO or NaI, allowing higher count rate capabilities.

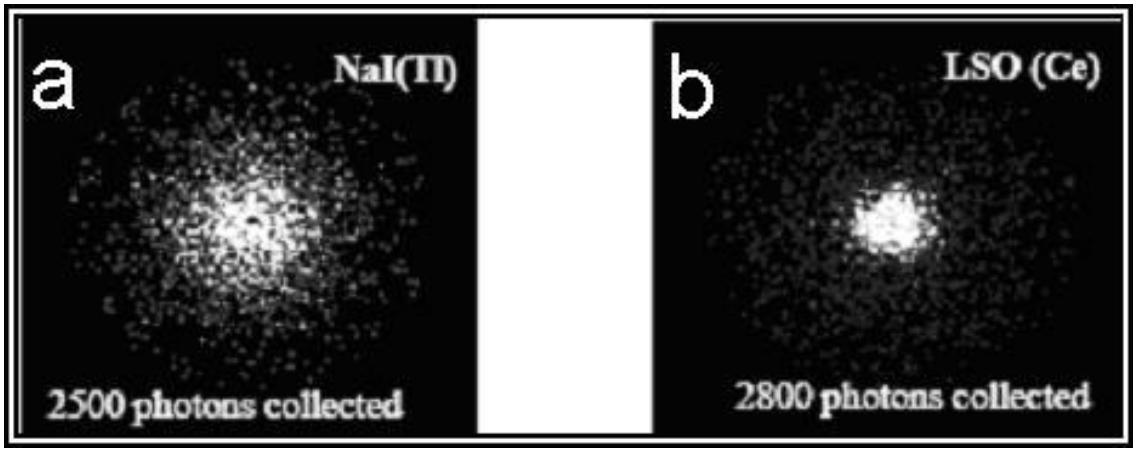

Figure 4.4 Results showing light distribution for **(a)** NaI(Tl), **(b)** LSO(Ce) [39].

LSO crystal is used in most of design today but we preferred continuous NaI crystal in our design. We preferred NaI, because it still has the highest light output, and it is cheaper. Hygroscopic nature of NaI is not very important for us because we don't use tiny discrete crystals.

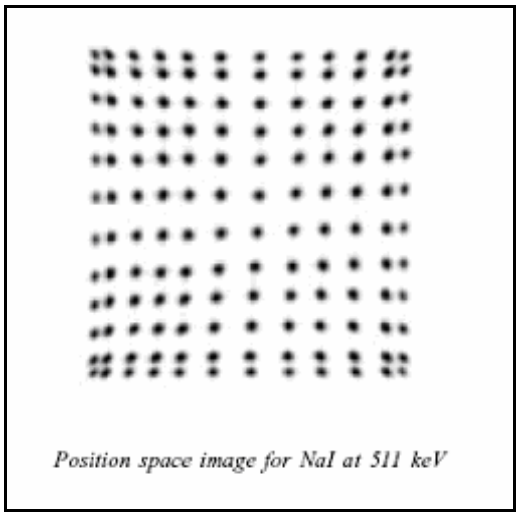

Figure 4.5 NaI crystal at 511 keV [38].

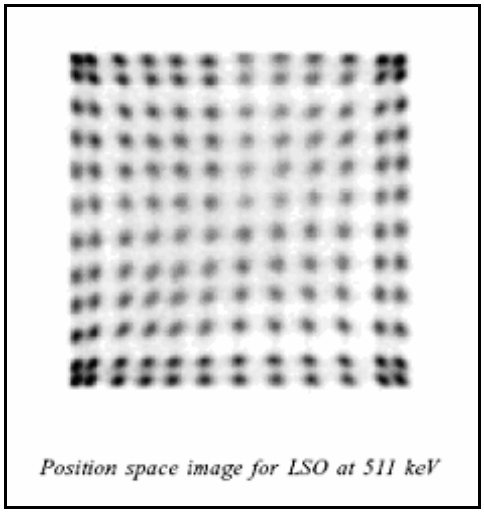

Figure 4.6 LSO crystal at 511 keV [38].

4.4.2 Detector Choice

Today, most designs have consisted of detectors assembled using tiny discrete crystal elements identified by position sensitive photomultiplier tubes (PSPMTs). This is

because the desired spatial resolution can be achieved by selecting the size of the crystal element, crystals with dimensions of <2 mm are typical. However, as described in detail by Siegel [40], making the crystals narrower for higher resolution causes several problems including inter crystal scatter, light collection difficulty, practical difficulties of accurate and consistent crystal size, and high cost associated with fine crystal cutting and treatment.

When PET detectors are constructed from individual scintillator crystals, the size of the scintillation crystals largely determines the spatial resolution. For low energy positron emitting isotopes (e.g. fluorine-18, copper-64 and carbon-11) in a small diameter PET system, this remains the case even for crystals approaching 1 mm in size [41]. Thus further improvements in spatial resolution for PET systems may be realized by developing detectors based upon very small scintillator elements. A limiting factor with this approach, particularly if the overall length of the scintillator elements is maintained, is the collection of scintillation light from these small crystals and accurate identification of each element in a tightly packed array of individual crystals. Light collection is important for spatial resolution, timing resolution and energy resolution. For example, the signal amplitude measured from 1 × 1 × 10 mm BGO crystals is less than one half of that obtained from 2 × 2 × 10 mm BGO crystals [42].

The development of PMTs with large numbers of closely packed channels has allowed the direct readout of small crystal arrays [43, 44, 45], but even the latest generation of multi-channel and position-sensitive PMTs still have significant dead space around their active area [46]. For geometrical reasons, it has therefore been difficult to use the direct coupling approach to develop small PET scanners with continuous ring geometry and with multiple rings of detector modules stacked in the axial direction. An advantage of fixed ring systems that require no motion is that very rapid dynamic imaging can be performed, while the stacking of modules in the axial direction enables better coverage and leads to higher sensitivity. One solution to this problem is the coupling of the scintillation light from the crystal array to the PMTs via optical fibers, an approach that initially used to develop microPET, a prototype small animal PET scanner [47, 48]. Despite the substantial loss of scintillation light in the optical fiber coupling, arrays of 2 × 2 × 10 mm LSO crystals could be clearly resolved and these detectors were incorporated into a small diameter

detector ring with very high packing fraction. However, using individual fibers for each crystal becomes difficult for scintillator element sizes approaching 1 mm, and coherent optical fiber bundles offer a more convenient approach.

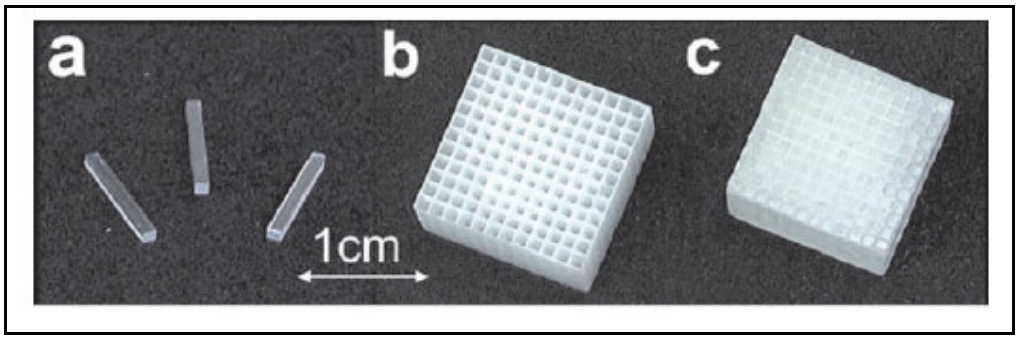

Figure 4.7 Photograph of **(a)** 1×1×10mm LSO crystals, **(b)** polyurethane grid and **(c)** completed 12 × 12 scintillator array [49].

There have been several previous efforts for using continuous crystal PET applications [40, 50]. The feasibility of using a continuous crystal detector for high resolution PET imaging has been investigated in order to reduce the system cost; improve the system performance; and increase the system's sampling density without requiring detector motion. Further, new positioning algorithms have been implemented to address linearity and edge effect artifacts that are inherent with conventional Anger style positioning schemes.

Our approach for acquiring high resolution is also to use continuous crystal while improving the performance by implementing a new algorithm. We aim to overcome the problems appear when using Anger algorithm and obtain a considerable resolution.

5. ALGORITHMS

Photon detection is one of the most important parts in nuclear imaging. Precisely detected photons will give better resolution. Different designs use different type of crystals with varying thickness and different configuration of PMTs or photodiodes for photon detection. There are also different algorithms developed.

The first scintillation point estimation algorithm based on centroid arithmetic is disclosed in US patent No. 3011057. This algorithm named after its inventor H. O. Anger. Another algorithm disclosed in European patent No. 0450388, uses a coarse estimation algorithm to calculate the scintillation point approximately. The PMTs and the weighting factors involved in the scintillation point estimation are identified as a function of the approximated point; the location of the scintillation point is calculated using only the outputs of the identified PMTs and the identified weighting factors. The identification stage is carried out using correction matrices.

One other approach uses maximization of likelihood function. In US patent No. 5293044, the use of maximum likelihood estimator based on Poisson model for statistical fluctuations of the output signals of the PMTs is proposed. An iterative refinement scheme is implemented according to the proposed algorithm. First, a rough pre-localization of the scintillation point is done using a coarse grid, then a finer grid is superimposed around the point of the coarse grid with the highest value of the probability function.

US patent No. 5444253 and No. 5285072 are two other versions of maximum likelihood approach. The latter one defines an algorithm to resolve multiple scintillation events using pattern recognition approach. According to the proposed scheme, PMT output signals are compared with multiple comparative signal sets. The locations of the multiple scintillation events in question are then registered as coinciding with the known origins belonging to the multiple comparative signals set which generates the greatest similarity value with the output signals in question.

5.1 Anger Algorithm

Anger algorithm is the most common algorithm used in PET systems. This algorithm is based on centroid arithmetic. [51, 52] Estimation of the scintillation point is calculated as the weighted sum of PMT signal amplitudes V_i:

$$x_{est} = \sum_i \omega_i V_i \tag{5.1}$$

The choice of weights depends on the detector geometry. They reflect the distance of PMTs from the center. Basic Anger algorithm has fixed weights, optimized to obtain the best resolution at a certain point of the detector array. Usually the choice of such point is the center.

Estimation error of basic Anger algorithm consists primarily of the statistical error due to fluctuations in the number of photoelectrons detected by a PMT and bias, which is especially significant in case of events distant from the center.

One of the main drawbacks of basic Anger algorithm is that it treats all signals, both from close and distant events equally, without considering the positional uncertainty of the distant event. Positional uncertainty of distant events are greater than the near ones, this phenomenon is explained by Barrett and Swindell [53]. To overcome this problem, a threshold is applied for the signals below a certain amount.

Another problem with the Anger algorithm is the linearity distortion. Unless the scintillation takes place in the center of the detector, there is always a bias. Therefore, it is necessary to make a linearity correction to determine the spatial resolution.

5.1.1 Edge Effects

As mentioned previously, camera geometry has a significant influence on the spatial resolution. The solid angle subtended by the PMT gets smaller when the distance of the interaction point to the PMT center is more. For instance, amount of photoelectrons

produced will be less, and since the number of photoelectrons obeys Poisson statistics, uncertainty in the received signal from distant events will be high, and this will influence the position estimation. Since distant events are more close to the edges of the crystal, multiple reflections and refractions distort the light distribution and lead to losses of scintillation photons. As the result, the spatial resolution in regions close to the camera edges significantly degrades.

Limitations of a detection algorithm can also degrade spatial resolution because of providing less precise estimation of distant events. For example, centroid arithmetic algorithms are unable to estimate position of events, which occur outside a circle limited by the centers of the outermost PMTs.

5.1.2 Parallax Effect and the Depth of Interaction

Crystal thickness leads to another geometric effect that degrades spatial resolution. It is especially significant in low density crystals such as NaI, where the mean penetration depth is relatively large. This effect, called as parallax effect, is caused by the fact that the annihilation photons can interact at any depth in the scintillator material. In other words, for gamma beams coming from the same location with the same angle of incidence, X and Y coordinate of the interaction can change depending on the DOI. In addition to this, even if the angle of incidence is 0° (perpendicular to the surface), depending on the DOI the PMT outputs varies due to light distribution in the crystal. This variation causes problems when using Anger algorithm. We simulated two cases to show the effect of DOI on resolution, details can be found in section 7.4, page 68.

Parallax shift would cause, even an absolutely precise detection algorithm to introduce bias in position estimation. Photon detection algorithms used currently do not provide an effective solution to the presented problem. In this work, we describe the lookup table based nearest neighbor positioning algorithm and provide results comparing the performance of a continuous NaI(Tl) crystal PET detector using our algorithm versus conventional Anger algorithm. We believe that our method give information about the DOI therefore it will reduce the parallax effect.

$$\delta_p = \Delta z \cdot \sin\theta \tag{5.2}$$

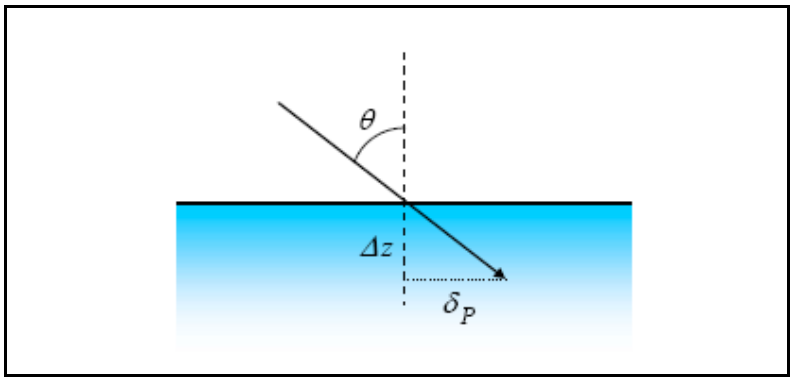

Figure 5.1 Parallax effect in a thick scintillation crystal. [17]

5.2 Iterative Algorithms

As scintillation event location calculations using Anger arithmetic do not provide accurate position estimates near crystal edges nor good position estimates especially in small detectors, it has been necessary to develop alternative event location estimation schemes.

Iterative algorithms are based on the attempt to maximize or minimize a target function determined by the particular algorithm used. The target is reached through several analytic processes called iterations. A major advantage of this type of algorithm is the possibility of incorporating different a priori information. However, it must be pointed out that inclusion of additional parameters means increase in processing times. Different iterative algorithms are present in literature, some based on the methodologies of numeric linear algebra and others based on statistical approaches. The maximum likelihood algorithm is based on statistical approach. Maximum likelihood position estimation schemes are two stage mappings from measured PMT signals to position estimates. The first stage is a mapping from measured signal space to an event characterization space. The second is a mapping from the event characterization space to the space of position estimates. The maximum likelihood algorithms are still being used in various studies.

5.3 Lookup Table Based Positioning Algorithm

Suppose that for a photoelectric scintillation at position (*x, y, z)*, PMT outputs P_1; P_2; P_3; P_4; .....; P_n are independent with a mean, μ_i. The nearest neighbor position estimation for any event with PMT outputs p_1; p_2; p_3; p_4; .....; p_n is achieved by minimizing the quantity inside of the bracket.

$$(\hat{x}, \hat{y}, \hat{z}) = \underset{\forall(x,y,z)}{\arg} \min_{x=\hat{x}, y=\hat{y}, z=\hat{z}} \sum_{i=1}^{n} (p_i - \mu_i(\hat{x}, \hat{y}, \hat{z}))^2 \tag{5.3}$$

Nearest neighbor position estimation in small scintillation detectors often involves the use of a lookup table to map event characterization vectors to the associated position estimates. As the number of possible characterization vectors determines the size of the lookup table, the number of these vectors needs to be manageable. This implies that a critical component of an implementation of nearest neighbor position estimation is the mapping of detector output signals to characterization vectors.

To validate this algorithm, we formed a sample lookup table. This lookup table is composed of mean PMT outputs from a set of points in a $1000mm^3$ volume. Since there will be infinitely many points in this defined volume, we needed to shrink the size of our sample set. From Nyquist frequency criteria, we know that, it is necessary to use a sampling rate at least twice the desired resolution. Hence, we decided to take sample points that are 0.5 mm apart in all three directions (x, y, z). For each point, 100 independent photoelectric interactions were simulated with simulation program called DETECT2000. Then, for each point, the mean PMT outputs of these 100 events are calculated. The coordinates of the point and the mean PMT outputs at this point are tabulated and stored in a file. In reality it is impossible to create a photoelectric interaction at a specific depth inside a crystal but DETECT2000 is a very realistic simulation program and it allows us to simulate such cases.

A program is written, to match the data from the lookup table with the real case PMT outputs in order to find the interaction point. This program uses the above algorithm and has an option to set a threshold, named as Proximity value, to set the maximum

acceptable error, in other words, it is possible to eliminate events for which the nearest neighbor is not close enough to accept. These eliminated events are above the energy threshold; however the inconsistency in their PMT distribution made us to believe that these events are most likely to be scattered interactions.

6. MONTE CARLO SIMULATIONS AND ADDITIONAL MODULES

Monte Carlo numerical simulation method can be described as a statistical method that uses random numbers as a base to perform simulation of any specified situation. Random number generator is a fundamental part of Monte Carlo simulations. Most Monte Carlo applications, only requires the known probability density functions of the system and the physical processes that are going to be modeled.

The design of scintillator counters assisted by Monte Carlo simulations is an effective approach. It leads to important cost savings in the detector prototyping phase, because no real expense has been made other than processing time. It also reduces the time to build a working prototype by providing realistic design specifications. For a given model, it is very easy to change different parameters and investigate the effect of these changes on the performance of the system. Even the effects of the parameters that cannot be measured experimentally can be studied. Therefore, it is convenient to use Monte Carlo simulations for the optimization of an imaging system.

6.1 The Simulation Platform

In our simulations, we used Monte Carlo simulation programs developed by TRIUMF PET detector group. These are BUILDER, GRIT and DETECT2000. BUILDER and DETECT2000 were downloaded from http://www.gel.ulaval.ca/detect/ and GRIT was provided by Emile Michael Hoskinson from Department of Physics at University of California Berkeley (For more information about the installation and use of the simulation programs please refer to the user manuals).

In addition to these programs, we wrote a program to match the PMT outputs with the average outputs stored in the lookup table file.

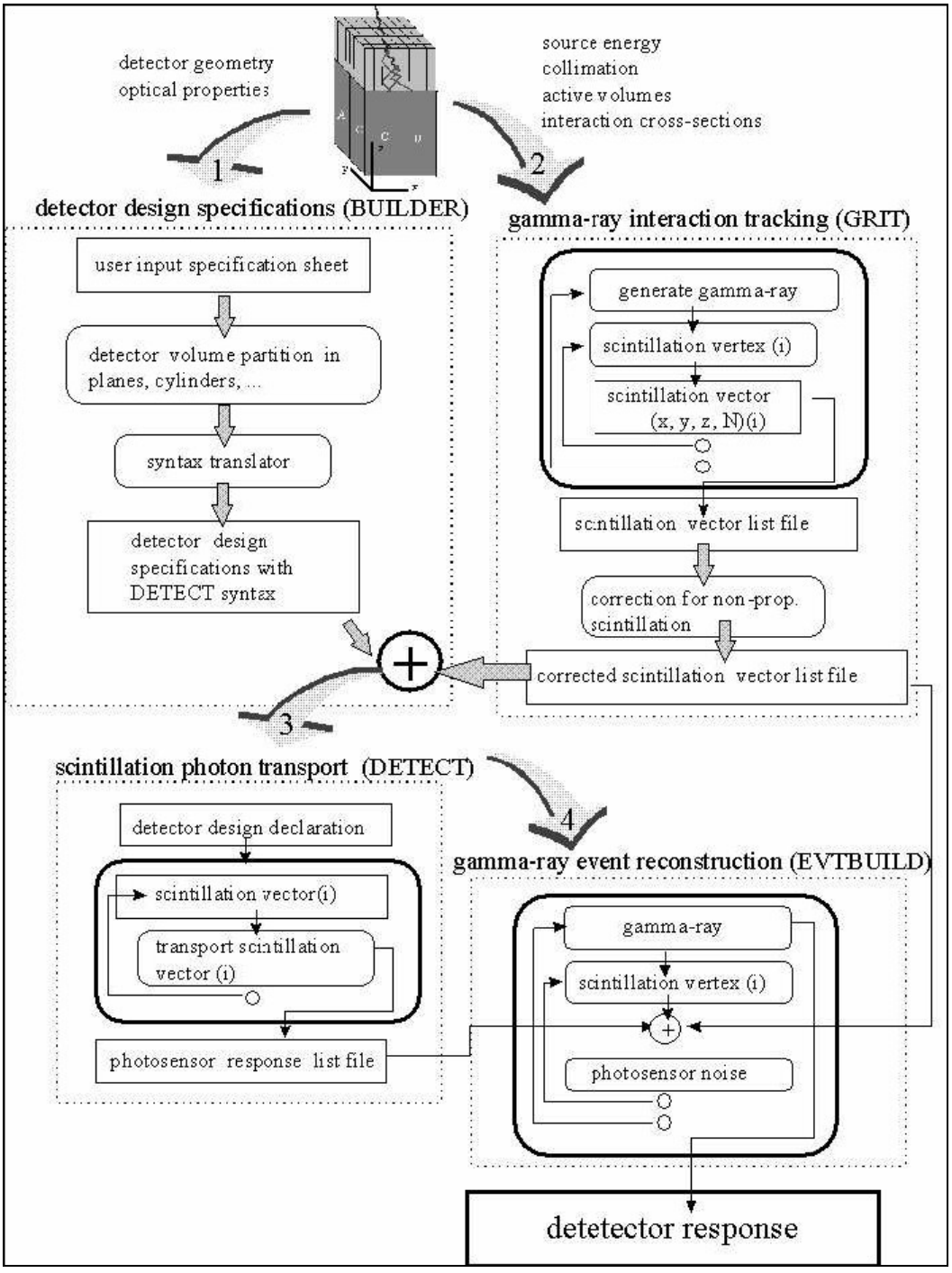

Figure 6.1 Schematic flow diagram of the steps involved in the simulations [54].

These simulation programs are validated by a lot of researchers through many simulations. One of them is simulating HR PLUS block detector manufactured by Siemens-CTI. It has dimensions of 36 mm in the transaxial direction X, 38 mm in the axial direction

Y, and is 30 mm in Z direction. The scintillation light from photons interacting in BGO is converted to pulses by an assembly of four circular PMTs of 19 mm in diameter coupled to the back face of the block. Saw cuts perpendicular to the PMTs' entrance windows segment the block into an 8 x 8 crystal matrix for position encoding. The edge and corner crystals of the block are cut slightly smaller than the inner ones to maintain uniform cut spacing between blocks in a ring detector.

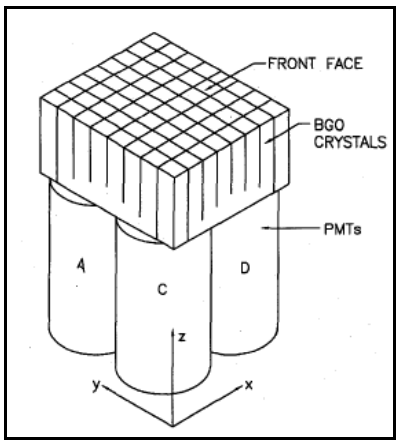

Figure 6.2 Schematic of diagram of HR PLUS [52].

According to studies published by Moisan et al. [52, 55, 56, 57] the results were in agreement of ±0.5 mm between simulated and the measured line spread functions (LSFs) for the detector. The crystal dependent energy resolution was also accurate to an average of ±1%. Some other studies related for the validation of the simulations can be found in the references section [12, 39, 52, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64].

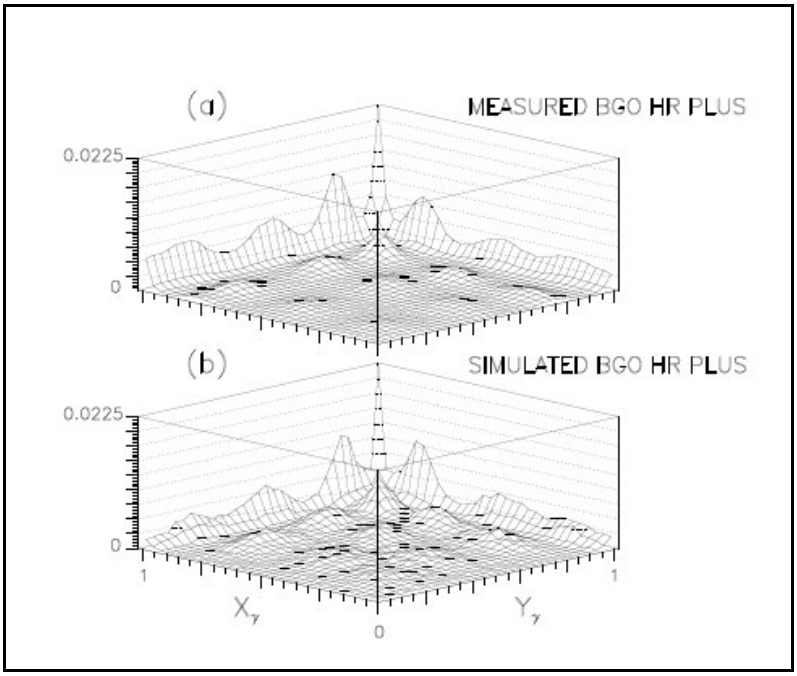

Figure 6.3 The results published by Moisan et al. **(a)** measured flood response of the detector **(b)** measured flood response of the detector [52].

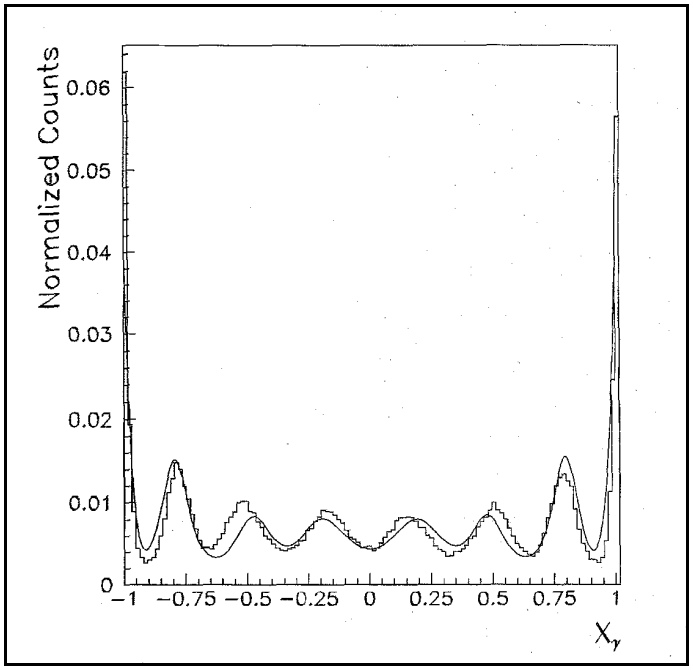

Figure 6.4 Comparison between measured (histogram) and simulated (smooth line) X, distributions for single crystals in a row at the axial center of the block [52].

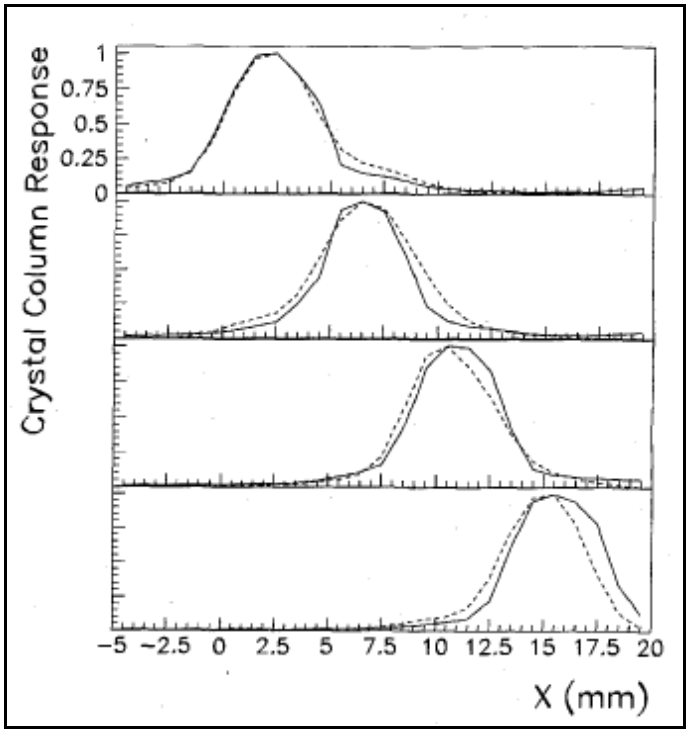

Figure 6.5 Comparison between the measured (full lines) and simulated (dashed lines) LSFs for four columns of crystals, stepping from the center (top LSF) to the edge (bottom LSF) of the HR PLUS block [52].

Table 6.1
Comparison of the measured and simulated photopeak relative channels and FWHMs [52].

crystal id.	$P_{(r,c)}/P_{(3,3)}$ (%)		$FWHM_{(r,c)}/P_{(r,c)}$ (%)	
(row,col.)	meas.	sim.	meas.	sim.
(1,1)	69±7	73	28±2	28
(1,2)	74±7	82	23±1	19
(1,3)	87±8	89	24±1	22
(1,4)	68±8	74	24±1	33
(2,1)	72±7	83	25±1	23
(2,2)	79±7	92	22±1	17
(2,3)	93±5	97	22±1	16
(2,4)	72±7	76	23±1	27
(3,1)	75±6	88	25±1	25
(3,2)	82±9	96	22±1	16
(3,3)	100	100	21±1	17
(3,4)	79±4	81	22±0	21
(4,1)	54±5	54	29±2	35
(4,2)	56±7	61	26±1	26
(4,3)	66±6	71	26±1	22
(4,4)	42±3	29	33±2	36

6.1.1 BUILDER

BUILDER is a utility for managing model geometry definitions for input to the light transport simulator DETECT2000, specializing in the modeling of crystal block detectors for PET. BUILDER was designed to extend the functionality of DETECT2000 as part of the simulation platform developed by the PET group at TRIUMF for modeling Positron Emission Topography detectors. Conceptually BUILDER sits on top of DETECT2000, and is an interface between the user's model definition and the DETECT2000 simulation driver. BUILDER simplifies the task of defining the geometrical properties of a model to DETECT2000 by translating a higher level component definition syntax into the language imposed by DETECT. The user can define sophisticated scintillation detector designs by connecting, for example, simple box components, cylindrical or rectangular PMTs, a block detector, or a “stack collection". Once each component has been defined, BUILDER connects them, checking for overlaps and incompatible surface finishes, and then translates these higher level definitions into the lower level language DETECT2000 can understand.

BUILDER takes the geometrical and optical properties from a component file and the scintillation event coordinates along with the number of optical photons to be generated at each position from a source file. The source file contains a list of scintillation event coordinates along with the number of optical photons generated at each point. It then creates an output file that can be used as an input file for DETECT2000.

6.1.2 GRIT

GRIT simulates the interactions of gamma rays in a box shaped scintillator. Type and dimensions of the scintillator can be defined by the user. In GRIT, gamma rays originate from either a point source or a spherical source. The energy of the gamma rays can be set by the user. If wanted, a pencil or line collimator can be placed between source and the scintillator. GRIT program creates an output file containing information about each interaction occurred in the block which are coordinates of the interaction, number of photons given off, a sequential index which is 0 for each incoming ray and incremented for each interaction took place. This file used as the input source file for BUILDER.

6.1.3 DETECT2000

DETECT2000 is a Monte Carlo program for modeling the behavior of optical systems with a special emphasis on scintillation detectors. It generates individual scintillation photons in specified portions of the scintillator, follows each photon in its passage through the various components and interactions with surfaces, allows for the possible absorption and re-emission by a wave-shifting component, and records the fate (absorption, escape, or detection) of each.

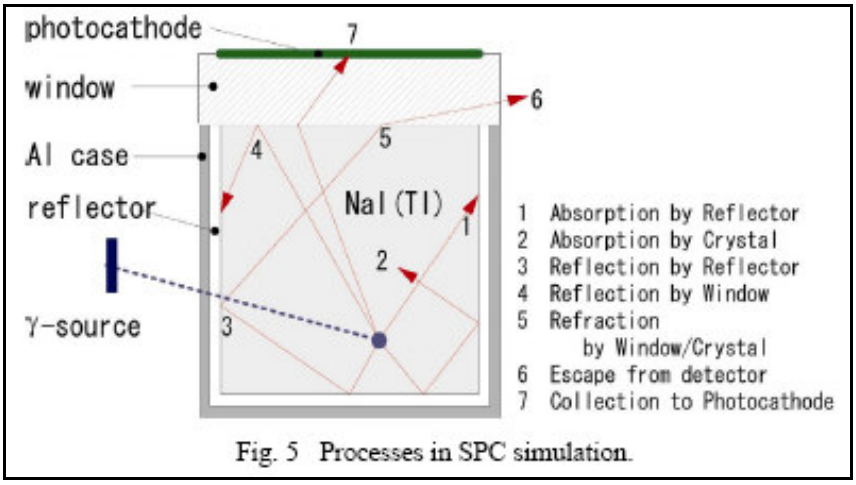

Figure 6.6 Possible interactions [65].

A very general syntax for geometry specification exists to allow the representation of complex systems consisting of composites of many different scintillator and wave shifting elements. Any element may consist of a volume specified by multiple planar, cylindrical, conical, or spherical surfaces with arbitrary orientation. More complex shapes can be built up by specifying contiguous elements with common “pseudo-surfaces" that are ignored by the simulation. The optical behavior of real surfaces may be specified to simulate possible reflection under polished, ground, painted, or metalized conditions. Surfaces in optical contact are treated using Snell's law of refraction. Within each optical element, bulk absorption, scattering, and wavelength shifting are simulated by specifying a mean distance of photon travel for each process.

The program uses initial definition statements to specify the optical properties of all materials and surface finishes used in the system. Abstract geometrical objects, planes, cylinders, cones and spheres in convex or concave orientations are also defined. Components are then built out of these materials, finishes and objects. The optical behavior of each surface is chosen by selecting one of a set of previously defined surface finishes, including a photosensor surface representing the photocathode of a PMT. Surfaces may either be external (assumed to be an interface with a vacuum) or shared with another component. An extensive set of consistency checks are incorporated to flag errors where possible.

Photons are isotropically generated within a defined material and volume subset of the system. Photons are tracked on an individual basis until they are absorbed, detected, or have escaped from the system. At each photon reflection or scattering, the program logic determines the new direction of the photon, identifies the component in which it is traveling, and computes the next intersection with a surface. A random sampling is then made to determine if the photon is bulk absorbed, scattered, or wavelength shifted over this path. If none of these processes occur, the optical properties of the next surface determine whether the photon is reflected, refracted, detected, or absorbed. This process is then repeated for all subsequent paths in the history. A maximum flight time per history is specified to abort those cases in which a photon becomes internally trapped. In order to simulate the photosensor response time, a delay before detection is recorded for photons intersecting with detection type surfaces.

At the end, an output file containing the number photons counted by each component with a DETECT type of finish is generated. If desired, a report is prepared that summarizes the probability of occurrence and statistical uncertainty estimate of each of possible fates. Data are also reported on the probability of wavelength shift, mean age and mean number of surfaces encountered. This data is tabulated both for all photons and for just the subset that are ultimately detected. A histogram of photon ages can also be generated.

6.2 Additional Modules to GRIT and DETECT2000 Software

During our simulations we found that the softwares used could be improved and became more realistic if some additional corrections were made. For this purpose we wrote additional modules to make our simulations more realistic. We used the random number generator libraries created by Agner Fog [66] in writing the correction modules.

6.2.1 Non-proportional Light Yield Correction

Scintillator's gamma ray energy resolution is dependent on a number of factors. First and most important is the number of charge carriers produced, which are photoelectrons in the case of photomultiplier. Second factor is intrinsic energy resolution that is connected with imperfections and non-homogeneities of the crystal and properties of the crystal itself. Deviations, which are known as non-proportionality (or non-linearity) in the scintillation efficiency or response, result in difficulties in the determination of the energy of the detected radiation [67, 68].

There are some theories accounting for the observed non-proportionality and energy resolution of the scintillators. Iredale pointed at electron response function as a more fundamental measure of non-proportionality [69, 70]. Valentine et al. has proposed Compton coincidence technique to measure directly the electron response function avoiding the limitation of surface excitation only [71, 72].

The simulation program we used does not take this non-proportionality into account. It assumes the light yield is constant and not changing. This affects the results negatively, and a correction has to be made. The phenomenon of non-proportional response for NaI(Tl) has been studied by Zerby et al., Porter et al., and Hill et al. [73, 74, 75]. Fig. 6.7 shows the result of the three studies and a recent study [76].

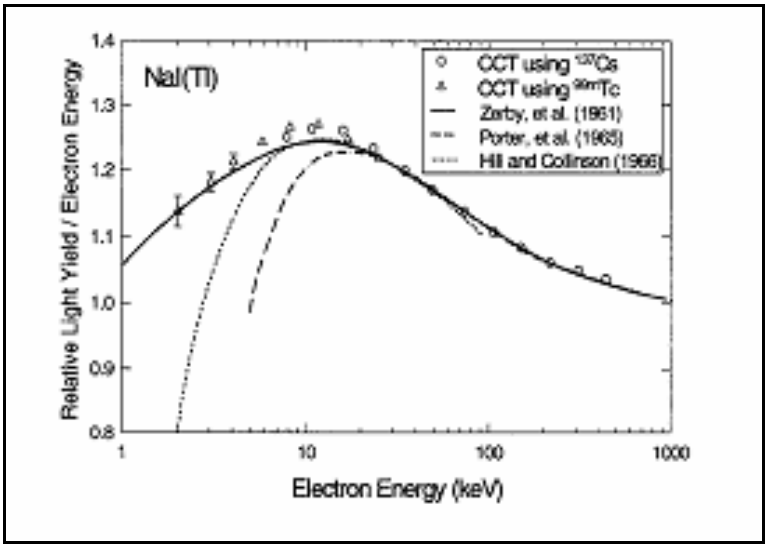

Figure 6.7 Results of the previous studies for NaI(Tl) [76].

In order to correct the absence of non-proportionality in the GRIT simulation program, we used the graph above and divided it into segments, where we can assume the function changes linearly. Then, we wrote a program in C++ to correct the light yield of our data by using these linear relationships.

6.2.2 Statistical Correction

In addition to this, to simulate photoelectron generation more realistic, we followed the statistical model of Barrett and Swindell [53], and made some additional corrections. Barret and Swindell [53] prove that Poisson statistics is preserved if a Poisson process is used as input to a binomial process. As the result, the number of photoelectrons in the PMT also has a Poisson distribution.

The total number of optical photons created by monoenergetic gamma rays can be described as Poisson stochastic process. On the other hand, the probability that an optical photon will reach the photocathode has binomial distribution, which depends mainly upon the solid angle subtended by the photocathode from the scintillation source. Another binomial process is that of photoelectron generation in the PMT. The probability of

incident photon conversion into a photoelectron is a statistically independent event with probability equal to the quantum efficiency of the PMT.

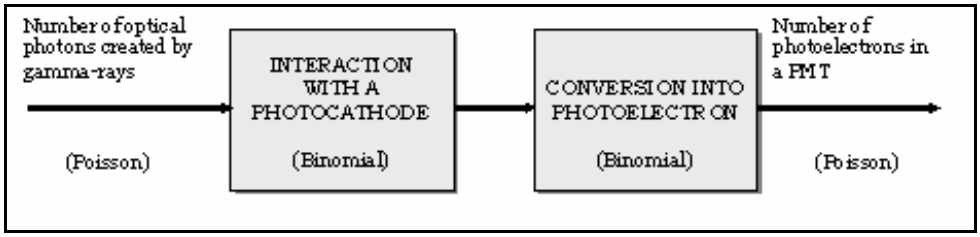

Figure 6.8 Scheme of photoelectron generation statistical model.

DETECT2000 simulation program was good at applying binomial distribution to the interaction of optical photons with the photocathode, while the other two distributions were not realistically applied. Moreover, DETECT2000 assumes quantum efficiency of PMTs as 1. Therefore, we made a modification in the light yield correction program and added a random number generator to correct the number of optical photons generated obeying Poisson distribution.

Secondly, for the photoelectron generation in PMTs, we wrote another program that will correct the quantum efficiency of the PMTs and this correction is going to obey binomial distribution. Again, we modified and used our random number generator, and we assumed quantum efficiency of PMTs as 20%.

After corrections, our model became more close to the real cases. Fig. 6.9 shows the energy resolution of NaI(Tl) from another study [65] and Fig. 6.10 shows the energy resolution from our study.

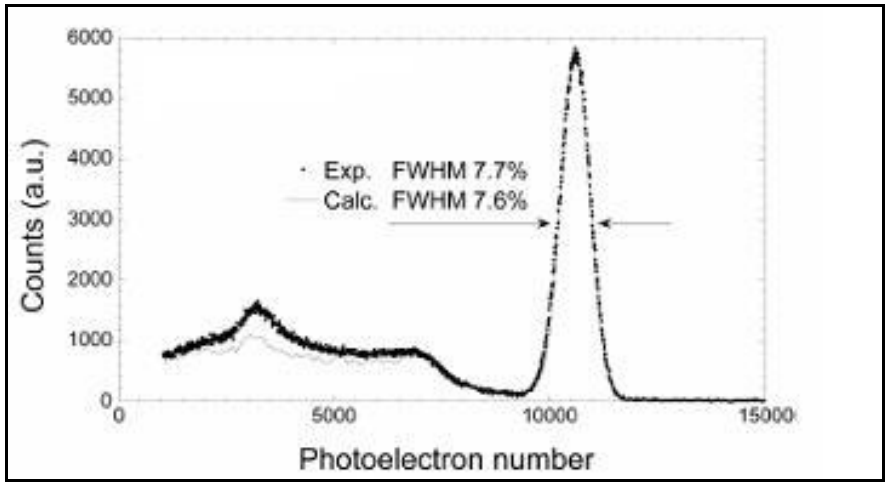

Figure 6.9 Energy resolution of NaI(Tl) at 662 keV [65].

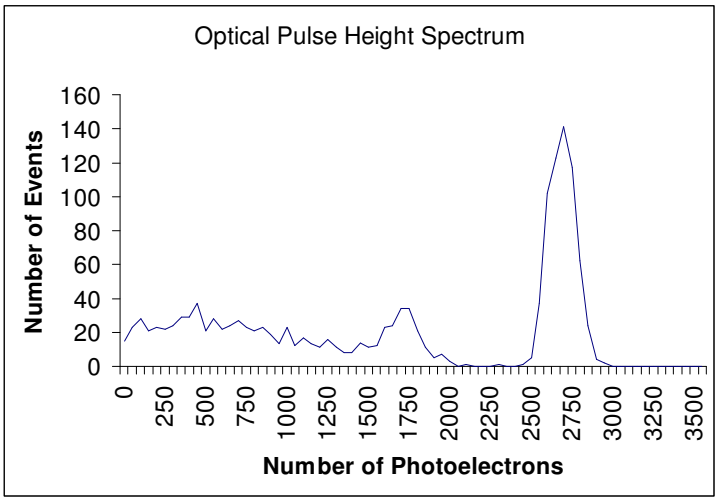

Figure 6.10 Energy resolution of NaI(Tl) at 511 keV from our study.

6.2.3 Scattered Photon Correction

DETECT2000 program simulates each interaction event separately; therefore successive scattered events seem like single events in the output file. In real life, this is not the case, PMTs response time is not enough to differentiate successive scattered events from a single events. We wrote a program which uses the information in the DETECT2000's input file if the event is a single or scattered event, to combine the PMT outputs of scattered events as a single event.

7. SIMULATIONS AND RESULTS

We designed a detector composed of a NaI(Tl) crystal with dimensions 60 mm x 60 mm x 10 mm. The NaI(Tl) crystal is coupled to 16 PMTs, they are square PMTs with dimensions 15 mm x 15 mm, but the live area is only 12.5 mm x 12.5 mm. The overall active area is 64 %. There is a 2 mm thick light guide between the crystal and the PMTs and the inter-tube area was assumed consisting of an ideal light absorbing material. Optical properties of different surfaces were selected according to DETECT2000 syntax as painted and ground for side and face surfaces respectively. Reflection coefficient for ground face was set to 0.85 and for painted sides they were set to 0.95. Refraction coefficients for NaI(Tl) was assumed to be 1.85 and for the glass it was 1.52. Refraction and reflection coefficients were based on the coefficients used in previous works performed by Bronstein et al. [5]. The whole system was defined under BUILDER using necessary commands.

As mentioned previously, lookup table has been formed in a 1000 mm^3 volume between the following coordinates [(15,35,0), (25,35,0), (25,45,0), (15,45,0), (15,35,10), (25,35,10), (25,45,10), (15,45,10)]. To calculate resolution, three points were chosen. Two of them were in side the defined volume [(20, 40), (22, 38)] and the other point is the center of the crystal (30, 30) which was out of the defined volume.

Narrow beam of 5000 gamma photons with energy of 511 keV sent from 50 mm away from each point perpendicular to the surface under GRIT. First, light yield correction has been made. The corrected file used together with the BUILDER and input file for DETECT2000 has been obtained. Secondly, PMT quantum efficiency correction was applied. Then, the scattered events in the PMT output file was merged. Finally, the output file was subjected to matching with the lookup table if the point of interests could be matched within the defined volume. Different thresholds were tried during matching, each result set was recorded. After the results were obtained, necessary calculations have been made; the resolutions (resolutions only in x direction) were found and compared. Besides intrinsic resolution, energy resolution is also calculated and compared to theoretical energy resolution.

It takes about 37.3 eV of energy to produce each optical photon, and about 64% of these will strike the photocathode in the detector considered. Assuming quantum efficiency about 20%, theoretical energy resolution can be calculated as follows:

$$N = \frac{511000}{37.3} \times 0.20 \times 0.64 \approx 1753$$
$$\%SD = \frac{1}{\sqrt{N}} = \frac{1}{\sqrt{1753}} \approx 2.4\% \quad (7.1)$$
$$2.4 \times 2.35 = 5.6\%$$

Energy threshold is taken as 10% below the total incoming energy that is 511 keV. Thus, it is found to be 460 keV, this is almost equal to energy of 2400 photoelectrons detected.

$$\frac{460000 \times 0.20}{37.3} \approx 2400 \quad (7.2)$$

Details and statistics for each point can be found below.

7.1 Point (30, 30) (Center of the Crystal)

1. Number of Gamma Rays Sent: 5000
2. Number of Hits: 1404
3. Number of Interactions (Photoelectric): 262
4. Number of Interactions (Single Interaction and Escaped): 675
5. Number of Interactions (Compton Scatter): 467
   - Number of Interactions with 5 Scatterings: 0
   - Number of Interactions with 4 Scatterings: 6
   - Number of Interactions with 3 Scatterings: 22
   - Number of Interactions with 2 Scatterings: 120
   - Number of Interactions with 1 Scattering: 319

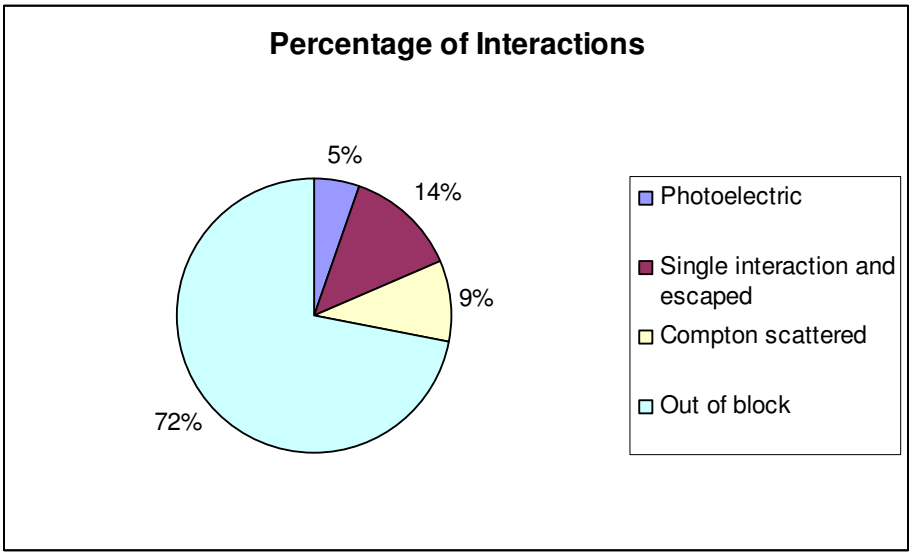

Figure 7.1 Percentage of interactions at (30, 30).

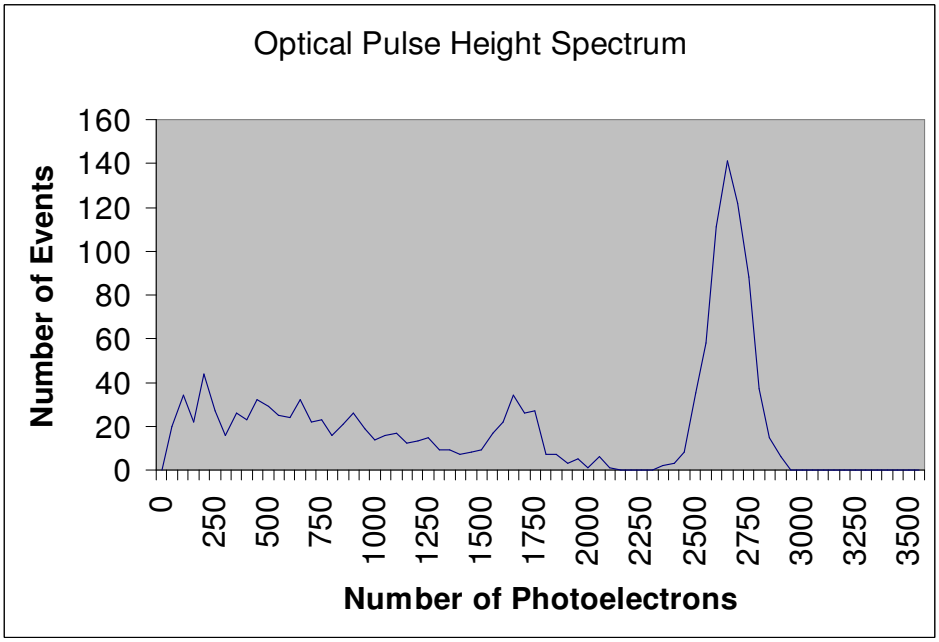

Figure 7.2 Optical pulse height spectrum at (30, 30).

6. Number of Interactions Above Energy Threshold (2400): 621
7. Average PMT Output Above Energy Threshold (Peak) : 2637
8. Standard Deviation : 88
9. FWHM : 207

10. Energy Resolution : 7.84%

7.1.1 Resolution at (30, 30) by Using Anger Algorithm

Since this point is out of the defined volume we cannot use the lookup table, we only include the Anger result for this point in order to compare with our results because this point is the center of the detector and we know that Anger algorithm has fewer problems in the center of the detector. The results at this point may need little correction but we neglect this and use them without any linearity correction.

The calculated average X coordinate is 30.03 with a standard deviation of 1.26 and calculated Y coordinate is 29.92 with a standard deviation of 1.22. It is clear that Anger algorithm has no bias in the center. Resolution in X direction without any correction is found to be 2.96 mm.

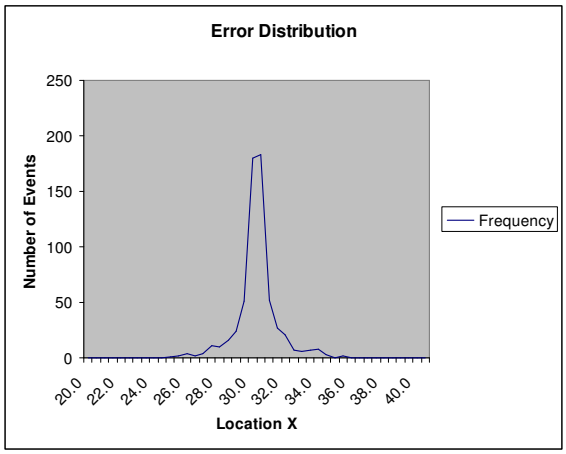

Figure 7.3 Error distribution at (30, 30) when using Anger algorithm.

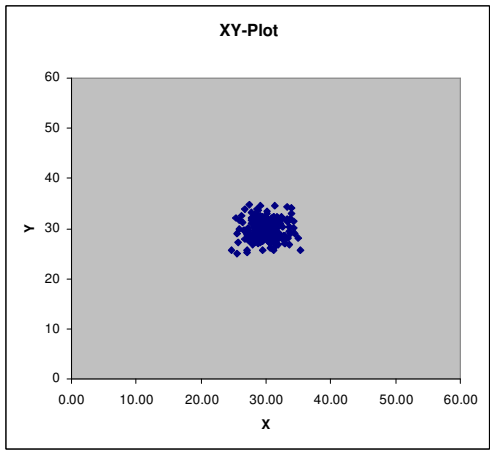

Figure 7.4 XY-plotting at (30, 30) when using Anger algorithm.

7.2 Point (20, 40)

1. Number of Gamma Rays Sent: 5000
2. Number of Hits: 1410
3. Number of Interactions (Photoelectric): 247

4. Number of Interactions (Single Interaction and Escaped): 676
5. Number of Interactions (Compton Scatter): 487
   - Number of Interactions with 5 Scatterings: 2
   - Number of Interactions with 4 Scatterings: 1
   - Number of Interactions with 3 Scatterings: 27
   - Number of Interactions with 2 Scatterings: 100
   - Number of Interactions with 1 Scattering: 357

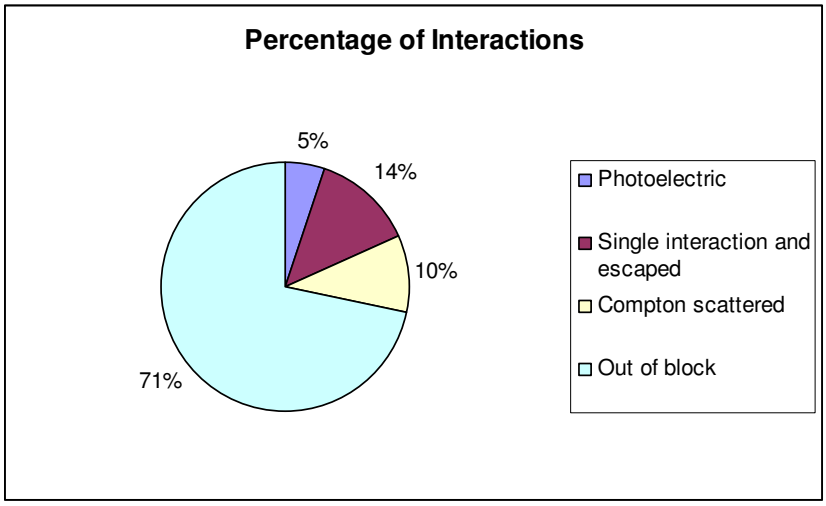

Figure 7.5 Percentage of interactions at (20, 40).

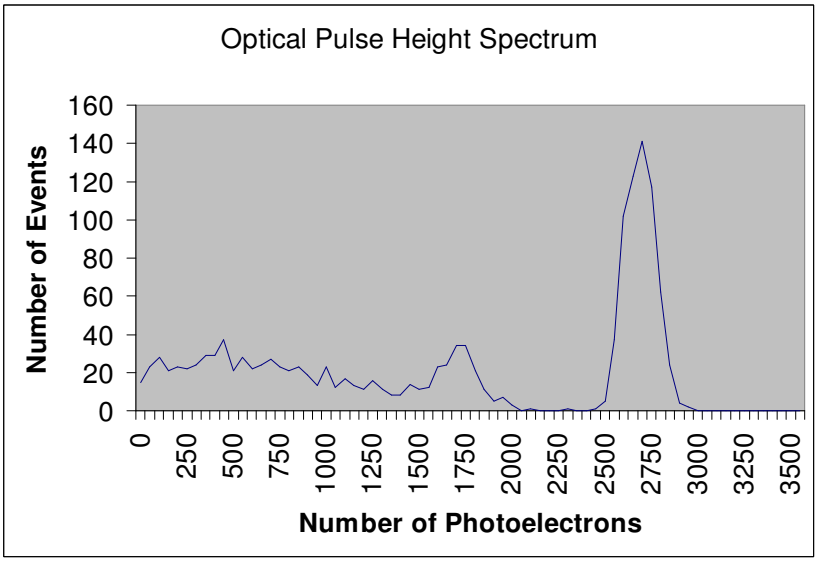

Figure 7.6 Optical pulse height spectrum at (20, 40).

6. Number of Interactions Above Energy Threshold (2400): 618
7. Average PMT Output Above Energy Threshold (Peak) : 2666
8. Standard Deviation : 80
9. FWHM : 188
10. Energy Resolution : 7.05%

7.2.1 Resolutions at (20, 40) by Using Our Algorithm

7.2.1.1 Case I (Proximity value = 5000) When Proximity value is set to 5000, 493 events out of 618 interactions could be matched with the table. From 493 events, average X coordinate is calculated as 19.89 with a standard deviation of 1.22 and Y coordinate is calculated as 39.99 with a standard deviation of 1.34. Resolution in X direction is found to be 2.87 mm.

7.2.1.2 Case II (Proximity value = 4500) When Proximity value is set to 4500, 476 events out of 618 interactions could be matched with the table. From 476 events, average X coordinate is calculated as 19.91 with a standard deviation of 1.21 and Y coordinate is calculated as 39.95 with a standard deviation of 1.26. Resolution in X direction is found to be 2.85.

7.2.1.3 Case III (Proximity value = 4000) When Proximity value is set to 4000, 444 events out of 618 interactions could be matched with the table. From 444 events, average X coordinate is calculated as 19.91 with a standard deviation of 1.15 and Y coordinate is calculated as 39.95 with a standard deviation of 1.20. Resolution in X direction is found to be 2.71.

7.2.1.4 Case IV (Proximity value = 3500) When Proximity value is set to 3500, 409 events out of 618 interactions could be matched with the table. From 409 events, average X coordinate is calculated as 19.96 with a standard deviation of 1.10 and Y coordinate is

calculated as 39.95 with a standard deviation of 1.19. Resolution in X direction is found to be 2.58 mm.

7.2.1.5 Case V (Proximity value = 3000) When Proximity value is set to 3000, 363 events out of 618 interactions could be matched with the table. From 363 events, average X coordinate is calculated as 19.95 with a standard deviation of 1.06 and Y coordinate is calculated as 39.98 with a standard deviation of 1.09. Resolution in X direction is found to be 2.49 mm.

7.2.1.6 Case VI (Proximity value = 2500) When Proximity value is set to 2500, 304 events out of 618 interactions could be matched with the table. From 304 events, average X coordinate is calculated as 19.98 with a standard deviation of 1.03 and Y coordinate is calculated as 40.00 with a standard deviation of 1.06. Resolution in X direction is found to be 2.43 mm.

7.2.1.7 Case VII (Proximity value = 2000) When Proximity value is set to 2000, 222 events out of 618 interactions could be matched with the table. From 222 events, average X coordinate is calculated as 20.00 with a standard deviation of 0.91 and Y coordinate is calculated as 40.02 with a standard deviation of 1.02. Resolution in X direction is found to be 2.13 mm.

7.2.1.8 Case VIII (Proximity value = 1500) When Proximity value is set to 1500, 123 events out of 618 interactions could be matched with the table. From 123 events, average X coordinate is calculated as 20.05 with a standard deviation of 0.89 and Y coordinate is calculated as 40.07 with a standard deviation of 0.93. Resolution in X direction is found to be 2.09 mm.

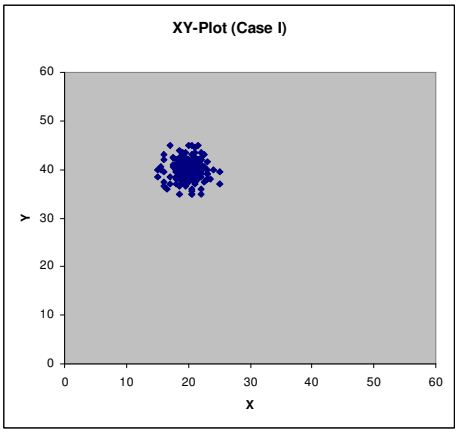

Figure 7.7 XY-plotting of Case I.

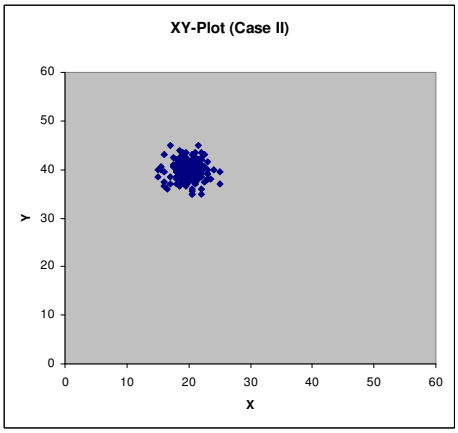

Figure 7.8 XY-plotting of Case II.

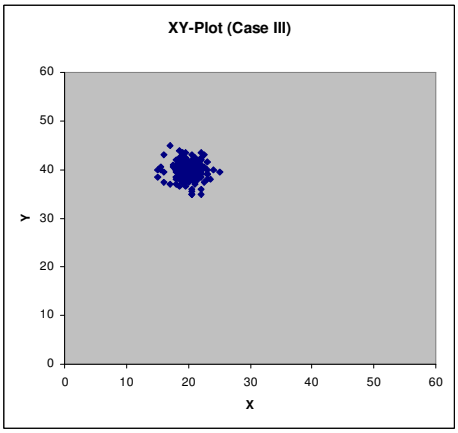

Figure 7.9 XY-plotting of Case III.

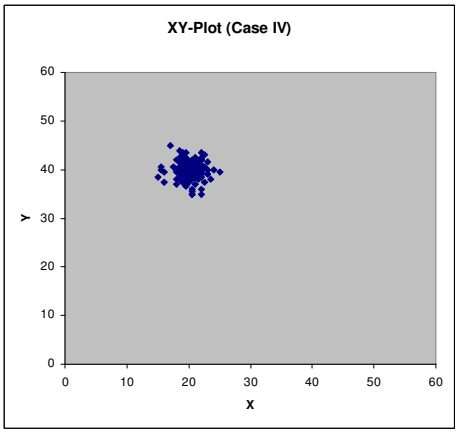

Figure 7.10 XY-plotting of Case IV.

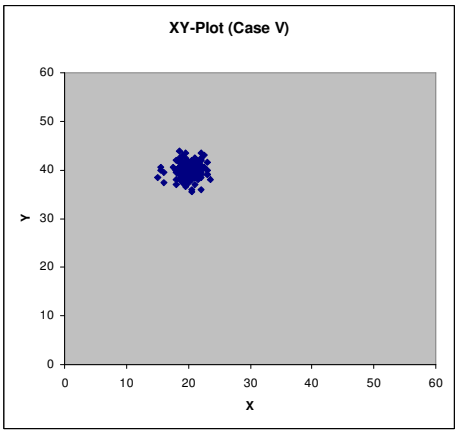

Figure 7.11 XY-plotting of Case V.

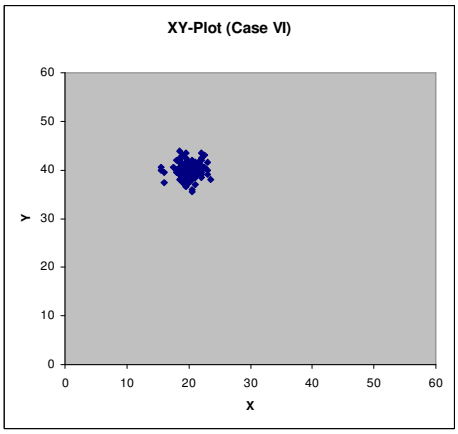

Figure 7.12 XY-plotting of Case VI.

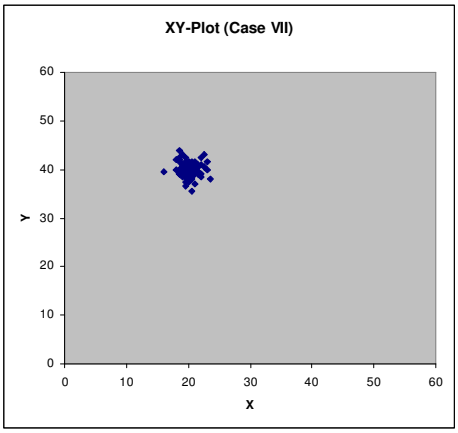

Figure 7.13 XY-plotting of Case VII.

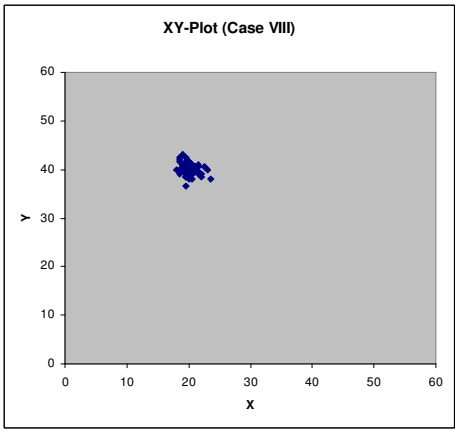

Figure 7.14 XY-plotting of Case VIII.

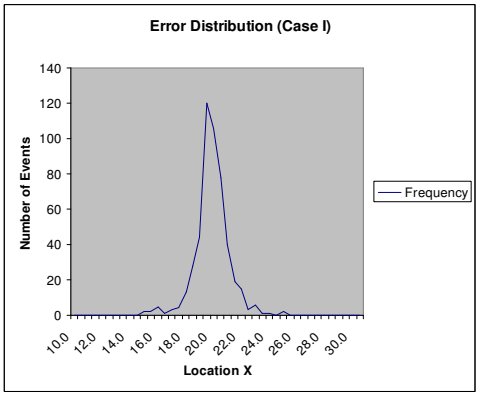

Figure 7.15 Error distribution for Case I.

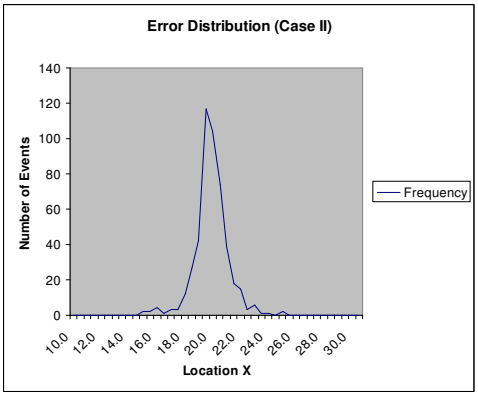

Figure 7.16 Error distribution for Case II.

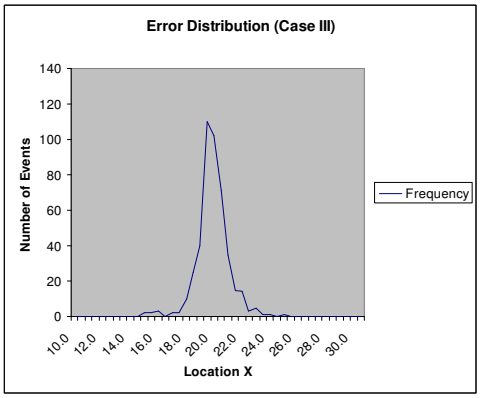

Figure 7.17 Error distribution for Case III.

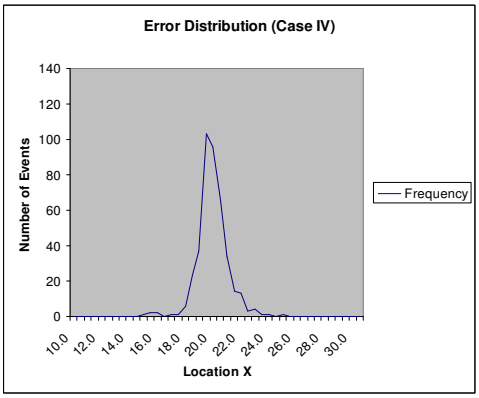

Figure 7.18 Error distribution for Case IV.

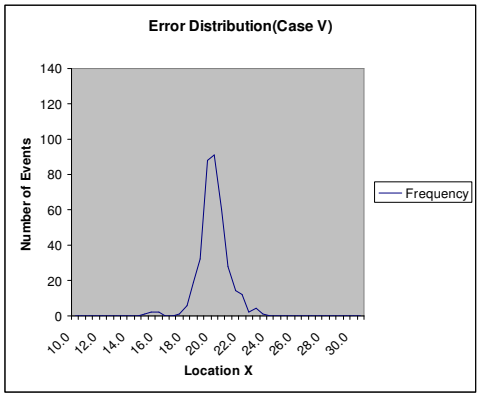

Figure 7.19 Error distribution for Case V.

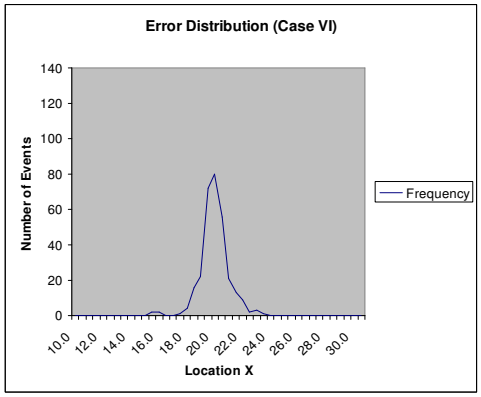

Figure 7.20 Error distribution for Case VI.

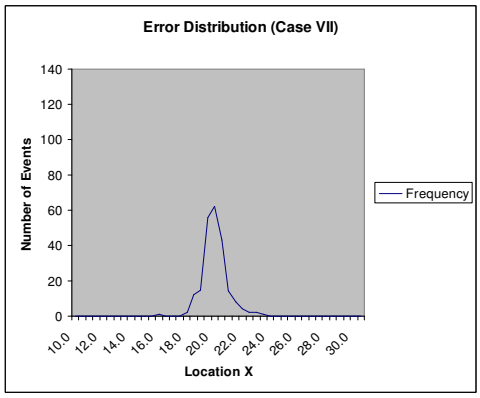

Figure 7.21 Error distribution for Case VII

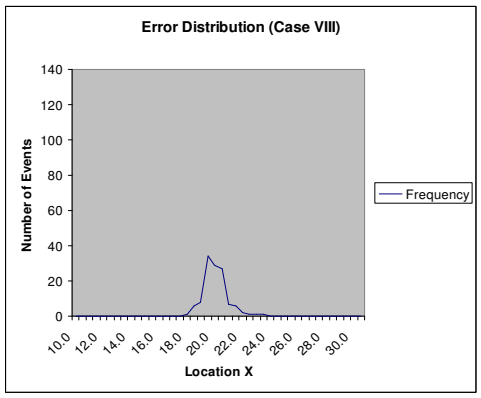

Figure 7.22 Error distribution for Case VIII.

7.2.2 Resolution at (20, 40) by Using Anger Algorithm

All the events above the threshold are included in the calculations. From centroid arithmetic, the calculated average X coordinate is 25.66 with a standard deviation of 1.06 and calculated Y coordinate is 34.34 with a standard deviation of 1.01. This results show the linearity distortion due to Anger algorithm. Since our algorithm is unbiased, linearity correction has to be made for Anger algorithm in order to make an appropriate comparison between the resolution performances of both algorithms. Resolution in X direction without any correction is found to be 2.49 mm.

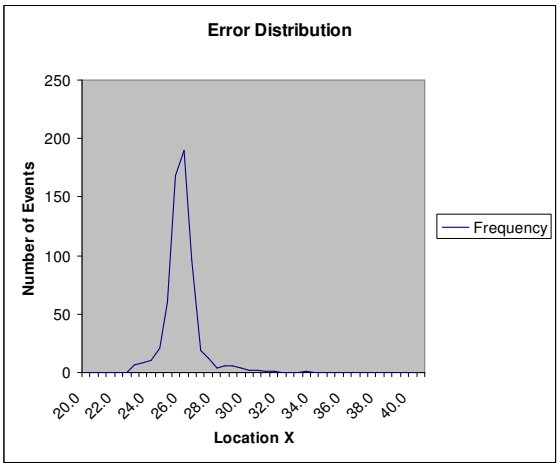

Figure 7.23 Error distribution at (20, 40) when using Anger algorithm.

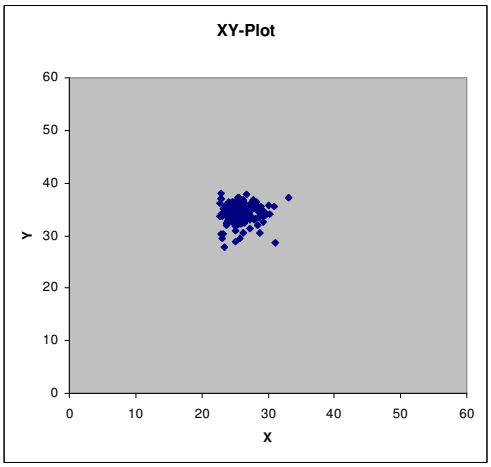

Figure 7.24 XY-plotting at (20, 40) when using Anger algorithm.

7.3 Point (22, 38)

1. Number of Gamma Rays Sent: 5000
2. Number of Hits: 1351
3. Number of Interactions (Photoelectric): 230
4. Number of Interactions (Single Interaction and Escaped): 641
5. Number of Interactions (Compton Scatter): 480
   - Number of Interactions with 5 Scatterings: 0
   - Number of Interactions with 4 Scatterings: 7
   - Number of Interactions with 3 Scatterings: 26
   - Number of Interactions with 2 Scatterings: 136
   - Number of Interactions with 1 Scattering: 311

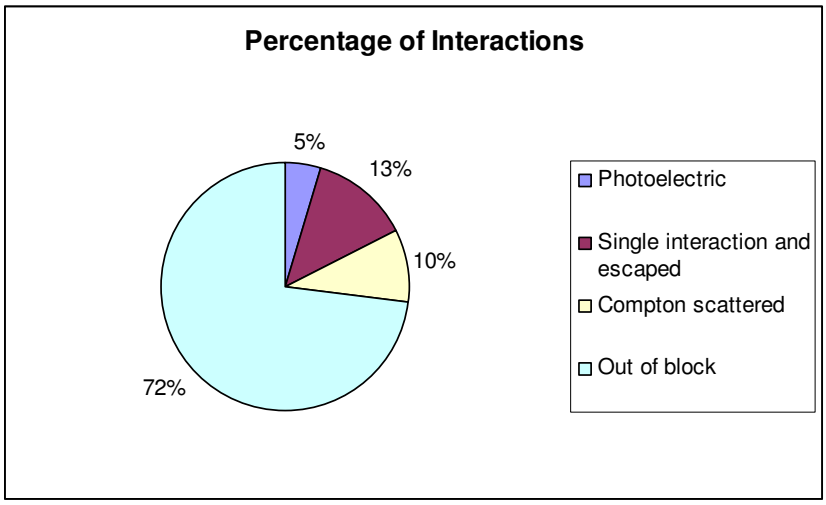

Figure 7.25 Percentage of interactions at (22, 38).

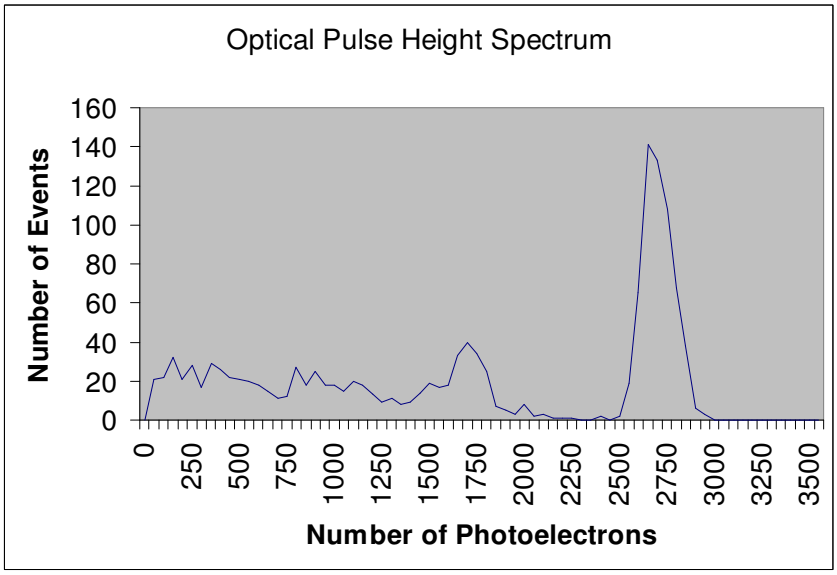

Figure 7.26 Optical pulse height spectrum at (22, 38).

6. Number of Interactions Above Energy Threshold (2400): 583
7. Average PMT Output Above Energy Threshold (Peak) : 2681
8. Standard Deviation : 80
9. FWHM : 188
10. Energy Resolution : 7.01%

7.3.1 Resolutions at (22, 38) by Using Our Algorithm

7.3.1.1 Case I (Proximity value = 5000) When Proximity value is set to 5000, 460 events out of 583 interactions could be matched with the table. From 460 events, calculated average X coordinate is 21.76 with a standard deviation of 1.52 and calculated Y coordinate is 38.18 with a standard deviation of 1.46. Resolution in X direction is found to be 3.57 mm.

7.3.1.2 Case II (Proximity value = 4500) When Proximity value is set to 4500, 440 events out of 583 interactions could be matched with the table. From 440 events, calculated average X coordinate is 21.81 with a standard deviation of 1.42 and calculated Y coordinate is 38.18 with a standard deviation of 1.44. Resolution in X direction is found to be 3.34 mm.

7.3.1.3 Case III (Proximity value = 4000) When Proximity value is set to 4000, 412 events out of 583 interactions could be matched with the table. From 412 events, calculated average X coordinate is 21.82 with a standard deviation of 1.34 and calculated Y coordinate is 38.16 with a standard deviation of 1.42. Resolution in X direction is found to be 3.15 mm.

7.3.1.4 Case IV (Proximity value = 3500) When Proximity value is set to 3500, 382 events out of 583 interactions could be matched with the table. From 382 events, calculated average X coordinate is 21.86 with a standard deviation of 1.25 and calculated Y coordinate is 38.15 with a standard deviation of 1.33. Resolution in X direction is found to be 2.94 mm.

7.3.1.5 Case V (Proximity value = 3000) When Proximity value is set to 3000, 344 events out of 583 interactions could be matched with the table. From 344 events, calculated average X coordinate is 21.91 with a standard deviation of 1.12 and calculated Y

coordinate is 38.15 with a standard deviation of 1.29. Resolution in X direction is found to be 2.63 mm.

7.3.1.6 Case VI (Proximity value = 2500) When Proximity value is set to 2500, 307 events out of 583 interactions could be matched with the table. From 307 events, calculated average X coordinate is 21.96 with a standard deviation of 0.96 and calculated Y coordinate is 38.10 with a standard deviation of 1.25. Resolution in X direction is found to be 2.26 mm.

7.3.1.7 Case VII (Proximity value = 2000) When Proximity value is set to 2000, 227 events out of 583 interactions could be matched with the table. From 227 events, calculated average X coordinate is 21.98 with a standard deviation of 0.99 and calculated Y coordinate is 38.06 with a standard deviation of 1.21. Resolution in X direction is found to be 2.32 mm.

7.3.1.8 Case VIII (Proximity value = 1500) When Proximity value is set to 1500, 127 events out of 583 interactions could be matched with the table. From 127 events, calculated average X coordinate is 22.02 with a standard deviation of 0.86 and calculated Y coordinate is 38.17 with a standard deviation of 0.99. Resolution in X direction is found to be 2.03 mm.

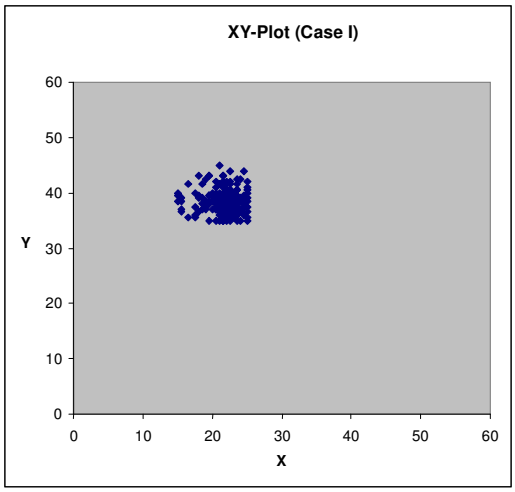

Figure 7.27 XY-plotting of Case I.

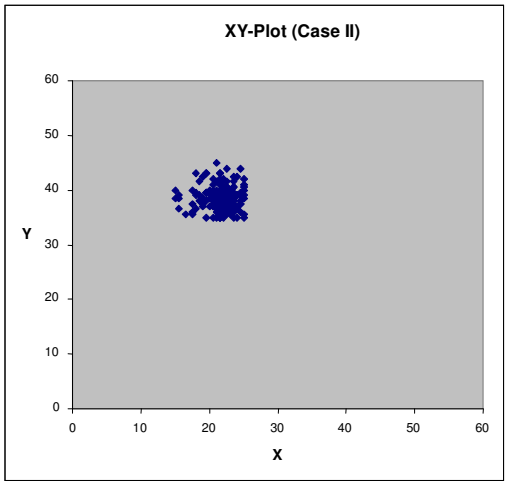

Figure 7.28 XY-plotting of Case II.

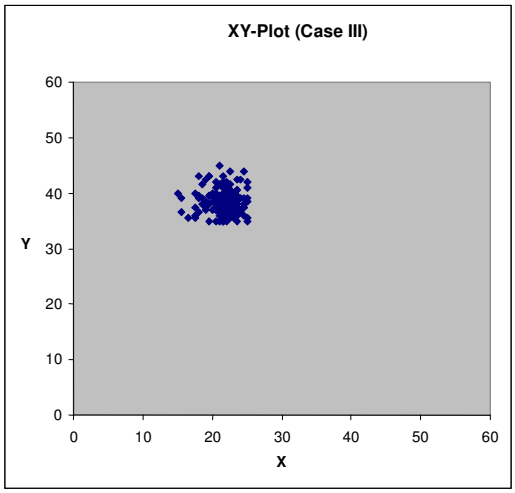

Figure 7.29 XY-plotting of Case III.

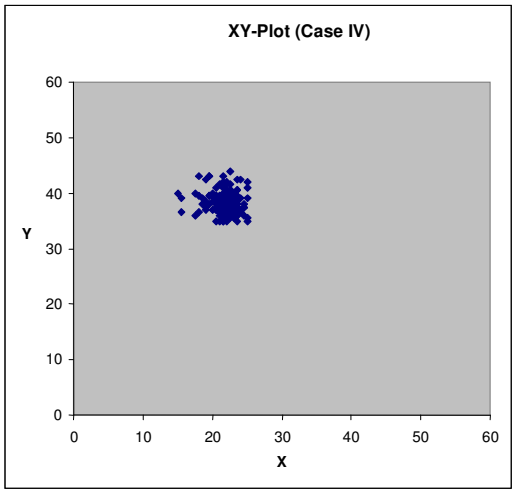

Figure 7.30 XY-plotting of Case IV.

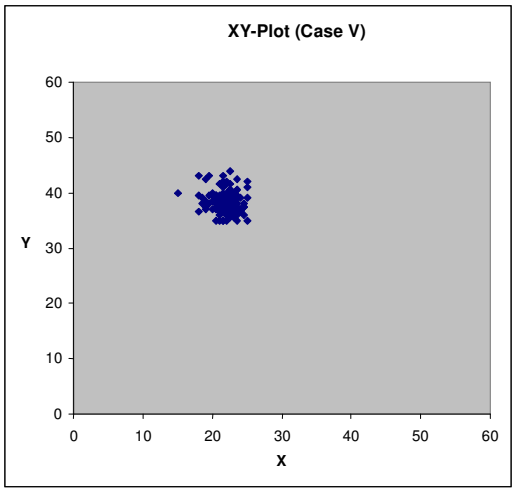

Figure 7.31 XY-plotting of Case V.

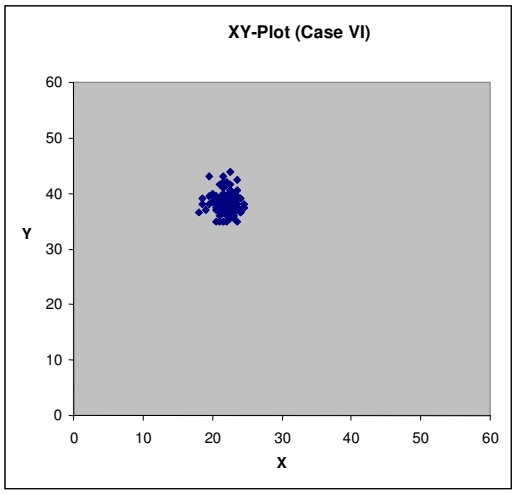

Figure 7.32 XY-plotting of Case VI.

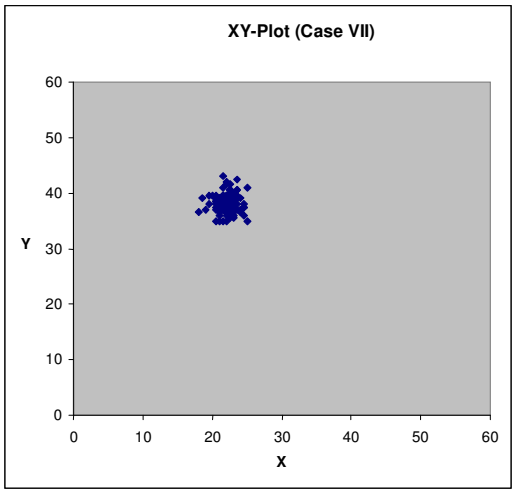

Figure 7.33 XY-plotting of Case VII.

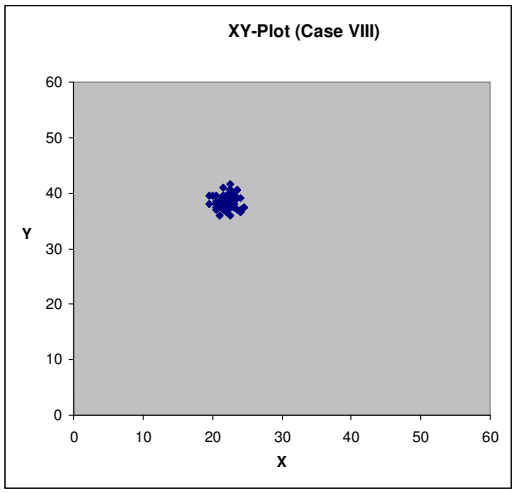

Figure 7.34 XY-plotting of Case VIII.

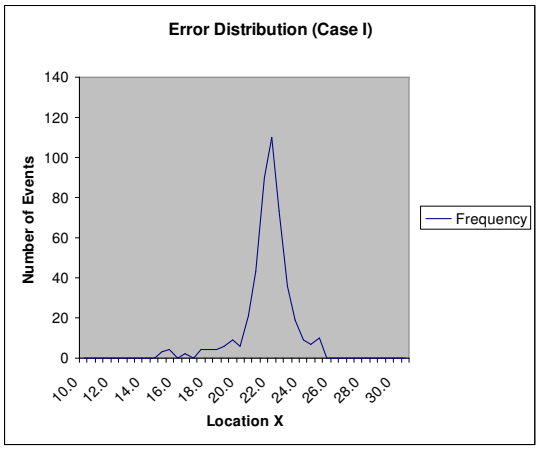

Figure 7.35 Error distribution for Case I.

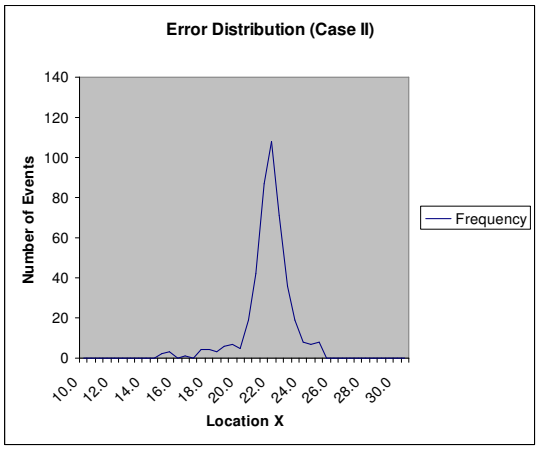

Figure 7.36 Error distribution for Case II.

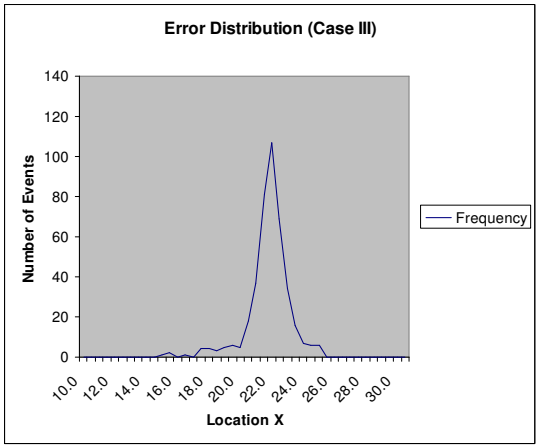

Figure 7.37 Error distribution for Case III.

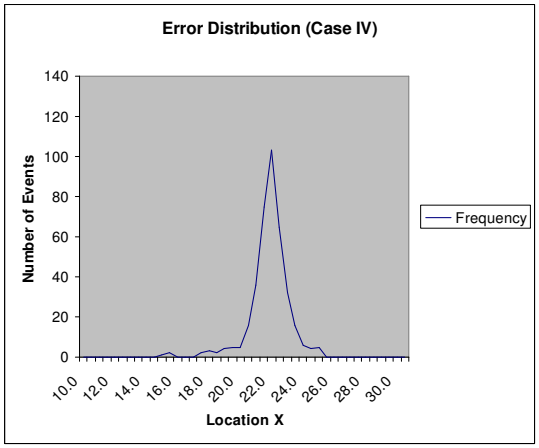

Figure 7.38 Error distribution for Case IV.

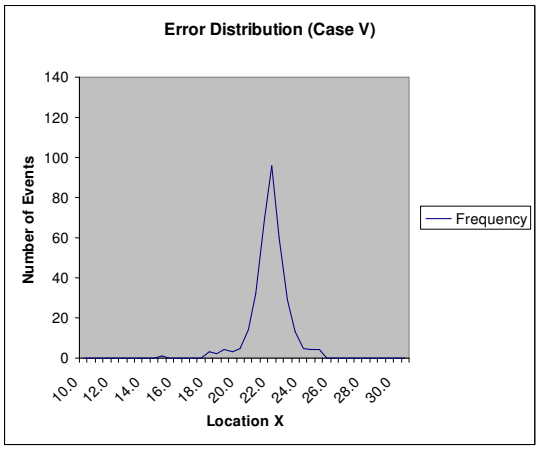

Figure 7.39 Error distribution for Case V.

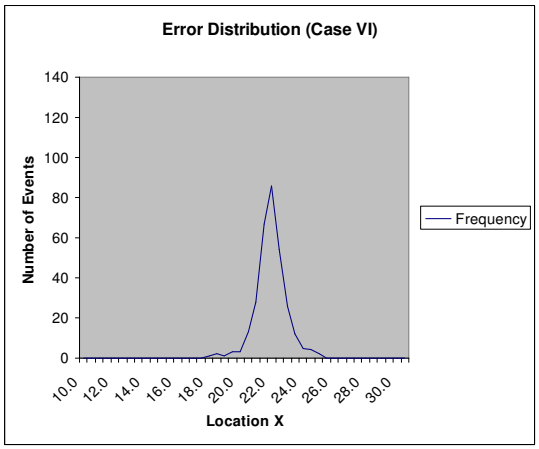

Figure 7.40 Error distribution for Case VI.

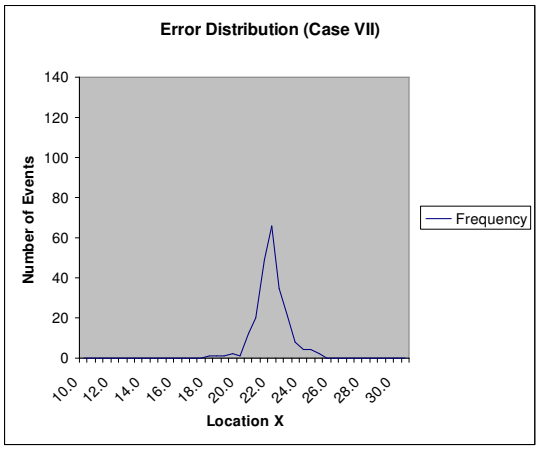

Figure 7.41 Error distribution for Case VII.

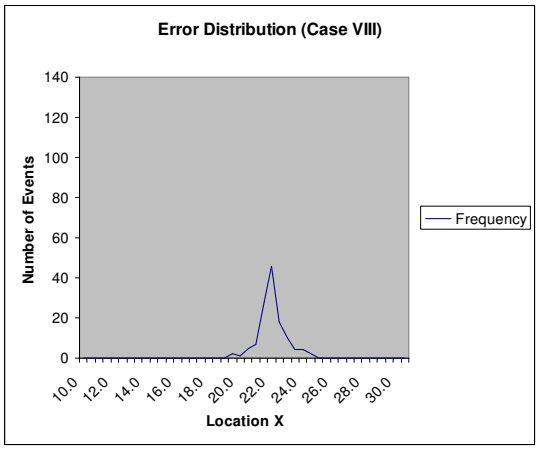

Figure 7.42 Error distribution for Case VIII.

7.3.2 Resolution at (22, 38) by Using Anger Algorithm

All the events above the threshold are included in the calculations. From centroid arithmetic, the calculated average X coordinate is 26.36 with a standard deviation of 1.06 and calculated Y coordinate is 33.67 with a standard deviation of 1.04. On the other hand the real coordinate is (22, 38), there is about 4.5 mm bias. Resolution in X direction without any correction is found to be 2.48. Anger algorithm compresses the events to the center, therefore distant two events become closer than they are in the real case.

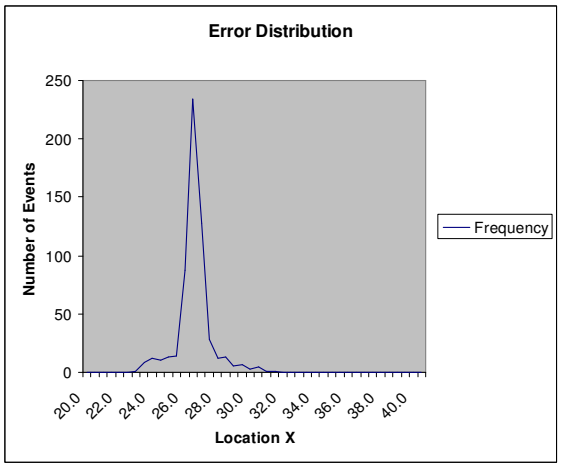

Figure 7.43 Error distribution at (22, 38) when using Anger algorithm.

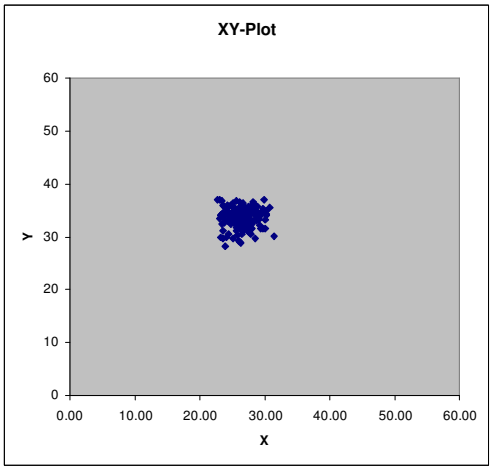

Figure 7.44 XY-plotting at (22, 38) when using Anger algorithm.

7.4 Effect of Depth of Interaction When Using Anger Algorithm

We also tried to show how the DOI effects the location estimation when using Anger algorithm even the gamma ray was coming perpendicular to the crystal surface.

We simulated 500 photoelectric interactions at 3 different depths (at 0.1 mm, at 5.0 mm, and at 9.9 mm) with same X and Y coordinate inside our 10mm crystal. We estimate the location of interaction by using Anger algorithm. We performed the simulation at two locations [(20, 40) and (30, 30)]. First location could be anywhere away from the center, and the second location was chosen to check the simulation results because that point is the center of the crystal and we know from symmetry that the distribution of light photons will be almost equal in all directions depending on the DOI.

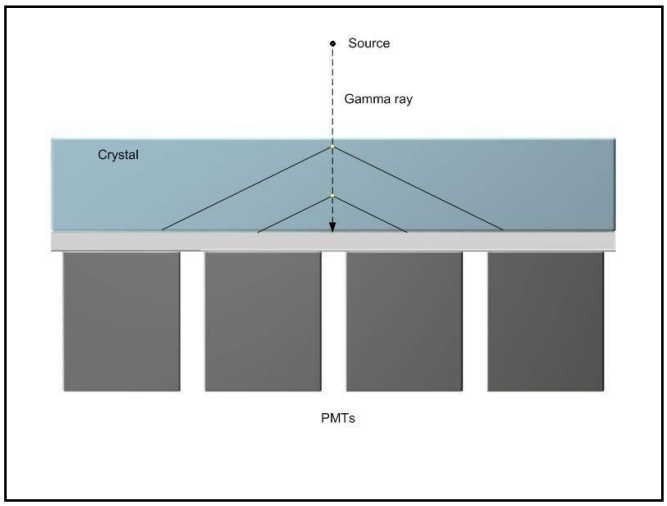

Figure 7.45 Effect of thickness and depth of interaction

7.4.1 Results at (20, 40)

When z = 0.1 mm, the average X coordinate is calculated to be 26.07 with Anger algorithm. For z = 5.0 mm, the average X coordinate becomes 25.65 and at z = 9.9 mm, the average X coordinate is found to be 25.05. There is 1 mm difference in the estimation of X coordinate between the shallow point of interaction and the deep point of interaction. Moreover, these results are biased, when corrected the difference will be more than 1 mm.

7.4.2 Results at (30, 30)

When z = 0.1 mm, the average X coordinate is calculated to be 29.98 with Anger algorithm. For z = 5.0 mm, the average X coordinate becomes 30.01 and at z = 9.9 mm, the average X coordinate is found to be 30.02. Here, there is no difference in the estimation of X coordinate between the shallow point of interaction and the deep point of interaction due to symmetry in the light photon distribution over the PMTs.

8. CONCLUSION

In our study, we designed a small detector which is suitable for PEM. Using Monte Carlo simulation we obtained reasonable resolutions by employing a nearest neighbor algorithm and utilizing a lookup table. We showed that Anger algorithm is not very suitable when dealing with small detectors and needs secondary corrections in order to be used meaningfully. The validity of centroid calculation depends on the functional shape of light distribution received by PMTs. Actual light distribution is far from Gaussian function which rapidly decreases as the point moves away from its mean; instead it is close to Couchy's distribution that decreases slowly. Therefore, centroid calculation causes bias in the estimation that result in the non-linearity and the degradation of spatial resolution.

These results also prove that the DOI effects the location estimation and therefore the resolution when using Anger algorithm. Our results indicated that using lookup table can be an alternative algorithm and it can provide a better resolution. It is obvious that as the proximity value gets smaller the resolution improves but on the contrary the number of events matched also decreases. This affects sensitivity because some interactions which are above the energy threshold are no longer contributing.

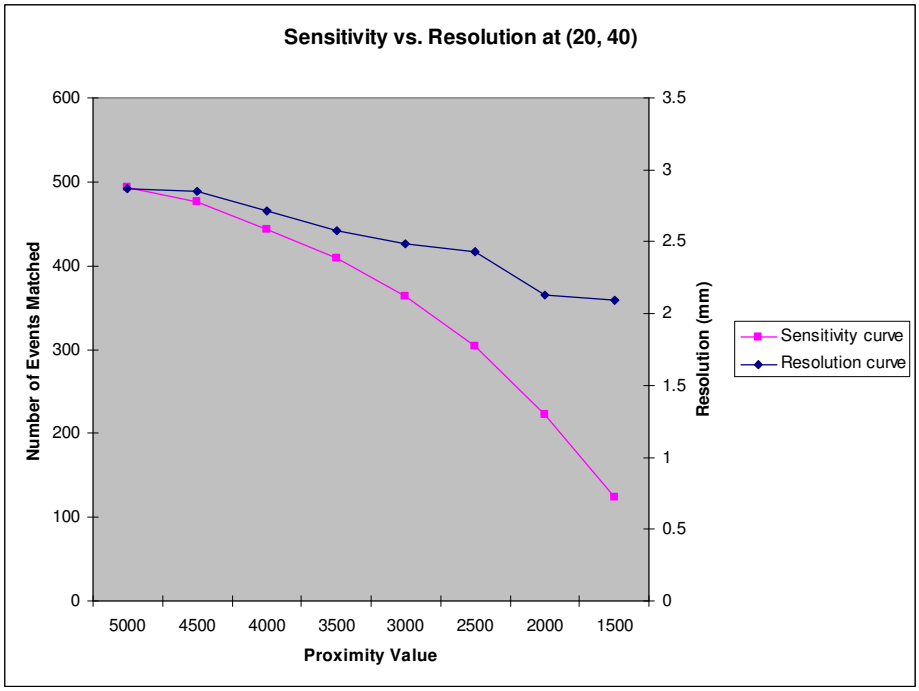

Figure 8.1 Graph above shows the change in Resolution vs. Sensitivity with respect to Proximity value at (20, 40).

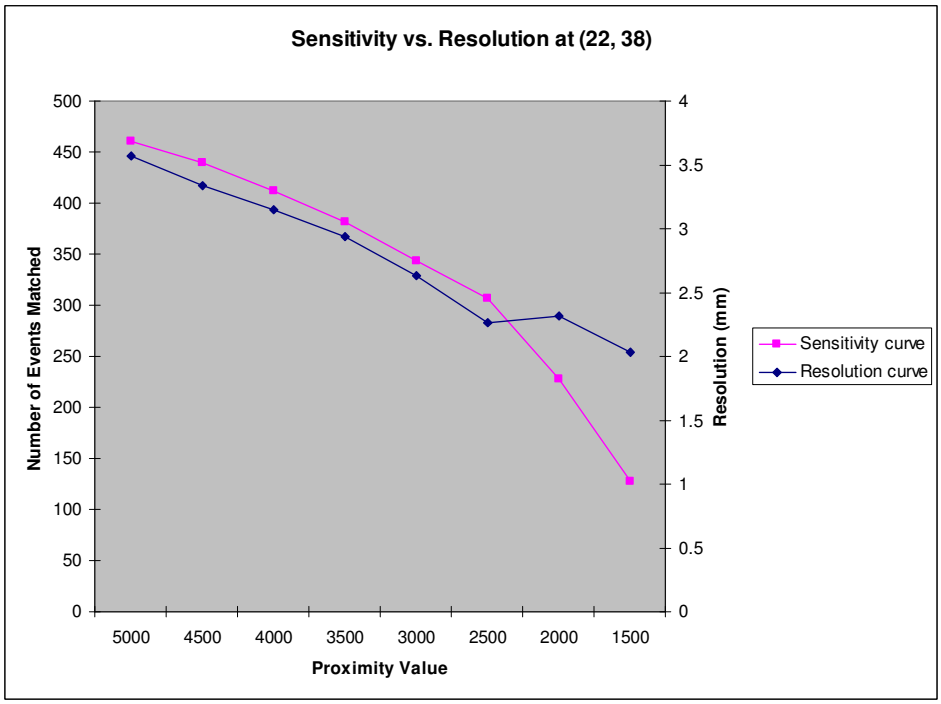

Figure 8.2 Graphs above shows the change in Resolution vs. Sensitivity with respect to Proximity value at (22, 38).

At this point we need to make an assumption to find an optimum value for the proximity value. From the simulation results at three different points in our detector configuration, we see that number of photoelectric events form only 5% of the total interactions and only 40 % of the interactions above the energy threshold are photoelectric. Since we do not want any Compton interactions contributing, the optimum value for the proximity value is the case where 60 % of the interactions above the energy threshold are eliminated. For the detector configuration we proposed, this case is obtained when the proximity value is between 2500 and 2000. In order to be on the safe side optimum value for the proximity value is recommended to be 2500. Once again if we check the resolutions, it is obvious that resolution is improved with our algorithm. It is better than the best resolution that can be obtained by Anger which is in the center of the crystal.

Table 8.1
Resolutions found in 3 locations.

Algorithm	Location	Resolution (mm)
Anger	(30, 30)	2.96
Nearest Neighbor @2500	(22, 38)	2.43
Nearest Neighbor @2500	(20, 40)	2.26

Our study only includes a specific region inside the crystal therefore the resolution can vary in different regions; the areas over a PMT can display better resolutions than areas over dead space. For future work, the whole crystal can be included in the lookup table and other algorithms especially the ones based on artificial neural networks can be investigated. Furthermore, one needs to assess the accuracy of a simulation based lookup table for representing a physical detector.

APPENDIX A. CODES FOR THE SIMULATION PROGRAMS

A.1 BUILDER Configuration Code

```
start_box
boundaries 0.0 60.0 0.0 60.0 0.0 10.0
n 1.85
absorption_coeff 4000
scattering_coeff 4000
fin_xmin PAINT 0.85RC
fin_xmax PAINT 0.85RC
fin_ymin PAINT 0.85RC
fin_ymax PAINT 0.85RC
fin_zmin POLISH
fin_zmax GROUND 0.95RC
end_box

***# PMT1 #***
# start_round_pmt #
start_square_pmt
backwards FALSE
position 15.0 60.0 0.0
width 15
height 15
window_depth 2
holder_depth 0.1
pmt_depth 1.5
pmt_gap 0.1
# eff_radius 7.5 #
pmt_width 12.5
pmt_height 12.5
offset 0.0 0.0
dummy_plane_depth 100

dummy_fin PAINT 0.85RC

window_side_fins
PAINT 0.85RC
PSEUDO
PSEUDO
PAINT 0.85RC

holder_side_fins
PAINT 0.85RC
PSEUDO
PSEUDO
PAINT 0.85RC

faceplate_interface_fin POLISH

# cyl_wall_fin POLISH 1.0RC #

pmt_sides_fin POLISH 1.0RC
window_n 1.52
```

```
window_ad -1.0
window_sd -1.0
holder_n 1.52
holder_ad -1.0
holder_sd -1.0
pmt_n 1.52
pmt_ad -1.0
pmt_sd -1.0
detect_n -1.0

connecting_fin POLISH

holder_hole_fin PAINT 0.85RC

holder_bottom_fin PAINT 0.85RC

# window_pl #
# window_wsd #
# window_wsl #
# holder_pl #
# holder_wsd #
# holder_wsl #
# pmt_pl #
# pmt_wsd #
# pmt_wsl #
end_square_pmt
# end_round_pmt #

***# PMT2 #***
# start_round_pmt #
start_square_pmt
backwards FALSE
position 30.0 60.0 0.0
width 15
height 15
window_depth 2
holder_depth 0.1
pmt_depth 1.5
pmt_gap 0.1
# eff_radius 7.5 #
pmt_width 12.5
pmt_height 12.5
offset 0.0 0.0
dummy_plane_depth 100

dummy_fin PAINT 0.85RC

window_side_fins
PSEUDO
PSEUDO
PSEUDO
PAINT 0.85RC


holder_side_fins
PSEUDO
PSEUDO
PSEUDO
PAINT 0.85RC
```

```
faceplate_interface_fin POLISH

# cyl_wall_fin POLISH 1.0RC #

pmt_sides_fin POLISH 1.0RC
window_n 1.52
window_ad -1.0
window_sd -1.0
holder_n 1.52
holder_ad -1.0
holder_sd -1.0
pmt_n 1.52
pmt_ad -1.0
pmt_sd -1.0
detect_n -1.0

connecting_fin POLISH

holder_hole_fin PAINT 0.85RC

holder_bottom_fin PAINT 0.85RC

# window_pl #
# window_wsd #
# window_wsl #
# holder_pl #
# holder_wsd #
# holder_wsl #
# pmt_pl #
# pmt_wsd #
# pmt_wsl #
end_square_pmt
# end_round_pmt #

***# PMT3 #***
# start_round_pmt #
start_square_pmt
backwards FALSE
position 45.0 60.0 0.0
width 15
height 15
window_depth 2
holder_depth 0.1
pmt_depth 1.5
pmt_gap 0.1
# eff_radius 7.5 #
pmt_width 12.5
pmt_height 12.5
offset 0.0 0.0
dummy_plane_depth 100

dummy_fin PAINT 0.85RC

window_side_fins
PSEUDO
PSEUDO
PSEUDO
PAINT 0.85RC
```

```
holder_side_fins
PSEUDO
PSEUDO
PSEUDO
PAINT 0.85RC

faceplate_interface_fin POLISH

# cyl_wall_fin POLISH 1.0RC #

pmt_sides_fin POLISH 1.0RC
window_n 1.52
window_ad -1.0
window_sd -1.0
holder_n 1.52
holder_ad -1.0
holder_sd -1.0
pmt_n 1.52
pmt_ad -1.0
pmt_sd -1.0
detect_n -1.0

connecting_fin POLISH

holder_hole_fin PAINT 0.85RC

holder_bottom_fin PAINT 0.85RC

# window_pl #
# window_wsd #
# window_wsl #
# holder_pl #
# holder_wsd #
# holder_wsl #
# pmt_pl #
# pmt_wsd #
# pmt_wsl #
end_square_pmt
# end_round_pmt #

***# PMT4 #***
# start_round_pmt #
start_square_pmt
backwards FALSE
position 60.0 60.0 0.0
width 15
height 15
window_depth 2
holder_depth 0.1
pmt_depth 1.5
pmt_gap 0.1
# eff_radius 7.5 #
pmt_width 12.5
pmt_height 12.5
offset 0.0 0.0
dummy_plane_depth 100
```

```
dummy_fin PAINT 0.85RC

window_side_fins
PSEUDO
PAINT 0.85RC
PSEUDO
PAINT 0.85RC


holder_side_fins
PSEUDO
PAINT 0.85RC
PSEUDO
PAINT 0.85RC

faceplate_interface_fin POLISH

# cyl_wall_fin POLISH 1.0RC #

pmt_sides_fin POLISH 1.0RC
window_n 1.52
window_ad -1.0
window_sd -1.0
holder_n 1.52
holder_ad -1.0
holder_sd -1.0
pmt_n 1.52
pmt_ad -1.0
pmt_sd -1.0
detect_n -1.0

connecting_fin POLISH

holder_hole_fin PAINT 0.85RC

holder_bottom_fin PAINT 0.85RC

# window_pl #
# window_wsd #
# window_wsl #
# holder_pl #
# holder_wsd #
# holder_wsl #
# pmt_pl #
# pmt_wsd #
# pmt_wsl #
end_square_pmt
# end_round_pmt #

***# PMT5 #***
# start_round_pmt #
start_square_pmt
backwards FALSE
position 15.0 45.0 0.0
width 15
height 15
window_depth 2
holder_depth 0.1
pmt_depth 1.5
```

```
pmt_gap 0.1
# eff_radius 7.5 #
pmt_width 12.5
pmt_height 12.5
offset 0.0 0.0
dummy_plane_depth 100

dummy_fin PAINT 0.85RC

window_side_fins
PAINT 0.85RC
PSEUDO
PSEUDO
PSEUDO


holder_side_fins
PAINT 0.85RC
PSEUDO
PSEUDO
PSEUDO

faceplate_interface_fin POLISH

# cyl_wall_fin POLISH 1.0RC #

pmt_sides_fin POLISH 1.0RC
window_n 1.52
window_ad -1.0
window_sd -1.0
holder_n 1.52
holder_ad -1.0
holder_sd -1.0
pmt_n 1.52
pmt_ad -1.0
pmt_sd -1.0
detect_n -1.0

connecting_fin POLISH

holder_hole_fin PAINT 0.85RC

holder_bottom_fin PAINT 0.85RC

# window_pl #
# window_wsd #
# window_wsl #
# holder_pl #
# holder_wsd #
# holder_wsl #
# pmt_pl #
# pmt_wsd #
# pmt_wsl #
end_square_pmt
# end_round_pmt #

***# PMT6 #***
# start_round_pmt #
start_square_pmt
```

```
backwards FALSE
position 30.0 45.0 0.0
width 15
height 15
window_depth 2
holder_depth 0.1
pmt_depth 1.5
pmt_gap 0.1
# eff_radius 7.5 #
pmt_width 12.5
pmt_height 12.5
offset 0.0 0.0
dummy_plane_depth 100

dummy_fin PAINT 0.85RC

window_side_fins
PSEUDO
PSEUDO
PSEUDO
PSEUDO


holder_side_fins
PSEUDO
PSEUDO
PSEUDO
PSEUDO

faceplate_interface_fin POLISH

# cyl_wall_fin POLISH 1.0RC #

pmt_sides_fin POLISH 1.0RC
window_n 1.52
window_ad -1.0
window_sd -1.0
holder_n 1.52
holder_ad -1.0
holder_sd -1.0
pmt_n 1.52
pmt_ad -1.0
pmt_sd -1.0
detect_n -1.0

connecting_fin POLISH

holder_hole_fin PAINT 0.85RC

holder_bottom_fin PAINT 0.85RC

# window_pl #
# window_wsd #
# window_wsl #
# holder_pl #
# holder_wsd #
# holder_wsl #
# pmt_pl #
# pmt_wsd #
```

```
# pmt_wsl #
end_square_pmt
# end_round_pmt #

***# PMT7 #***
# start_round_pmt #
start_square_pmt
backwards FALSE
position 45.0 45.0 0.0
width 15
height 15
window_depth 2
holder_depth 0.1
pmt_depth 1.5
pmt_gap 0.1
# eff_radius 7.5 #
pmt_width 12.5
pmt_height 12.5
offset 0.0 0.0
dummy_plane_depth 100

dummy_fin PAINT 0.85RC

window_side_fins
PSEUDO
PSEUDO
PSEUDO
PSEUDO


holder_side_fins
PSEUDO
PSEUDO
PSEUDO
PSEUDO

faceplate_interface_fin POLISH

# cyl_wall_fin POLISH 1.0RC #

pmt_sides_fin POLISH 1.0RC
window_n 1.52
window_ad -1.0
window_sd -1.0
holder_n 1.52
holder_ad -1.0
holder_sd -1.0
pmt_n 1.52
pmt_ad -1.0
pmt_sd -1.0
detect_n -1.0

connecting_fin POLISH

holder_hole_fin PAINT 0.85RC

holder_bottom_fin PAINT 0.85RC

# window_pl #
```

```
# window_wsd #
# window_wsl #
# holder_pl #
# holder_wsd #
# holder_wsl #
# pmt_pl #
# pmt_wsd #
# pmt_wsl #
end_square_pmt
# end_round_pmt #

***# PMT8 #***
# start_round_pmt #
start_square_pmt
backwards FALSE
position 60.0 45.0 0.0
width 15
height 15
window_depth 2
holder_depth 0.1
pmt_depth 1.5
pmt_gap 0.1
# eff_radius 7.5 #
pmt_width 12.5
pmt_height 12.5
offset 0.0 0.0
dummy_plane_depth 100

dummy_fin PAINT 0.85RC

window_side_fins
PSEUDO
PAINT 0.85RC
PSEUDO
PSEUDO


holder_side_fins
PSEUDO
PAINT 0.85RC
PSEUDO
PSEUDO

faceplate_interface_fin POLISH

# cyl_wall_fin POLISH 1.0RC #

pmt_sides_fin POLISH 1.0RC
window_n 1.52
window_ad -1.0
window_sd -1.0
holder_n 1.52
holder_ad -1.0
holder_sd -1.0
pmt_n 1.52
pmt_ad -1.0
pmt_sd -1.0
detect_n -1.0
```

```
connecting_fin POLISH

holder_hole_fin PAINT 0.85RC

holder_bottom_fin PAINT 0.85RC

# window_pl #
# window_wsd #
# window_wsl #
# holder_pl #
# holder_wsd #
# holder_wsl #
# pmt_pl #
# pmt_wsd #
# pmt_wsl #
end_square_pmt
# end_round_pmt #

***# PMT9 #***
# start_round_pmt #
start_square_pmt
backwards FALSE
position 15.0 30.0 0.0
width 15
height 15
window_depth 2
holder_depth 0.1
pmt_depth 1.5
pmt_gap 0.1
# eff_radius 7.5 #
pmt_width 12.5
pmt_height 12.5
offset 0.0 0.0
dummy_plane_depth 100

dummy_fin PAINT 0.85RC

window_side_fins
PAINT 0.85RC
PSEUDO
PSEUDO
PSEUDO


holder_side_fins
PAINT 0.85RC
PSEUDO
PSEUDO
PSEUDO

faceplate_interface_fin POLISH

# cyl_wall_fin POLISH 1.0RC #

pmt_sides_fin POLISH 1.0RC
window_n 1.52
window_ad -1.0
window_sd -1.0
holder_n 1.52
```

```
holder_ad -1.0
holder_sd -1.0
pmt_n 1.52
pmt_ad -1.0
pmt_sd -1.0
detect_n -1.0

connecting_fin POLISH

holder_hole_fin PAINT 0.85RC

holder_bottom_fin PAINT 0.85RC

# window_pl #
# window_wsd #
# window_wsl #
# holder_pl #
# holder_wsd #
# holder_wsl #
# pmt_pl #
# pmt_wsd #
# pmt_wsl #
end_square_pmt
# end_round_pmt #

***# PMT10 #***
# start_round_pmt #
start_square_pmt
backwards FALSE
position 30.0 30.0 0.0
width 15
height 15
window_depth 2
holder_depth 0.1
pmt_depth 1.5
pmt_gap 0.1
# eff_radius 7.5 #
pmt_width 12.5
pmt_height 12.5
offset 0.0 0.0
dummy_plane_depth 100

dummy_fin PAINT 0.85RC

window_side_fins
PSEUDO
PSEUDO
PSEUDO
PSEUDO


holder_side_fins
PSEUDO
PSEUDO
PSEUDO
PSEUDO

faceplate_interface_fin POLISH
```

```
# cyl_wall_fin POLISH 1.0RC #

pmt_sides_fin POLISH 1.0RC
window_n 1.52
window_ad -1.0
window_sd -1.0
holder_n 1.52
holder_ad -1.0
holder_sd -1.0
pmt_n 1.52
pmt_ad -1.0
pmt_sd -1.0
detect_n -1.0

connecting_fin POLISH

holder_hole_fin PAINT 0.85RC

holder_bottom_fin PAINT 0.85RC

# window_pl #
# window_wsd #
# window_wsl #
# holder_pl #
# holder_wsd #
# holder_wsl #
# pmt_pl #
# pmt_wsd #
# pmt_wsl #
end_square_pmt
# end_round_pmt #

***# PMT11 #***
# start_round_pmt #
start_square_pmt
backwards FALSE
position 45.0 30.0 0.0
width 15
height 15
window_depth 2
holder_depth 0.1
pmt_depth 1.5
pmt_gap 0.1
# eff_radius 7.5 #
pmt_width 12.5
pmt_height 12.5
offset 0.0 0.0
dummy_plane_depth 100

dummy_fin PAINT 0.85RC

window_side_fins
PSEUDO
PSEUDO
PSEUDO
PSEUDO


holder_side_fins
```

```
PSEUDO
PSEUDO
PSEUDO
PSEUDO

faceplate_interface_fin POLISH

# cyl_wall_fin POLISH 1.0RC #

pmt_sides_fin POLISH 1.0RC
window_n 1.52
window_ad -1.0
window_sd -1.0
holder_n 1.52
holder_ad -1.0
holder_sd -1.0
pmt_n 1.52
pmt_ad -1.0
pmt_sd -1.0
detect_n -1.0

connecting_fin POLISH

holder_hole_fin PAINT 0.85RC

holder_bottom_fin PAINT 0.85RC

# window_pl #
# window_wsd #
# window_wsl #
# holder_pl #
# holder_wsd #
# holder_wsl #
# pmt_pl #
# pmt_wsd #
# pmt_wsl #
end_square_pmt
# end_round_pmt #

***# PMT12 #***
# start_round_pmt #
start_square_pmt
backwards FALSE
position 60.0 30.0 0.0
width 15
height 15
window_depth 2
holder_depth 0.1
pmt_depth 1.5
pmt_gap 0.1
# eff_radius 7.5 #
pmt_width 12.5
pmt_height 12.5
offset 0.0 0.0
dummy_plane_depth 100

dummy_fin PAINT 0.85RC

window_side_fins
```

```
PSEUDO
PAINT 0.85RC
PSEUDO
PSEUDO


holder_side_fins
PSEUDO
PAINT 0.85RC
PSEUDO
PSEUDO

faceplate_interface_fin POLISH

# cyl_wall_fin POLISH 1.0RC #

pmt_sides_fin POLISH 1.0RC
window_n 1.52
window_ad -1.0
window_sd -1.0
holder_n 1.52
holder_ad -1.0
holder_sd -1.0
pmt_n 1.52
pmt_ad -1.0
pmt_sd -1.0
detect_n -1.0

connecting_fin POLISH

holder_hole_fin PAINT 0.85RC

holder_bottom_fin PAINT 0.85RC

# window_pl #
# window_wsd #
# window_wsl #
# holder_pl #
# holder_wsd #
# holder_wsl #
# pmt_pl #
# pmt_wsd #
# pmt_wsl #
end_square_pmt
# end_round_pmt #


***# PMT13 #***
# start_round_pmt #
start_square_pmt
backwards FALSE
position 15.0 15.0 0.0
width 15
height 15
window_depth 2
holder_depth 0.1
pmt_depth 1.5
pmt_gap 0.1
# eff_radius 7.5 #
```

```
pmt_width 12.5
pmt_height 12.5
offset 0.0 0.0
dummy_plane_depth 100

dummy_fin PAINT 0.85RC

window_side_fins
PAINT 0.85RC
PSEUDO
PAINT 0.85RC
PSEUDO



holder_side_fins
PAINT 0.85RC
PSEUDO
PAINT 0.85RC
PSEUDO

faceplate_interface_fin POLISH

# cyl_wall_fin POLISH 1.0RC #

pmt_sides_fin POLISH 1.0RC
window_n 1.52
window_ad -1.0
window_sd -1.0
holder_n 1.52
holder_ad -1.0
holder_sd -1.0
pmt_n 1.52
pmt_ad -1.0
pmt_sd -1.0
detect_n -1.0

connecting_fin POLISH

holder_hole_fin PAINT 0.85RC

holder_bottom_fin PAINT 0.85RC

# window_pl #
# window_wsd #
# window_wsl #
# holder_pl #
# holder_wsd #
# holder_wsl #
# pmt_pl #
# pmt_wsd #
# pmt_wsl #
end_square_pmt
# end_round_pmt #

***# PMT14 #***
# start_round_pmt #
start_square_pmt
backwards FALSE
```

```
position 30.0 15.0 0.0
width 15
height 15
window_depth 2
holder_depth 0.1
pmt_depth 1.5
pmt_gap 0.1
# eff_radius 7.5 #
pmt_width 12.5
pmt_height 12.5
offset 0.0 0.0
dummy_plane_depth 100

dummy_fin PAINT 0.85RC

window_side_fins
PSEUDO
PSEUDO
PAINT 0.85RC
PSEUDO


holder_side_fins
PSEUDO
PSEUDO
PAINT 0.85RC
PSEUDO

faceplate_interface_fin POLISH

# cyl_wall_fin POLISH 1.0RC #

pmt_sides_fin POLISH 1.0RC
window_n 1.52
window_ad -1.0
window_sd -1.0
holder_n 1.52
holder_ad -1.0
holder_sd -1.0
pmt_n 1.52
pmt_ad -1.0
pmt_sd -1.0
detect_n -1.0

connecting_fin POLISH

holder_hole_fin PAINT 0.85RC

holder_bottom_fin PAINT 0.85RC

# window_pl #
# window_wsd #
# window_wsl #
# holder_pl #
# holder_wsd #
# holder_wsl #
# pmt_pl #
# pmt_wsd #
# pmt_wsl #
```

```
end_square_pmt
# end_round_pmt #

***# PMT15 #***
# start_round_pmt #
start_square_pmt
backwards FALSE
position 45.0 15.0 0.0
width 15
height 15
window_depth 2
holder_depth 0.1
pmt_depth 1.5
pmt_gap 0.1
# eff_radius 7.5 #
pmt_width 12.5
pmt_height 12.5
offset 0.0 0.0
dummy_plane_depth 100

dummy_fin PAINT 0.85RC

window_side_fins
PSEUDO
PSEUDO
PAINT 0.85RC
PSEUDO


holder_side_fins
PSEUDO
PSEUDO
PAINT 0.85RC
PSEUDO

faceplate_interface_fin POLISH

# cyl_wall_fin POLISH 1.0RC #

pmt_sides_fin POLISH 1.0RC
window_n 1.52
window_ad -1.0
window_sd -1.0
holder_n 1.52
holder_ad -1.0
holder_sd -1.0
pmt_n 1.52
pmt_ad -1.0
pmt_sd -1.0
detect_n -1.0

connecting_fin POLISH

holder_hole_fin PAINT 0.85RC

holder_bottom_fin PAINT 0.85RC

# window_pl #
# window_wsd #
```

```
# window_wsl #
# holder_pl #
# holder_wsd #
# holder_wsl #
# pmt_pl #
# pmt_wsd #
# pmt_wsl #
end_square_pmt
# end_round_pmt #

***# PMT16 #***
# start_round_pmt #
start_square_pmt
backwards FALSE
position 60.0 15.0 0.0
width 15
height 15
window_depth 2
holder_depth 0.1
pmt_depth 1.5
pmt_gap 0.1
# eff_radius 7.5 #
pmt_width 12.5
pmt_height 12.5
offset 0.0 0.0
dummy_plane_depth 100

dummy_fin PAINT 0.85RC

window_side_fins
PSEUDO
PAINT 0.85RC
PAINT 0.85RC
PSEUDO


holder_side_fins
PSEUDO
PAINT 0.85RC
PAINT 0.85RC
PSEUDO

faceplate_interface_fin POLISH

# cyl_wall_fin POLISH 1.0RC #

pmt_sides_fin POLISH 1.0RC
window_n 1.52
window_ad -1.0
window_sd -1.0
holder_n 1.52
holder_ad -1.0
holder_sd -1.0
pmt_n 1.52
pmt_ad -1.0
pmt_sd -1.0
detect_n -1.0

connecting_fin POLISH
```

```
holder_hole_fin PAINT 0.85RC

holder_bottom_fin PAINT 0.85RC

# window_pl #
# window_wsd #
# window_wsl #
# holder_pl #
# holder_wsd #
# holder_wsl #
# pmt_pl #
# pmt_wsd #
# pmt_wsl #
end_square_pmt
# end_round_pmt #

max_life 10000
max_seed 5000
flag fast_qe 0
flag verbose 0
vms FALSE
ignore_outside_events FALSE
n_gen 1.85
absorption_coeff_gen 4000
scattering_coeff_gen 4000
```

A.2 GRIT Configuration Code

```
# -------------------- Source-Block Settings -------------------- #

units mm
source_type POINT
source_location 0.0000 0.0000 60.0000
source_radius 0.0000
initial_gamma_energy 511.0000
cutoff_energy 15.0000
source_collimation OFF
collimation_type NONE
collimator_position  0.0000 -9.9000 -9.9000
collimator_distance 0.0000
collimator_thickness 0.0000
line_collimation_alignment y
line_collimation_width 0.0000
pencil_collimation_radius 0.0000
block_plane NOT_SET


# -------------------- Block geometry -------------------- #

scintillator NAI
n_scintillator 1.8500
absorption_coeff 4000.0000
scattering_coeff 4000.0000
block_finish GROUND
```

```
rc_scintillator 1.0000
block_dim_axial 60.0000
block_dim_transaxial 60.0000
block_dim_thickness 10.0000
can_thickness 0.5000
cut_width 0.0000
num_crystals 1
num_cuts 0
cut_depths
n_detector_window 1.5200
rc_detector_window 1.0000
rc_pmt_holder 1.0000
det_centre_shift_x 0.0000
det_centre_shift_y 0.0000
det_window_z -2.0000
det_surf_z -3.5000
holder_bottom_z -2.1000
pmt_radius 12.5000
pmt_gap 0.1000
qe 0.2000
surf_fin_qe_f_factor 1.0000


# -------------------- GRIT Controls -------------------- #

vms FALSE
photoe_fmt "$x%10.4f$ $y%10.4f$ $z%10.4f$ $q%4i$ $h%2i$ $u%10.4f$ $v%10.4f$\n"
compton_fmt "$x%10.4f$ $y%10.4f$ $z%10.4f$ $q%4i$ $h%2i$ $u%10.4f$ $v%10.4f$\n"
low_e_fmt "$x%10.4f$ $y%10.4f$ $z%10.4f$ $q%4i$ $h%2i$ $u%10.4f$ $v%10.4f$\n"
left_block_fmt ""
gamma_start_fmt ""
gamma_end_fmt ""
gamma_file interactions.src
slice_file none
settings_file grt.f

rseed 16352 29781 46644
max_life 10000
max_rand_gen_seed 5000
run 5000

#-----------------------------------------------------------#
```

APPENDIX B. CODES FOR THE ADDITIONAL MODULES

B.1 Photon Correction Program (Written in C++)

```
#define _CRT_SECURE_NO_DEPRECATE

#include <stdio.h>
#include <MALLOC.H>
#include <math.h>
#include <iostream>
#include <stdlib.h>
#include <time.h>

#include "randomc.h"                // define classes for random number generators
#include "userintf.cpp"             // define system specific user interface

// define which random number generator to base random library on:

#include "mersenne.cpp"             // members of class TRandomMersenne
#define RANDOM_GENERATOR TRandomMersenne

// or:
//#include "ranrotw.cpp"            // members of class TRanrotWGenerator
//#define RANDOM_GENERATOR TRanrotWGenerator

// or:
//#include "mother.cpp"             // members of class TRanrotWGenerator
//#define RANDOM_GENERATOR TRandomMotherOfAll

#include "stocc.h"                  // define random library classes
#include "stoc1.cpp"


void FindFirstToRead(FILE *);

int main()
{
 char c;
 int iLines,i,j;
 FILE *fp,*f2;
 double **matrix;
 iLines = 0;

 int32 seed = time(0);              // random seed
 StochasticLib1 sto(seed);

 fp=fopen("interactions.txt","r");
 f2=fopen("inputsrc.txt","w");

 while(!feof(fp))
 {
        fscanf(fp,"%c",&c);
        if(c=='\n')
          iLines++;
```

```
};

fseek(fp,0,SEEK_SET);
matrix = (double **)malloc(sizeof(double*)*iLines);

for(i=0;i<iLines;i++)
        matrix[i] = (double *)malloc(sizeof(double)*7);

FindFirstToRead(fp);

for(i=0;i<iLines;i++){
        for(j=0;j<7;j++){
                FindFirstToRead(fp);
                matrix[i][j]=0.0;
                fscanf(fp,"%lf",&matrix[i][j]);
        }
}

for(i=0;i<iLines;i++){


         if(matrix[i][3] <=110)
                  matrix[i][3] = 0;
         else if(matrix[i][3] <= 167)
                  matrix[i][3] = matrix[i][3]*(0.04*(matrix[i][3]/37.3)+0.93);
         else if(matrix[i][3] <= 186)
                  matrix[i][3] = matrix[i][3]*((-0.07)*(matrix[i][3]/37.3)+1.425);
         else if(matrix[i][3] <= 223)
                  matrix[i][3] = matrix[i][3]*(0.009*(matrix[i][3]/37.3)+1.03);
         else if(matrix[i][3] <= 298)
                  matrix[i][3] = matrix[i][3]*(0.0165*(matrix[i][3]/37.3)+0.985);
         else if(matrix[i][3] <= 466)
                  matrix[i][3] = matrix[i][3]*(0.0096*(matrix[i][3]/37.3)+1.0406);
         else if(matrix[i][3] <= 652)
                  matrix[i][3] = matrix[i][3]*(0.003*(matrix[i][3]/37.3)+1.1225);
         else if(matrix[i][3] <= 932)
                  matrix[i][3] = matrix[i][3]*((-0.0015)*(matrix[i][3]/37.3)+1.2007);
         else if(matrix[i][3] <= 1268)
                  matrix[i][3] = matrix[i][3]*((-0.0027)*(matrix[i][3]/37.3)+1.2307);
         else if(matrix[i][3] <= 1865)
                  matrix[i][3] = matrix[i][3]*(0.0012*(matrix[i][3]/37.3)+1.0975);
         else if(matrix[i][3] <= 3730)
                  matrix[i][3] = matrix[i][3]*((-0.001)*(matrix[i][3]/37.3)+1.21);
         else if(matrix[i][3] <= 11190)
                  matrix[i][3] = matrix[i][3]*((-0.0004)*(matrix[i][3]/37.3)+1.1525);
         else if(matrix[i][3] <= 19097)
                  matrix[i][3] = matrix[i][3]*((-0.0001)*(matrix[i][3]/37.3)+1.0605);
         else if(matrix[i][3] <= 37300)
                  matrix[i][3] = matrix[i][3]*((-0.000033)*(matrix[i][3]/37.3)+1.0167);
         else matrix[i][3] = 0;


}


 for (i=0; i<iLines; i++){
                  matrix[i][3] = sto.Poisson(matrix[i][3]);
                  fprintf(f2,"%10.4lf%11.4lf%11.4lf%6.0lf%3.0lf%11.4lf%11.4lf\n",
matrix[i][0],matrix[i][1],matrix[i][2],matrix[i][3],matrix[i][4],matrix[i][5],matrix[i][6]);
```

```
            }


        printf("\n");

 fclose(fp);

 for(i=0;i<iLines;i++)
        free(matrix[i]);
 free(matrix);
 return 0;
}

void FindFirstToRead(FILE *fp)
{
 char c;
 fpos_t pos;
 fscanf(fp,"%c",&c);
 while(c==' ')
 {
        fscanf(fp,"%c",&c);
 };
 fgetpos(fp,&pos);
 pos--;
 fsetpos(fp,&pos);
 return;
        };
```

B.2 PMT Correction Program (Written in C++)

```
#define _CRT_SECURE_NO_DEPRECATE

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

#include "randomc.h"                // define classes for random number generators
#include "userintf.cpp"             // define system specific user interface

// define which random number generator to base random library on:

#include "mersenne.cpp"             // members of class TRandomMersenne
#define RANDOM_GENERATOR TRandomMersenne

// or:
//#include "ranrotw.cpp"            // members of class TRanrotWGenerator
//#define RANDOM_GENERATOR TRanrotWGenerator

// or:
//#include "mother.cpp"             // members of class TRanrotWGenerator
//#define RANDOM_GENERATOR TRandomMotherOfAll
```

```
#include "stocc.h"                  // define random library classes
#include "stoc1.cpp"                  // random library source code

void FindFirstToRead(FILE *);

int main()
{
  char c;
  int iLines,i,j;
  FILE *fp,*f2;
  long **matrix;
  iLines = -1;


int32 seed = time(0);               // random seed
StochasticLib1 sto(seed);           // make instance of random library


fp=fopen("pmt.dat","r");
f2=fopen("inputdat.txt","w");

 while(!feof(fp))
  {
          fscanf(fp,"%c",&c);
          if(c=='\n')
            iLines++;
  };

fseek(fp,0,SEEK_SET);
matrix = (long **)malloc(sizeof(long*)*iLines);

for(i=0;i<iLines;i++)
          matrix[i] = (long *)malloc(sizeof(long)*16);

FindFirstToRead(fp);

for(i=0;i<iLines;i++){
          for(j=0;j<16;j++){
                    FindFirstToRead(fp);
                    matrix[i][j]=0;
                    fscanf(fp,"%d",&matrix[i][j]);
          }
}

for (i=0; i<iLines; i++){
             for (j=0; j<16; j++){
                              matrix[i][j] = sto.Binomial(matrix[i][j], 0.2);
                    }
  }


for (i=0; i<iLines; i++){
          fprintf(f2,"%d %d %d %d %d %d %d %d %d %d %d %d %d %d %d %d\n",
matrix[i][0],matrix[i][1],matrix[i][2],matrix[i][3],matrix[i][4],matrix[i][5],matrix[i][6],matrix[i][7],matrix[i][
8],matrix[i][9],matrix[i][10],matrix[i][11],matrix[i][12],matrix[i][13],matrix[i][14],matrix[i][15]);
}

fclose(fp);
```

```
for(i=0;i<iLines;i++)
        free(matrix[i]);

free(matrix);

return 0;

}

void FindFirstToRead(FILE *fp)
{
 char c;
 fpos_t pos;
 fscanf(fp,"%c",&c);
 while(c==' ')
 {
        fscanf(fp,"%c",&c);
 };
 fgetpos(fp,&pos);
 pos--;
 fsetpos(fp,&pos);
 return;
};
```

B.3 Data Merging Program (Written in C#)

```
using System;
using System.Collections.Generic;
using System.Text;
using System.IO;
using System.Diagnostics;

namespace MergeLines
{
        class Program
        {
                static void Main(string[] args)
                {
                        String srcFile = "c:\\inputSrc.txt";
        String datFile = "c:\\inputdat.txt";
                        String outputFile = "c:\\output.txt";

                        srcFile = ReadOption(args, "-s:", srcFile);
                        datFile = ReadOption(args, "-d:", datFile);
        outputFile = ReadOption(args, "-o:", outputFile);

                        StreamReader src = new StreamReader(srcFile);
        StreamReader dat = new StreamReader(datFile);
                        StreamWriter output = new StreamWriter(outputFile);

        Lines accumulated = new Lines();
        accumulated.Add(new DatLine(dat.ReadLine()));

        // Skip one line from the source
        src.ReadLine();
```

```
        while (!src.EndOfStream && !dat.EndOfStream)
        {
            SrcLine line = new SrcLine(src.ReadLine());

            if (line.InteractionLevel != 0)
            {
                accumulated.Add(new DatLine(dat.ReadLine()));
            }
            else
            {
                accumulated.Merge().Write(output);
                accumulated.Reset();
                accumulated.Add(new DatLine(dat.ReadLine()));
            }
        }

        if (accumulated.HasLines)
        {
            accumulated.Merge().Write(output);
        }

        output.Flush();
                }

                private static String ReadOption(String[] args, String option, String defaultValue)
                {
                        foreach (String s in args)
                        {
                                if (s.StartsWith(option))
                                {
                                        StringBuilder builder = new StringBuilder(s);
                                        return builder.Replace(option, "").ToString();
                                }
                        }

                        return defaultValue;
                }

                private static bool ExistsOption(String[] args, String option)
                {
                        foreach (String s in args)
                        {
                                if (s.Equals(option))
                                {
                                        return true;
                                }
                        }

                        return false;
                }
}

internal class SrcLine
{
    public SrcLine(String line)
    {
        values = new Double[7];
        int i = 0;
```

```
            foreach (String s in line.Split(new char[] { ' ' }, 7))
            {
                Double.TryParse(s, out values[i++]);
            }
        }

        public SrcLine(Double x, Double y, Double z, Double photonCount, Double interactionLevel, Double
xEntry, Double yEntry)
        {
            values = new Double[7];
            values[0] = x;
            values[1] = y;
            values[2] = z;
            values[3] = photonCount;
            values[4] = interactionLevel;
            values[5] = xEntry;
            values[6] = yEntry;
        }

        public Double X
        {
            get
            {
                return values[0];
            }
        }

        public Double Y
        {
            get
            {
                return values[1];
            }
        }

        public Double Z
        {
            get
            {
                return values[2];
            }
        }

        public Double PhotonCount
        {
            get
            {
                return values[3];
            }
        }

        public Double InteractionLevel
        {
            get
            {
                return values[4];
            }
        }
```

```
    public Double XEntry
    {
        get
        {
            return values[5];
        }
    }

    public Double YEntry
    {
        get
        {
            return values[6];
        }
    }

    public Double WeightedX
    {
        get
        {
            return X * PhotonCount;
        }
    }

    public Double WeightedY
    {
        get
        {
            return Y * PhotonCount;
        }
    }

    public Double WeightedZ
    {
        get
        {
            return Z * PhotonCount;
        }
    }

    public void Write(StreamWriter writer)
    {
        int i = 0;
        foreach (Double value in values)
        {
            String format = (i < 3) ? "F" : "F0";
            writer.Write(value.ToString(format));
            writer.Write(' ');
            ++i;
        }

        writer.WriteLine();
    }

    Double[] values;
}

internal class DatLine
{
```

```
    public DatLine()
    {
        values = new Double[16];
    }

    public DatLine(String line)
    {
        values = new Double[16];
        int i = 0;
        foreach (String s in line.Split(new char[] { ' ' }, 16))
        {
            Double.TryParse(s, out values[i++]);
        }
    }

    public static DatLine operator+(DatLine oper1, DatLine oper2)
    {
        Double[] result = new Double[16];

        for (int i = 0; i < 16; ++i)
        {
            result[i] = oper1.values[i] + oper2.values[i];
        }

        return new DatLine(result);
    }

    public void Write(StreamWriter writer)
    {
        int i = 0;
        foreach (Double value in values)
        {
            writer.Write(value.ToString("F0"));
            writer.Write(' ');
            ++i;
        }

        writer.WriteLine();
    }

    private DatLine(Double[] values)
    {
        this.values = values;
    }

    Double[] values;
}

internal class Lines
{
    public Lines()
    {
        Reset();
    }

    public bool HasLines
    {
        get
        {
```

```
            return (lines != null) && (lines.Count > 0);
        }
    }

    public void Add(DatLine line)
    {
        lines.Add(line);
    }

    public DatLine Merge()
    {
        DatLine result = new DatLine();

        foreach (DatLine line in lines)
        {
            result += line;
        }

        return result;
    }

    public void Reset()
    {
        lines = new List<DatLine>();
    }

    private List<DatLine> lines;
}

/*
internal class Lines
{
    public Lines()
    {
        Reset();
    }

    public bool HasLines
    {
        get
        {
            return (lines != null) && (lines.Count > 0);
        }
    }

    public void Add(Line line)
    {
        lines.Add(line);
    }

    public Line Merge()
    {
        return new Line(ComputeX(), ComputeY(), ComputeZ(), TotalPhotonCounts, 0, lines[0].XEntry,
lines[0].YEntry);
    }

    public void Reset()
    {
        lines = new List<Line>();
```

```
            totalPhotonCounts = 0;
        }

        private Double ComputeX()
        {
            Double newX = 0;

            foreach (Line line in lines)
            {
                newX += line.WeightedX;
            }

            return newX / TotalPhotonCounts;
        }

        private Double ComputeY()
        {
            Double newY = 0;

            foreach (Line line in lines)
            {
                newY += line.WeightedY;
            }

            return newY / TotalPhotonCounts;
        }

        private Double ComputeZ()
        {
            Double newZ = 0;

            foreach (Line line in lines)
            {
                newZ += line.WeightedZ;
            }

            return newZ / TotalPhotonCounts;
        }

        private Double TotalPhotonCounts
        {
            get
            {
                if (totalPhotonCounts == 0)
                {
                    foreach (Line line in lines)
                    {
                        totalPhotonCounts += line.PhotonCount;
                    }
                }

                return totalPhotonCounts;
            }
        }

        private List<Line> lines;
        private Double totalPhotonCounts;
    } */
}
```

B.4 Data Matching Program (Written in C#)

```
using System;
using System.Collections.Generic;
using System.Text;
using System.IO;
using System.Diagnostics;

namespace ConsoleApplication2
{
	class Program
	{
		static void Main(string[] args)
		{
			int distanceThreshold = 5;
			int threshold = 0;
			String gridFile = null;
			String inputFile = "c:\\input.txt";
			String outputFile = "c:\\output.txt";
			String allOutputFile = null;

			if (args.Length < 2)
			{
				System.Console.WriteLine("Usage: FindPoint -g:gridfile [-t:threshold] [-
d:distance] [-i:inputfile] [-o:outputfile] [-a:alloutputfile]");
			}
			else
			{
				Int32.TryParse(ReadOption(args, "-t:", threshold.ToString()), out
threshold);
				Int32.TryParse(ReadOption(args, "-d:", distanceThreshold.ToString()), out
distanceThreshold);
				gridFile = ReadOption(args, "-g:", gridFile);
				if (gridFile == null)
				{
					System.Console.WriteLine("Usage: FindPoint -g:gridfile [-
t:threshold] [-d:distance] [-i:inputfile] [-o:outputfile] [-a:alloutputfile]");
					return;
				}

				inputFile = ReadOption(args, "-i:", inputFile);
				outputFile = ReadOption(args, "-o:", outputFile);
				allOutputFile = ReadOption(args, "-a:", allOutputFile);

				Grid grid = new Grid(gridFile);

				StreamReader input = new StreamReader(inputFile);
				StreamWriter output = new StreamWriter(outputFile);
				StreamWriter allOutputs = null;

				if (allOutputFile != null)
				{
					allOutputs = new StreamWriter(allOutputFile);
				}

				Int32[] values = new Int32[16];
				Char[] splitters = {','};
```

```
                                    while (!input.EndOfStream)
                                    {
                                            String line = input.ReadLine();
                                            Int32 i = 0;
                                            foreach (String current in line.Split(splitters))
                                            {
                                                    Int32.TryParse(current, out values[i++]);
                                            }

                                            Int32 total = 0;
                                            foreach (Int32 v in values)
                                            {
                                                    total += v;
                                            }

                                            if (allOutputs != null)
                                            {
                                                    allOutputs.Write(line);
                                                    allOutputs.Write(": ");
                                            }

                                            Cell found = grid.FindClosest(values, threshold,
distanceThreshold);

                                            if (total >= threshold)
                                            {
                                                    output.Write(line);
                                                    output.Write(": ");

                                                    if (found == null)
                                                    {
                                                            output.Write("No matching points found");
                                                            if (allOutputs != null)
                                                            {
                                                                    allOutputs.Write("No matching points
found");

                                                            }
                                                    }
                                                    else
                                                    {
                                                            found.Write(output);
                                                            if (allOutputs != null)
                                                            {
                                                                    found.Write(allOutputs);
                                                            }
                                                    }

                                                    output.WriteLine();
                                                    output.Flush();
                                            }
                                            else
                                            {
                                                    if (allOutputs != null)
                                                    {
                                                            allOutputs.Write("Total below threshold");
                                                    }
                                            }

                                            if (allOutputs != null)
```

```
                                {
                                        allOutputs.WriteLine();
                                        allOutputs.Flush();
                                }
                        }
                }
        }

        private static String ReadOption(String[] args, String option, String defaultValue)
        {
                foreach (String s in args)
                {
                        if (s.StartsWith(option))
                        {
                                StringBuilder builder = new StringBuilder(s);
                                return builder.Replace(option, "").ToString();
                        }
                }

                return defaultValue;
        }

        private class Cell
        {
                private Int32[] coordinates = new Int32[3];
                private List<IndexedValue> orderedValues = new List<IndexedValue>();
                private IndexedValue[] values = new IndexedValue[16];
                private Int32 total;

                public Cell(String row)
                {
                        Char[] splitters = {','};
                        String[] strings = row.Split(splitters);

                        Int32.TryParse(strings[0], out coordinates[0]);
                        Int32.TryParse(strings[1], out coordinates[1]);
                        Int32.TryParse(strings[2], out coordinates[2]);

                        for (Int32 i = 3; i < 19; ++i)
                        {
                                Int32 a;
                                Int32.TryParse(strings[i], out a);
                                IndexedValue temp = new IndexedValue(i-3, a);
                                orderedValues.Add(temp);
                                values[temp.Index] = temp;
                                total += a;
                        }

                        orderedValues.Sort();
                }

                public void Write(StreamWriter writer)
                {
                        StringBuilder builder = new StringBuilder();
                        builder.AppendFormat(
                                        "{0}, {1}, {2}: {3}, {4}, {5}, {6}, {7}, {8}, {9}, {10},
{11}, {12}, {13}, {14}, {15}, {16}, {17}, {18}: {19}",
                                        coordinates[0],
                                        coordinates[1],
```

```
                                        coordinates[2],
                                        values[0].Value,
                                        values[1].Value,

                                        values[2].Value,
                                        values[3].Value,

                                        values[4].Value,

                                        values[5].Value,

                                        values[6].Value,

                                        values[7].Value,

                                        values[8].Value,

                                        values[9].Value,

                                        values[10].Value,

                                        values[11].Value,

                                        values[12].Value,

                                        values[13].Value,

                                        values[14].Value,

                                        values[15].Value,

                                        total
                                        );

        writer.Write(builder.ToString());
}

public Int32 Total
{
        get
        {
                return total;
        }
}

public Int32 Distance(Int32 index, Int32 value)
{
        Int32 distance = values[index].Value - value;
        return (distance >= 0) ? distance : -distance;
}

private class IndexedValue : IComparable
{
        private Int32 index;
        private Int32 value;

        public IndexedValue(Int32 index, Int32 value)
        {
                this.index = index;
```

```
                        this.value = value;
                }

                public Int32 Index
                {
                        get
                        {
                                return index;
                        }
                }

                public Int32 Value
                {
                        get
                        {
                                return value;
                        }
                }

                public int CompareTo(Object other)
                {
                        IndexedValue temp = (IndexedValue)other;
                        return Value.CompareTo(temp.Value);
                }
        }
}

private class Grid
{
        private List<Cell> cells = new List<Cell>();

        public Grid(String fileName)
        {
                StreamReader stream = new StreamReader(fileName);

                while (!stream.EndOfStream)
                {
                        cells.Add(new Cell(stream.ReadLine()));
                }
        }

        public Cell FindClosest(Int32[] values, Int32 threshold, Int32 distanceThreshold)
        {
                List<Cell> result = cells;

                result = Eliminate(threshold, result);

                Int32 i;
                for (i = 0; i < 16; ++i)
                {
                        if (result.Count <= 1)
                        {
                                break;
                        }

                        result = Eliminate(i, values[i], result, distanceThreshold);
                }

                if (result.Count == 0)
```

```
            {
                return null;
            }

            return PickClosest(values, result);
        }

        private List<Cell> Eliminate(Int32 threshold, List<Cell> current)
        {
            List<Cell> result = new List<Cell>();

            foreach (Cell cell in current)
            {
                if (cell.Total >= threshold)
                {
                    result.Add(cell);
                }
            }

            return result;
        }

        private List<Cell> Eliminate(Int32 index, Int32 value, List<Cell> current, Int32
distanceThreshold)
        {
            List<Cell> result = new List<Cell>();

            foreach (Cell cell in current)
            {
                if (cell.Distance(index, value) <= distanceThreshold)
                {
                    result.Add(cell);
                }
            }

            // Return the current list if this elimination returned 0 items
            if (result.Count == 0)
            {
                result = current;
            }

            return result;
        }

        private Cell PickClosest(Int32[] values, List<Cell> current)
        {
            Int32 closestDistance = Int32.MaxValue;;
            Cell closestCell = current[0];

            foreach (Cell cell in current)
            {
                Int32 total = 0;
                for (int i = 0; i < 16; ++i)
                {
                    total += cell.Distance(i, values[i]);
                }

                if (total < closestDistance)
                {
```

```
                                        closestDistance = total;
                                        closestCell = cell;
                                    }
                                }

                                return closestCell;
                            }
                        }
                    }
                }
```

APPENDIX C. DRAWINGS

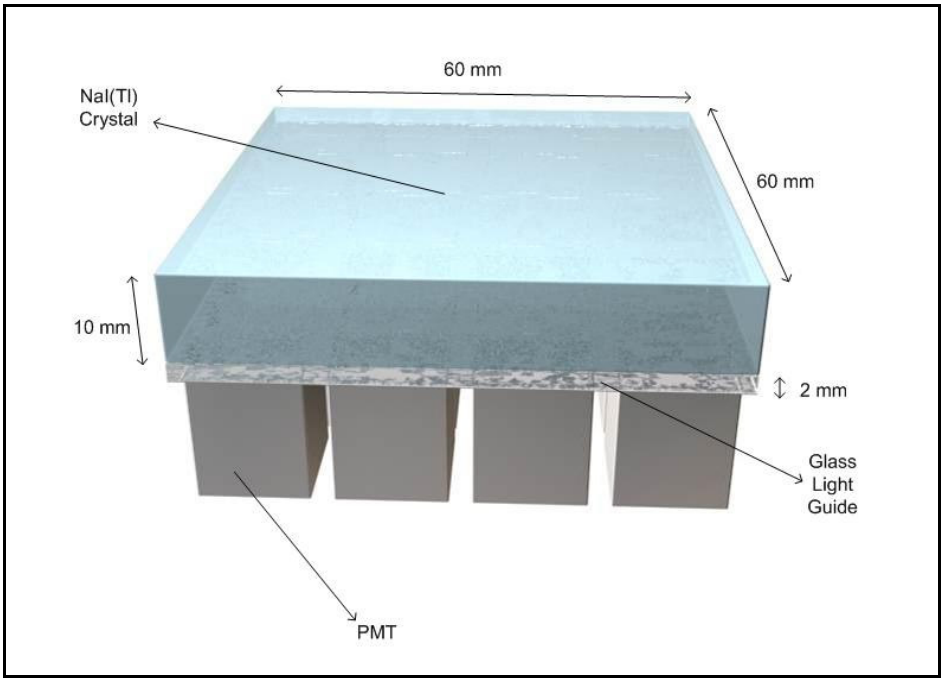

Figure C.1 Side view of the detector.

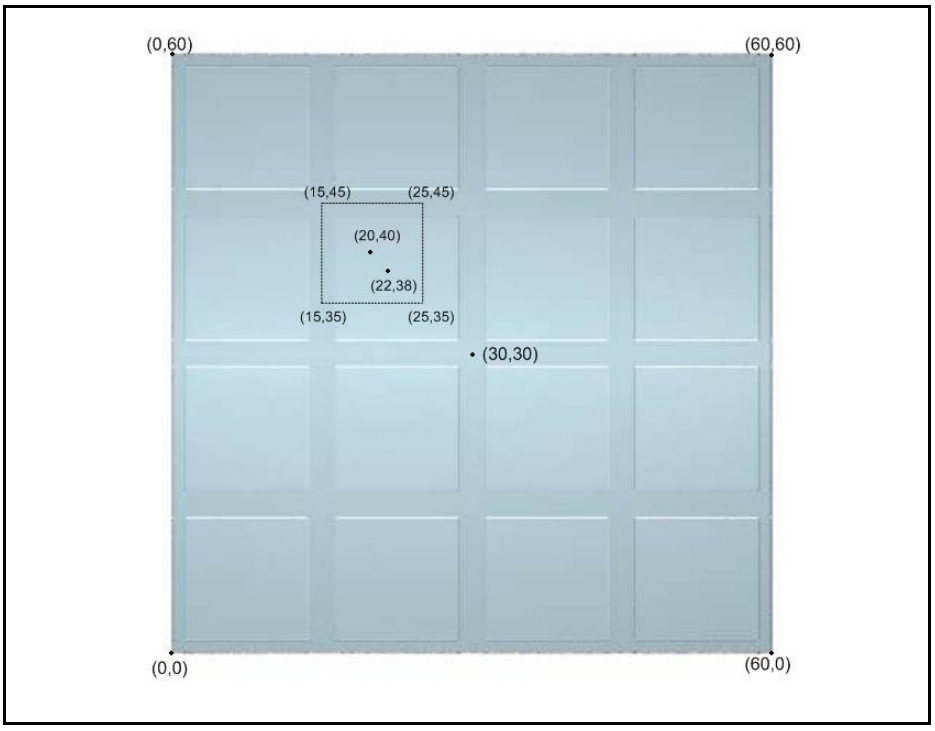

Figure C.2 Top view of the detector (Coordinates of the sample points).

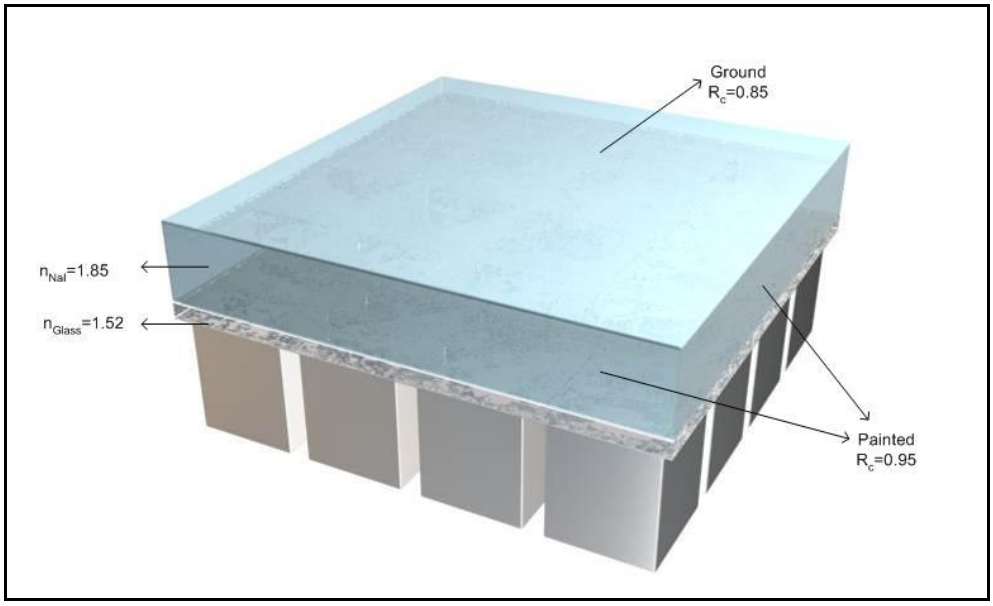

Figure C.3 Reflection and refraction coefficients.

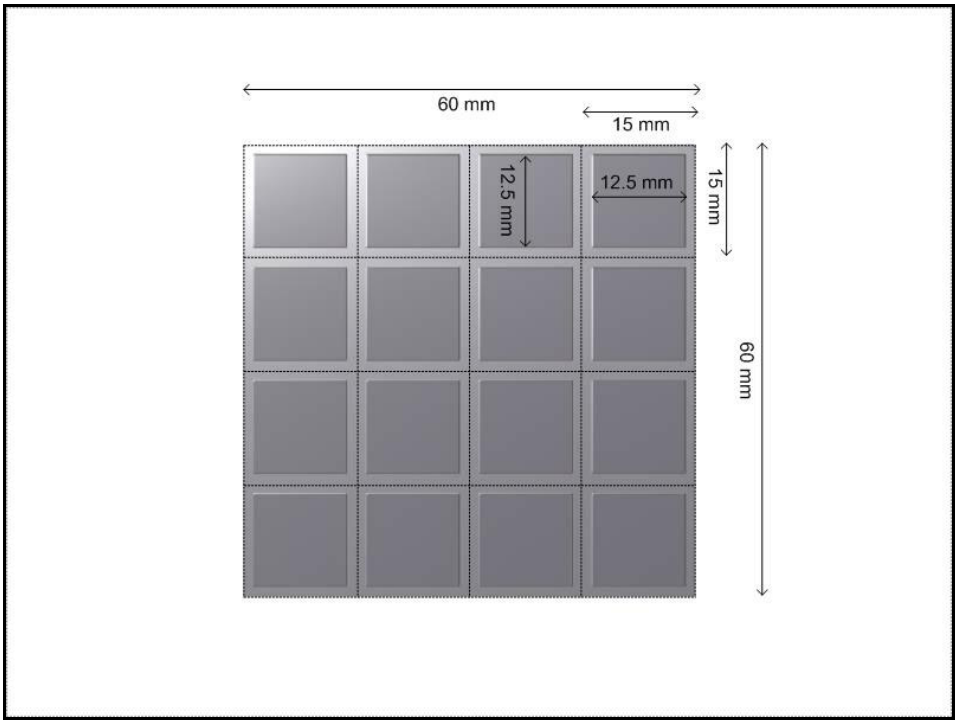

Figure C.4 Bottom view of the detector.

REFERENCES

1. Avril, N., M. Schelling, J. Dose, W.A. Weber, and M. Schwaiger, "Utility of PET in breast cancer," *Clinical Positron Imaging*, Vol. 2, pp. 261-271, Oct 1999.

2. Townsend, D. W., "Medical Imaging," *CERN Academic Training Lecture Series*, Vol. 325, pp. 1-108, Apr 1996.

3. Derenzo, S. E., Moses W. W., Huesman R. H., and T. F. Budinger, *Critical instrumentation issues for <2mm resolution, high sensitivity brain PET*, in *Quantification of Brain Function: Tracer Kinetics and Image Analysis in Brain PET*, pp. 25-37, Amsterdam: Elsevier Science Publishers, 1993.

4. Lerche, Ch. W., J. M. Benlloch, F. Sánchez, N. Pavón, E. N. Gimenez, M. Giménez, M. Fernández, J. Cerda, J. Martínez and A. Sebastià, "Depth of interaction measurement in gamma ray imaging detectors with continuous scintillation crystals," *IEEE Nuclear Science Symposium Conference Record,* Vol. 3, pp. 2169-2173, Oct 2003.

5. Bronstein, A., M. Bronstein, M. Zibulevsky and Y. Y. Zeevi, "Optimal nonlinear line-of-flight estimation in positron emission tomography," *IEEE Trans. Med. Imag.*, Vol. 50, pp. 421-426, Jun 2002.

6. Gagnon, D., N. Pouliot, L. Laperriére , M. Therrien and P. Oliver, "Maximum likelihood positioning in the scintillation camera using depth of interaction," *IEEE Trans. Med. Imag.*, Vol. 12, pp. 101-107, Mar 1993.

7. Moses, W. M., and S. E. Derenzo, "Design studies for a PET detector module using a PIN photodiode to measure depth of interaction," *IEEE Trans. Nucl. Sci.*, Vol. 41, pp. 1441-1445, Aug 1994.

8. Lui, H., T. Omura, M. Watanabe, and T. Yamashita, "Development of a depth of interaction detector for gamma-rays," *Nucl. Inst. Meth.*, Vol. 459, pp. 182-190, Feb 2001.

9. Crosetto, D., *400+ Times Improved PET Efficiency for Lower-Dose Radiation, Lower-Cost Cancer Screening*, DeSoto: 3D-Computing, 1st ed., 2000.

10. Rogers, J. G., C. Moisan, E. M. Hoskinson, M. S. Andreaco, C. W. Williams, and R. Nutt, "A practical block detector for a depth-encoding PET camera," *IEEE Trans. Nucl. Sci.*, Vol. 43, pp. 3240-3248, Dec 1996.

11. Moisan, C., M. S. Andreaco, J. G. Rogers, S. Paquet, and D. Vozza, "Segmented LSO crystal for depth-of-interaction encoding PET," *IEEE Trans. Nucl. Sci.*, Vol. 45, pp. 3030–3035, Dec 1998.

12. Lerche, Ch. W., J. M. Benlloch, F. Sánchez, N. Pavón, B. Escat, E. N. Gimenez, M. Fernández, I. Torres, M. Giménez, A. Sebastià, and J. Martínez, "Depth of gamma-ray interaction within continuous crystals from the width of its scintillation light-distribution," *IEEE Trans. Nucl. Sci.*, Vol. 52, pp. 560-572, Jun 2005.

13. Joung, J., R. S. Miyaoka, S. G. Kohlmyer, and T. K. Lewellen, "Investigation of bias-free positioning estimators for the scintillation cameras," *IEEE Trans. Nucl. Sci.*, Vol. 48, pp. 715-719, Jun 2001.

14. Tomitani, T., Y. Futami, Y. Iseki, S. Kouda, T. Nishio, T. Murakami, A. Kitagawa, M. Kanazawa, E. Urakabe, M. Shinbo, and T. Kanai, "Depth encoding of point-of-interaction in thick scintillation cameras," *IEEE Nuclear Science Symposium Conference Record*, Vol. 3, pp. 1182-1186, Oct 1999.

15. Delorme, S., R. Frei, C. Joseph, J. F. Loude, and C. Morel, "Use of a neural network to exploit light division in a triangular scintillating crystal," *Nucl. Inst. Meth.*, Vol. 373, pp. 111-118, Feb 1996.

16. Clément, D., R. Frei, J. F. Loude, and C. Morel, "Development of a 3D position sensitive scintillation detector using neural networks," *IEEE Nuclear Science Symposium Conference Record*, Vol. 3, pp. 1448-1452, Nov 1998.

17. Bronstein, A., M. Bronstein, M. Zibulevsky, and Y. Y. Zeevi, "High-energy photon detection in positron emission tomography using adaptive non-linear parametric estimation algorithms," preprint, Technion, Feb 2002.

18. Shung, K. K., M. B. Smith, and B. M. W. Tsui, *Principles of Medical Imaging*, California: Academic Press, 1992.

19. Anger, H. O., *Radioisotope Cameras*, Vol. 2 of *Instrumentation in Nuclear Medicine*, pp. 478-552, New York: Academic Press, 1974.

20. Anger, H. O., *Tomography and Other Depth-Discrimination Techniques*, Vol. 2 of *Instrumentation in Nuclear Medicine*, pp. 61-100, New York: Academic Press, 1974.

21. Vancouver Hospital and Health Sciences Centre's Medical Imaging Research Group's Spect Tutorial Website. Available: http://www.physics.ubc.ca/~mirg/home/tutorial/tutorial.html.

22. Anger, H. O., "A new instrument for mapping gamma-ray emitters," Berkeley, UCRL-3653-38, 1957.

23. Anger, H. O., "Scintillation camera with multichannel collimators," *J. Nucl. Med.*, Vol. 5, pp. 515-531, Jul 1964.

24. Sharp, P. F., et al., *Practical Nuclear Medicine*, Oxford: IRL Press, 1998.

25. Steén, M., and P. Uhlén, "Development of a time-of-flight and 3D demonstration set-up for positron emission tomography," Master Thesis, Royal Institute of Technology, Stockholm, Sweden, 1998.

26. Knoll, G. F., *Radiation Detection and Measurement*, New York: John Wiley & Sons, 2nd ed., 1989.

27. National Electrical Manufacturers Association, "Performance measurements of positron emission tomographs," NEMA, NU-2, 2001.

28. Saint-Gobain Crystals Website. Available: http://www.bicron.com.

29. Karp, J. S., and G. Muehllehner, "Performance of a position-sensitive scintillation detector," *Phys. Med. Biol.*, Vol. 30, pp. 643-655, Jan 1985.

30. Casey, M. E., and R. Nutt, "A multicrystal two dimensional BGO detector system for positron emmision tomography," *IEEE Trans. Nucl. Sci.*, Vol. 33, pp. 460-463, Jan 1986.

31. Pichler, B. J., G. Boning, M. Rafecas, W. Pimpl, E. Korenz, M. Schwaiger, and S. I. Ziegler, "Feasibility study of a compact high resolution dual layer LSO-APD detector module for positron emission tomography," *IEEE Nuclear Science Symposium Conference Record*, Vol. 2, pp. 1199-1203, Nov 1998.

32. Shao, Y., R. W. Silverman, R. Farell, L. Cirignano, R. Grazioso, K. S. Shah, G. Vissel, M. Clajus, T. O. Tumer, and S. R. Cherry, "Design studies of a high resolution PET detector using APD arrays," *IEEE Trans. Nucl. Sci.*, Vol. 47, pp. 1051-1057, Jun 2000.

33. Lecomte, R., J. Cadorete, S. Rodrigue, D. Lapointe, D. Rouleau, M. Bentourkia, R. Yao, and P. Msaki, "Initial results from the Sherbrooke avalanche photodiode positron tomography," *IEEE Trans. Nucl. Sci.*, Vol. 43, pp. 1952-1957, Jun 1996.

34. Ziegler, S. I., B. J. Pichler, G. Boening, M. Rafecas, W. Pimpl, E. Lorenz, N. Schmitz, and M. Schwaiger, "Prototype high resolution animal positron tomograph with avalanche photodiode arrays and LSO crystals," *Eur. J. Nucl. Med.*, Vol. 28, pp. 136-143, Feb 2001.

35. Saoudi, A., C. Pepin, F. Dion, M. Bentourkia, R. Lecomte, M. Andreaco, M. Casey, R. Nutt, and H. Dautet, "Investigation of depth-of-interaction by pulse shape discrimination in multicrystal detectors read out by avalanche photodiode," *IEEE Nuclear Science Symposium Conference Record*, Vol. 2, pp. 1078-1082, Nov 1998.

36. Schmand, M., L. Eriksson, M. E. Casey, M. S. Andreaco, C. Melcher, K. Wienhard, G. Flugge, and R. Nutt, "Performance results of a new DOI detector block for a high resolution PET-LSO Research Tomograph HRRT," *IEEE Trans. Nucl. Sci.*, Vol. 45, pp. 3000-3006, Dec 1998.

37. Huber, J. S., W. W. Moses, S. E. Derenzo, M. H. Ho, M. S. Andreaco, M. J. Paulus, and R. Nutt, "Characterization of a 64 channel PET detector using photodiodes for crystal identification," *IEEE Trans. Nucl. Sci.*, Vol. 44, pp. 1197-1201, Jun 1997.

38. Bendriem, B., M. Casey, L. Eriksson, M. Schmand, M. Eriksson, J. Frey, and R. Nutt, "From PET to PET/SPECT," *Revue de l'ACOMEN*, Vol. 5, pp. 156-159, Mar 1999.

39. Dhanasopon, A. P., C. S. Levin, A. M. K. Foudray, P. D. Olcott, and F. Habte, "Scintillation crystal design features for a miniature gamma ray camera," *IEEE Trans. Nucl. Sci.*, Vol. 52, pp. 1443-1446, Oct 2005.

40. Siegel, S., S. R. Cherry, A. R. Ricci, Y. Shao, and M. E. Phelps, "Development of continuous detectors for a high resolution animal PET system," *IEEE Nuclear Science Symposium and Medical Imaging Conference Record*, Vol. 4, pp. 1662-1666, Nov 1994.

41. Levin, C. S., and E. J. Hoffman, "Calculation of positron range and its effect on the fundamental limit of positron emission tomography system spatial resolution," *Phys. Med. Biol.*, Vol. 44, pp. 781-799, Mar 1999.

42. Cherry, S. R., Y. Shao, M. P. Tornai, S. Siegel, A. R. Ricci, and M. E. Phelps, "Collection of scintillation light from small BGO crystals," *IEEE Trans. Nucl. Sci.*, Vol. 42, pp. 1058-1063, Aug 1995.

43. Del Guerra, A., F. De Notaristefani, G. Di Domenico, M. Giganti, R. Pani, A. Piffanelli, A. Turra, and G. Zavattini, "Use of a YAP: Ce matrix coupled to a position-sensitive photomultiplier for high resolution positron emission tomography," *IEEE Nuclear Science Symposium and Medical Imaging Conference Record*, Vol. 2, pp. 1016-1020, Oct 1995.

44. Siegel, S., J. J. Vaquero, L. Aloj, J. Seidel, E. Jagoda, W. R. Gandler, W. C. Eckelman, and M. V. Green, "Initial results from a PET/planar small animal imaging system," *IEEE Nuclear Science Symposium Conference Record*, Vol. 2, pp. 1274–1276, Nov 1998.

45. Vaquero, J. J., J. Seidel, S. Siegel, W. R. Gandler, and M. V. Green, "Performance characteristics of a compact position-sensitive LSO detector module," *IEEE Trans. Med. Imag.*, Vol. 17, pp. 967-978, Dec 1998.

46. Shao, Y. P., R. W. Silverman, and S. R. Cherry, "Evaluation of Hamamatsu R5900 series PMTs for readout of high-resolution scintillator arrays," *Nucl. Inst. Meth.*, Vol. 454, pp. 379-388, Nov 2000.

47. Chatziioannou, A. F., S. R. Cherry, Y. Shao, R. W. Silverman, K. Meadors, T. H. Farquhar, M. Pedarsani, and M. E. Phelps, "Performance evaluation of microPET: a high-resolution lutetium oxyorthosilicate PET scanner for animal imaging," *J. Nucl. Med.*, Vol. 40, pp. 1164-1175, Jul 1999.

48. Cherry, S. R., Y. Shao, S. Siegel, R. W. Silvermad, K. Meadors, J. Young, W. F. Jones, D. Newport, C. Moyers, E. U. Mumcuoglu, A. F. Chatziioannou, T. Farquhar, M. Andreaco, M. Paulus, D. Binkley, R. Nutt, and M. E. Phelps, "MicroPET: a high resolution PET scanner for imaging small animals," *IEEE Nuclear Science Symposium Conference Record*, Vol. 2, pp. 1120-1124, Nov 1996.

49. Chatziioannou, A. F., Y. C. Tai, N. Doshi, and S. R. Cherry, "Detector development for microPET II: a 1 μl resolution PET scanner for small animal imaging," *Phys. Med. Biol.*, Vol. 46**,** pp**.** 2899-2910, Nov 2001.

50. Seidel, J., W. R. Gandler, and M. V. Green, "Characteristics of a pair of small field-of-view LSO scintillation cameras," *IEEE Trans. Nucl. Sci.*, Vol. 43, pp. 1968-1973, Jun 1996.

51. Sundermann, E., and I. Lemahieu, "PET image reconstruction using simulated annealing," *Proceedings of SPIE*, Vol. 2734, pp. 378-386, May 1995.

52. Tsang, G., C. Moisan, and J. G. Rogers, "A Simulation to model position encoding multicrystal PET detectors," *IEEE Trans. Nucl. Sci.*, Vol. 42, pp. 2236-2243, Dec 1995.

53. Barrett, H., and W. Swindell, *Radiological Imaging*, New York: Academic Press, 1981.

54. Moisan, C., E. M. Hoskinson, A. Levin, and D. Vozza, "A public domain platform to model scintillation counters for gamma-ray imaging applications," *Proceedings of SPIE*, Vol. 3115, pp. 21-29, Jul 1997.

55. Moisan, C., D. Vozza, and M. Loope, "Simulating the performances of an LSO based position encoding detector for PET," *IEEE Trans. Nucl. Sci.*, Vol. 44, pp. 2450-2458, Dec 1997.

56. Moisan, C., G. Tsang, J. G. Rogers, and E. M. Hoskinson, "Performance studies of a depth encoding multicrystal detector for PET," *IEEE Nuclear Science Symposium and Medical Imaging Conference Record*, Vol. 2, pp. 1064-1068, Oct 1995.

57. Vozza, D., C. Moisan, and S. Paquet, "An improved model for the energy resolution of multicrystal encoding detectors for PET," *IEEE Trans. Nucl. Sci*., Vol. 44, pp. 179-183, Apr 1997.

58. Levin, A., and C. Moisan, "A more physical approach to model the surface treatment of scintillation counters and its implementation into DETECT," *IEEE Nuclear Science Symposium Conference Record*, Vol. 2, pp. 702-706, Nov 1996.

59. Shao, Y., S. R. Cherry, S. Siegel, and R. W. Silverman, "A study of inter crystal scatter in small scintillator arrays for high resolution PET imaging," *IEEE Nuclear Science Symposium and Medical Imaging Conference Record*, Vol. 2, pp. 1006-1010, Oct 1995.

60. Spisar, M., J. N. Aarsvold, and R.A. Mintzer, "DETECT97 simulation studies of light output in a full field-of-view small gamma camera," *IEEE Nuclear Science Symposium Conference Record*, Vol. 2, pp. 1188-1192, Nov 1997.

61. Moisan, C., A. Levin, and H. Laman, "Testing scintillation transport models with photoelectron yields measured under different surface finishes," *IEEE Nuclear Science Symposium Conference Record*, Vol. 1, pp. 824-828, Nov 1997.

62. Saoudi, A., C. M. Pepin, and R. Lecomte, "Study of light collection in multi-crystal detector," *IEEE Trans. Nucl. Sci*., Vol. 47, pp. 1634-1639, Aug 2000.

63. Takacs, G. J., A. B. Rosenfeld, and M. L. F. Lerch, "Design and simulation of continuous scintillator with pixellated photodetector," *IEEE Trans. Nucl. Sci*., Vol. 48, pp. 1412-1417, Aug 2001.

64. Cayouette, F., C. Moisan, N. Zhang, and C. J. Thompson, "Monte Carlo modeling of scintillator crystal performance for stratified PET detectors with DETECT2000," *IEEE Trans. Nucl. Sci*., Vol. 49, pp. 624-628, Jun 2002.

65. Tawara, H., S. Sasaki, K. Saito1, and E. Shibamura, "A Monte-Carlo method for determining absolute scintillation-photon yields and energy resolution of scintillators for gamma rays," *Proceedings of KEK*, Vol. 1, pp. 152-160, Aug 2000.

66. C++ Class Library Containing Uniform Random Number Generators. Available: http://www.agner.org/random/randomc.zip

67. Aitken, D. W., B. L. Beron, G. Yenicay, and H. R. Zulliger, "The fluorescent response of NaI(Tl+), CsI(Tl+), CsI(Na+) and CaF2(Eu2+) to X-rays and low energy gamma rays," *IEEE Trans. NucI. Sci.*, Vol. 14, pp. 468-477, Feb 1967.

68. Murray, R. B., and A. Meyer, "Scintillation response of activated inorganic crystals to various charged particles," *Physical Review*, Vo1. 122, pp. 815-826, May 1961.

69. Iredale, P., "The non-proportional response of NaI(Tl) crystals to gamma-rays," *Nucl. Inst. Meth.*, Vol. 11, pp. 336-339, 1961.

70. Iredale, P., "The effect of the non-proportional response of NaI(Tl) crystals to electrons upon the resolution for gamma-rays," *Nucl. Inst. Meth.*, Vol. 11, pp. 340-346, 1961.

71. Valentine, J. D., B. D. Rooney and J.Li, "The light yield nonproportionality component of scintillator energy resolution," *IEEE Nuclear Science Symposium Conference Record*, Vol. 1, pp. 833-837, Nov 1997.

72. Rooney, B. D., and J. D. Valentine, "Scintillator light yield nonproportionality: calculating photonresponse using measured electron response," *IEEE Trans. Nucl. Sci.*, Vol. 44, pp. 509-516, Jun 1997.

73. Zerby, C. D., A. Meyer and R. B. Murray, "Intrinsic line broadening in NaI(Tl) gamma-ray spectrometers," *Nucl. Instr. Meth.*, Vol. 12, pp. 115-123, 1961.

74. Hill, R., and A. J. L. Collinson, "The effect on the scintillation efficiency of NaI(Tl) of changes in the thallium concentration and strain," *Brit. J. Appl. Phys.*, Vol. 17, pp. 1377-1383, 1966.

75. Porter, F. T., M. S. Freedman, F. Wagner, Jr. Sherman, and I. S. Sherman, "Response of NaI, anthracene and plastic scintillators to electrons and the problems of detecting low energy electrons with scintillation counters," *Nucl. Instr. Meth.*, Vol. 39, pp. 35-44, 1966.

76. Rooney, B. D., and J. D. Valentine, "Benchmarking the Compton coincidence technique for measuring electron response nonproportionality in inorganic scintillators," *IEEE Trans. Nucl. Sci.*, Vol. 43, pp. 1271-1276, Jun 1996.