

**EDWARD FLÓREZ PACHECO**

**ANÁLISE DA DINÂMICA E QUANTIFICAÇÃO METABÓLICA DE IMAGENS  
DE MEDICINA NUCLEAR NA MODALIDADE PET/CT**

**São Paulo**

**2016**

**EDWARD FLÓREZ PACHECO**

**ANÁLISE DA DINÂMICA E QUANTIFICAÇÃO METABÓLICA DE IMAGENS  
DE MEDICINA NUCLEAR NA MODALIDADE PET/CT**

**Tese de Doutorado apresentada à Escola  
Politécnica da Universidade de São Paulo, como  
parte dos requisitos para a obtenção do título de  
Doutor em Ciências.**

**Área de Concentração: Engenharia Biomédica**

**Orientador: Prof. Dr. Sérgio Shiguemi Furuie**

**São Paulo**

**2016**

Este exemplar foi revisado e corrigido em relação à versão original, sob responsabilidade única do autor e com a anuência de seu orientador.

São Paulo, \_\_\_\_\_ de \_\_\_\_\_ de \_\_\_\_\_

Assinatura do autor: \_\_\_\_\_

Assinatura do orientador: \_\_\_\_\_

### Catálogo-na-publicação

Flórez Pacheco, Edward

Análise da Dinâmica e Quantificação Metabólica de imagens de Medicina Nuclear na modalidade PET/CT / E. Flórez Pacheco -- versão corr. -- São Paulo, 2016.

168 p.

Tese (Doutorado) - Escola Politécnica da Universidade de São Paulo. Departamento de Engenharia de Telecomunicações e Controle.

1.Medicina Nuclear 2.Tomografia por Emissão de Pósitrons (PET)  
3.Quantificação Metabólica 4.Phantoms Antropomórficos 3D 5.Imagens PET/CT reais I.Universidade de São Paulo. Escola Politécnica. Departamento de Engenharia de Telecomunicações e Controle II.t.

## FOLHA DE APROVAÇÃO

Edward Flórez Pacheco. Análise da Dinâmica e Quantificação Metabólica de imagens de Medicina Nuclear na modalidade PET/CT [Tese de Doutorado]. Escola Politécnica da Universidade de São Paulo, Área de Engenharia Biomédica.

São Paulo, 28 de março de 2016.

### Banca Examinadora

1. Prof. Dr. Sérgio Shiguemi Furuie  
Escola Politécnica (EP)  
Universidade de São Paulo – USP
2. Prof. Dr. Marco A. Gutierrez  
Instituto do Coração (InCor-HCFM)  
Universidade de São Paulo – USP
3. Prof. Dr. Mauricio Moralles  
Instituto de Pesquisas Energéticas e Nucleares (IPEN)  
Universidade de São Paulo – USP
4. Prof. Dr. Carlos Buchpiguel  
Centro de Medicina Nuclear – Instituto de Radiologia (CMN-HCFM)  
Universidade de São Paulo – USP
5. Prof. Dr. Nelson Delfino d'Ávila Mascarenhas  
Universidade Federal de São Carlos – UFSCar

## DEDICATÓRIA

A Deus, por me dar saúde, paciência, confiança e motivação para a batalha do dia-a-dia durante todo o meu doutorado. Ele literalmente foi quem me carregou no colo durante tantos momentos difíceis, não me permitindo que fique afastado do propósito que ele tem para mim. Porque sempre preencheu minha vida de coisas maravilhosas e não deixou de me conduzir pelo seu caminho, aumentando sempre a minha fé e a necessidade dele e da sua palavra. Obrigado Deus pela vida e pela maravilhosa família que me deste!

Aos meus amados pais: Emperatriz e Edgard, duas pessoas únicas e insubstituíveis para mim, exemplos de pais e amigos. Vocês me educaram sempre usando as vias do amor e me deram muito mais do que sempre precisei. Vocês fizeram de mim a pessoa que hoje sou, e eu só tenho motivos para lhes agradecer. Sou eternamente grato por tê-los comigo. Que Deus os conserve sempre assim! Amo muito vocês!

Ao meu irmão Roosevelt e a minha irmã Milagros, pelo carinho sincero, apoio constante e por sempre serem aqueles parceiros ao longe da minha existência. Sempre me deram aquele espaço tão especial em suas vidas, acompanhando de perto meu desenvolvimento e comemorando como suas cada uma das minhas conquistas.

A minha adorada Susana. Á quem merece o meu amor hoje e sempre. Aproveito a culminação desta importante fase da minha vida para te expressar a minha gratidão e alegria por estar ao teu lado. Você é uma grande companheira e uma pessoa maravilhosa que me aceita, me impulsa e me ajuda a ver o futuro com otimismo. Deus me presenteou grandemente por ter você comigo e para mim, e porque você é a principal responsável por tudo de bom que acontece comigo e com a nossa linda família. Obrigado por tanto amor, pela maravilhosa convivência e por me complementar tão perfeitamente.

Ao meu amado André Eduardo. Motivo de vida, motivo de felicidade, motivo de esperança, motivo de desafio e luta constante. Você representa a minha maior inspiração e conquista. Filho, um dia você apareceu nas nossas vidas e magicamente se fez dono absoluto do nosso tempo, do nosso espaço e do nosso amor. Obrigado pela oportunidade de conhecer aspectos da vida que são simplesmente únicos, sendo seu pai e amigo. Agradeço muito a Deus por fazê-lo existir e por preencher os nossos dias de vida e alegria.

## AGRADECIMENTOS

Ao meu orientador, Prof. Dr. Sérgio Shiguemi Furuie, pela bela orientação, dedicação, suporte, paciência e, principalmente pela amizade durante todos esses anos de pesquisa.

À Escola Politécnica da Universidade de São Paulo – USP, na pessoa de seu Diretor, Prof. Dr. José Roberto Castilho Piqueira.

Aos Professores do Programa de Pós-Graduação em Engenharia Biomédica, pela contribuição na minha formação acadêmica e científica.

À Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP) pelo importante apoio financeiro e institucional durante a realização deste projeto, aqui no Brasil e no exterior.

Ao Prof. Felix Mas Milian pelo fornecimento do *phantom* MASH que tem sido muito valioso na fase de simulação deste projeto.

Ao Centro de Medicina Nuclear do Hospital das Clínicas da Faculdade de Medicina da Universidade de São Paulo (HC-FMUSP), na pessoa do Sr. Alexandre Teles Garcez, pelo suporte e facilidades no fornecimento dos manuais e ficha de dados do PET/CT Biograph™ – Siemens.

Ao Instituto do Coração (InCor) do Hospital das Clínicas da Faculdade de Medicina da Universidade de São Paulo (HC-FMUSP), na pessoa do Dr. José Cláudio Meneghetti – Diretor do Serviço de Medicina Nuclear e Imagem Molecular e, do Sr. Marco Antônio de Oliveira – Físico Chefe e Supervisor de Radioproteção, pelo suporte e facilidades no fornecimento das imagens médicas PET/CT para a elaboração deste projeto.

Ao Laboratório de Computação Científica Avançada da Universidade de São Paulo, na pessoa do Sr. Francisco Ribacionka, pelo suporte na etapa de instalação dos softwares e durante o tempo de uso do cluster AGUIA.

*To University of Mississippi Medical Center, in the person of the Chair of the Dept. of Radiology, Dr. Timothy C. McCowan, MD.*

*Special thanks to Dr. Vani Vijayakumar, Professor (T) of the Department of Radiology and Chief of the Division of Nuclear Medicine at University of Mississippi Medical Center who contributed in my day-to-day learning regarding the clinical life in the field we work, and enhanced my personal life by immersing in the American culture. She supervised and guided me in developing a part of my PhD project, that was a period of much growth and improvement.*

## RESUMO

A presença da Medicina Nuclear como modalidade de obtenção de imagens médicas é um dos principais procedimentos utilizados hoje nos centros de saúde, tendo como grande vantagem a capacidade de analisar o comportamento metabólico do paciente, traduzindo-se em diagnósticos precoces. Entretanto, sabe-se que a quantificação em Medicina Nuclear é dificultada por diversos fatores, entre os quais estão a correção de atenuação, espalhamento, algoritmos de reconstrução e modelos assumidos. Neste contexto, o principal objetivo deste projeto foi melhorar a acurácia e a precisão na análise de imagens de PET/CT via processos realísticos e bem controlados. Para esse fim, foi proposta a elaboração de uma estrutura modular, a qual está composta por um conjunto de passos consecutivamente interligados começando com a simulação de *phantoms* antropomórficos 3D para posteriormente gerar as projeções realísticas PET/CT usando a plataforma GATE (com simulação de Monte Carlo), em seguida é aplicada uma etapa de reconstrução de imagens 3D, na sequência as imagens são filtradas (por meio do filtro de Anscombe/Wiener para a redução de ruído Poisson característico deste tipo de imagens) e, segmentadas (baseados na teoria Fuzzy Connectedness). Uma vez definida a região de interesse (ROI) foram produzidas as Curvas de Atividade de Entrada e Resultante requeridas no processo de análise da dinâmica de compartimentos com o qual foi obtida a quantificação do metabolismo do órgão ou estrutura de estudo. Finalmente, de uma maneira semelhante imagens PET/CT reais fornecidas pelo Instituto do Coração (InCor) do Hospital das Clínicas da Faculdade de Medicina da Universidade de São Paulo (HC-FMUSP) foram analisadas. Portanto, concluiu-se que a etapa de filtragem tridimensional usando o filtro Anscombe/Wiener foi relevante e de alto impacto no processo de quantificação metabólica e em outras etapas importantes do projeto em geral.

**Palavras-chave:** Medicina Nuclear, Tomografia por Emissão de Pósitrons (PET), Tomografia Computadorizada (CT), Processamento de imagens tridimensionais, Segmentação de imagens tridimensionais, Quantificação Metabólica, *Phantoms* antropomórficos, Imagens PET/CT reais.

## ABSTRACT

The presence of Nuclear Medicine as a medical imaging modality is one of the main procedures utilized nowadays in medical centers, and the great advantage of that procedure is its capacity to analyze the metabolic behavior of the patient, resulting in early diagnoses. However, the quantification in Nuclear Medicine is known to be complicated by many factors, such as degradations due to attenuation, scattering, reconstruction algorithms and assumed models. In this context, the goal of this project is to improve the accuracy and the precision of quantification in PET/CT images by means of realistic and well-controlled processes. For this purpose, we proposed to develop a framework, which consists in a set of consecutively interlinked steps that is initiated with the simulation of 3D anthropomorphic phantoms. These phantoms were used to generate realistic PET/CT projections by applying the GATE platform (with Monte Carlo simulation). Then a 3D image reconstruction was executed, followed by a filtering process (using the Anscombe/Wiener filter to reduce Poisson noise characteristic of this type of images) and, a segmentation process (based on the Fuzzy Connectedness theory). After defining the region of interest (ROI), input activity and output response curves are required for the compartment analysis in order to obtain the Metabolic Quantification of the selected organ or structure. Finally, in the same manner real images provided from the Heart Institute (InCor) of Hospital das Clínicas, Faculty of Medicine, University of São Paulo (HC-FMUSP) were analysed. Therefore, it is concluded that the three-dimensional filtering step using the Ascombe/Wiener filter was preponderant and had a high impact on the metabolic quantification process and on other important stages of the whole project.

**Keywords:** Nuclear Medicine, Positron Emission Tomography (PET), Computed Tomography (CT), tri-dimensional image processing, segmentation of three-dimensional images, Metabolic Quantification, anthropomorphic phantoms, real PET/CT Images.

# SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b> .....	12
1.1	Considerações Gerais .....	12
1.2	Objetivos do Projeto.....	14
<b>2</b>	<b>REVISÃO DA LITERATURA</b> .....	15
2.1	Descrição da modalidade PET .....	15
2.2	Sistemas Híbridos PET/CT .....	20
2.3	GATE (Geant4 Application for Tomography Emission) .....	22
2.4	STIR (Software for Tomographic Image Reconstruction) .....	23
2.5	Conectividade Fuzzy de Imagens Digitais .....	27
2.6	Análise e Dinâmica de Compartimentos .....	28
<b>3</b>	<b>METODOLOGIA</b> .....	31
3.1	Conceitos Iniciais.....	31
3.2	Simulação de <i>phantoms</i> antropomórficos 3D e aquisição de imagens dinâmicas PET 3D .....	33
3.3	Geração das projeções realísticas usando o GATE .....	35
3.4	Reconstrução 3D das projeções adquiridas .....	46
3.5	Filtragem das imagens PET 3D .....	51
3.6	Segmentação 3D das regiões de interesse .....	54
3.7	Estimação dos parâmetros dos Modelos Compartimentais .....	58
<b>4</b>	<b>RESULTADOS</b> .....	74
4.1	Avaliação da sequência de processamento aplicada aos volumes simulados do <i>phantom</i> MASH .....	74

4.2 Estimação dos parâmetros metabólicos característicos do Modelo Compartimental que representa o coração do MASH .....	75
4.3 Estimação dos parâmetros metabólicos do Modelo Compartimental utilizando exames PET reais .....	76
<b>5 DISCUSSÃO</b> .....	79
5.1 Considerações relevantes .....	79
5.2 Importância e contribuições da pesquisa .....	83
<b>6 CONCLUSÕES</b> .....	85
<b>PUBLICAÇÕES</b> .....	86
<b>REFERÊNCIAS BIBLIOGRÁFICAS</b> .....	87
<b>APÊNDICE A</b> .....	94
MANUAL DE INSTALAÇÃO: GATE v.6.2 .....	94
<b>APÊNDICE B</b> .....	101
MANUAL DE INSTALAÇÃO: STIR v.2.4 .....	101
<b>APÊNDICE C</b> .....	104
CONECTIVIDADE FUZZY (FUZZY CONNECTEDNESS - FC) .....	104
<b>APÊNDICE D</b> .....	107
USO DA MACRO NA OBTENÇÃO DOS <i>PHANTOMS</i> MASH .....	107
<b>APÊNDICE E</b> .....	114
<i>PHANTOMS</i> 3D: FORMATO DE ENTRADA ASCII .....	114
<b>APÊNDICE F</b> .....	116
PROCESSO PARA GERAR OS FORMATOS ANALYSE/INTERFILE .....	116
<b>APÊNDICE G</b> .....	120
LINHAS DE COMANDO GATE: TOMÓGRAFO PET SIEMENS BIOGRAPH + PHANTOM ANTROPOMÓRFICO MASH (TÓRAX) .....	120

<b>APÊNDICE H</b> .....	128
INTERFACE GATE / STIR .....	128
<b>APÊNDICE I</b> .....	150
STIR: COMANDOS DE RECONSTRUÇÃO 3D .....	150
<b>APÊNDICE J</b> .....	154
ALGORITMO DE CORREÇÃO DE ATENUAÇÃO (PETac) .....	154
<b>APÊNDICE K</b> .....	158
NORMALIZAÇÃO .....	158
<b>APÊNDICE L</b> .....	165
COMKAT: MODELO PARA ESTUDO DO METABOLISMO EM IMAGENS PET REAIS DE PERFUSÃO MIOCÁRDICA COM RB-82 .....	165

# Capítulo 1

## INTRODUÇÃO

### 1.1 Considerações Gerais

Nos últimos anos e especialmente na década passada se têm dado avanços tecnológicos bastante significativos em todas as modalidades de imagiologia médica, com os quais se fez possível melhorar a acúcia do diagnóstico e consequentemente a prognose e o tratamento mais acertado das doenças. Com a Medicina Nuclear o panorama não tem sido diferente, foram desenvolvidos novos materiais detectores mais eficientes e mais precisos, máquinas com velocidade e capacidade superior estão permitindo reconstruções volumétricas em tempo real, assim como elementos radiofármacos mais específicos estão sendo continuamente desenvolvidos com o fim de se terem agentes diagnósticos específicos para cada tipo de tumor.

No entanto, uma das coisas que trouxe maior ganho foi o uso de sistemas combinados (híbridos) como o caso do PET/CT, propiciando uma interação interessante entre médicos nucleares e radiologistas, permitindo ter uma visualização mais clara das estruturas anatômicas e uma melhor compreensão das suas inter-relações funcionais [Robilotta, 2006].

Clinicamente, sabe-se da importância que representa para o médico especialista ter conhecimento da localização e posteriormente detectar alterações funcionais e metabólicas em um setor específico do corpo, sobretudo porque elas precedem às alterações anatômicas. O grande valor disso, é que as anomalias funcionais podem estar presentes em várias alterações neurológicas e psiquiátricas sem a presença de defeitos estruturais [De Almeida et al., 2002; Robilotta, 2006].

Nesse sentido, a Medicina Nuclear tem um destacado diferencial em muitos aspectos principalmente porque outorga a possibilidade de reconhecer alterações em um órgão, função e/ou estrutura específica do corpo, com peculiar sensibilidade e especificidade na detecção de doenças, conseguindo demonstrar precocemente desordens ou alterações bioquímicas mesmo onde não existe (ainda) uma anormalidade estrutural evidente, permitindo um diagnóstico mais precoce e, consequentemente, terapias com maior possibilidade de sucesso [Almeida, 2004; Ragheb, 2007; Blankenberg, 2002].

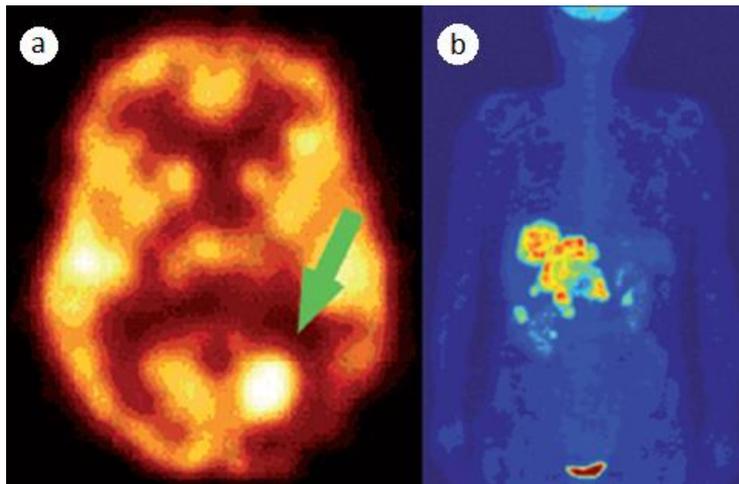
Assim, a função da Medicina Nuclear atualmente é diagnóstica, prognóstica e de controle, com um potencial muito importante em aspectos terapêuticos na tentativa de curar doenças ou câncer através da radioterapia [Carrió et al., 2003].

No diagnóstico, sua principal aplicação consiste na investigação de tumores e metástases, que naturalmente aumenta o consumo de glicose devido ao maior nível metabólico. Se o estudo é realizado em vários intervalos sucessivos de tempo, obtém-se uma distribuição temporal de imagens do marcador possibilitando estudos dinâmicos [Tinois, 2005].

Uma condição clínica que descreve claramente esse fato é, por exemplo, quando o Ventrículo Esquerdo do coração exibe uma diminuição na sua funcionalidade a qual é conhecida como disfunção do Ventrículo Esquerdo. Esse quadro clínico poderia levar à insuficiência cardíaca congestiva ou infarto do miocárdio, entre outras doenças cardiovasculares. Então, através da captação da glicose miocárdica (MGU), alterada em diversas doenças cardíacas, é possível avaliar quantitativamente a utilização focalizada de glicose miocárdica em pacientes com disfunção ventricular esquerda através da dinâmica FDG - PET [Morita et al., 2005; Hicks et al., 1991]. O  $^{18}\text{F}$  – fluordesoxiglucose ou simplesmente FDG, tem uma implicância especial porque possui uma funcionalidade biológica similar à da glicose convencional, nutriente das células. Uma vez no interior das células, a FDG se acumula nelas. As células que exibem um metabolismo acelerado, por exemplo, as células cancerosas em uma divisão descontrolada, o músculo cardíaco em contínua contração e as regiões mais ativas do cérebro, acumulam mais  $^{18}\text{F}$  e dão um sinal diferenciado nas imagens PET (Figura 1).

Por outro lado, existem também propósitos terapêuticos tal como o tratamento de hipertireoidismo, câncer de tireoide, desbalanceamento no sangue e o alívio da dor de certos cânceres de osso. Nesse sentido, a Medicina Nuclear é um método seguro e eficaz de obter informações que contrariamente estariam indisponíveis, ou só poderiam ser obtidas por técnicas sumamente invasivas e arriscadas tais como cirurgia ou biópsias [Ragheb, 2007; Blankenberg, 2002].

Atualmente a repercussão da engenharia na área da Medicina, especificamente na Medicina Nuclear, está em franca expansão tomando cada vez mais força e importância. Porém, ainda há inúmeros mecanismos fisiológicos que são desconhecidos ou pouco esclarecidos e a Medicina Nuclear como modalidade é um importante instrumento para investigar hipóteses e modelos de fenômenos biológicos.



**FIGURA 1** – (a) Imagem PET de um cérebro humano depois da administração de glicose marcada com FDG. Observa-se um tumor na parte inferior direita da imagem, indicada pela seta verde. (b) Imagem PET inteira do corpo com administração de FDG. As zonas que tem cor vermelho indicam maior captação de FDG, por exemplo, na bexiga (parte inferior) canal habitual de eliminação deste radiofármaco [Ballester, 2008].

Em virtude à carência de técnicas e ferramentas computacionais que permitam avaliar e estudar a dinâmica de processos a partir de imagens de Medicina Nuclear pretende-se contribuir com novas perspectivas, novas abordagens e ferramentas que permitam auxiliar, facilitar e melhorar o trabalho dos especialistas durante os procedimentos clínicos.

## 1.2 Objetivos do Projeto

Tendo como premissa a busca constante de um diagnóstico preciso no dia a dia da Medicina Nuclear, o objetivo principal do projeto é a pesquisa e o desenvolvimento de métodos mais precisos para a análise da dinâmica de processos fisiológicos baseado em PET. Em especial, pretendemos contribuir na análise da dinâmica de transporte e quantificação do metabolismo de imagens 3D de Medicina Nuclear na modalidade PET pelo uso de radiofármacos, os quais fornecem informações preciosas sobre a perfusão e o consumo destas substâncias pelos tecidos.

O projeto terá como foco de estudo o cérebro e o coração (miocárdio). No entanto, como ferramenta de aplicação clínica não ficará limitada a estes órgãos.

## Capítulo 2

# REVISÃO DA LITERATURA

Neste capítulo, tentaremos conhecer de perto quais são os processos atuais que são usados para avaliar e obter a quantificação metabólica tanto no cérebro quanto do coração, definidos como foco do nosso estudo. Uma visão geral do tema, assim como quais são seus principais problemas, desafios, aplicações atuais e suas perspectivas no futuro serão abordadas. Conhecer estes aspectos nos permitirá propor métodos alternativos, precisos, rápidos, robustos e com menos nível de invasividade comparados com os procedimentos atuais. Para facilitar o entendimento dos estudos relacionados a este tema de investigação, a revisão de literatura foi dividida nos seguintes tópicos:

### 2.1 Descrição da modalidade PET

A Tomografia por Emissão de Pósitrons ou também conhecida simplesmente como PET é uma modalidade que pertence à família de exames tomográficos da Medicina Nuclear. O PET é uma modalidade de imagem médica funcional, que proporciona informação sobre a fisiologia dos diferentes sistemas biológicos que conformam o corpo vivo, permitindo a identificação de perturbações metabólicas em células neoplásicas com capacidade de detecção tumoral, prevendo anormalidades anatômicas [De Almeida et al., 2002; Robilotta, 2006]. Esta modalidade é complementada com outras modalidades de obtenção de imagens que conseguem capturar informação morfológica com detalhes anatômicos e estruturais, permitindo uma melhor localização do problema identificado pelo PET [Gispert et al., 2010].

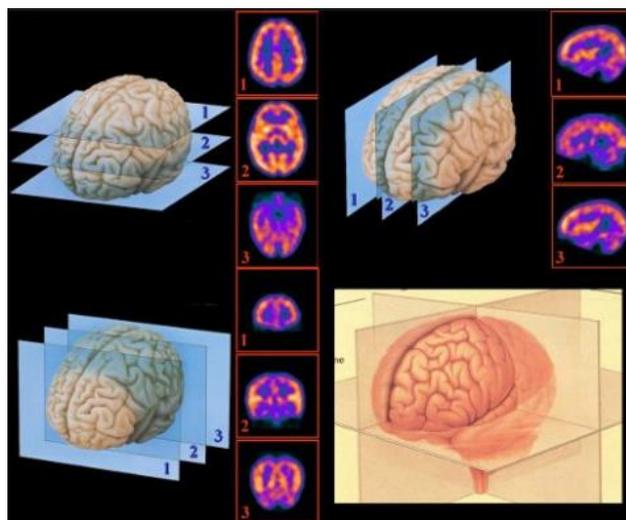
O principal fundamento e particularidade da modalidade PET consiste em utilizar uma substância metabolicamente ativa, que carrega um isótopo emissor de pósitron (por exemplo,  $^{11}\text{C}$ ,  $^{13}\text{N}$ ,  $^{15}\text{O}$  ou  $^{18}\text{F}$ ), cujo comportamento deseja-se seguir ou “traçar” [Oliveira et al., 2006]. Assim, dependendo do radiofármaco ou traçador utilizado, faz-se possível estudar distintos processos biológicos.

O radioisótopo mais utilizado é o  $^{18}\text{F}$  (Flúor 18) e o traçador, um análogo da glicose chamado fluor-desoxi-glicose (FDG) [Maisey, 2005]. Uma vez introduzida

no fluxo sanguíneo (por via endovenosa), a FDG é armazenada pela célula; no entanto, a diferença da glicose normal, como ela não pode ser utilizada para a produção de energia, fica presa dentro dela. Desta maneira, a captação do FDG das células é proporcional ao nível metabólico delas. Enquanto o FDG fica preso na célula, o radioisótopo  $^{18}\text{F}$  como é instável, sofre uma desintegração nuclear que origina a liberação de um pósitron o qual vai se aniquilar quase instantaneamente ao se combinar com um elétron perto dele (por terem cargas iguais e opostas), emitindo sua energia de repouso sob a forma de dois fótons de 511 keV que se propagam em sentidos opostos.

Para a detecção da emissão destes fótons, são utilizadas as denominadas câmaras de cintilação, conformadas por uma série de anéis de fotodetectores que circundam o paciente. Estes elementos tornam os fótons emitidos em luz visível e como estão rodeados por um conjunto de tubos fotomultiplicadores, os quais, no momento em que os fótons são detectados coincidentemente a  $180^\circ$  no anel, determinam a linha da aniquilação do pósitron [Wagner, 1998].

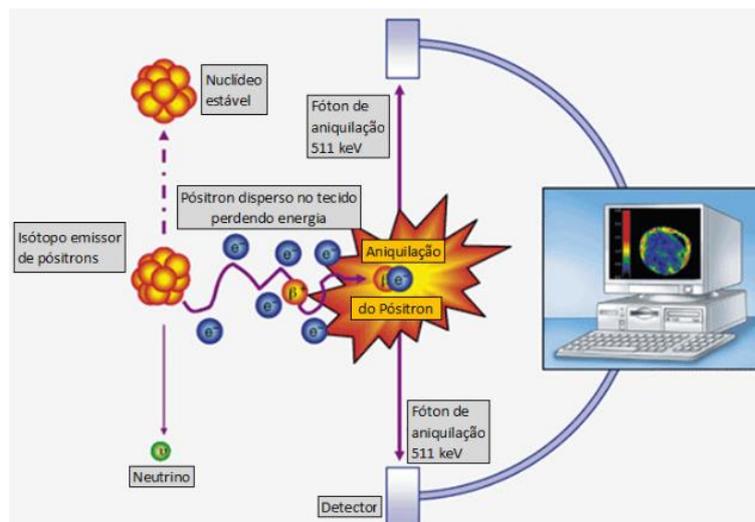
Partindo desses fatos coincidentes no anel de detectores, a distribuição do traçador é computacionalmente calculada a partir dos dados de projeção registrados. Essa distribuição é processada em um computador, e por meio dos procedimentos de reconstrução é possível reconstruir um volume formado por voxels que mostram o metabolismo da zona estudada e que são representados nas respectivas imagens transversas, coronais e sagitais [Fabey, 1996] como se mostra na Figura 2.



**FIGURA 2** – Exemplo do estudo cerebral PET de Medicina Nuclear na modalidade PET. Pode-se diferenciar as vistas coronal, sagital e axial do estudo ([www.alasbimnjournal.cl/](http://www.alasbimnjournal.cl/)).

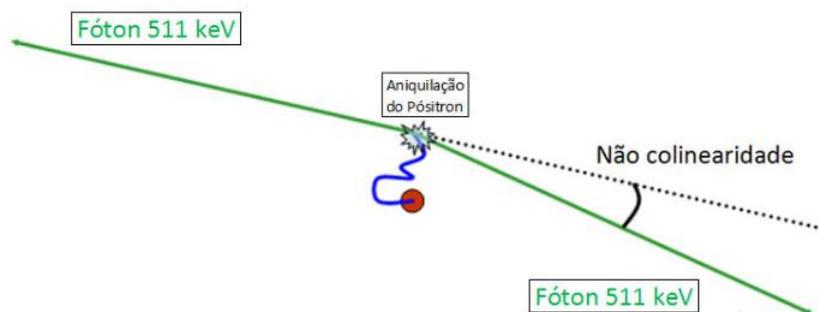
No entanto as imagens de Medicina Nuclear e em particular as imagens PET estão caracterizadas por apresentar uma pobre relação sinal-ruído, causada principalmente pela baixa contagem de fótons e pelo ruído Poisson, representativo deste tipo de imagens [King et al., 1983]. A seguir vamos mencionar algumas das principais peculiaridades desta modalidade, com o propósito de entender os pormenores no processo de geração das imagens PET:

- (a) **Alcance do pósitron** – constitui um fator limitante na resolução da imagem obtida por meio desta modalidade. Este parâmetro estabelece a distância percorrida entre o ponto de emissão do pósitron, uma vez que abandona o núcleo, e o ponto de aniquilação com o elétron produzindo dois fótons colineares e de igual energia, como apresentado na Figura 3 [Pozzo, 2005].



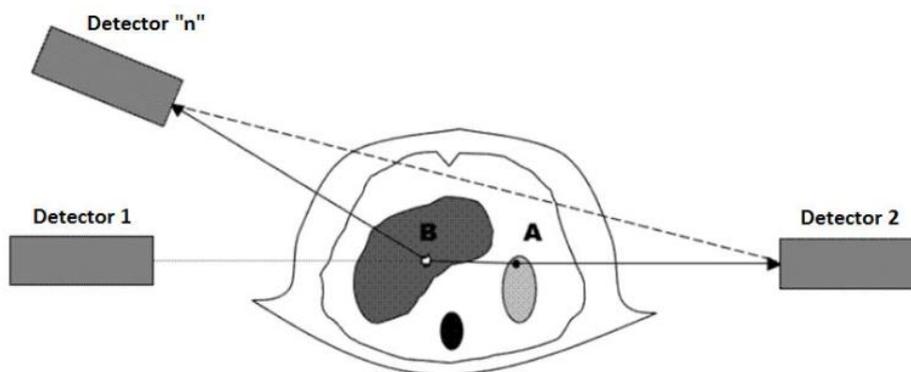
**FIGURA 3** – Esquema da emissão e aniquilação do pósitron. A modalidade PET é baseada neste processo físico (Imagem adaptada de <http://www.hardmob.com.br/threads/438088-Aniquila%C3%A7%C3%A3o-de-mat%C3%A9ria-e-antimat%C3%A9ria-poder%C3%A1-criar-laser-de-Raios-Gama/>).

- (b) **Não colinearidade** – é produzida uma vez que ocorre a aniquilação entre o pósitron e o elétron como se mostra na Figura 4, mas como o pósitron não tem perdido toda sua energia cinética, o momento linear e a energia vão ser conservados, com perda da colinearidade entre os dois fótons emitidos [López, 2008].



**FIGURA 4** – Esquema do processo de aniquilação com representação do efeito da não colinearidade [López, 2008].

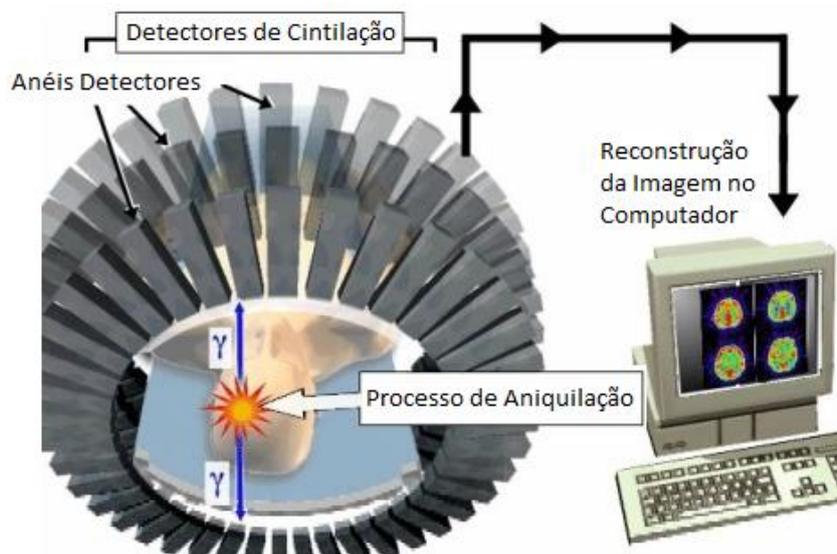
(c) **Dispersão Compton** – também conhecida como scattering é resultado de alguma interferência experimentada pelos fótons de aniquilação com algum órgão ou estrutura intermediária, originando-se uma queda da energia inicial e o desvio deste fóton com um determinado ângulo  $\alpha$ ; conseqüentemente, ativarão um detector diferente ao detector que teria incidido sem dispersão [López, 2008]. A ocorrência destes processos físicos degenera a qualidade da imagem resultante, devido às coincidências em linhas de resposta errôneas (Figura 5).



**FIGURA 5** – Dispersão Compton no Objeto emissor [López, 2008].

(d) **Atenuação** – Além do efeito Compton, produz-se também o que é conhecido como atenuação da imagem devido a que o fóton emitido pode ser absorvido, produto da interação com os elementos do meio, reduzindo o número de fótons detectados. Este fenômeno constitui também um fator degradante da imagem resultante [Pozzo, 2005].

**(e) Instrumentação** – A eletrônica da modalidade PET tem evoluído realmente de uma forma notável nos últimos anos tornando possível a obtenção de imagens PET de qualidade. No início, eram utilizados sistemas compostos por anéis simples com alguns detectores, produzindo imagens ralas e de apenas uma fina secção do paciente. Posteriormente, começaram a serem utilizados novos sistemas com múltiplos anéis (de 3 a 8 anéis de detectores como na Figura 6), e com diâmetro suficiente para aquisição de imagens de corpo inteiro. Hoje em dia, é comum o uso de tomógrafos PET de até 32 anéis de detectores, podendo-se gerar um conjunto de imagens tomográficas adquiridas simultaneamente constituídas pelas projeções tomográficas originadas por cada um dos anéis [Borges, 2010]. Projeções unidimensionais obtidas em diferentes ângulos tornam possível reconstruir uma imagem bidimensional, assim como também se pode fazer reconstruções tridimensionais a partir de projeções oblíquas [Sabbatini, 1997]. Outrossim, existe uma corrente de pesquisa muito forte na tentativa de testar novos materiais de detecção com a finalidade de melhorar a resposta temporal dos detectores, o qual significaria um incremento da resolução na imagem [Matej et al., 2008].



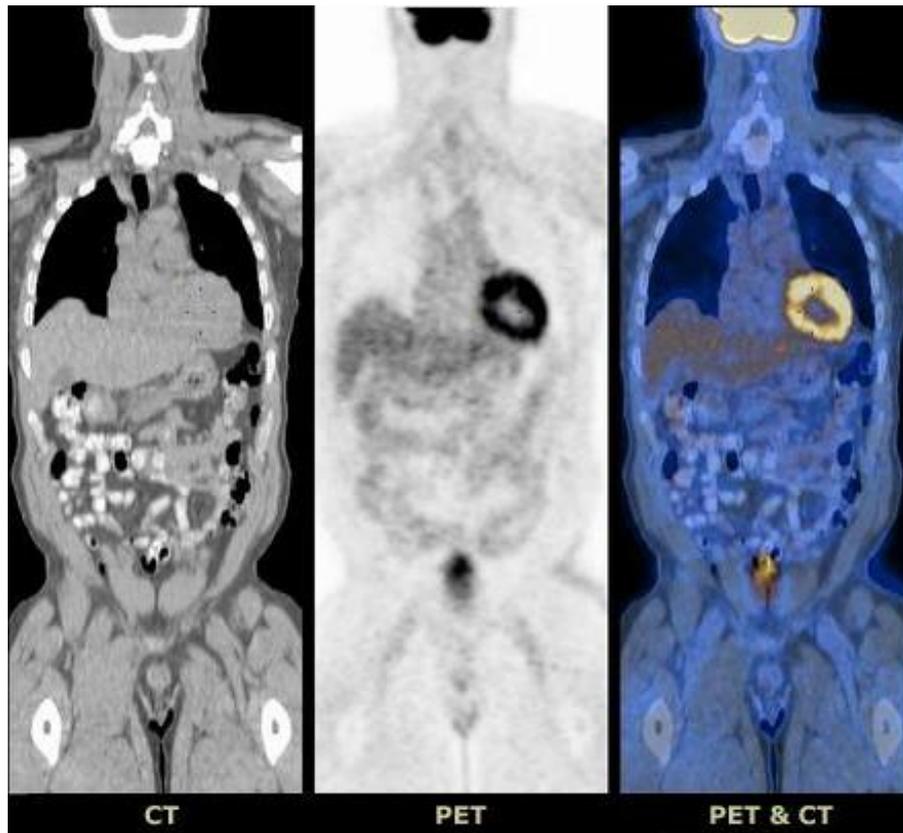
**FIGURA 6** – Os cristais convertem a radiação gama, emitidos do paciente, a fótons de luz, e os tubos fotomultiplicadores convertem e amplificam os fótons a sinais elétricos. Estes sinais elétricos então, são processados pelo computador para finalmente gerar a imagem dessa região (Imagem adaptada de <http://www.radiolog.pl/publikacje/7300/>).

(f) **Radioquímica** – Como fruto do desenvolvimento da física nuclear foi factível o surgimento e evolução da modalidade PET, que tornou possível a criação artificial dos radioisótopos que são utilizados na obtenção das imagens, como por exemplo, a FDG que é um análogo da glicose. Assim, uma série de traçadores podem ser produzidos a partir de um ciclotron (acelerador) ou utilizando um gerador. A principal diferença entre estes equipamentos é que os ciclotrons têm a capacidade de produzir vários tipos de radiofármacos ao mesmo tempo, enquanto os geradores produzem um único radiofármaco [Knapp; Mirzadeh, 1994]. Os radioisótopos mais utilizados na modalidade PET produzidos por ciclotrons são: o  $^{18}F$ , o  $^{11}C$  e o  $^{13}N$ , com tempos de meia-vida de 110, 20 e 10 minutos, respectivamente. Além desses, o  $^{82}Rb$  com um tempo de meia-vida de 76 segundos é outro agente amplamente usado nos estudos do fluxo miocárdico, sendo produzido por meio de um gerador [De Amorin, 2007].

## 2.2 Sistemas Híbridos PET/CT

No começo, a fusão de imagens médicas teve como propósito correlacionar diferentes tipos de imagens médicas adquiridas de forma não simultânea. Não obstante, esta abordagem originava erros de co-registo, principalmente devido às diferenças existentes no posicionamento do paciente nas várias aquisições, o que não permitiu explorar na sua totalidade as potencialidades de cada modalidade envolvida [Silva, 2012]. A fim de transpor este problema foi proposto o desenvolvimento de sistemas de imagem híbridos. É assim que durante a década de 90, os sistemas PET/CT foram os primeiros sistemas de imagem híbridos a serem propostos e serem introduzidos na prática clínica, constituindo no presente uma ferramenta relevante para a medicina moderna.

Os sistemas híbridos PET/CT combinam informações funcionais (procedente do PET) e anatômicas (procedente do CT) em uma única aquisição para oferecer uma localização aprimorada e uma perspectiva molecular das anomalias da estrutura analisada, como apresentado na Figura 7. A eficiência clínica deste tipo de sistemas fica evidenciada com o aproveitamento da mesma cama de aquisição, o mesmo computador de controle e a mesma estrutura, permitindo que os diferentes tipos de informação sejam adquiridos sequencialmente com o paciente na mesma posição e durante o mesmo procedimento, o que simplifica o processo de reconstrução, o registro e a fusão das imagens [Borges, 2010].



**FIGURA 7** – Exemplo de imagens de CT e PET do corpo inteiro, e a fusão PET/CT. Fonte: Multimodality Imaging in Cancer Management – <https://healthmanagement.org/>

Como vantagens, as imagens PET/CT possibilitam uma melhor interpretação das imagens obtidas na diferenciação de alterações patológicas; viabilizam diagnósticos de alta qualidade e imagens com um alto nível de detalhamento; permitem a detecção precoce da doença com possível redução de procedimentos de diagnósticos invasivos ou intervenção médica; permitem o monitoramento de recorrências da doença e a diferenciação entre tumores benignos e malignos; possibilitam um melhor planejamento e análise da eficácia do tratamento; fornecem um diagnóstico mais rápido em um único exame; propiciam um mapeamento preciso (localização anatômica precisa) da região a ser irradiada, especialmente nos casos em que o tratamento se baseia em radioterapia ou em sua associação à quimioterapia; assim como possibilitam a obtenção de um mapa de coeficientes de atenuação específico, entre outros. Contudo, o alto custo associado, as modificações necessárias nas instalações, as altas doses de radiação, e os artefatos provocados por metais são algumas das suas maiores desvantagens [Storne, 2011].

Apesar disso a utilização do PET/CT está sendo muito favorecido nas diversas aplicações médicas atuais, permitindo fazer diagnósticos mais precisos e planejar tratamentos mais adequados com menos efeitos secundários [Bruce, 2000]. O uso de PET-CT em Planejamento para Radioterapia permite o estudo da atividade metabólica para uma melhor definição da área a tratar por Radioterapia, minimizando a dose de radiação induzida a tecidos circundantes e normais. Assim, os importantes benefícios que o PET/CT confere à oncologia, neurologia, cardiologia e outras disciplinas são muito amplos e estão em constante evolução.

### 2.3 GATE (Geant4 Application for Tomography Emission)

As técnicas de simulação Monte Carlo (MC), tornaram-se conhecidas como um método numérico acurado devido ao modelamento detalhado dos processos físicos envolvidos nos tópicos da Medicina Nuclear tais como a determinação da distribuição de dispersão (scatter), desenho do colimador, e os efeitos de vários parâmetros que regulam a qualidade da imagem [Ljungberg, 2013]. Existem vários códigos de simulação Monte Carlo, sendo alguns dos mais conhecidos e utilizados os seguintes: **SimSET** (Simulation System for Emission Tomography), **EGS** (Electron Gamma Shower), **GEANT4** (GEometry ANd Tracking), **MCNP** (Monte Carlo NParticles), **PENELOPE** (PENetration and Energy LOss of Positrons and Electrons), **SIMIND**, **ASIM**, entre vários outros.

Nesse trabalho foi utilizado código de simulação Monte Carlo através da ferramenta **GATE** (*Geant4 Application for Tomography Emission*) que é bastante interessante já que integra um conjunto de pacotes com o objetivo de possibilitar a simulação realística das projeções PET com uma destacada proximidade com aquelas geradas com o próprio equipamento PET [Jan et al., 2004].

Em particular, o GATE foi desenvolvido para UNIX/LINUX e integra centenas de classes em C++ em uma estrutura de várias camadas. Atualmente a plataforma GATE pode ser instalada em quase todos os sistemas operacionais como Windows, Linux (Fedora, Ubuntu, RedHat), MacOSX, entre outros, cada uma delas com vantagens e desvantagens específicas no processo da instalação.

Uma vez avaliado o nosso cenário e propósito, optou-se por instalar a versão 6.2 do GATE sob o sistema operativo Linux – Fedora, com o qual foi possível reduzir

o tempo e inconvenientes no processo de instalação. A instalação está constituída basicamente por cinco (5) pacotes ou elementos:

- Geant4 (<http://geant4.web.cern.ch/geant4/collaboration/index.shtml/>), que engloba um conjunto de ferramentas para a simulação da passagem de partículas através da matéria. Suas áreas de aplicação incluem alta energia, física nuclear e do acelerador, bem como estudos em medicina e ciência espacial.
- CLHEP (<http://projclhep.web.cern.ch/proj-clhep/>) “Class Library for High Energy Physics”, biblioteca onde são alojadas as ferramentas de álgebra linear, geração de números pseudo aleatórios, vetores, matrizes, geometria, entre muitos outros.
- Root (<http://root.cern.ch/drupal/>) é um programa orientado a objetos que foi originalmente projetado para análise de dados de física com funcionalidades específicas para esse campo, mas também é utilizado em outras aplicações. O Root é uma ferramenta com mais de 1200 classes para acesso, análise, processamento e visualização de dados.
- LMF3 (<http://www.opengatecollaboration.org/>) “List Mode Format” é uma biblioteca de ferramentas para trabalhar com os arquivos de modo lista.
- ECAT7 (<http://www.opengatecollaboration.org/ECAT/>) é uma biblioteca de ferramentas, e constitui um dos arquivos de saída gerados pelo GATE contendo as projeções e sinogramas resultantes da simulação.

No Apêndice A está sendo disponibilizado o Manual de Instalação detalhado (passo a passo) da plataforma GATE na sua versão 6.2.

## **2.4 STIR (Software for Tomographic Image Reconstruction)**

STIR (<http://stir.sourceforge.net/>) é um software “open source” com uma biblioteca que reúne um conjunto de classes, funções e utilidades para o processo de conversão, manipulação, exibição e reconstrução de imagens PET 3D. O conjunto de bibliotecas que conformam o STIR são modulares e muito flexíveis. Isso torna relativamente fácil adicionar novos algoritmos, filtros, etc.

Com esse fim, temos instalado a versão 2.4 do STIR no Sistema Operativo Linux – Fedora 15. Desta forma, os arquivos de saída do GATE (ROOT, Interfile,

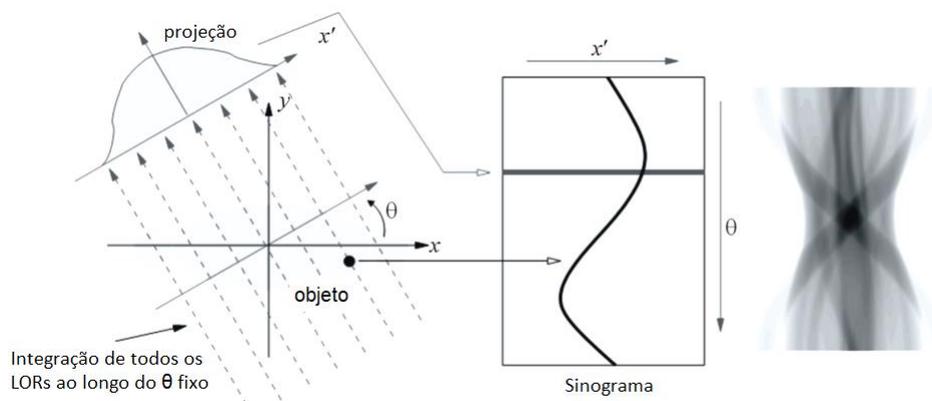
sinogram format, ECAT7, List Mode Format, entre outros) que contém as projeções e/ou os sinogramas podem ser reconstruídos pelo STIR através de diferentes tipos de algoritmos de reconstrução tanto 2D quanto 3D.

No Apêndice B está sendo disponibilizado o Manual de Instalação detalhado (passo a passo) do STIR na sua versão 2.4.

### 2.4.1 Técnicas de Reconstrução

As projeções representam o produto do processo de aquisição e são precisamente os dados utilizados por algoritmos específicos a fim de gerar as imagens tridimensionais (volume) das quais pode-se obter cortes tomográficos em qualquer nível e/ou plano, assim como avaliar a informação da atividade no espaço analisado.

No entanto, na maioria das vezes os dados são armazenados também em forma de sinogramas. Este tipo de armazenamento consiste na representação de projeções em linhas, ou seja, as LORs são descritas em função da orientação angular e da sua posição relativamente ao campo-de-visão do tomógrafo como é mostrado na Figura 8.



**FIGURA 8** – Ilustração de uma projeção e o seu correspondente sinograma obtidos em um processo PET. Um modelo de sinograma completo é mostrado à direita [Noori-Asl, 2013].

Os métodos de reconstrução se dividem em dois grupos bem diferenciados:

- (a) Métodos de reconstrução analíticos e,
- (b) Métodos de reconstrução iterativos.

### **a. Reconstrução Analítica**

Um dos métodos analíticos de reconstrução de imagens tridimensionais mais usados é a filtragem das projeções retrojetadas ou também conhecido como FBP (do inglês *Filtered Back Projection*). O FBP baseia-se em implementações discretas, onde a imagem estimada é diretamente calculada a partir dos dados. Este método de reconstrução soma as contribuições individuais a um valor de pixel (voxel) das contagens detectadas de todas as linhas de resposta que tem passado através desse pixel (voxel), no entanto, o benefício desse processo rápido de reconstrução se vê mitigado devido às propriedades superiores de supressão de ruído alcançado com os métodos de reconstrução iterativos [Alzimami; Sassi; Spyrou, 2009].

As imagens em Medicina Nuclear representam a distribuição espacial de emissões de radionuclídeo em termos de contagens em uma coordenada específica (x,y). Em processamento de imagens, é conveniente, muitas vezes, transformar as informações da imagem do domínio espacial para o domínio de frequências. Por exemplo, a transformada de Fourier pode ser usada para representar essas imagens como funções trigonométricas caracterizadas pela variação de frequências e amplitudes. Em outras palavras, a transformada de Fourier é uma ferramenta que permite que uma imagem seja dividida em diversos componentes, cada componente representando uma determinada frequência [Andrade, 2003].

Este método aplica a transformada de Fourier às projeções e, em seguida, um filtro-rampa no domínio da frequência. Finalmente é feita a retroprojeção das projeções filtradas [Tarantola, 2003]. Retroprojetar significa tomar o valor de um elemento adquirido em um ângulo  $\theta$  e colocar este valor em todos os elementos ao longo de uma linha no ângulo  $\theta$  da matriz da imagem. Todos os elementos na linha são preenchidos com este mesmo valor, porque, neste estágio, a distribuição real da fonte ao longo da linha é desconhecida. Este processo é repetido para todos os elementos captados em todos os ângulos  $\theta$  e os valores retrojetados reforçam uns aos outros em determinada localização.

Uma desvantagem das imagens em Medicina Nuclear reconstruídas por FBP é a amplificação do ruído Poisson inerente às imagens, que é uma componente de alta frequência sempre presente [Andrade, 2003]. Este problema pode ser solucionado pela combinação do filtro rampa com uma função “janela” que reduz o peso dos componentes das altas frequências.

## **b. Reconstrução Iterativa**

O princípio dos algoritmos iterativos se baseia na busca de uma solução por meio de sucessivas estimativas (iterações). As projeções da estimativa atual são comparadas com as projeções medidas (originais) e o resultado desta comparação é utilizado para modificar a estimativa corrente. Os diversos algoritmos iterativos diferem no modo como as projeções são comparadas e no tipo de correção aplicada a cada uma [Andrade, 2003].

Os métodos iterativos, como o caso do OSEM (Ordered Subsets Expectation Maximization), são métodos não lineares com múltiplos parâmetros e incorporam a natureza discreta da amostragem de dados e, normalmente, algum modelo estatístico do processo de aquisição de dados. Os sistemas de equações, no entanto, são tão grandes que estes métodos tentam se aproximar a uma solução razoável por meio de uma série de estimativas sucessivas. Isto significa que os métodos são menos previsíveis e menos bem compreendido do que os métodos analíticos, mas a grande vantagem destes métodos é a melhor remoção de artefatos indesejáveis [Alzimami; Sassi; Spyrou, 2009].

Os métodos iterativos são métodos de reconstrução mais lentos e durante o processo geram uma série de estimativas sucessivas que convergem na solução do problema, ou seja, neste caso, após cada iteração feita, o valor da atividade calculado é cada vez mais próximo da realidade.

Para cada iteração são usados os valores calculados nas iterações anteriores. Uma das grandes vantagens dos métodos iterativos é o fato de poderem incorporar modelos de correção e poderem corrigir os dados durante a reconstrução, obrigando que todo tipo de correção dos dados seja aplicado antes da reconstrução [Ferreira, 2001].

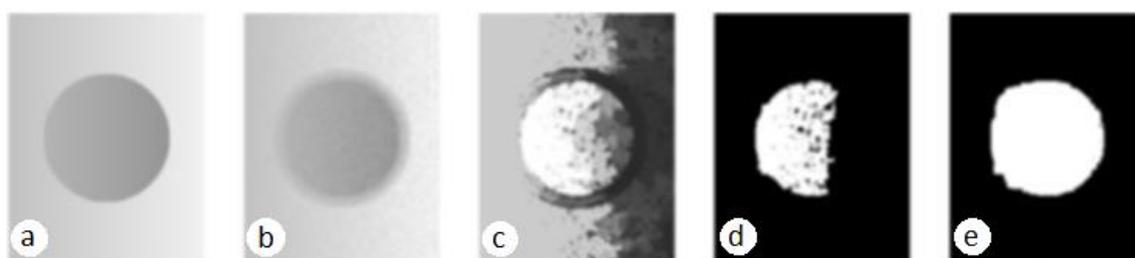
O método ML-EM (Maximum-Likelihood by Expectation Maximization) [Shepp; Vardi, 1982] é o método de reconstrução iterativo mais conhecido. Porém existe o algoritmo OSEM (Ordered Subsets Expectation-Maximization) que usa subgrupos (subsets) de dados e aplica o método EM em cada um dos grupos de forma sucessiva, sendo usado o resultado do subgrupo anterior aplicado no subgrupo seguinte. O método é tão rápido quanto maior o número de subgrupos, no entanto quando os subgrupos não têm informação suficiente, a convergência pode resultar em uma imagem diferente da imagem correta [Ferreira, 2001].

## 2.5 Conectividade Fuzzy de Imagens Digitais

O processo de segmentar um objeto dentro de uma imagem é uma tarefa complicada para um computador, quando a característica que difere o objeto do background é alguma propriedade da textura em vez da intensidade, ou quando a imagem está corrompida por ruído ou por uma iluminação não homogênea [Carvalho et al., 1999]. A existência dessas interferências na imagem exige dos algoritmos de segmentação um esforço adicional para identificar e separar as regiões de interesse presentes nas imagens [Gonzalez; Woods, 2008]. Na literatura encontramos uma variedade de métodos para este fim: Snakes [Kass et al., 1988; Xu, 1998], Level Set [Osher, 2002], Fuzzy Connectedness [Nyúl et al., 2002; Udupa, 2003; Pednekar, 2006], IFT [Falcão et al., 2004], Region Growing [Gonzalez; Woods, 2008], thresholding [Otsu, 1979], entre outros.

A Figura 9 mostra um exemplo de segmentação utilizando os métodos Region Growing e Fuzzy Connectedness. Em (a) se tem a imagem original simulada, em (b) a imagem inicial com ruído gaussiano incorporado, em (c) é mostrado o mapa de afinidades dos elementos da imagem. Na sequência, (d) foi aplicado o Region Growing, que apresentou ter perda de informação, já em (e) pode-se observar o resultado da segmentação utilizando o Fuzzy Connectedness [Udupa, 2003].

O Fuzzy Connectedness tem se mostrado eficiente na segmentação de imagens médicas, mesmo comparado com outros métodos [Kass et al., 1988; Xu, 1998], e dependendo da aplicação, ele é mais interessante, pois é um método robusto, principalmente quando se trata de imagens com intensidades não uniformes [Pednekar, 2006]. As definições da Teoria Fuzzy, assim como as funções, propriedades e relações que engloba esta teoria são detalhadas no Apêndice C.



**FIGURA 9** – Exemplo de segmentação do círculo na imagem simulada para demonstrar a efetividade da Conectividade Fuzzy em relação a outros métodos (a) Imagem simulada original, (b) Imagem com ruído Gaussiano, (c) Relação de afinidade de cada spel (spatial element), (d) Círculo segmentado por crescimento de região (percebe-se que uma parte dele foi perdida), (e) Círculo segmentado por Conectividade Fuzzy [Udupa; Saha, 2003].

## 2.6 Análise e Dinâmica de Compartimentos

Tecidos biológicos trocam material com outros tecidos (e.g. sangue), e eventualmente, com o meio ambiente de forma complexa. Estas trocas obedecem às leis de conservação de massa e energia, permitindo uma descrição matemática simplificada do sistema, o qual pode ser modelado como um conjunto de subsistemas chamados de compartimentos [Carson, 2005]. O compartimento é um conceito abstrato que não necessariamente está vinculado a um órgão. Ele pode representar um espaço físico separado por membranas, ou mesmo marcadores em uma forma específica, como o FDG aprisionado nas células e distribuídas pelo tecido. Supõe-se que cada compartimento é homogêneo [Carson; Cobelli, 2001] e pode ser caracterizado por um conjunto de equações diferenciais ordinárias e poucos parâmetros.

Num exame de PET dinâmico, as informações de interesse são obtidas pela estimativa de parâmetros que governam os modelos compartimentais. Estes parâmetros são calculados com base em certas informações da concentração do radiofármaco no plasma sanguíneo do paciente e nas regiões de interesse que estão sendo avaliadas, conhecidas como Curvas de Atividade [Huang, 1980].

A figura 10 ilustra um modelo de 3 compartimentos que reflete o transporte e metabolismo de glicose por um tecido (cérebro, miocárdio, etc.). A glicose é disponibilizada aos tecidos pelos vasos sanguíneos ( $C_a$ ). O compartimento 1 representa a concentração de glicose não-metabolizada no tecido e o compartimento 2 a glicose metabolizada pelas células do tecido de interesse. Os parâmetros  $K_1$  a  $k_4$  representam as taxas (constantes) de entrada/saída de marcadores nos compartimentos. Especificamente  $K_1$  representa a extração de glicose pelo miocárdio,  $k_2$  a relação de transporte devolvido aos vasos sanguíneos,  $k_3$  a relação metabólica convertida e  $k_4$  a constante do processo de desfosforilação. [Burckhardt, 2009; Carson; Cobelli, 2001].

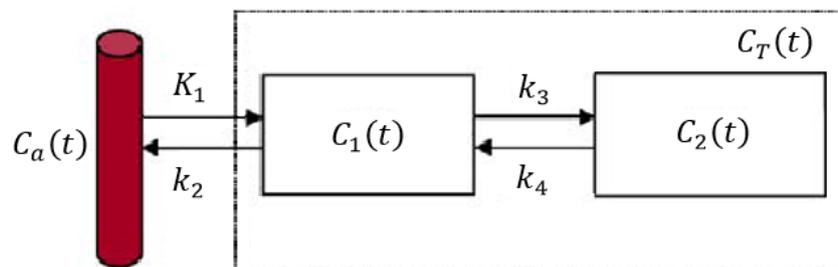


FIGURA 10 – Modelo Compartimental com 3 compartimentos [Burckhardt, 2009].

A utilização do FDG, que é um análogo da glicose, apresenta uma leve diferença bioquímica em relação à glicose, ainda assim permite também medir o metabolismo. A razão disto é que o FDG entra nas células de forma análoga à glicose, por meio das mesmas enzimas. No entanto, no caso do FDG o produto resultante se acumula na célula, enquanto que o da glicose segue o processo da transformação em energia. Este acúmulo pode ser medido, pois o  $^{18}\text{F}$  emite pósitrons e contribui para as contagens de fótons.

Tendo em conta a definição de conservação de massa e considerando que a taxa de saída do compartimento  $C_2$  é desprezível,  $k_4 \approx 0$ , as equações que descrevem este modelo são:

$$\frac{dC_1(t)}{dt} = K_1 C_a(t) - k_2 C_1(t) - k_3 C_1(t) \quad (1)$$

$$\frac{dC_2(t)}{dt} = k_3 C_1(t) \quad (2)$$

nos quais  $C_1(t)$  e  $C_2(t)$  representam as concentrações do marcador nos respectivos compartimentos.

A concentração de FDG é proporcional à atividade radioativa (TAC) medida. A soma das concentrações dos compartimentos  $C_1$  e  $C_2$  é obtida medindo-se o TAC em uma região do miocárdio (tecido) nas imagens PET ao longo do tempo ( $C_{\text{TAC}}(t) = C_1(t) + C_2(t)$ ), em virtude de que o tomógrafo não tem a faculdade de discernir entre cada um dos compartimentos e apenas consegue captar a concentração de atividade radioativa total no tecido alvo. A concentração de atividade de entrada  $C_a(t)$  pode ser obtida se medindo o TAC no plasma arterial [Simoncic; Jeraj, 2011].

Portanto, a estimativa dos parâmetros  $K_1$  a  $k_3$  é um problema de otimização envolvendo um sistema de equações diferenciais ordinárias, no qual, em um protocolo convencional, tem-se em torno de 20 pontos para  $C_T(t)$ . O critério pode ser a minimização do erro quadrático médio entre o  $C_{\text{TAC}}(t)$  medido e o  $C_{\text{TAC}}(t)$  estimado [Matis; Wehrly; Gerald, 1983].

Pela própria modelagem, percebe-se que os parâmetros denotam tanto o transporte de glicose para o tecido (perfusão), quanto o consumo do mesmo (metabolismo), que podem, portanto, ser estudados e quantificados.

Vários outros modelos podem ser projetados e desenvolvidos, dependendo do tecido biológico, da bioquímica do radiofármaco, e dos parâmetros físicos de interesse.

No mercado existem alguns grupos que desenvolveram softwares de avaliação para modelagem de compartimentos, entre os mais conhecidos na área da Medicina Nuclear podemos citar a BLD, RFit, Pk-Fit, KMZ/PKIN, SAAM/SAAM II e COMKAT [Muzic; Cornelius, 2001]. Muitos destes softwares têm algumas desvantagens que limitam sua utilidade (pagamentos de licença, entre outros). No caso do COMKAT (*Compartment Model Kinetic Analysis Tool*), é uma plataforma aberta, feita em MatLab, pode ser descarregada diretamente pela internet sem custo para os pesquisadores e principalmente se tem disponibilidade do código fonte o qual permite aos usuários adicionar recursos ou melhorias para atender às suas aplicações específicas. Pelas razões acima citadas, no projeto foi estabelecido a utilização do COMKAT (Figura 11).

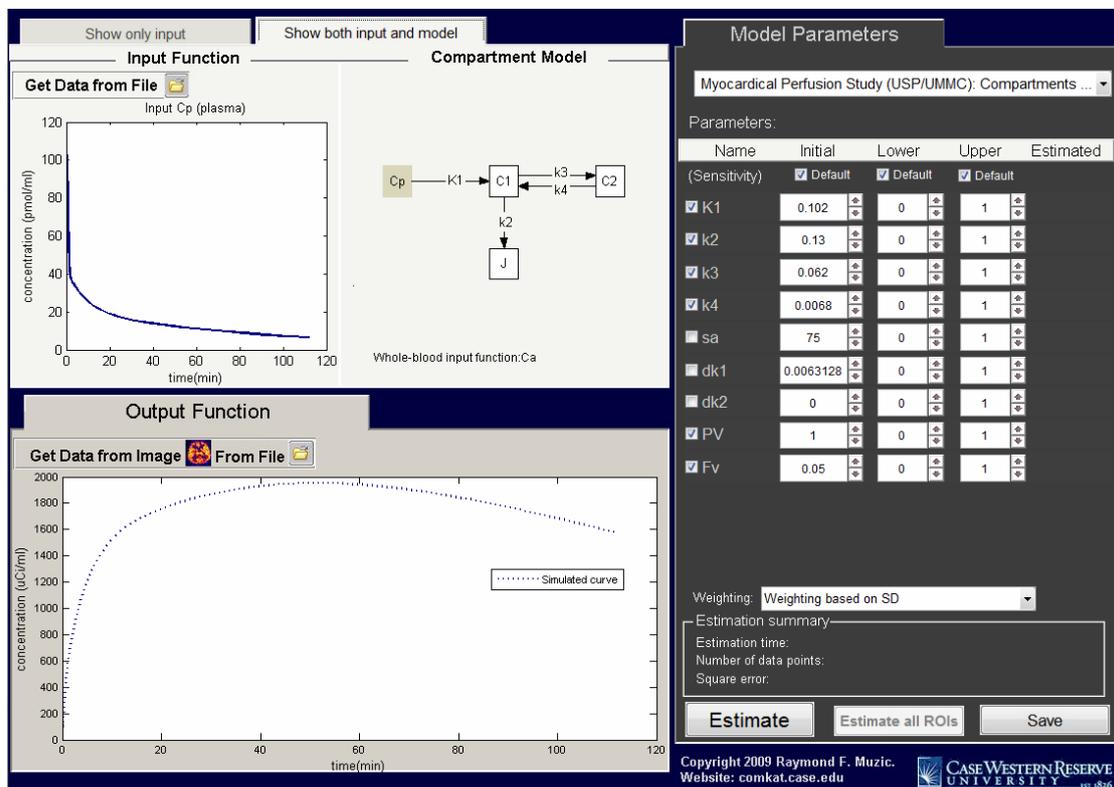


FIGURA 11 – Entorno da plataforma COMKAT v3.2. Enlace: <http://comkat.case.edu/>.

# Capítulo 3

## METODOLOGIA

### 3.1 Conceitos Iniciais

Como indicado na secção 1.2 o propósito principal do projeto é o desenvolvimento de métodos para a análise da quantificação da dinâmica de transporte e de metabolismo de radiofármacos. Por meio deste processo será factível realizar uma avaliação do consumo das substâncias pelos tecidos, com o suporte da opinião dos especialistas. Para esse fim, nosso projeto está integrado por uma sequência de passos modulares e inter-relacionados, os quais foram implementados como parte do projeto e serão pormenorizados a seguir no presente capítulo.

O diagrama de blocos da Figura 12 resume de forma geral as partes que fazem parte do trabalho proposto, esboçando de forma global e estruturada a sequência dos módulos que compõem a plataforma para a Análise da Dinâmica e Quantificação Metabólica de imagens de Medicina Nuclear na modalidade PET/CT.

Os principais tópicos de interesse que serão abordados como parte deste projeto são:

- a) Simulação de *phantoms* Antropomórficos 3D dinâmicos e aquisição de imagens funcionais PET 3D reais ao longo de um período de tempo.
- b) Geração das projeções realísticas que seriam obtidas em um equipamento PET/CT, utilizando a plataforma GATE que utiliza simulação de Monte Carlo para a detecção de fótons.
- c) Reconstrução de imagens 3D dinâmicas.
- d) Pré-processamento por meio de Filtros 3D específicos.
- e) Processo de Segmentação de estruturas de interesse em imagens PET 3D.
- f) Geração da Curva de Atividade de Entrada e Curva de Atividade Resultante.

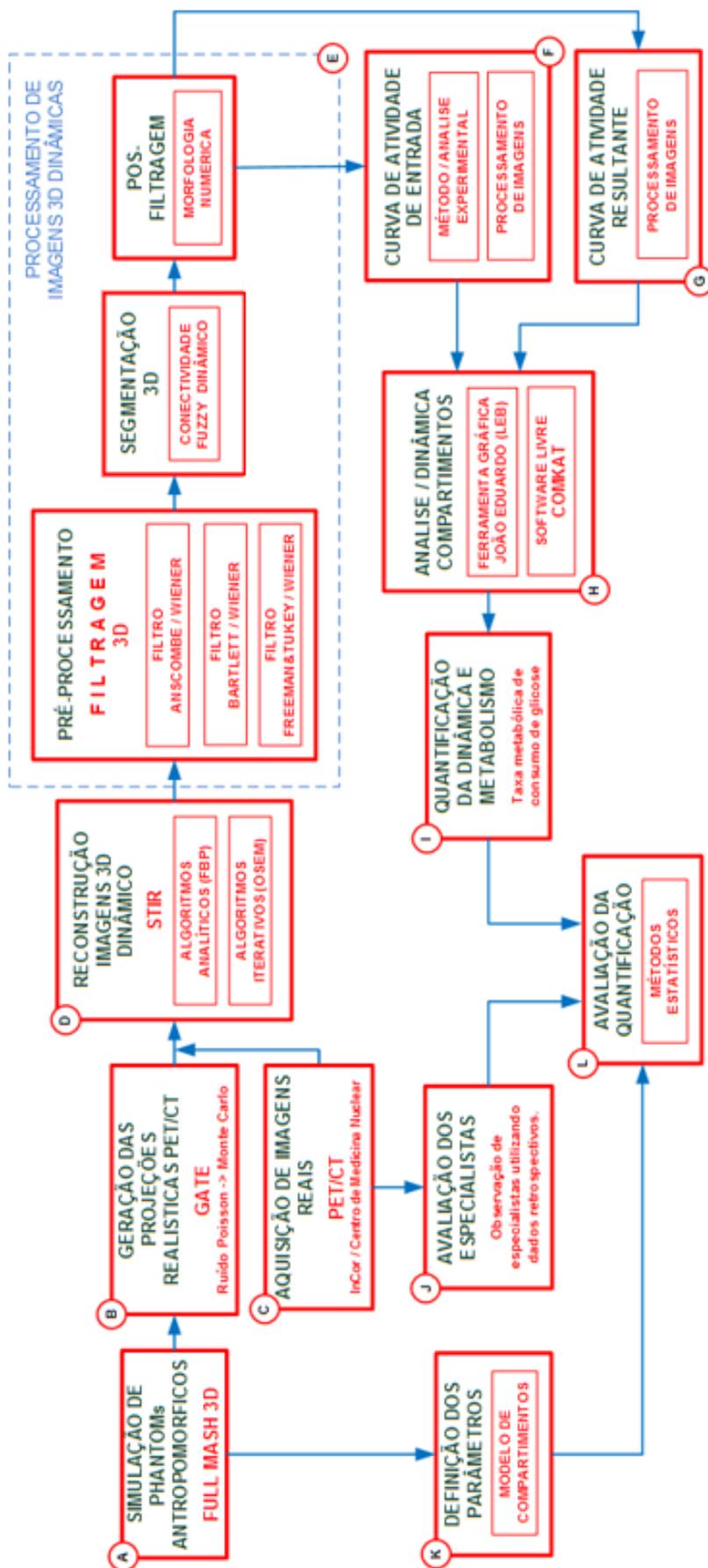


FIGURA 12 – Diagrama de Blocos Geral do Projeto.

- g) Análise da dinâmica de compartimentos a partir dos sinais (séries) de entrada e saída.
- h) Quantificação da dinâmica e do metabolismo.
- i) Avaliação das quantificações metabólicas e das análises dinâmicas.

Para o desenvolvimento dos algoritmos representados por cada um dos módulos da Figura 12, foi utilizado um computador Marca Dell, Modelo Precision T7600, Intel® Xeon® E5-2620, 6 núcleos @ 2.00GHz, Sistema Operacional de 64 Bits, com 32 GBytes de memória RAM e 1 TBytes de disco rígido, propriedade do Laboratório de Engenharia Biomédica da EPUSP.

Além disso, os softwares utilizados no desenvolvimento deste projeto foram basicamente GATE, ImageJ, (X)MedCon, STIR, C/C++, COMKAT, Java/Eclipse e MatLab®.

### **3.2 Simulação de *phantoms* antropomórficos 3D e aquisição de imagens dinâmicas PET 3D**

A investigação proposta não envolve novos experimentos com animais ou seres humanos, uma vez que utilizaremos dados retrospectivos, isto é, sequência de imagens 3D dinâmicas arquivadas no prontuário eletrônico do InCor, além de *phantoms* numéricos 3D.

Como indicado, o projeto tem como foco de estudo o cérebro e o coração (miocárdio), no entanto, como ferramenta de aplicação clínica poderá ser utilizada em qualquer outro órgão ou região do corpo humano ou animal.

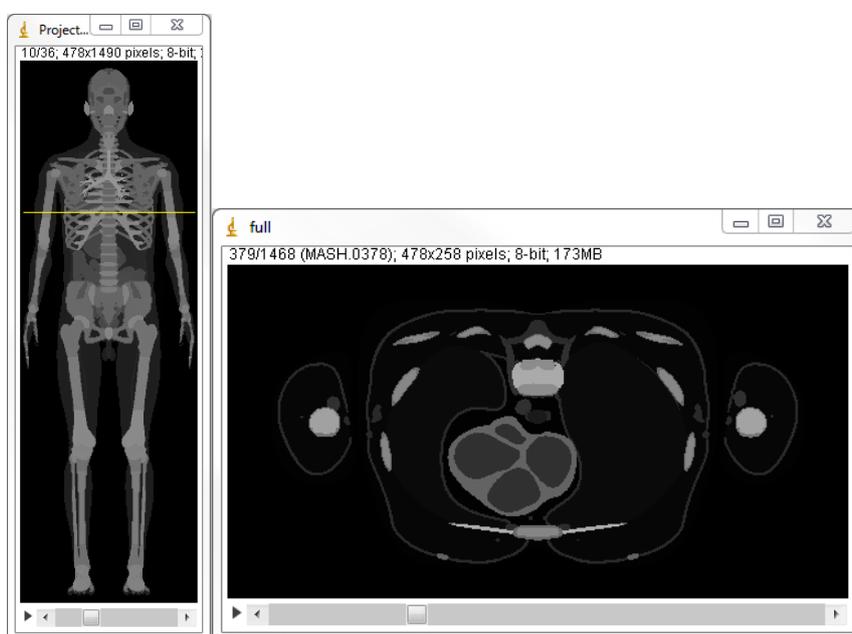
Existe uma variedade de processos e/ou elementos que poderiam ser utilizados nesta primeira etapa, no entanto, para nosso estudo serão considerados só dois conjuntos de bases de imagens 3D que estão detalhados a seguir:

#### **3.2.1 *Phantom* antropomórfico 3D**

Os *phantoms* desempenham um papel preponderante nas pesquisas com imagens médicas, já que têm a capacidade de simular um número ilimitado de anatomias conhecidas dos pacientes, oferecendo meios práticos para avaliar, comparar e melhorar estratégias usando imagens médicas [Segars; Tsui, 2009].

Existem diversos *phantoms* que são utilizados na pesquisa, alguns deles foram testados ao longo do nosso projeto tal como o *Phantom* 3D do cérebro [Hoffman et al., 1990], o *Phantom* 3D do ventrículo esquerdo do coração (NCAT-4D) [Sergars, 2009], o *Phantom* 3D do corpo inteiro (MASH), entre outros.

No entanto, neste projeto será pormenorizado o uso do *Phantom* antropomórfico 3D do corpo inteiro chamado MASH (Male Adult meSH). O MASH (Figura 13), é um *phantom* com superfície de malha poligonal que contém os mais relevantes dados anatômicos, incluindo a maioria dos órgãos e tecidos internos, assim como também o sistema esquelético integral formado por osso cortical, esponjoso, medula óssea e cartilagem [Kramer et al., 2010].



**FIGURA 13** – *Phantom* antropomórfico MASH do corpo completo de um homem adulto. Fonte: do artigo *FASH and MASH: female and male adult human phantoms based on polygon mesh surfaces: II. Dosimetric Calculations*. [Kramer et al., 2010].

Na secção 3.3.3, apresenta-se o *phantom* MASH completo no interior do tomógrafo PET modelado para exemplificar que o estudo pode ser realizado em qualquer região do corpo.

No entanto, para não sobrecarregar a simulação nem o tempo de aquisição dos órgãos de interesse, pode-se separar a região específica que pretendemos estudar (e.g. cérebro ou coração) junto com as estruturas vizinhas nessas regiões. No caso da região do coração que temos utilizado, o *phantom* tem uma

resolução de 478 x 258 x 90, largura do pixel 1,2 mm/pixels, espaçamento entre fatias 1,2 mm e volume do voxel de 1,728  $mm^3$ .

No Apêndice D é detalhado o uso da macro em MatLab® na obtenção do *phantom* MASH completo assim como os passos para a simulação de uma região específica do corpo (i.e. cabeça para estudo do cérebro, ou da região do tórax para o estudo do coração).

### **3.2.2 Imagens PET 3D Reais**

O segundo grupo está constituído por imagens PET/CT reais, do coração, que foram fornecidas pelo Instituto do Coração (InCor) do Hospital das Clínicas da Faculdade de Medicina da Universidade de São Paulo (HC-FMUSP).

Neste ponto, é importante indicar que temos conseguido a permissão do Diretor do Serviço de Medicina Nuclear e Imagem Molecular do InCor-HC-FMUSP para utilizar um grupo de imagens PET/CT que correspondem ao projeto interno de pesquisa do InCor CAPPESQ 30/2011 cujo título é Estudo de Perfusão Miocárdica utilizando  $^{82}Rb$ .

Todo o material fornecido pelo InCor conta com a devida permissão da Comissão de Ética do Hospital das Clínicas da FMUSP, CONEP N° 16814 realizado na Faculdade de Medicina da Universidade de São Paulo (FMUSP).

## **3.3 Geração das projeções realísticas usando o GATE**

### **3.3.1 MODELAGEM DO TOMÓGRAFO PET REAL**

Partindo da premissa que no nosso projeto é contemplada a análise de dois grupos de imagens PET 3D: Simulações usando *Phantoms* Antropomórficos e Imagens Reais, é indispensável considerar a modelagem de um tomógrafo PET real para realizar o processo de aquisição das projeções simuladas. Com o suporte do pessoal do Centro de Medicina Nuclear (CMN) foi possível conseguir os dados técnicos e de arquitetura do aparelho PET/CT real. Trata-se do tomógrafo PET/CT marca Siemens, modelo BiographTM.

Nos manuais e dados técnicos do aparelho foram identificados os dados mais relevantes da arquitetura do tomógrafo PET. Os dados (dimensões dos detectores, material dos detectores, distribuição dos anéis, módulos e dos

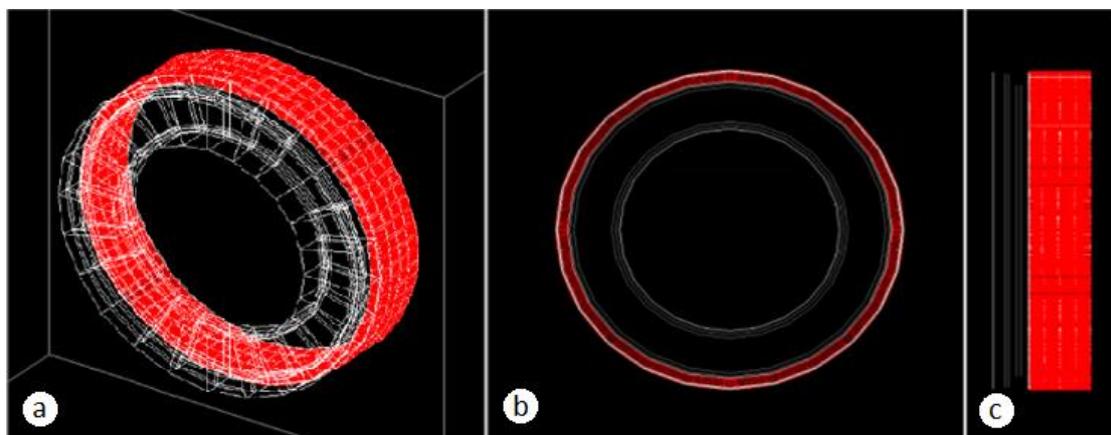
detectores, etc.) foram incorporados nos arquivos de simulação gerando um tomógrafo próximo ao usado no processo clínico real.

Na Tabela 1 são mostrados os principais parâmetros do tomógrafo Siemens Biograph™ utilizados (dados indicados dentro da caixa vermelha) tais como o material do detector que é do tipo BGO, o número de anéis, o número de cristais em cada anel, as dimensões dos cristais e o campo de visão axial (FOV).

**TABELA 1** – Dados da arquitetura do tomógrafo Siemens Biograph™, a imagem foi tirada de um dos manuais do tomógrafo Siemens onde aparecem as especificações de dois tomógrafos diferentes: BGO e LSO. A arquitetura usada neste projeto está dentro da caixa vermelha.

Specifications	BGO	LSO
Number of crystal rings	32	24
Number of crystals/ring	576	384
Total number of crystals	18,432	9,216
Crystal size, mm	4.05 x 4.39 x 30	6.45 x 6.45 x 25
Axial FOV, mm	155	162

Já no Apêndice G são apresentadas as linhas de comando que foram utilizadas na modelagem do tomógrafo Siemens Biograph™ no GATE. Na sequência, a Figura 14 mostra três diferentes vistas (diagonal, frontal e lateral) do tomógrafo PET modelado.



**FIGURA 14** – Tomógrafo Siemens Biograph™ modelado usando a plataforma GATE: (a) vista diagonal, (b) vista frontal, e (c) vista lateral.

### 3.3.2 VOXELIZAÇÃO DOS *PHANTOMS* ANTROPOMÓRFICOS

O primeiro passo para gerar as projeções realísticas através do GATE é realizar um processo de acoplamento do *phantom* antropomórfico no interior do tomógrafo PET modelado. Posteriormente, por meio de um processo de voxelização foi determinado o objeto de interesse onde foi alojado o elemento radioativo ou radiofármaco com o objetivo de simular o processo de *tracking* dos fótons emitidos desde a região escolhida, com a posterior obtenção das respectivas projeções as quais foram utilizadas subsequentemente em um processo de reconstrução.

Para atender o primeiro requisito, devemos notar que o *phantom* antropomórfico MASH está representado por uma matriz de coeficientes (em formato ASCII) que conotam os valores das intensidades de cada um dos elementos que conformam as diversas estruturas. No Apêndice E foram especificados minuciosamente os passos para gerar volumes de entrada no formato ASCII.

Embora os formatos dos arquivos de entrada aceitos pela plataforma GATE são o formato **ASCII** (\*.txt / \*.dat), o formato **Analyse** (\*.hdr / \*.img) e o formato **Interfile** (\*.h33 / \*.i33), é recomendável usar o formato Interfile para garantir o sucesso de etapas subsequentes, principalmente porque é necessário rever se os parâmetros do *phantom* estão corretos (i.e. dimensões dos voxels, tipo das imagens, ordem dos bytes, etc.). Para esse fim, no Apêndice F estão pormenorizados os passos para gerar volumes de entrada no formato Interfile.

Por outro lado, deve-se indicar que o *phantom* utilizado está formado por um conjunto de materiais diferentes, pelo que é necessário identificar cada um destes materiais (associados a seus valores de ID conforme à lista apresentada na tabela 12) e diferencia-los através de uma cor particular utilizando coeficientes RGB. No caso do processo de voxelização da caixa torácica do *phantom* antropomórfico MASH que contém o coração, foi utilizado o arquivo **range\_mash\_heart.dat** como é mostrado na Tabela 2, o qual representa o volume da Figura 45 mostrado no Apêndice E.

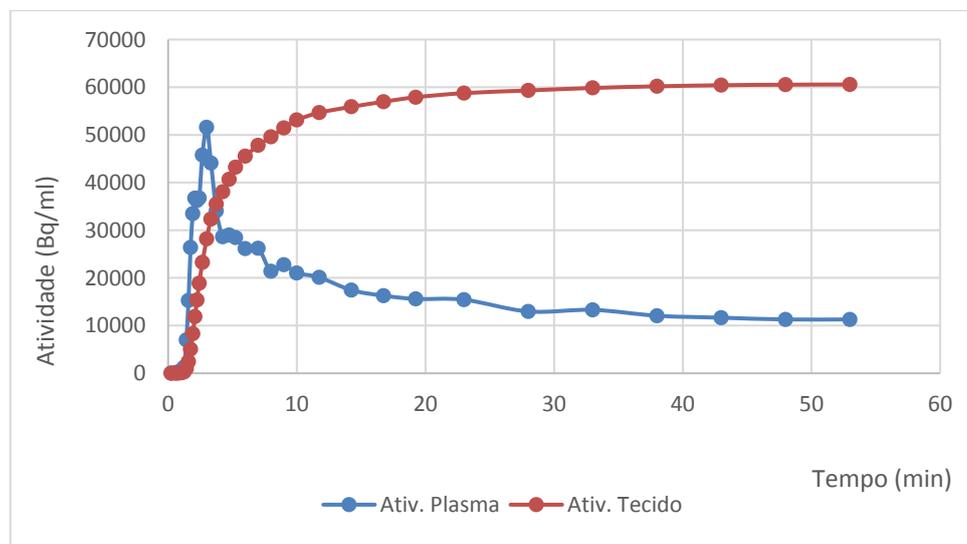
A primeira linha da tabela indica o número total de materiais (ou seja, o número de linhas que apareceram subsequentemente). As linhas seguintes contêm os dados e/ou parâmetros ligados a cada um dos materiais: valor de intensidade do material, nome do material, opção de visibilidade do material (verdadeiro ou falso) e valores para atribuição de cor (vermelho [R], verde [G], azul [B], alfa).

**TABELA 2** – Dados do arquivo **range\_mash\_heart.dat** que indica o conjunto de elementos ou regiões que fazem parte do processo de simulação PET.

range_mash_heart.dat							
12							
0	0	Air	false	0.0	0.0	0.0	0.2
10	10	Lung	true	0.0	0.0	1.0	0.2
42	42	Skin_1	true	0.5	0.2	0.4	0.2
43	43	Adipose_tissue_1	true	0.3	0.3	0.3	0.2
48	48	Heart_blood_filled	true	0.0	1.0	0.0	0.2
99	99	Heart	true	1.0	0.0	0.0	0.2
104	104	Cortical_bone	true	0.1	0.2	0.3	0.2
117	117	Muscle_skeletal_2	true	0.3	0.2	0.1	0.2
134	134	Cartilage	true	0.3	0.5	0.3	0.2
140	140	SpineBone	true	0.3	0.3	0.3	0.2
163	163	Muscle_skeletal_1	true	0.8	0.1	0.4	0.2
174	174	RibBone	true	0.4	0.6	0.2	0.2

Posteriormente, cada material foi associado a um valor de atividade (em becquerel [Bq.]) selecionado pelo usuário, assim como a valores de atenuação que o GATE aplica dentro do processo de simulação para cada tipo de material de acordo com a sua biblioteca interna de materiais, juntamente com a biblioteca de materiais que formam parte do *phantom* MASH.

Com a finalidade de verificar que as simulações PET realísticas e os processos posteriores estão corretos foi tomada uma lista de valores de atividade real ao longo do tempo no plasma  $C_p(t)$  e no tecido  $C_{TAC}(t)$ , representadas pela Figura 15 e listadas na Tabela 3.

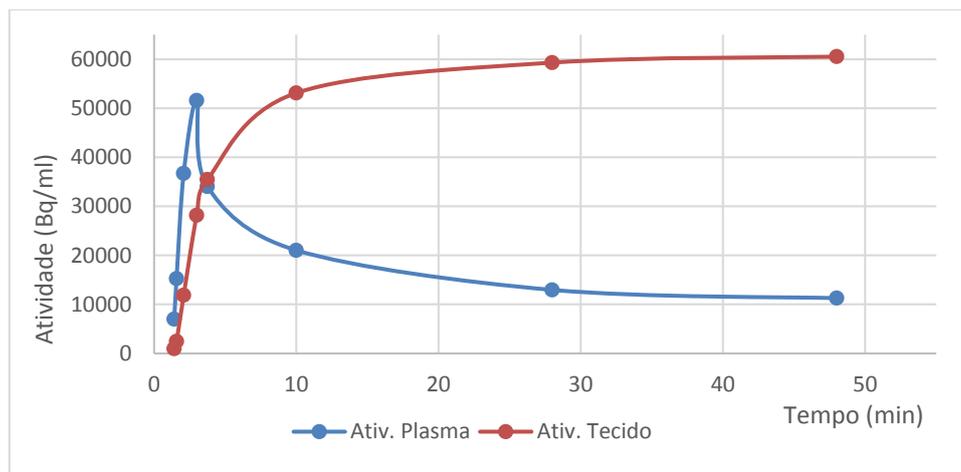


**FIGURA 15** – Curvas de Atividade de Entrada  $C_p(t)$  e Curva de Atividade no Tecido  $C_{TAC}(t)$ .

**TABELA 3 –** Curva de Atividade de Entrada no plasma e Curva de Atividade Resultante no tecido (miocárdio), ambas curvas em função do tempo.

Tempo (min)	Curva de Atividade de Entrada $C_p(t)$ (Bq/ml)	Curva de Atividade Resultante $C_{ta}(t)$ (Bq/ml)
0.250	34.828	3.448
0.583	48.149	8.171
0.750	216.250	29.498
0.917	404.411	69.210
1.083	675.894	135.420
1.250	1261.590	259.030
1.417	6984.485	948.100
1.583	15269.469	2449.300
1.750	26362.637	5033.900
1.917	33484.479	8299.900
2.083	36731.350	11862.000
2.250	36233.318	15350.000
2.417	36746.580	18867.000
2.667	45766.102	23255.000
3.000	51600.446	28187.000
3.333	44104.127	32343.000
3.750	34033.075	35480.000
4.250	28619.404	38069.000
4.750	28979.496	40682.000
5.250	28455.617	43226.000
6.000	26158.438	45522.000
7.000	26205.307	47800.000
8.000	21387.696	49573.000
9.000	22751.693	51452.000
10.000	21028.367	53123.000
11.750	20101.691	54664.000
14.250	17423.367	55897.000
16.750	16260.698	56963.000
19.250	15583.554	57904.000
23.000	15418.741	58766.000
28.000	12950.850	59321.000
33.000	13287.180	59851.000
38.000	12053.021	60194.000
43.000	11640.720	60425.000
48.000	11270.424	60537.000
53.000	11280.143	60575.000

Entretanto, devido ao tempo que as simulações levam para serem concluídas, seria inviável realizá-las para todos os instantes presentes na lista de valores de atividade de entrada e resultante. Por esse motivo, na Figura 16 estão mostrados oito pontos selecionados que são capazes de esboçar de maneira satisfatória a forma das curvas de atividade no plasma e no miocárdio para serem simulados, sendo que cada instante corresponde a uma simulação diferente.



**FIGURA 16** – Curvas de Atividade (Entrada e Resultante) reduzidas a 8 instantes de tempo.

Com os valores de atividade da Figura 16, o próximo passo foi calcular as atividades para cada estrutura no processo de simulação. Para isso, foi necessário realizar primeiro uma conversão das unidades, ou seja, passar de Bq/ml para Bq/voxel. Esta conversão foi realizada de acordo com as informações fornecidas pelos desenvolvedores do *phantom* MASH, fundamentalmente utilizando as dimensões de cada voxel que são 0.12x0.12x0.12 cm (Tabela 4).

**TABELA 4** – Curva de Atividade de Entrada no plasma e Curva de Atividade Resultante no tecido para cada instante a ser simulado.

Tempo (min)	Curva de Atividade no Plasma (Bq/voxel)	Curva de Atividade no Miocárdio (Bq/voxel)
1.417	301.730	40.958
1.583	659.641	105.810
2.083	1586.794	512.438
3.000	2229.139	1217.678
3.750	1470.229	1532.736
10.000	908.425	2294.914
28.000	559.477	2562.667
48.000	486.882	2615.198

Analisando os valores da Tabela 4, é possível notar que em alguns instantes a atividade no sangue e/ou no miocárdio são extremamente elevadas. Considerar esses valores, inviabilizaria a execução de simulações, devido ao tempo necessário para completá-las. Na tentativa de solucionar essa dificuldade, a atividade em cada instante foi dividida por um mesmo fator arbitrário de modo

que as atividades diminuíssem para valores possíveis de serem simulados em um tempo razoável, como está mostrado na Tabela 5.

**TABELA 5** – Atividades no sangue e no miocárdio, considerando o fator de redução (x8) para cada uma das 8 simulações.

Simulação	Sangue (Plasma)	Miocárdio	Fator
1	38	5	8
2	83	13	8
3	198	64	8
4	279	153	8
5	184	192	8
6	114	287	8
7	70	320	8
8	61	327	8

No processo de simulação, os valores de atividade são dados pelo usuário por meio do arquivo **activity\_range\_mash\_heart.dat**. Assim, para o caso do estudo relatado, temos um conjunto de 12 diferentes materiais, cada um deles com um único valor de intensidade (podendo ter uma faixa de intensidades com o qual utilizamos a primeira e segunda coluna), sendo que cada um dos materiais está associado a um valor de atividade colocado na terceira coluna da tabela, assim como é mostrado na Tabela 6 a seguir:

**TABELA 6** – Dados do arquivo **activity\_range\_mash\_heart.dat** da simulação No. 7.

```

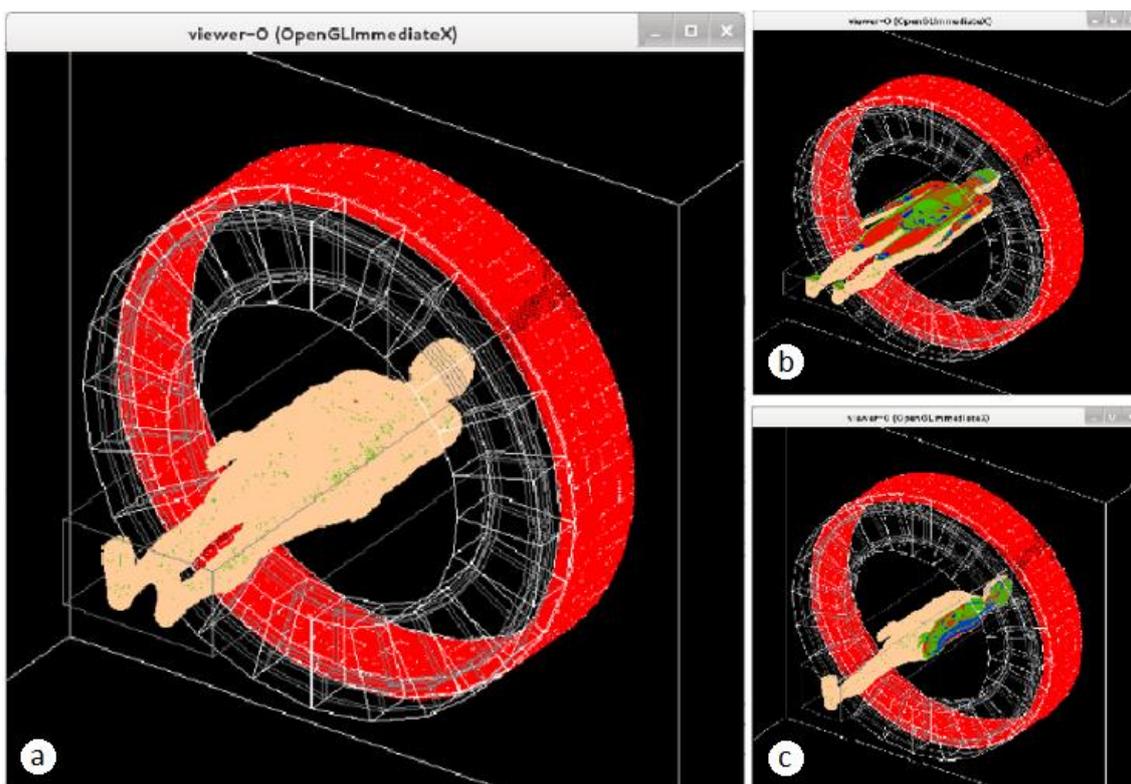
11
10 10 50
42 42 30
43 43 30
48 48 70
99 99 320
104 104 30
117 117 30
134 134 30
140 140 40
163 163 30
174 174 30

```

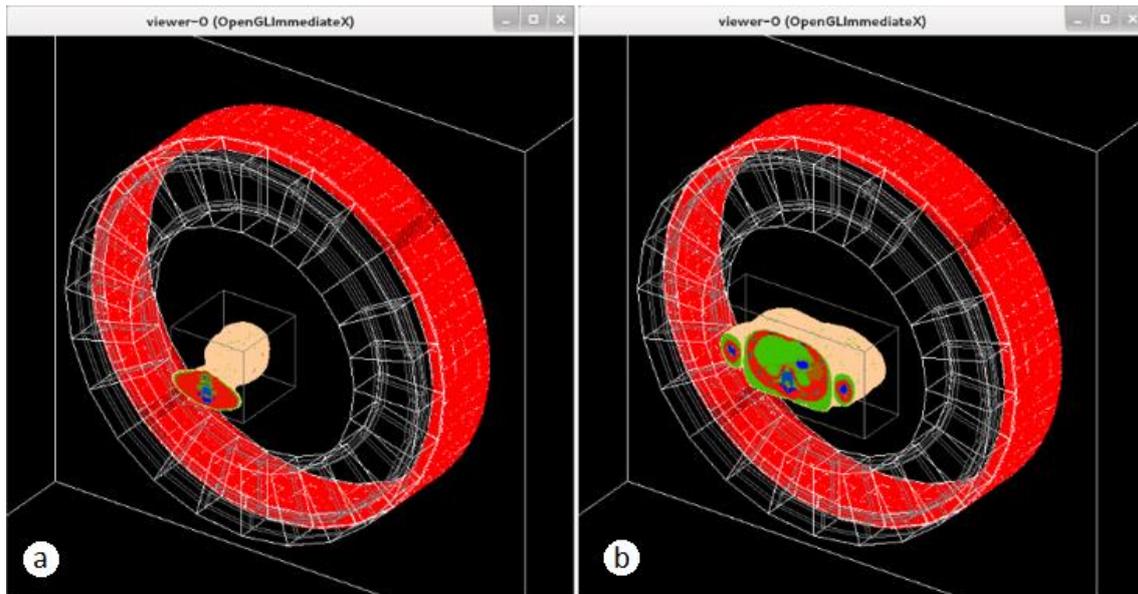
### 3.3.3 ACOPLAMENTO DOS PHANTOMS NO GATE

O *phantom* antropomórfico mostrado na secção 3.2.1, independente do formato utilizado para sua representação (Analyze, Interfile ou ASCII), foi acoplado no GATE e foi posicionado no centro do tomógrafo modelado com o intuito de obter as projeções como produto do exame PET simulado (processo de aquisição).

Então, foi possível fazer o acoplamento completo do *phantom* antropomórfico MASH dentro do tomógrafo simulado BiographTM como é visualizado na Figura 17. No entanto, conforme foi comentado anteriormente, é possível delimitar mais acuradamente a região de estudo para executar uma simulação mais apropriada e sem ter a necessidade de carregar o *phantom* completo no GATE. Por exemplo, sendo o nosso foco de estudo tanto o coração quanto o cérebro, pode-se delimitar as áreas de estudo a estas regiões como é mostrado nas Figuras 18 (a) e (b), respectivamente.

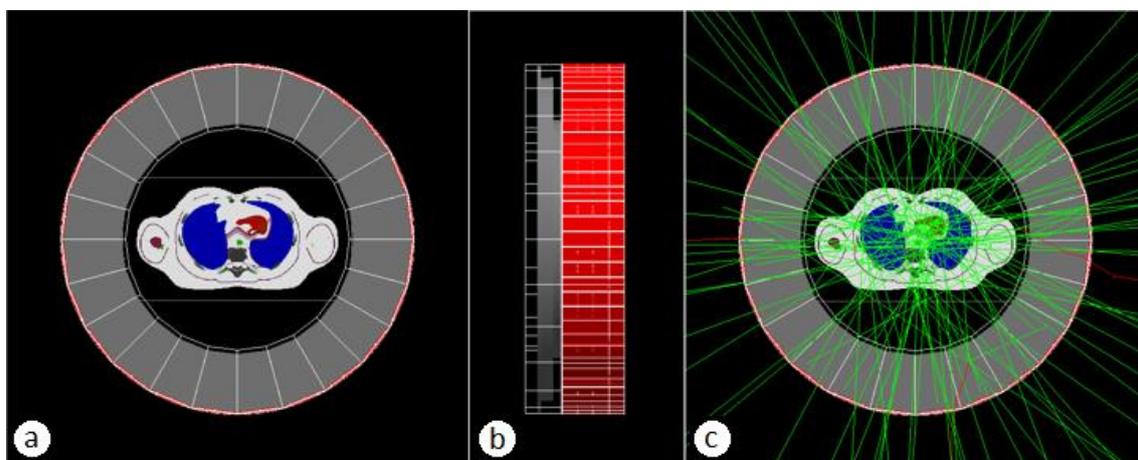


**FIGURA 17** – Acoplamento do MASH no GATE. (a) *Phantom* MASH completo em posição horizontal, (b) corte sagital, e (c) corte longitudinal do MASH com todas as estruturas internas (i.e. órgãos, músculos, esqueleto, etc.).



**FIGURA 18** – Acoplamento de regiões específicas do MASH dentro do tomógrafo BiographTM (a) região da cabeça para o estudo do cérebro, e (b) região do tronco para o estudo do coração.

Para o estudo do coração, o *phantom* antropomórfico correspondente à região torácica foi introduzido no meio do tomógrafo tanto na vista frontal quanto na vista lateral de tal forma que uma vez que as emissões dos fótons sejam geradas na simulação, o maior número deles possa ser detectado pelos cristais ao redor do objeto estudado. Nas Figuras 19(a) e 19(b), pode-se observar o posicionamento do *phantom* no interior do tomógrafo modelado. Já na Figura 19(c) se mostram os primeiros instantes de emissão dos fótons como parte do processo de aquisição.



**FIGURA 19** – *Phantom* antropomórfico MASH acoplado dentro do tomógrafo BiographTM: (a) vista frontal do posicionamento do *phantom*, (b) vista lateral do posicionamento do *phantom*, e (c) geração dos fótons no processo de simulação e aquisição através do GATE.

### 3.3.4 PROCESSO DE OBTENÇÃO DAS PROJEÇÕES

Por meio do GATE foram geradas as projeções realísticas PET com uma destacada proximidade com aquelas geradas com o próprio equipamento PET/CT, utilizando para esse fim a técnica de Monte Carlo essencial na exatidão da simulação [Ljungberg, 2013].

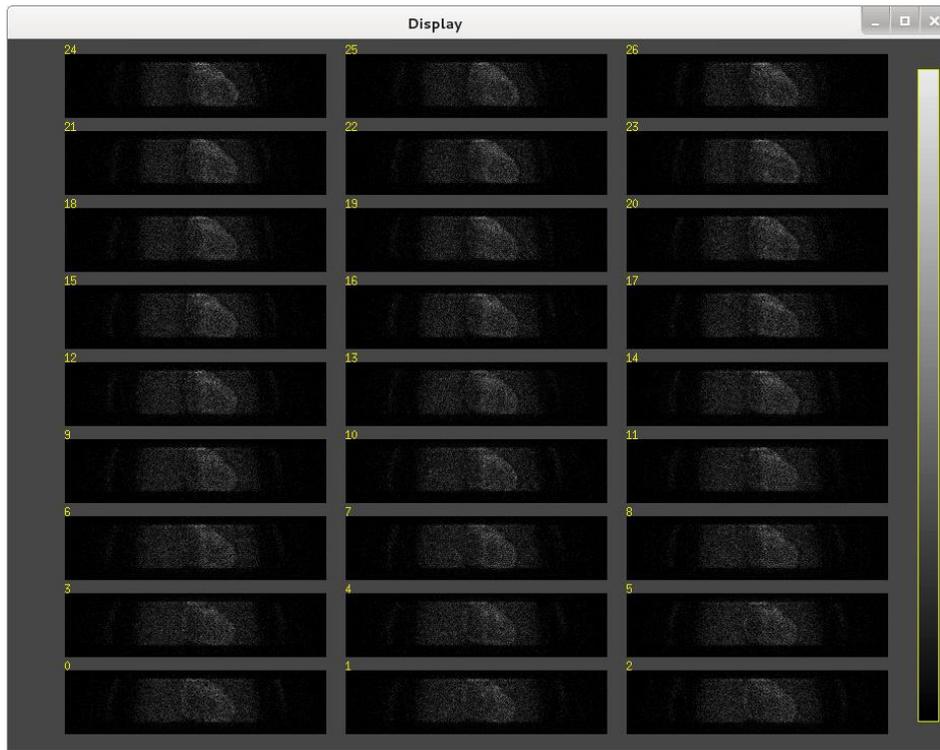
Neste ponto se contempla a execução completa da simulação na plataforma GATE (as linhas de comando estão contidas no Apêndice G), assim como a análise dos arquivos de saída contendo as informações dos sinogramas e/ou das projeções do objeto de interesse.

Como indicado na secção 3.3.2, foi simulado um processo de aquisição de um exame PET real de 48 minutos, o qual foi dividido em 8 *frames* considerando os instantes de tempo a seguir: 1.417, 1.583, 2.083, 3.0, 3.750, 10.0, 28.0 e 48.0 minutos. No entanto, o tempo máximo gasto no processo de simulação foi de 144 horas, 30 minutos e 41 segundos (~ 6 dias) em um computador de alto rendimento cujas características foram indicadas na secção 3.1.

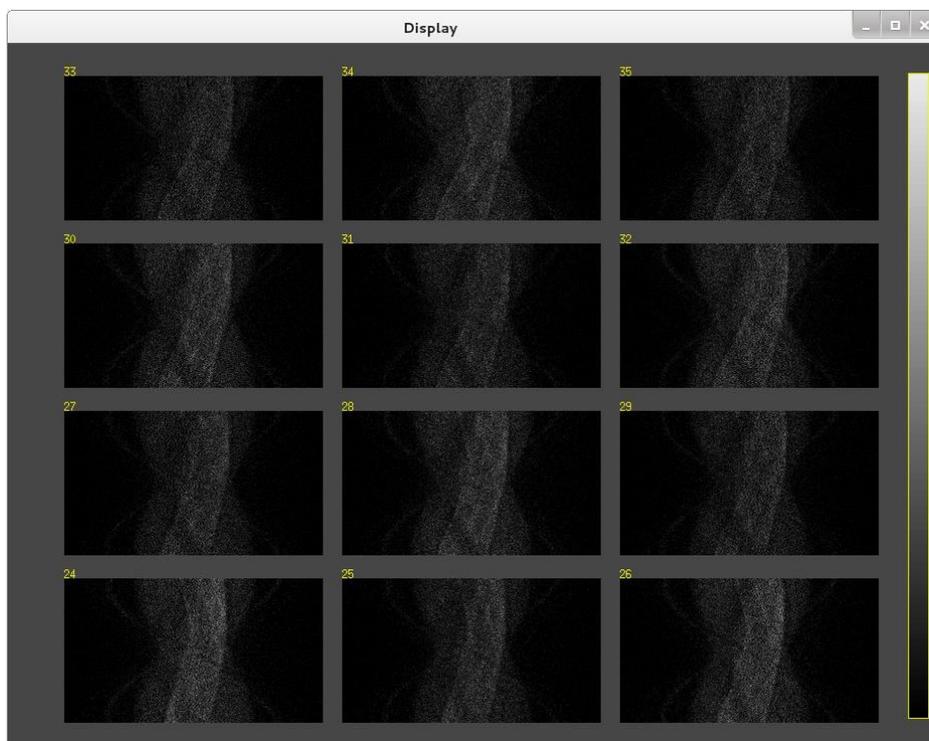
Como indicado também, o *phantom* tem uma resolução de  $478 \times 258$  com uma dimensão de  $1.2 \text{ mm}^3$  por voxel. Além disso, sabe-se que no GATE, aproximadamente 3 ou 4% dos fótons emitidos no processo de simulação são detectados pelo tomógrafo PET modelado durante o tempo de aquisição. E precisamente esses dados coletados foram parte dos arquivos (projeções e/ou sinogramas) que foram posteriormente utilizadas na fase de reconstrução.

É importante salientar que segundo especialistas aproximadamente 10% da quantidade de radiofármaco injetado no paciente atinge a região de interesse e, considerando que as atividades que estão sendo utilizadas nas simulações correspondem a esse 10%, a quantidade de radiofármaco que teria que ser injetada para atingir essa situação coincide com a faixa de substância marcadora permitida pela norma americana para exames PET reais (de 2mCi até 15mCi). Para termos uma ideia, uma dose aproximada de 10 mCi de FDG é injetada no paciente num procedimento clínico de rotina.

Como resultado do processo de simulação e aquisição através do GATE foram obtidas um total de 144 projeções (Figura 20) e 63 sinogramas (Figura 21) em cada uma das 8 simulações, utilizados posteriormente como dados de entrada na etapa de reconstrução dos volumes simulados.



**FIGURA 20** – Vista das projeções realísticas geradas no GATE como resultado do processo de aquisição simulado. De acordo com a arquitetura do tomógrafo PET modelado, foram obtidas um total de 144 projeções em cada uma das simulações realizadas.



**FIGURA 21** – Vista dos sinogramas gerados no GATE como resultado do processo de simulação. No total foram obtidas 63 sinogramas, cada uma delas gera uma fatia pertencente ao volume no processo de reconstrução 3D.

### **3.4 Reconstrução 3D das projeções adquiridas**

Etimologicamente, a palavra tomografia provem da palavra grega “tomos” que significa corte. O intuito do processo de reconstrução de uma imagem tomográfica é conseguir uma imagem seccional a partir da informação adquirida pelo aparelho utilizado, scanner ou tomógrafo [Borges, 2010].

O STIR (*Software for Tomographic Image Reconstruction*), software open-source, foi utilizado como ferramenta de reconstrução 3D no processo de visualização, conversão, manipulação e propriamente na reconstrução das imagens PET 3D. Esta ferramenta de reconstrução usa como elemento de entrada as projeções e/ou sinogramas, obtidas no processo de simulação através do GATE.

Complementarmente, uma Interface GATE/STIR pode ser utilizada para fazer a reconstrução das projeções do objeto 3D, através de arquivos do tipo \*.root. Os algoritmos utilizados nesta interface, assim como os detalhes e as linhas de comando utilizados no processo de reconstrução através do STIR podem ser consultados no Apêndice H e I, respectivamente.

Falando do processo de reconstrução propriamente dito, o STIR possui dois grupos de algoritmos de reconstrução: Analíticos (FBP2D e FBP3D) e Iterativos (OSMAPOLS).

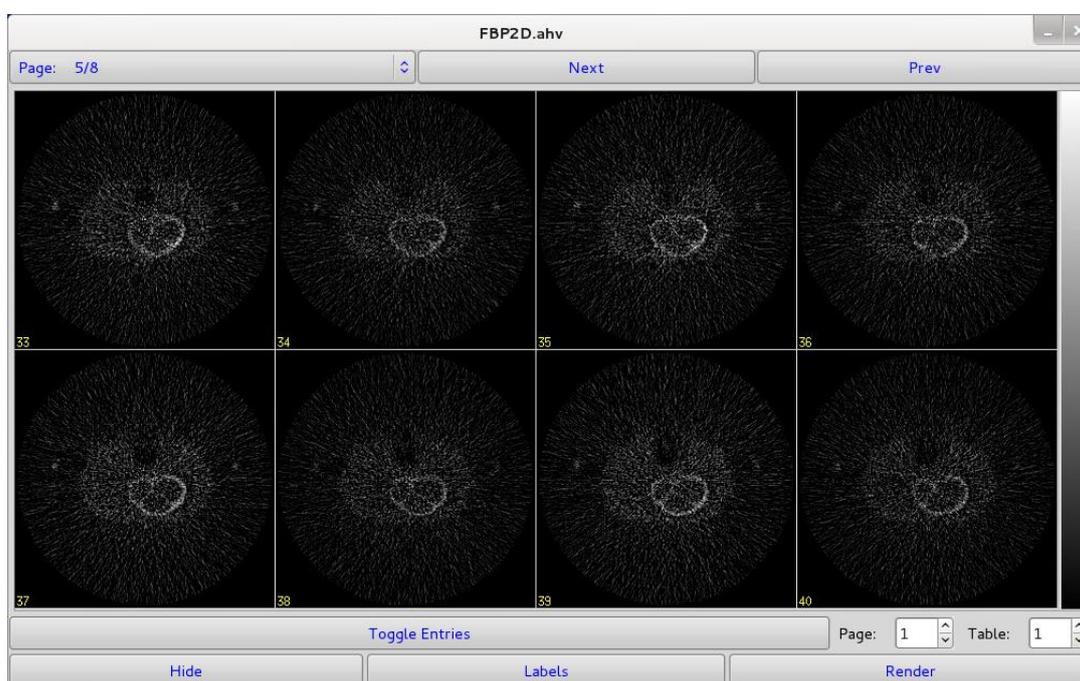
#### **3.4.1 Algoritmos ANALÍTICOS de Reconstrução 3D**

Neste grupo se encontram os algoritmos FBP (Filter Back Projection) 2D e 3D os quais se mostraram simples, eficientes e rápidos, mas apresentam algumas desvantagens para lidar com alguns fatores complicados como a dispersão e principalmente a atenuação. Este tipo de algoritmos de reconstrução são bastante empregados na maioria dos aparelhos médicos de Medicina Nuclear nas suas modalidades tomográficas PET e SPECT.

A seguir são mostrados alguns pormenores das reconstruções utilizando os algoritmos analíticos FBP2D e FBP3DRP. As linhas de comandos e/ou as opções de reconstrução utilizadas para cada uma destas abordagens estão listadas no Apêndice I.

## (a) FBP2D

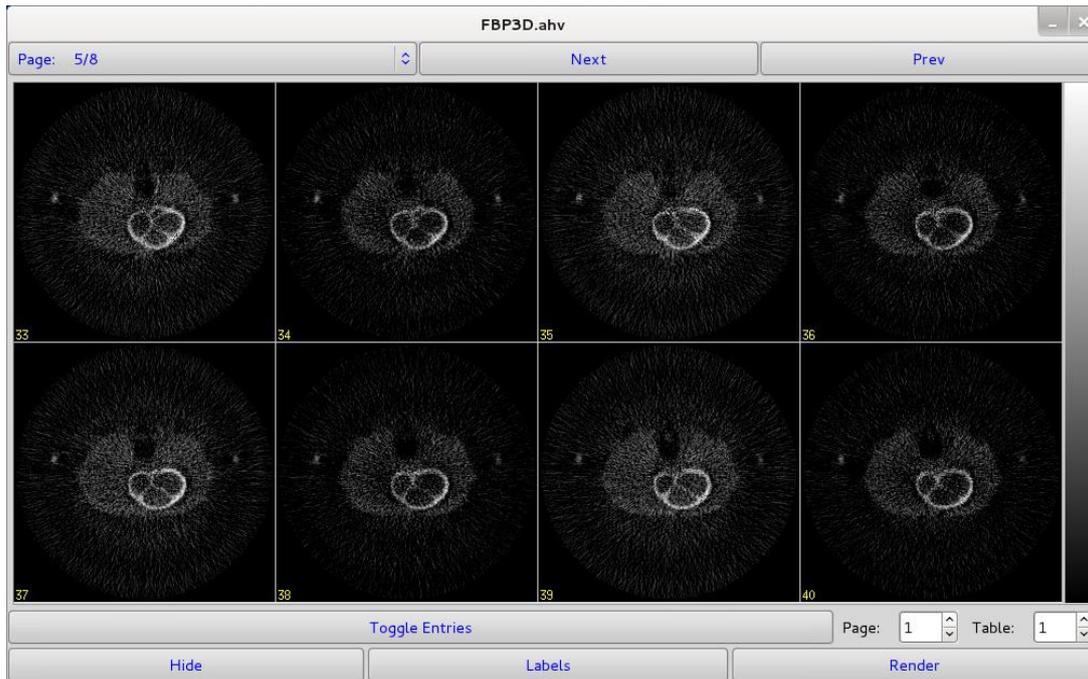
Após a execução do algoritmo de reconstrução FBP2D, com um tempo de processamento de aproximadamente 10 segundos, foram gerados os arquivos de saída no formato Interfile (formato escolhido): FBP2D.ahv, FBP2D.hv e FBP2D.v. Utilizando o software (X)MedCon foi possível visualizar a reconstrução do volume do coração o qual está composto por 63 fatias. Parte destas fatias são mostradas na Figura 22.



**FIGURA 22** – Reconstrução do volume simulado que contém o coração através do algoritmo analítico FBP2D. Esta abordagem se limita a usar unicamente as coincidências que foram detectadas pelos cristais que pertencem a um anel específico, caso contrário o evento é descartado do processo de reconstrução.

## (b) FBP3DRP

Após a execução do algoritmo de reconstrução FBP3DRP com um tempo de processamento aproximado de 58 segundos (~ 1 minuto) foram obtidos os arquivos de saída no formato Interfile (formato escolhido): FBP3D.ahv, FBP3D.hv e FBP3D.v. Através do software (X)MedCon foi possível visualizar a reconstrução do volume do coração o qual possui 63 fatias. Parte delas são mostradas na Figura 23.



**FIGURA 23** – Reconstrução do volume simulado que contém o coração através do algoritmo analítico FBP3DRP. Esta abordagem utiliza tanto as coincidências detectadas em cristais de um mesmo anel, quanto de anéis diferentes. Portanto, o resultado de reconstrução obtido por meio deste método se mostra superior ao processo utilizando o algoritmo FBP2D.

### 3.4.2 Algoritmos ITERATIVOS de Reconstrução 3D

Por outro lado, têm-se os algoritmos iterativos os quais são mais versáteis, no entanto são menos eficientes em termos de tempo de reconstrução (demorados em comparação com os Analíticos). Os algoritmos Iterativos estão atualmente em desenvolvimento tendo como objetivo ser cada vez mais utilizados nas aplicações tomográficas PET e SPECT, permitindo obter reconstruções mais quantitativas.

#### (a) OSMAPOS

Após a execução do algoritmo iterativo de reconstrução OSMAPOS, considerando 12 iterações e um tempo de processamento aproximado de 273 segundos (~ 4.5 minutos) foram obtidos os arquivos de saída no formato Interfile (formato escolhido): OSMAPOS.ahv, OSMAPOS.hv e OSMAPOS.v. Utilizando o software (X)MedCon foi possível visualizar a reconstrução do volume do coração o qual está formado por 63 fatias. Parte delas são mostradas na Figura 24.



**FIGURA 24** – Reconstrução do volume simulado que contém o coração através do algoritmo iterativo OSMAPOLS. Esta abordagem utiliza um número limitado de iterações que pode ser igual ao número total de projeções ( $N=144$ ), um número submúltiplo ou múltiplo dele.

No entanto, o processo de reconstrução 3D utilizando a abordagem iterativa possui ferramentas de correção de atenuação e normalização, as quais podem ser ativadas dentro do processo geral de reconstrução como se mostra nos Apêndices J e K, respectivamente. Por sua vez, os processos de correção de atenuação e normalização precisam de arquivos específicos para esse fim que serão detalhados a seguir:

- **Correção de Atenuação**

A correção de atenuação é um importante passo no processo de reconstrução PET devido a que pelo menos 60% dos fótons emitidos na etapa de aquisição interagem com os tecidos e órgãos internos do corpo do paciente [McQuaid, 2008]. Esse processo precisa de um mapa dos coeficientes de atenuação (mapa  $\mu$ ) a 511 keV, que pode ser calculado a partir do conhecimento dos materiais que integram o estudo assim como os limites de cada um deles. Neste projeto foi utilizado um algoritmo chamado PETac (PET attenuation correction). PETac foi escrito pelo Robbie Barnett, da Universidade de Sydney, Austrália, em 2012. No entanto, foi necessário dedicar muito trabalho na modificação e atualização do

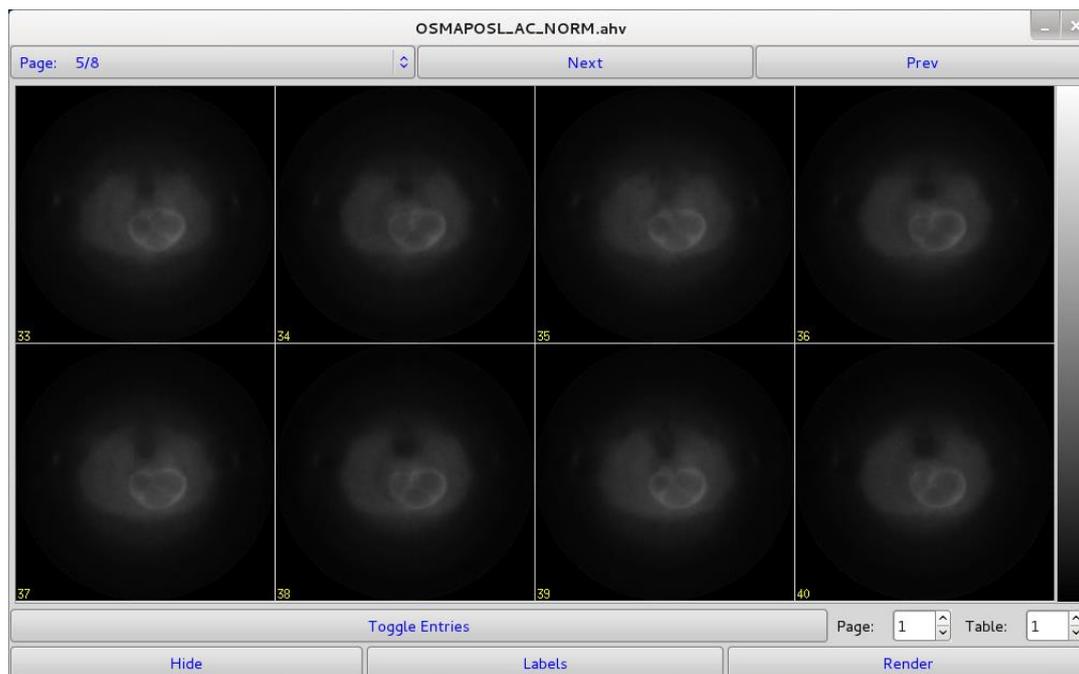
código original para concluir o processo de instalação e posteriormente obter o arquivo com o mapa de atenuação ( $\mu$ ). Todas as alterações do código original e os requisitos de instalação estão detalhados no Apêndice J. PETac escala uma imagem CT (ou conjunto de imagens tomográficas) e gera um mapa de atenuação ( $\mu$ -map) referenciado no coeficiente de atenuação da água que é igual a  $0.097 \text{ cm}^{-1}$ . Este algoritmo necessita de alguns parâmetros específicos do aparelho de PET que foram utilizados no processo de aquisição, a fim de completar a geração de mapa  $\mu$ . Finalmente, o arquivo com o mapa de atenuação gerado foi utilizado no processo de reconstrução 3D por meio do algoritmo OSMAPOSL.

#### ▪ Normalização

A normalização é um ingrediente indispensável no processo de reconstrução junto com a correção de atenuação. Assim, foi dedicado um grande esforço na elaboração e implementação de um procedimento adequado para obter a normalização 3D. Para este efeito, torna-se indispensável avaliar os fatores associados com os efeitos da geometria do scanner e dos anéis, a variação da sensibilidade nos diferentes cristais, e os efeitos que dependem da posição dos cristais dentro de um bloco, entre outros [Bai et al., 2002]. Na literatura, existem várias propostas para medir e corrigir esse efeito, como aquele que considera utilizar um cilindro uniforme.

Neste estudo, a normalização 3D que temos executado foi produto de uma longa simulação no GATE com uma fonte cilíndrica (sem material algum e conseqüentemente sem fator de atenuação, apenas vazio) cobrindo todo o diâmetro e a largura da câmara PET, usando a detecção coincidente de fótons. As linhas de código GATE utilizadas neste processo e a figura do cilindro no scanner PET estão no Apêndice K. Os dados brutos, geradas através do GATE, foram binarizados (usando o código do C-Ross) em um sinograma 3D (norma.hs). Em seguida, este sinograma foi usado no processo de reconstrução por meio do STIR. Finalmente, esse arquivo foi utilizado no processo de reconstrução 3D por meio do algoritmo OSMAPOSL.

A Figura 25 mostra o processo de reconstrução OSMAPOSL que inclui os processos de correção de atenuação e normalização. As linhas de comando da reconstrução iterativa junto com os processos adicionais de correção de atenuação e normalização, podem ser visualizadas no Apêndice I.



**FIGURA 25** – Reconstrução do volume simulado que contém o coração através do algoritmo iterativo OSMAPOLS. Esta abordagem incorporou os processos de correção de atenuação e normalização dentro da tarefa de reconstrução 3D.

### 3.5 Filtragem de imagens PET 3D

Define-se o ruído em uma imagem digitalizada, como toda mudança indesejada das intensidades dos pixels que pode comprometer a sensibilidade e principalmente a qualidade da imagem [Sarmiento; Carvalho; Luz, 2004].

De modo geral, os filtros são aplicados como recursos ou ferramentas que permitem aumentar o brilho e o contraste, bem como para adicionar uma grande variedade de texturas, tons e efeitos especiais em uma imagem. Este processo repercute no incremento da qualidade geral da imagem, o que pode ser interessante para fins de visualização, e ainda para ressaltar determinados objetos ou características da imagem tratada [Gonzalez; Woods, 2008].

O processo de filtragem em uma imagem fomenta a redução do ruído contido nela o qual é essencial em aplicações com imagens médicas, e consiste principalmente em utilizar informações sobre as degradações que afetam a imagem, a fim de aplicar um processo adequado para tirar essas distorções com o propósito de recuperar os detalhes anatômicos que podem estar escondidos nos dados [Rodrigues et al., 2008].

De acordo com a literatura, sabe-se que as imagens PET são corrompidas pelo ruído quântico que é dependente do sinal e pode ser modelado por uma distribuição estatística de Poisson [Bethea; Duran, 1995; Romualdo, 2009]. Além disso, por se tratar de um sinal aleatório, ferramentas estatísticas podem ser utilizadas na modelagem do ruído quântico [Wolbarst, 1993].

O processo de filtragem 3D foi aplicado sobre os volumes reconstruídos, para o qual foi assumindo que o ruído exibido nas imagens segue uma distribuição de probabilidade Poisson. Para esse propósito, o método de filtragem abordado, analisado, implementado e discutido neste projeto foi o Filtro de Anscombe / Wiener. Este método proposto efetua uma restauração nas imagens utilizando, em uma primeira etapa, a transformada de Anscombe (AT) junto com o filtro de Wiener pontual para redução do ruído quântico tipo Poisson. Posteriormente, foi utilizada a Transformada Inversa de Anscombe (IAT) para o realce das estruturas de interesse da imagem filtrada.

▪ **Transformada de Anscombe (AT)** – Uma transformação que permite estabilizar a variância (Variance Stabilizing Transformation) é uma transformação de dados que tem por objetivo simplificar considerações na análise de dados quando a hipótese de um modelo linear não é satisfeita [Allen, 2013]. A transformada de Anscombe é uma transformação não linear que permite transformar o ruído quântico de uma imagem digital, que é dependente do sinal, em um ruído aproximadamente independente do sinal, aditivo, gaussiano, com média zero e variância unitária [Inouye, 1971; Romualdo, 2009; Vieira et al., 2013]. Existem alternativas similares como as transformadas de Bartlett e de Freeman&Tukey que tem sido implementadas e testadas.

Assim, dada a variável aleatória  $\tilde{U}_i$ , com distribuição estatística de Poisson, podemos expressar a AT desta variável aleatória de acordo com [Anscombe, 1948], utilizando a equação (3).

$$\tilde{Z}_i = 2 \cdot \sqrt{\tilde{U}_i + \frac{3}{8}} \quad (3)$$

De igual forma, usando as transformadas de Bartlett [Bartlett, 1936] e de Freeman&Tukey [Freeman; Tukey, 1950], a transformada da variável  $\tilde{U}_i$  são expressas pelas equações (4) e (5), respectivamente.

$$\tilde{Z}_i = 2 \cdot \sqrt{\tilde{U}_i + \frac{1}{2}} \quad (4)$$

$$\tilde{Z}_i = \sqrt{\tilde{U}_i} + \sqrt{\tilde{U}_i + 1} \quad (5)$$

Esta nova variável  $\tilde{Z}_i$  da equação (3), poder ser representada por meio de um modelo aditivo [Inouye, 1971], conforme mostra a equação (6):

$$\tilde{Z}_i = 2 \cdot \sqrt{\tilde{U}_i + \frac{1}{8}} + \tilde{N}_i = \tilde{S}_i + \tilde{N}_i \quad (6)$$

sendo  $\tilde{N}_i$  um ruído aproximadamente independente do sinal  $\tilde{S}_i$ , descrito por uma distribuição Gaussiana com média zero e variância unitária.

Em outras modalidades de imagens médicas degradadas pelo ruído Poisson, a transformada de Anscombe tem proporcionado bons resultados para filtragem do ruído antes do processo de restauração e reconstrução de imagens [Mascarenhas; Santos; Cruvinel, 1999; Homen et al., 2002; Homen et al., 2004; Romualdo, 2009]. Após esta transformação, podem-se utilizar técnicas bem conhecidas aplicadas à nova variável, para a redução do ruído, que agora é aditivo e independente do sinal [Mascarenhas; Santos; Cruvinel, 1999].

▪ **Filtro de Wiener pontual (Lee)** - Sob o critério do mínimo erro quadrático médio, o Filtro de Wiener pontual [Lee, 1980] é um filtro linear ótimo e pode ser utilizado para a filtragem do ruído aditivo gaussiano. Assim, dado o vetor de observações  $u$ , para cada  $u_i$ , a Transformada de Anscombe é agora descrita pela equação (7)

$$z_i = 2 \cdot \sqrt{u_i + \frac{3}{8}} \quad (7)$$

Obtém-se uma estimativa sem ruído  $\hat{s}_i$  para  $z_i$  utilizando o Filtro de Wiener pontual dado pela equação (8)

$$\hat{s}_i = E[\tilde{S}_i] + \frac{\sigma_{\tilde{S}_i}^2}{\sigma_{\tilde{S}_i}^2 + 1} (z_i - E[\tilde{S}_i]) \quad (8)$$

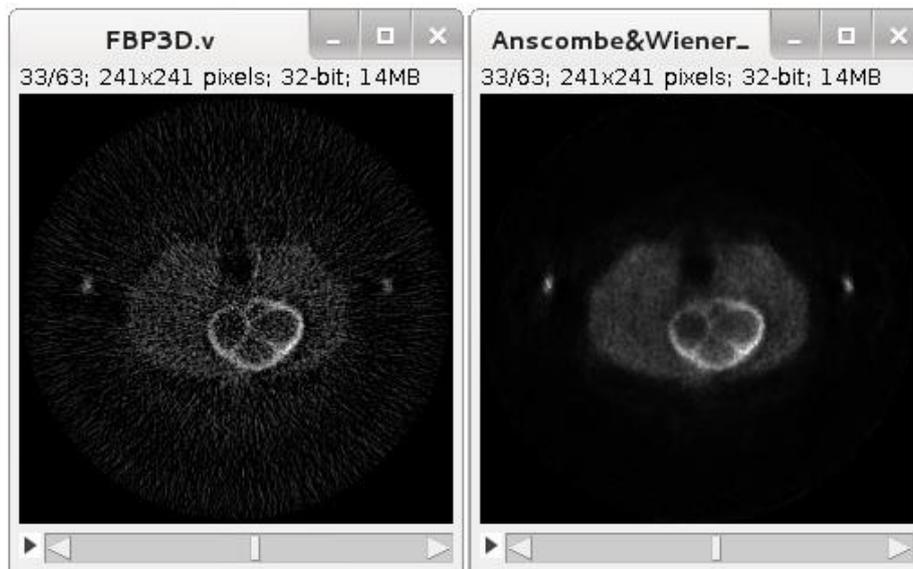
sendo que a média e a variância de  $\hat{s}_i$ , expressas respectivamente por  $E[\tilde{S}_i]$  e  $\sigma_{\tilde{S}_i}^2$ , são medidas locais e podem ser estimadas na prática a partir da imagem a ser filtrada.

▪ **Transformada Inversa de Anscombe (IAT)** - Após a filtragem do ruído, aplicamos a transformação inversa de Anscombe para termos uma estimativa da

imagem degradada sem o ruído quântico [Romualdo, 2009]. Podemos expressar a IAT sobre  $\hat{s}_i$  pela equação (9)

$$\hat{b}_i = \frac{1}{4} \cdot \hat{s}^2 - \frac{1}{8} \quad (9)$$

Na Figura 26, se mostra o volume reconstruído através do algoritmo FBP3D e, o resultado do processo de filtragem tridimensional do ruído quântico tipo Poisson utilizando o filtro Anscombe/Wiener. O tempo de processamento gasto no processo de filtragem foi de 27.6 segundos (0.46 minutos). Deve-se ressaltar que o processo de filtragem foi aplicado sobre todos os volumes reconstruídos por métodos analíticos e iterativos.



**FIGURA 26** – Resultado do processo de filtragem 3D através da abordagem Anscombe/Wiener: na esquerda se mostra o volume reconstruído (FBP3D) e na direita o volume finalmente filtrado.

### 3.6 Segmentação 3D das regiões de interesse

A segmentação é um processo que consiste em decompor uma imagem em regiões ou objetos que a compõem [Gonzales; Wood, 2008], efetuando uma identificação dos spels (pixels ou voxels) que fazem parte do objeto de interesse selecionado pelo usuário.

Como indicado, o processo de segmentação aplicado neste trabalho baseia-se no conceito de Conectividade Fuzzy (do inglês Fuzzy Connectedness - FC). O

processo de segmentação utilizando o método FC, tem-se mostrado eficiente quando aplicado em imagens médicas, mesmo comparado com outros métodos [Xu, 1998]. Dependendo da aplicação, ele é mais interessante, pois é um método robusto, principalmente quando se trata de imagens com intensidades não uniformes [Pednekar; Kakadiaris, 2006].

Fundamentado nos aspectos teóricos apresentados na seção 2.5 no capítulo anterior, a seguir serão detalhadas algumas definições a fim de estabelecer os critérios que foram utilizados na implementação e execução do processo de segmentação neste projeto.

É importante apontar que um dos conceitos fundamentais da teoria de Conetividade Fuzzy e do processo de segmentação é o fator de Afinidade, pelo fato de que por seu intermédio ficam estabelecidos os parâmetros com os quais a segmentação será aplicada. Três propriedades fundamentais integram o elemento Afinidade: adjacência ( $\mu_\alpha$ ), homogeneidade ( $\mu_\psi$ ), e intensidade ( $\mu_\phi$ ) [Gonzalez; Woods, 2008]. A afinidade, portanto, é uma função das propriedades anteriormente mencionadas:

$$\mu_k = \mu_\alpha(c, d) \cdot g(\mu_\psi(c, d), \mu_\phi(c, d)) \quad (10)$$

Posto que a relação de adjacência  $\mu_\alpha(c, d)$ , indica a proximidade espacial entre os elementos avaliados, se  $c$  e  $d$  são vizinhos a adjacência Fuzzy é unitária. Com isso a afinidade é descrita como:

$$\mu_k = g(\mu_\psi(c, d), \mu_\phi(c, d)) \quad (11)$$

A definição do componente de homogeneidade  $\mu_\psi(c, d)$  é dada pela seguinte expressão:

$$\mu_\psi(c, d) = \exp \left[ -\frac{1}{2} \left( \frac{|f(c) - f(d)| - m_1}{s_1} \right)^2 \right] \quad (12)$$

na qual  $m_1$  e  $s_1$  são a média e o desvio padrão das homogeneidades locais do objeto, respectivamente. Assim mesmo, a definição do componente de intensidade  $\mu_\phi(c, d)$  é dada por:

$$\mu_{\phi}(c, d) = \exp \left[ -\frac{1}{2} \cdot \left( \frac{\left( \frac{f(c) + f(d)}{2} \right) - m_2}{s_2} \right)^2 \right] \quad (13)$$

em que  $m_2$  e  $s_2$  são a média e o desvio padrão das intensidades do objeto, respectivamente. Tanto na equação de homogeneidade quanto na equação de intensidade,  $f(x)$  representa a intensidade do spel  $x$ .

Sendo conhecidas as expressões tanto do componente de homogeneidade quanto de intensidade que integram a equação 11, pode-se definir finalmente o termo de afinidade da seguinte maneira:

$$\mu_k = w_1 \cdot \mu_{\psi}(c, d) + w_2 \cdot \mu_{\phi}(c, d) \quad (14)$$

onde  $w_1$  e  $w_2$  são valores numéricos não negativos que representam os pesos das componentes que conformam a expressão de afinidade, dado que  $w_1 + w_2 = 1$  [Nyúl et al., 2002].

Baseados nessas definições teóricas e complementadas com as relações globais de Conectividade Fuzzy entre os spels de uma imagem vistas na secção 2.5, foram desenvolvidos cinco diferentes algoritmos de segmentação: kFOEMS (abordagem genérica), Dial\_LIFO, Dial\_FIFO, kTetaFOEMS e kFOEMS\_DyW (pesos dinâmicos) [Flórez Pacheco, 2012].

No entanto, a seguir será abordado o algoritmo de segmentação que utiliza pesos dinâmicos o qual foi aplicado no projeto e através do qual foram feitos os processos de segmentação das imagens PET 3D. O código do algoritmo de segmentação original com pesos dinâmicos foi modificado de modo que possa aceitar múltiplas sementes no processo de segmentação 3D.

Deve-se salientar que o processo de segmentação desempenhou um papel proeminente no processo de avaliação funcional da etapa de filtragem 3D.

### 3.6.1 Algoritmo de Conectividade Fuzzy com Pesos Dinâmicos

Este algoritmo apresenta uma particularidade interessante em relação aos algoritmos de segmentação FC clássicos exibidos em [Flórez Pacheco, 2012]. A particularidade desta abordagem consiste em realizar ajustes dinâmicos dos

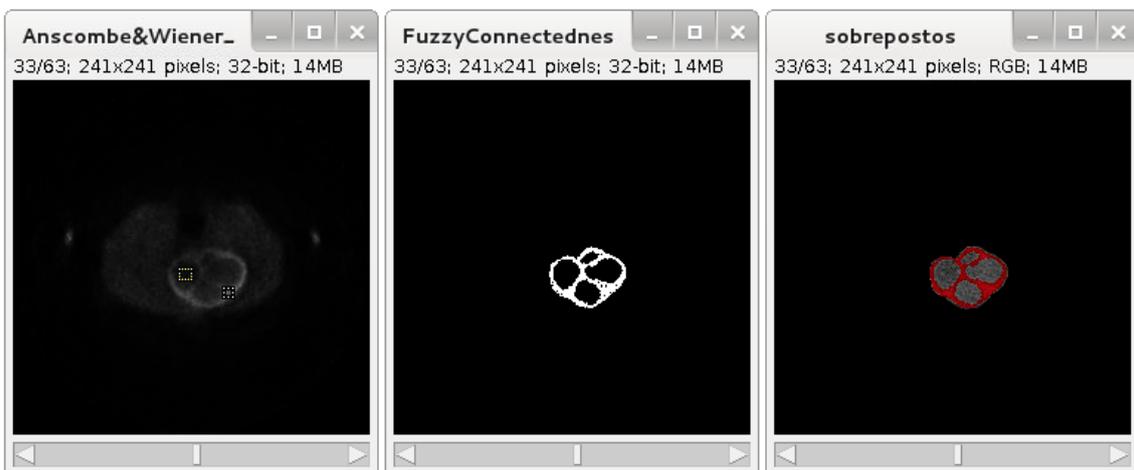
pesos  $w_1$  e  $w_2$  da equação 15, que relacionam os componentes de homogeneidade e intensidade, para cada par de spels  $(c,d)$  um caminho  $p_{c,d}$  qualquer [Nyúl et al., 2002; Udupa; Saha, 2003].

Assim, os pesos adotados por  $w_1$  e  $w_2$  para cada um dos pares de elementos,  $c$  e  $d$ , que pertencem ao mesmo caminho é expresso pelas seguintes equações [Lage, 2010]:

$$w_2 = \frac{\mu_\phi(c,d)}{\mu_\psi(c,d) + \mu_\phi(c,d)} ; \quad w_1 = 1 - w_2 \quad (15)$$

Na Figura 27, apresenta-se na esquerda o volume do coração junto com os outros órgãos da região torácica que foram reconstruídos com o FBP3D e filtrados utilizando o filtro Anscombe / Wiener 3D, na direita aparece o volume segmentado através do algoritmo de Conectividade Fuzzy que usa pesos dinâmicos e que permite utilizar múltiplas sementes. Para o processo de segmentação, usou-se sementes localizadas no miocárdio e em cada uma das cavidades sanguíneas. O tempo total de processamento gasto na segmentação foi de 67.2 segundos (1.12 minutos).

Deve-se ressaltar que o processo de segmentação foi aplicado sobre todos os volumes reconstruídos e filtrados.



**FIGURA 27** – Resultado do processo de segmentação 3D utilizando o algoritmo de Conectividade Fuzzy com Pesos Dinâmicos aplicado sobre o volume que hospeda o coração, o qual foi previamente reconstruído com o algoritmo FBP3D e filtrado com o Anscombe/Wiener 3D. Foram segmentadas as regiões do músculo cardíaco (miocárdio) e as quatro cavidades do coração com sangue.

### 3.7 Estimação dos parâmetros dos Modelos Compartimentais

Assume-se que a quantidade do radiofármaco introduzido é suficientemente pequeno de tal forma que o processo a ser estudado não fique perturbado pelo traçador introduzido (i.e., o traçador não exerce nenhum tipo de efeito sobre a massa) [Reivich, 1985]. Caso contrário, os resultados obtidos poderiam reflectir nos efeitos do traçador e não do processo original que se deseja avaliar.

Com base no acima exposto, a análise da concentração de radiofármacos nas imagens PET, produto da emissão de pósitrons, está afiliado ao nível metabólico da região avaliada e pode ser feita desde dois enfoques: qualitativa e/ou quantitativamente [Carson; Cobelli, 2001].

Para alguns estudos clínicos PET, a análise qualitativa é apropriada para dar resposta a questões biológicas, entre elas uma proposta interessante de estudo seria a localização de defeitos metabólicos. No entanto, a informação fornecida pela análise quantitativa é essencial como um critério de interpretação através de um modelo matemático. Aplicações como a medição do traçador PET nos processos fisiológicos ou a estimação da taxa metabólica da glicose, podem ser baseadas nesta abordagem quantitativa [Carson; Cobelli, 2001].

Na literatura existem vários modelos e/ou métodos para análise da concentração do traçador em imagens tomográficas PET em um órgão ou região de interesse. Podem-se ressaltar os seguintes: Modelos Compartimentais, modelos I/O e métodos gráficos. Como indicado na secção 2.6, neste projeto foi aplicada a análise baseada nos Modelos Compartimentais já que é uma técnica que permite desenvolver a modelagem de sistemas lineares e não lineares, assim como descrever o seu comportamento não estacionário. No caso dos outros dois métodos apontados anteriormente, eles são utilizados essencialmente na modelagem de sistemas lineares em estado estacionário [Carson; Cobelli, 2001].

A finalidade do uso do Modelo Compartimental é estimar os parâmetros referentes às constantes de fluxo e, conseqüentemente, a taxa metabólica da glicose. Portanto, devem-se estimar os valores das constantes de fluxo  $K_i$  (ver secção 2.6), que por determinarem os fluxos de troca de material entre os compartimentos, governam a dinâmica do sistema a ser estudado, por exemplo o miocárdio e/ou o cérebro.

Para a estimação dos parâmetros do modelo compartimentais com três compartimentos que foi utilizado neste estudo (constituído por quatro constantes

características:  $K_1$ ,  $k_2$ ,  $k_3$  e  $k_4$ ), foi indispensável iniciar o processo de estimação com alguns dados de entrada. Por um lado, a curva de função de entrada ou Curva de Atividade de Entrada (no plasma sanguíneo) foi um requisito fundamental para o processo de quantificação metabólica. No entanto, essa curva representa a quantidade de marcador no plasma e deve ser conhecido de forma tão precisa quanto possível, como uma função do tempo. Adicionalmente, precisa-se da curva de função de resposta ou Curva de Atividade Resultante, que representa a resposta do tecido a um radiofármaco entregue.

### **3.7.1 Curva de Atividade de Entrada (PTAC)**

A Curva de Atividade de Entrada é obtida como resultado da análise da concentração do radiofármaco ou traçador no sangue. No entanto, é necessário conhecer a concentração do marcador no plasma (i.e. o plasma representa aproximadamente o 55% do volume total do sangue) que será usado como função de excitação do modelo matemático para o cálculo dos parâmetros metabólicos da região de interesse.

A aquisição dos dados, que representam a Curva de Atividade de Entrada, podem ser obtidos através de análise experimental de amostragem sanguínea, por meio de imagens reconstruídas ou a partir de um banco de dados da população [Arno, 2001; Zaidi, 2006].

Para o caso da amostragem sanguínea, é utilizado um cateter através do qual é possível retirar pequenas quantidades do sangue da artéria ou veia do paciente. Essas amostras são colocadas em um contador de radiatividade para medir a concentração temporal de radiofármaco no tecido alvo. Vários estudos têm obtido a concentração plasmática a través deste procedimento [Sokoloff, 1976; Reivich, 1977; Sokoloff, 1977; Phelps, 1979; Huang, 1980; Hackett, 2013]. No entanto, a amostragem sanguínea arterial provoca desconforto para os pacientes e pode expor ao pessoal médico a radiação adicional [Kamasak et al., 2005], e, além disso poderia apresentar um alto risco associado a este procedimento invasivo como a trombose arterial, esclerose arterial, e isquemia tecidual irreversível [Mettler et al., 2008]. Adicionalmente e para termos uma ideia mais clara dessa técnica, em um estudo cardíaco seria bastante complexo, arriscado e invasivo o posicionamento de um cateter nas cercanias do miocárdio.

Portanto, a aplicação deste procedimento envolve basicamente dois tipos de erros de estimação: o primeiro devido à localização, pelo fato de que o tecido

alvo não se encontra próximo ao ponto de amostragem; e segundo devido ao tempo da amostragem, já que o contador obterá valores em instantes que não serão os mesmos instantes do exame.

Com base nestas desvantagens, um procedimento alternativo para estimar o nível de radioatividade no plasma foi considerado no nosso projeto, utilizando diretamente a informação contida nas imagens PET. Considera-se que este processo seja muito mais propício, não invasivo e confiável, uma vez que a informação será obtida diretamente dos valores dos pixels da série temporal das imagens.

No projeto, considerando tanto ao coração quanto ao cérebro como órgãos específicos de estudo, temos dois cenários para a obtenção da Curva de Atividade de Entrada:

(a) O primeiro, quando o estudo se baseia no coração (miocárdio). Nesta região, pode-se utilizar a informação do sangue dentro do coração (por exemplo, sangue do ventrículo esquerdo) ou o sangue da artéria aorta.

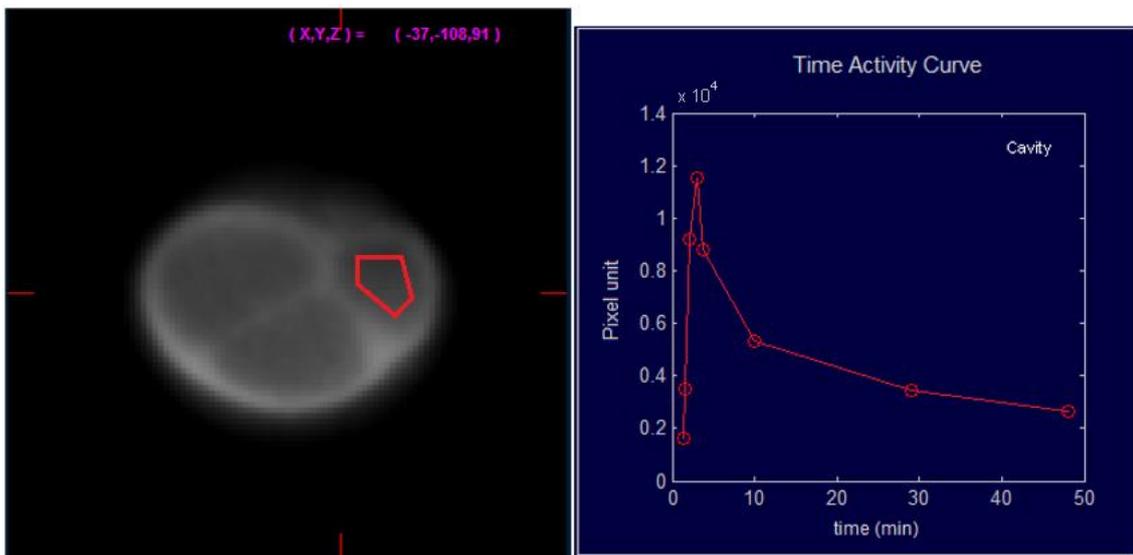
(b) O segundo, quando o estudo se baseia no cérebro. Neste caso, pode-se obter a informação diretamente da artéria carótida, mas existe um problema nesta segunda análise porque, por um lado, o diâmetro da artéria carótida é de ~ 5 mm e, por outro lado, sabe-se que as imagens de PET têm uma resolução moderada (~ 6 mm) e um elevado ruído associado a eles. Para resolver isto, foi utilizado um código em MatLab (Apêndice L), através do qual podemos construir a Curva de Atividade de Entrada utilizando alguns parâmetros específicos como por exemplo: o tipo do radiofármaco (tempo de meia-vida), a quantidade injetada do radiofármaco e o tempo de duração do exame PET.

Em geral, se o estudo tiver uma região contendo sangue (isto é, ter acesso a alguma artéria, veia ou no caso do coração a alguma câmara com sangue), podemos obter a Curva de Atividade de Entrada diretamente dessa região por meio de técnicas de processamento de imagem. O único requisito que o usuário precisa realizar é selecionar uma porção do objeto de interesse de qualquer fatia do volume. Como o estudo foi realizado sobre um conjunto de volumes ao longo do tempo (estudo dinâmico), o software localiza as mesmas coordenadas nos frames subsequentes para coletar a intensidade média dos pixels associada com cada um dos tempos de aquisição. Esses valores foram plotados de modo a se obter a Curva de Atividade de Entrada no sangue.

Tomando em consideração o nosso processo de simulação com todas as fases anteriores explicadas (geração das projeções, reconstrução tomográfica utilizando o algoritmo iterativo OSMAPOSL com correção de atenuação e normalização, filtragem e, segmentação do órgão de interesse) usando a região torácica do *phantom* antropomórfico MASH, foram gerados 8 volumes ou frames em um tempo total de simulação de 48 min. Cada instante simulado gerou um volume contendo o órgão estudado (coração), no qual foram definidas duas regiões, uma contendo sangue e outra o músculo cardíaco (miocárdio).

A Figura 28 mostra a Curva de Atividade de Entrada (PTAC) gerada pela concentração do radiofármaco no plasma. Para o processo simulado, os valores de atividade de entrada considerados inicialmente (Tabelas 4 e 5) representavam já o valor de atividade no plasma pelo qual não é necessário de se graduar os valores de atividade com base na correspondência que o plasma tem com respeito ao volume de sangue (~ 55%). Entretanto, para a análise das imagens PET reais, este ponto será considerado como parte do procedimento.

Um procedimento similar será executado para a obtenção da Curva de Atividade resultante no Tecido.

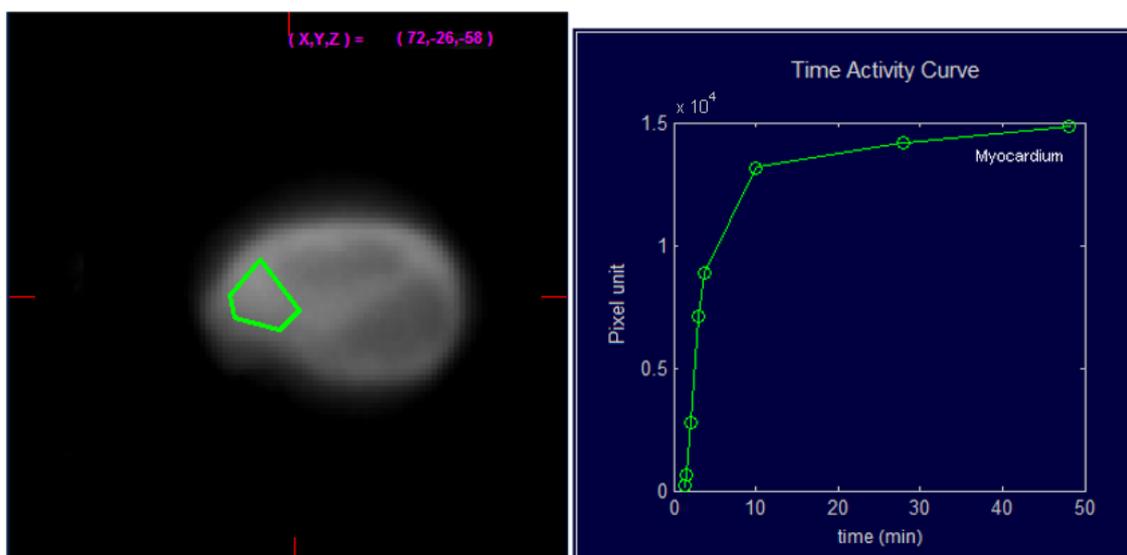


**FIGURA 28** – Curva de Atividade de Entrada contendo a informação da concentração do radiofármaco no plasma sanguíneo em cada um dos frames ao longo do tempo da simulação realística PET. O usuário deve escolher uma região contendo sangue em qualquer um dos volumes.

### 3.7.2 Curva de Atividade Resultante (TTAC)

Existem muitas considerações no processo de obtenção da Curva de Atividade Resultante ou Curva de Atividade no Tecido (TTAC), uma vez que é bastante complexo gerar a curva de resposta que possa representar todos os tecidos que conformam a região de interesse [Anderson, 1983].

A Curva de Atividade Resultante está realmente composta de várias curvas, já que a maioria dos tecidos contêm sangue e espaços intracelulares e extravasculares. Portanto, pode-se indicar que a Curva de Atividade Resultante (TTAC) contém uma parte da Curva de Atividade do Plasma (PTAC). Esta fracção da PTAC é, muitas vezes referida como transbordamento ou *spillover* [Phelps, 2004]. Na prática, um compartimento vascular pode ser incluído no modelo cinético para explicar o relacionamento da PTAC com o tecido vascular [Carson; Cobelli, 2001]. No entanto, uma das regras fundamentais da modelagem matemática indica que aumentar a complexidade do modelo (incremento de compartimentos) propiciaria que a estimativa dos parâmetros se torne muito imprecisa e, por outro lado, se o modelo for muito simples, pode-se perder alguns detalhes importantes da dinâmica estudada [Carson, 2005]. A Figura 29 mostra a curva de Atividade Resultante gerada pelas informações da concentração do radiofármaco no miocárdio (coração).



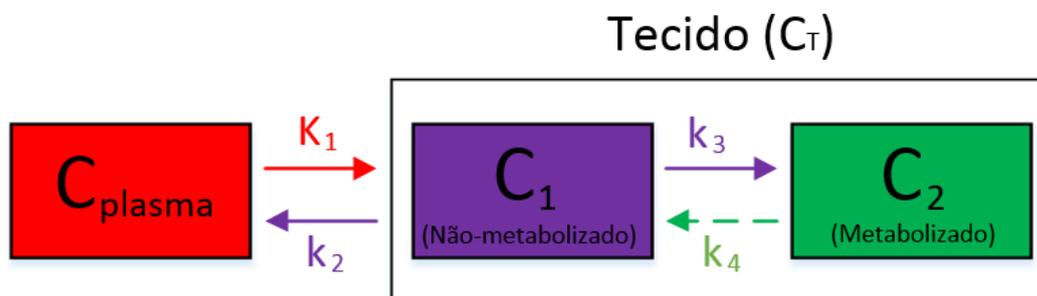
**FIGURA 29** – Curva de Atividade Resultante (no miocárdio) contendo a informação da concentração do radiofármaco no tecido em cada um dos frames ao longo do tempo da simulação realística PET. O usuário deve escolher uma região contendo músculo cardíaco em qualquer um dos volumes.

### 3.7.3 Modelo Compartmental

Uma vez que o radiofármaco entra nas câmaras cardíacas, fica bem misturado com o sangue e posteriormente é distribuído pela circulação arterial. Quando atinge o leito capilar, é onde finalmente ocorre a troca com o tecido. Uma fração do traçador é conduzida dentro do tecido e metabolizado, o restante é transportado de volta para o coração, onde um novo processo de circulação será iniciado [Carson; Cobelli, 2001].

Por meio de um modelo PET-FDG foi analisado a absorção do FDG, o qual tem um comportamento semelhante a glucose, com a finalidade de calcular a taxa metabólica da glicose na região estudada.

O processo metabólico (neste caso do FDG) pode ser modelado como um sistema com três setores onde o traçador irá transitar. Este sistema é denominado mais formalmente como um modelo de três compartimentos, e pode ser esquematizado como mostrado na Figura 30:



**FIGURA 30** – Modelo matemático com três compartimentos utilizado para o estudo da quantificação da taxa metabólica do coração e/ou do cérebro.

Sendo que:

- $C_{plasma}$  corresponde à concentração de radiofármaco no plasma sanguíneo e também à função de excitação do sistema.
- $C_1$  corresponde à concentração de radiofármaco no compartimento 1.
- $C_2$  corresponde à concentração de radiofármaco no compartimento 2.
- $K_1$ ,  $k_2$ ,  $k_3$  e  $k_4$  correspondem aos parâmetros que regulam o modelo e não variam com o tempo.

Existe uma membrana justamente entre o compartimento que representa o sangue ou plasma ( $C_{plasma}$ ) e o compartimento do tecido ( $C_1$  e  $C_2$ ), através do qual o material vai se propalar. Tanto a glicose quanto o FDG vem do sangue, passam pela membrana, até a célula ROI (seta vermelha), indicada pela constante  $K_1$ . Uma vez lá, esta sofre uma única etapa metabólica,  $k_3$  (seta roxa para a direita). O movimento ou reação inverso é indicada por,  $k_2$  e  $k_4$ , respectivamente (setas apontando para a esquerda). Todas as constantes,  $K_1 - k_4$ , representam a taxa de "movimento" de um compartimento para outro. No caso específico das constantes  $K_1$  e  $k_2$ , representam um movimento através de uma barreira física. No entanto, as constantes  $k_3$  e  $k_4$ , representam na realidade uma reação química [Anderson, 1983].

A seguir vamos continuar com a sequência para entender como esse sistema de compartimentos trabalha. Uma vez que o processo tem início, o sangue vai perder continuamente substância radioativa, exponencialmente. O compartimento,  $C_1$  (Não-metabolizado), vai apresentar em primeiro lugar um pico um pouco atrasado em relação ao pico do compartimento sanguíneo. Depois  $C_1$  vai perder atividade já que o traçador vai retornar ao sangue e também uma parte vai vazar no compartimento metabolizado,  $C_2$ . Posteriormente,  $C_1$  vai seguir de perto a curva de atividade do sangue no tempo. Na sua vez, o compartimento Metabolizado tem um pico que atinge o seu valor máximo 45 minutos após a injeção. Se  $k_4$  for ignorado (ou definido como 0),  $C_2$  continuaria subindo lentamente. Neste modelo, desde que  $k_4$  é diferente de zero toda a radioatividade, eventualmente, vai acabar no organismo. Em um exame PET real com FDG, a maior parte do elemento radioativo é eliminado através dos rins e da bexiga [Matis; Wehrly; Gerald, 1983].

Utilizando a lei de conservação de massa para análise da situação representada pelo modelo, têm-se:

$$\frac{dC_1(t)}{dt} = K_1 \cdot C_{plasma}(t) - k_2 \cdot C_1(t) - k_3 \cdot C_1(t) + k_4 \cdot C_2(t) \quad (16)$$

$$\frac{dC_2(t)}{dt} = k_3 \cdot C_1(t) - k_4 \cdot C_2(t) \quad (17)$$

Note-se que a concentração de radiofármaco nos compartimentos inicialmente é nula  $C_1(t = 0) = 0$  e  $C_2(t = 0) = 0$ . Após a resolução das equações (16) e (17), pode-se obter as expressões que descrevem a troca de material nos compartimentos  $C_1$  e  $C_2$ :

$$C_1(t) = \frac{K_1}{\alpha_2 - \alpha_1} [(k_4 - \alpha_1)e^{-\alpha_1 t} + (\alpha_2 - k_4)e^{-\alpha_2 t}] \otimes C_{plasma}(t) \quad (18)$$

$$C_2(t) = \frac{K_1 k_3}{\alpha_2 - \alpha_1} [e^{-\alpha_1 t} - e^{-\alpha_2 t}] \otimes C_{plasma}(t) \quad (19)$$

onde  $\alpha_1$  e  $\alpha_2$  são representados por:

$$\alpha_1 = \frac{k_2 + k_3 + k_4 - \sqrt{(k_2 + k_3 + k_4)^2 - 4k_2 k_4}}{2} \quad (20)$$

$$\alpha_2 = \frac{k_2 + k_3 + k_4 + \sqrt{(k_2 + k_3 + k_4)^2 - 4k_2 k_4}}{2} \quad (21)$$

Entretanto, a tomografia por emissão de pósitrons não tem uma resolução suficientemente alta para permitir a distinção entre os compartimentos  $C_1$  e  $C_2$ . Por esse motivo as concentrações da substância marcadora nos dois compartimentos costumam ser somadas e consideradas como a concentração de atividade radioativa total tecidual ( $C_{TAC}$ ). Portanto, tem-se:

$$C_{TAC}(t) = C_1(t) + C_2(t) \quad (22)$$

$$C_{TAC}(t) = \frac{K_1}{\alpha_2 - \alpha_1} [(k_3 + k_4 - \alpha_1)e^{-\alpha_1 t} + (\alpha_2 - k_3 - k_4)e^{-\alpha_2 t}] \otimes C_{plasma}(t) \quad (23)$$

Se considerarmos que  $k_4 = 0$  em (20) e (21), teríamos que:

$$\alpha_1 = 0 \quad (24)$$

$$\alpha_2 = k_2 + k_3 \quad (25)$$

Com o qual a Equação 23 ficaria finalmente expressa da seguinte maneira:

$$C_{TAC}(t) = \frac{K_1}{k_2 - k_3} [k_3 + k_2 \cdot e^{-(k_2 + k_3)t}] \otimes C_{plasma}(t) \quad (26)$$

Ora, em posse dos valores iniciais de atividade no sangue e no tecido, conforme apresentado na Tabela 3, podemos estimar os valores das constantes metabólicas que justamente inter-relacionam esses valores de atividade por meio da expressão 26.

Os valores referência utilizados na simulação para o conjunto de constantes  $K$ 's que representam a troca de material na região estudada foram os seguintes:

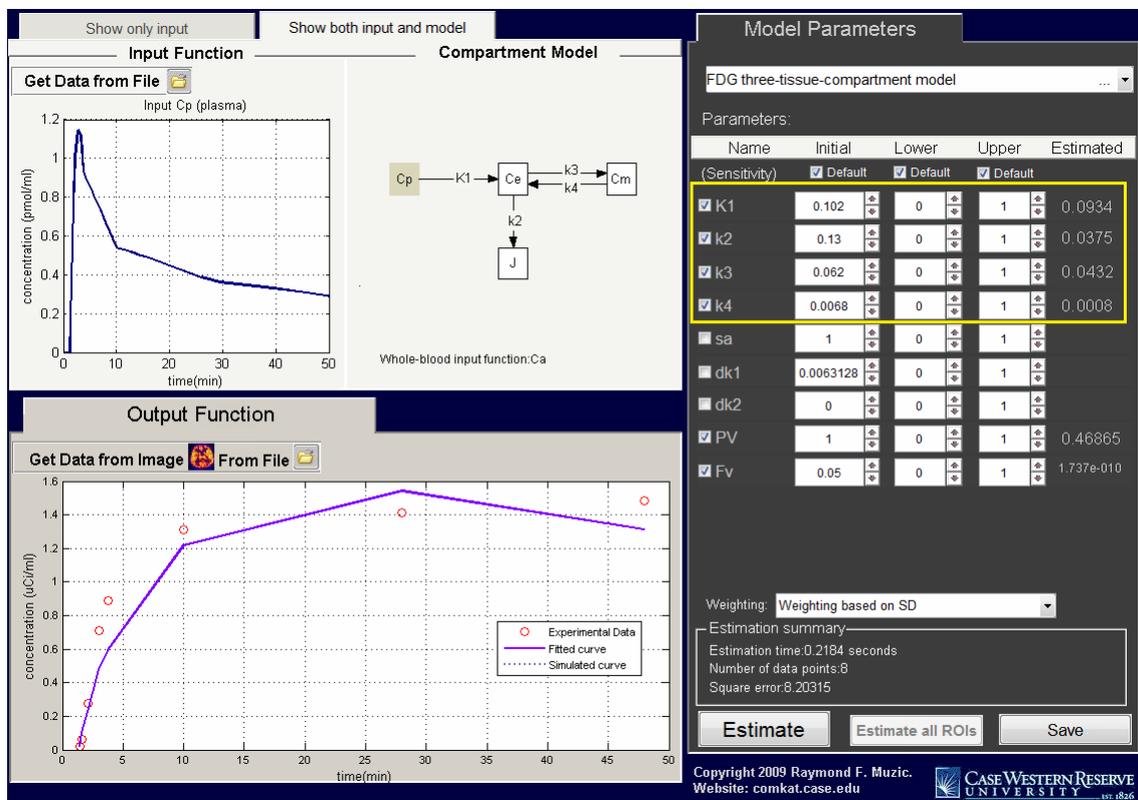
$$K_1 = 0.10; k_2 = 0.04; k_3 = 0.05; k_4 = 0.0 \quad (27)$$

O objetivo é estimar estas constantes, supondo-as desconhecidas, a partir de  $C_{plasma}(t)$  e  $C_{TAC}(t)$ . Então, foram utilizadas as Curvas de Atividade de Entrada no sangue e a Curva de Atividade Resultante no tecido obtidas diretamente das imagens 3D, como exibidas anteriormente. Uma vez que as curvas de atividade

foram geradas, o COMKAT será capaz de fornecer uma estimativa dos valores dos parâmetros do modelo compartimental definido (três compartimentos).

Na Figura 31 são mostradas as curvas de atividades, o modelo compartimental e os valores das constantes  $K_i$  estimados destacados dentro da caixa amarela.

Nota-se que os valores estimados são consistentes com os valores atribuídos na simulação, mostrando que todo o processo desde a geração das emissões até a estimativa dos parâmetros da modelagem compartimental está essencialmente correto.



**FIGURA 31** – Janela principal do COMKAT com as Curvas de Atividade de Entrada (plasma) e Resultante (miocárdio) e os parâmetros metabólicos característicos estimados:  $K_1 = 0.0934$ ,  $k_2 = 0.0375$ ,  $k_3 = 0.0432$  e  $k_4 = 0.0008$ .

Uma vez que os parâmetros metabólicos foram calculados, o último passo é calcular a taxa metabólica de glucose MRGlc (do inglês *Metabolic Rate of Glucose consumption*).

MRGlc é calculado utilizando as constantes metabólicas como se indica na equação seguinte:

$$MRGlc \left( \frac{\mu\text{moles}}{100\text{g}} \right) \frac{1}{\text{min}} = \frac{gl(\text{mg of glucose}/100\text{ml of plasma})}{LC \times 0.18(\text{mg}/\mu\text{moles})} \times \frac{K_1(\text{ml/g/min}) \cdot k_3(\text{min}^{-1})}{k_2(\text{min}^{-1}) + k_3(\text{min}^{-1})} \quad (28)$$

A unidade da MRGlc é  $\mu\text{moles}$  de glicose / 100 g de tecido / min.

Pode-se ver que no denominador da equação (28) se tem considerado um fator igual a 0.18. Esse fator é inferido pelo fato que 1  $\mu\text{moles}$  de  $^{18}\text{F} - \text{FDG}$  = 0.180 mg, equivalência necessária para converter MRGlc em  $\mu\text{moles}$ .

Por outro lado, sabe-se que o metabolismo do FDG é análogo ao metabolismo da glicose, pelo menos nas primeiras fases. No entanto, já que são moléculas distintas, elas vão experimentar marcadas diferenças como por exemplo as velocidades de captação (*uptake*), fosforilação, desfosforilação, e principalmente a difusão deles para fora da célula [Jadvar, 2010].

Baseado nessas diferenças e a fim de não ignorá-las, é utilizada a constante LC na equação (26), a qual é definida como a razão entre a fracção de extração artério-venosa de  $^{18}\text{F} - \text{FDG}$  ao da glicose em condições de estado estacionário e quando  $k_4$  é desprezível. LC tem sido calculada por alguns investigadores, por exemplo, para cálculos feitos usando o cérebro humano LC = 0.42 [Phelps, 1979], para outros órgãos pode ser considerado LC = 0.52 [Reivich et al., 1985].

Além disso, precisa-se de mais um ingrediente, do valor da glicemia (*gl*) o qual é definido como a concentração de glicose no plasma em estado estacionário. Esse valor de glicemia é obtido clinicamente por meio de duas ou três amostras de sangue retiradas do paciente, considerando-se a média dos valores de glicemia de cada uma das amostras [Reivich et al., 1985].

Assim, considerando-se os valores da glicemia, da constante LC, juntamente com os valores das constantes paramétricas  $K_1 - k_4$ , pode ser estimada o valor da MRGlc através da Equação 28.

### 3.7.4 Análise de Estudos Clínicos PET Reais

Como indicado na secção 3.2.2, tivemos acesso aos casos clínicos do Estudo de Perfusão Miocárdica com Rb-82 desenvolvido no InCor. Esse estudo utilizou o  $^{82}\text{Rb}$ , cujo radisótopo tem uma meia-vida de 76 segundos, no tomógrafo PET/CT da marca PHILIPS.

Segundo os autores, “a técnica se apresentou superior à do SPECT em termos de sensibilidade e especificidade além de fornecer dados sobre débito coronário, reserva coronária, fração de ejeção, motilidade e espessamento miocárdico em apenas alguns minutos para exames em estresse e repouso. Do mesmo modo, foi possível observar se o paciente tinha falta de sangue durante o estresse”.

Durante o Estudo de Perfusão Miocárdica com Rb-82 foram realizados 750 estudos perfusionais em repouso e estresse, dos quais foram viabilizados alguns casos clínicos (N=4). Esses exames PET foram selecionados aleatoriamente e passaram por um processo de anonimização como parte das exigências da instituição de origem. Cada um dos exames PET tem dois estudos dinâmicos (estresse e repouso).

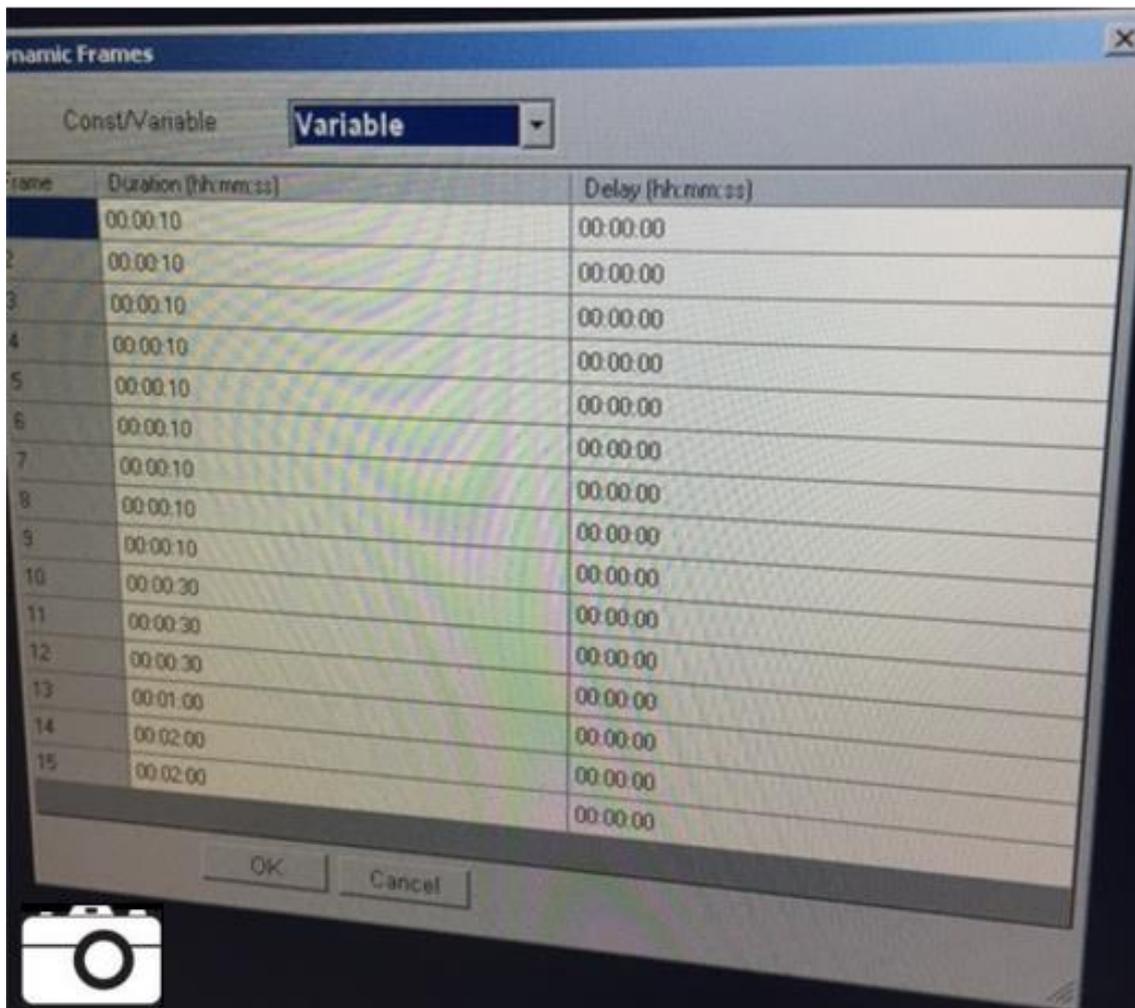
No entanto, devemos salientar que os estudos dinâmicos obtidos com o uso do rubídeo-82 apresentam algumas desvantagens, tais como um aumento de ruído devido à limitação da dose injetada e à meia-vida que tem o radioisótopo. De fato, isso é bastante conhecido sobre tudo quando comparadas com estudos dinâmicos feitos com  $^{18}F - FDG$ .

Assim, o processo de avaliação dos dados PET reais através do COMKAT precisou da incorporação de dois detalhes sumamente importantes. Por um lado, foi necessário criar um novo modelo (*template*) com 3 compartimentos o qual considera o uso de  $^{82}Rb$  como radiofármaco. Para esse fim, foram utilizadas linhas de comando em MatLab junto com alguns recursos (programas, funções, etc.) disponíveis no COMKAT. No Apêndice L, pode-se ver todos os detalhes do novo modelo e como carregá-lo dentro do COMKAT.

Além disso, outro detalhe relevante considerado no momento de avaliar os dados reais, foi a correção do tempo de cada um dos frames. Isto é, por alguma razão o equipamento da Philips quando gera o arquivo DICOM não grava corretamente o tempo de cada frame e isso foi corrigido.

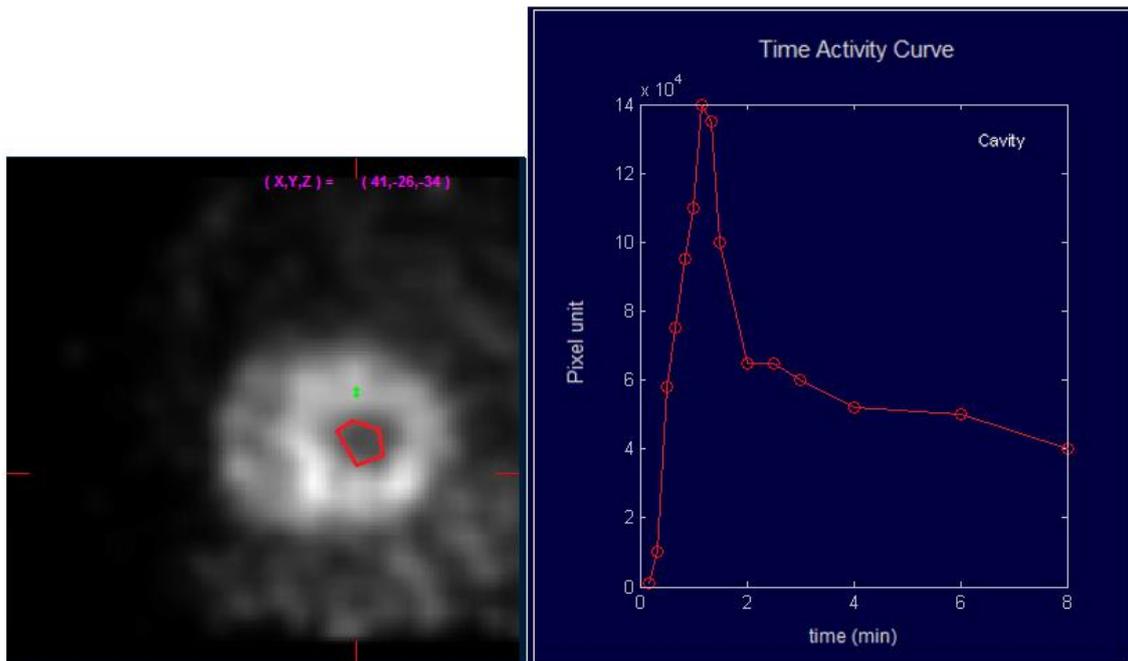
Portanto, dentro do novo modelo foram incorporados os tempos certos para os 15 frames que conformam o estudo PET dinâmico tal como aparece na workstation vinculada ao aparelho PET/CT da Philips.

Essa lista com os tempos corretos é mostrado na Figura 32.



**FIGURA 32** – Imagem capturada da workstation vinculada ao aparelho PET/CT da PHILIPS, onde se mostram os tempos dos 15 frames que conformam o estudo dinâmico.

Uma vez com o modelo implementado e carregado no COMKAT, a Curva de Atividade de Entrada (no sangue) foi gerada conforme indicado anteriormente (Figura 33).



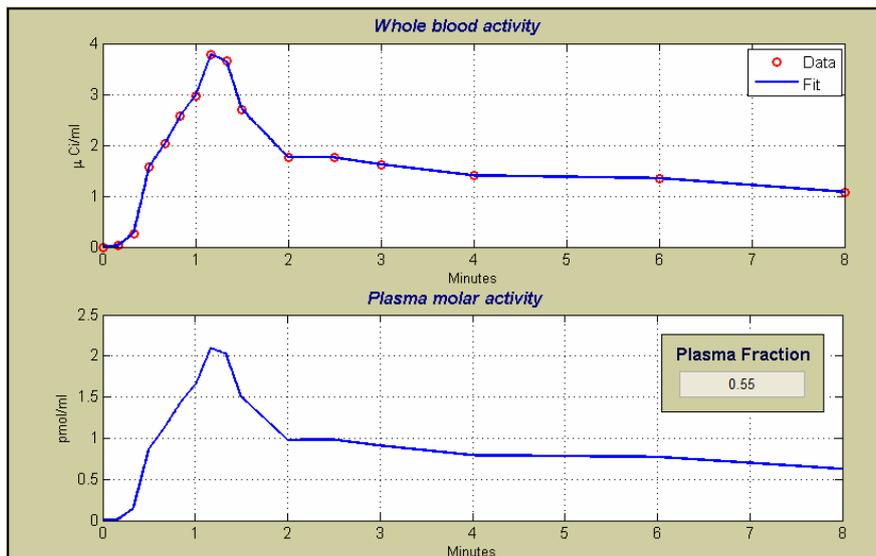
**FIGURA 33** – Curva de Atividade de Entrada contendo a informação da concentração do radiofármaco no sangue em cada um dos frames ao longo do tempo do exame PET dinâmico. Para isso, o usuário precisa definir manualmente a região implicada em qualquer um dos volumes que fazem parte do estudo dinâmico.

No entanto, temos que utilizar a concentração do radiofármaco no plasma sanguíneo no processo de quantificação dos parâmetros metabólicos.

Para isto, considera-se que o plasma representa o 55% do volume total do sangue e por meio de uma ferramenta do próprio COMKAT foi gerada a Curva de Atividade de Entrada no plasma sanguíneo.

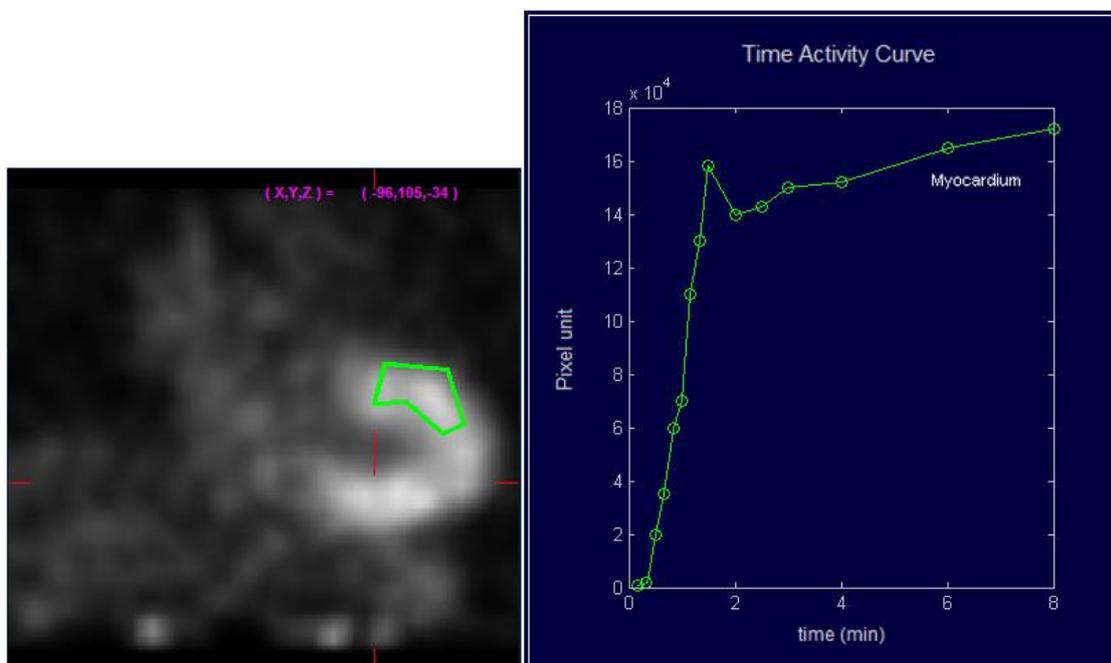
A Figura 34 mostra, na parte superior a Curva de Atividade no sangue e, na parte inferior a Curva de Atividade no plasma.

Note que o software faz uma conversão das unidades para  $\mu Ci/ml$ .



**FIGURA 34** – Curva de Atividade de Entrada contendo a informação da concentração de radiofármaco no sangue (curva de cima) e no plasma sanguíneo (curva de baixo).

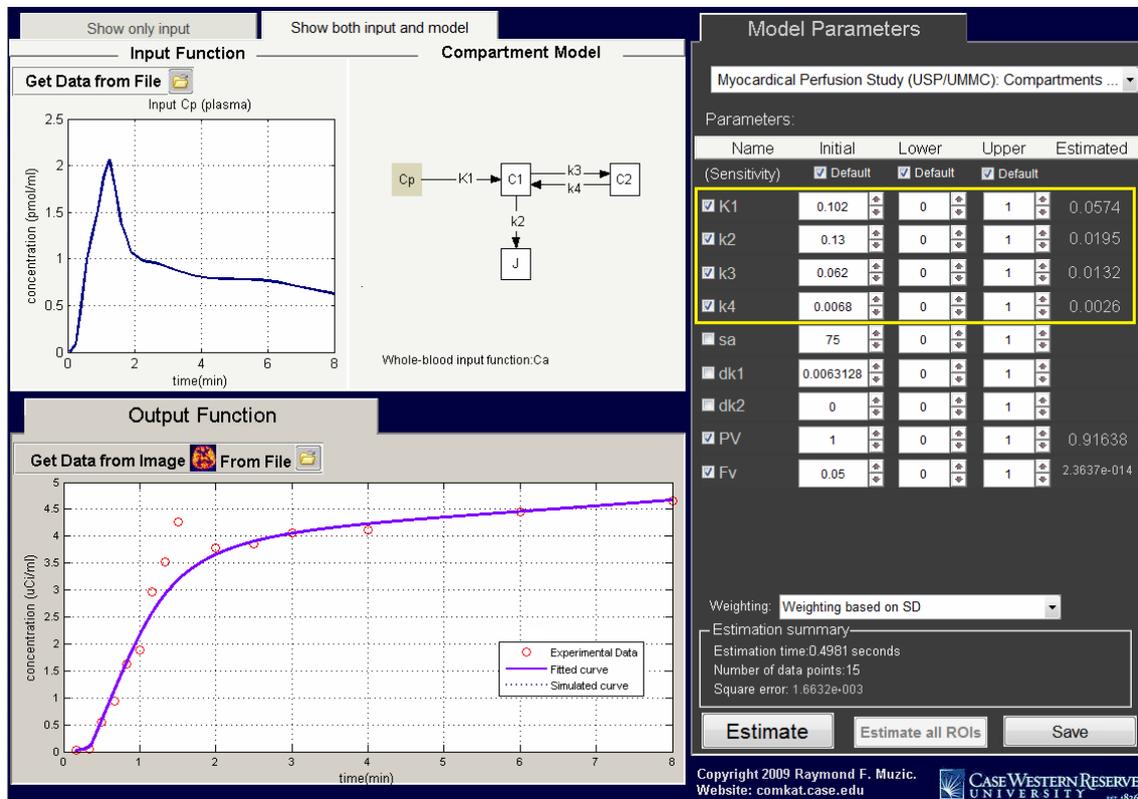
Da mesma forma, a Curva de Atividade Resultante (miocárdio) foi gerada selecionando manualmente uma porção do músculo cardíaco em um dos 15 volumes que conformam o estudo PET dinâmico (Figura 35).



**FIGURA 35** – Curva de Atividade de Resultante contendo a informação da concentração do radiofármaco no tecido cardíaco em cada um dos frames ao longo do tempo do exame PET. Para isso, o usuário precisa definir manualmente a região implicada em qualquer um dos volumes que fazem parte do estudo dinâmico.

Na sequência, a Curva de Atividade de Entrada no plasma sanguíneo e a Curva de Atividade Resultante no tecido (miocárdio) foram carregadas na plataforma COMKAT e, usando o modelo criado “Myocardical Perfusion Study (USP/UMMC): Compartments model(3) using Rb-82” foram calculadas todas as constantes paramétricas que definem os fluxos de troca de material entre os compartimentos e em consequência a dinâmica do sistema.

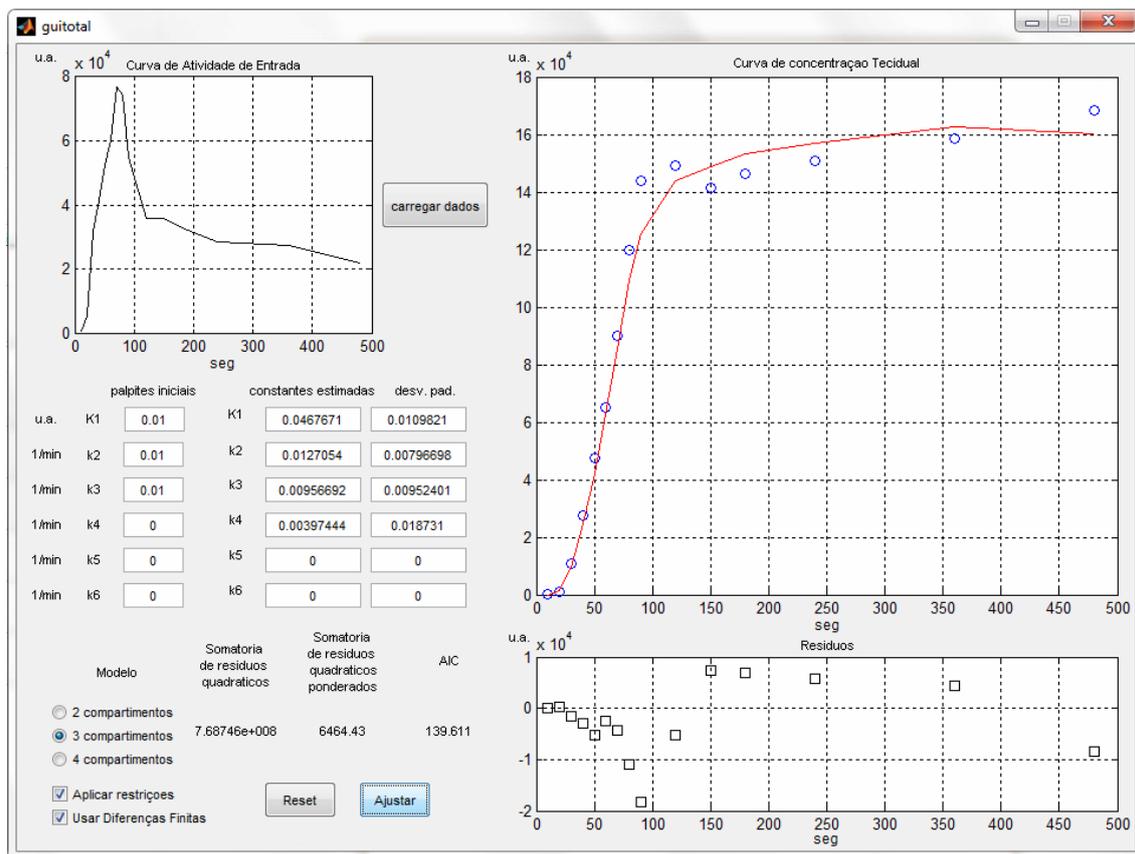
Na Figura 36 são mostradas as curvas de atividade de entrada e resultante, o modelo compartimental e os valores das constantes  $K_1$ ,  $k_2$ ,  $k_3$  e  $k_4$  que foram estimados (destacadas dentro da caixa amarela).



**FIGURA 36** – Janela principal do COMKAT com as Curvas de Atividade de Entrada (plasma) e Resultante (miocárdio) e os parâmetros metabólicos característicos estimados para o caso do Estudo de Perfução Miocárdica PET com  $^{82}\text{Rb}$ . Os parâmetros metabólicos calculados são os seguintes:  $K_1 = 0.0574$ ,  $k_2 = 0.0195$ ,  $k_3 = 0.0132$  e  $k_4 = 0.0026$ .

Complementarmente, uma ferramenta gráfica em MatLab foi utilizada para o cálculo dos parâmetros metabólicos característicos  $K_1$ ,  $k_2$ ,  $k_3$  e  $k_4$ . Essa ferramenta foi desenvolvida no nosso laboratório como parte do trabalho titulado “Estimação de Parâmetros de Modelos Compartimentais para Tomografia por Emissão de Pósitrons” [Silva, 2010].

Porém, uma vez que essa ferramenta não tem a capacidade de gerar as Curvas de Atividade diretamente das imagens PET, foi necessário introduzir esses dados manualmente por meio de um arquivo texto. Com os dados incorporados, foram calculadas as constantes metabólicas do modelo de 3 compartimentos como indicado a seguir na Figura 37.



**FIGURA 37** – Janela principal da ferramenta gráfica GuiTotal com as Curvas de Atividade de Entrada (plasma) e Resultante (miocárdio) e os parâmetros metabólicos característicos estimados para o caso do Estudo de Perfusão Miocárdica PET com  $^{82}Rb$ . Os parâmetros metabólicos calculados são os seguintes:  $K_1 = 0.0467871$ ,  $k_2 = 0.0127054$ ,  $k_3 = 0.00956692$  e  $k_4 = 0.00397444$ .

# Capítulo 4

## RESULTADOS

### 4.1 Avaliação da sequência de processamento aplicada aos volumes simulados do *phantom* MASH

Como ponto inicial, foi considerado o músculo cardíaco (miocárdio) como ROI nesta avaliação. O coração fica localizado dentro da região torácica do *phantom* antropomórfico MASH. Só para citar alguns dados, sabe-se que a estrutura do miocárdio está composta por 181855 voxels e a do sangue por 164892 voxels.

Para esta avaliação, foram executados os processos de simulação, reconstrução 3D, filtragem 3D e segmentação 3D. Na etapa de reconstrução foram aplicadas a abordagem analítica (algoritmo FBP3D) e iterativa (OSMAPOS). Na etapa de filtragem foram consideradas três diferentes abordagens: Filtro Anscombe / Wiener, Bartlett / Wiener e Freeman&Tukey / Wiener. A segmentação foi através do algoritmo FC com Pesos Dinâmicos. Nas Tabelas 7 e 8, são apresentados os resultados aplicados sobre os 8 volumes simulados utilizando 3 métricas de avaliação: Verdadeiro Positivo (TP), Falso Positivo (FP) e Máxima Distância [Udupa, 2002; Florez Pacheco, 2012], onde a segmentação desempenhou um papel relevante no processo de avaliação das diferentes abordagens de filtragem.

**TABELA 7 –** Resultados da sequência de processamento aplicado sobre os volumes simulados da região torácica do *phantom* MASH reconstruídos por meio do método analítico FBP3D.

F B P 3 D		Anscombe / Wiener			Bartlett / Wiener			Freeman & Tukey / Wiener		
Frame	Tipo de Volume	TP (%)	FP (%)	MaxDist	TP (%)	FP (%)	MaxDist	TP (%)	FP (%)	MaxDist
1	Vol. c/ruído Poisson	85.64	5.84	5	85.64	5.84	5	85.64	5.84	5
	Vol. filtrado	92.81	4.32	3	92.32	5.51	4	92.39	4.87	3
2	Vol. c/ruído Poisson	83.98	6.71	5	83.98	6.71	5	83.98	6.71	5
	Vol. filtrado	92.89	4.17	3	91.47	5.30	4	92.42	4.51	4
3	Vol. c/ruído Poisson	84.41	6.20	5	84.41	6.20	5	84.41	6.20	5
	Vol. filtrado	93.76	4.89	3	92.54	5.24	3	92.87	4.96	3
4	Vol. c/ruído Poisson	87.38	5.25	5	87.38	5.25	5	87.38	5.25	5
	Vol. filtrado	92.76	4.24	3	92.57	5.28	3	93.12	4.78	3
5	Vol. c/ruído Poisson	86.44	6.52	5	86.44	6.52	5	86.44	6.52	5
	Vol. filtrado	92.82	4.11	3	91.28	4.97	3	91.59	4.47	3
6	Vol. c/ruído Poisson	88.59	6.15	5	88.59	6.15	5	88.59	6.15	5
	Vol. filtrado	92.08	4.02	2	90.47	4.53	3	91.12	4.25	3
7	Vol. c/ruído Poisson	89.25	5.74	4	89.25	5.74	4	89.25	5.74	4
	Vol. filtrado	93.91	3.84	2	92.15	4.57	2	92.86	4.21	2
8	Vol. c/ruído Poisson	89.67	5.42	4	89.67	5.42	4	89.67	5.42	4
	Vol. filtrado	94.52	3.76	2	91.58	4.25	2	92.67	3.94	2
	Vol. c/ruído Poisson	<b>86.92 ± 2.16</b>	<b>5.98 ± 0.51</b>	<b>5 pixels</b>	<b>86.92 ± 2.16</b>	<b>5.98 ± 0.51</b>	<b>5 pixels</b>	<b>86.92 ± 2.16</b>	<b>5.98 ± 0.51</b>	<b>5 pixels</b>
	Vol. filtrado	<b>93.19 ± 0.79</b>	<b>4.17 ± 0.34</b>	<b>3 pixels</b>	<b>91.80 ± 0.73</b>	<b>4.96 ± 0.45</b>	<b>4 pixels</b>	<b>92.38 ± 0.69</b>	<b>4.45 ± 0.36</b>	<b>4 pixels</b>

**TABELA 8 –** Resultados da sequência de processamento aplicado sobre os volumes simulados da região torácica do *phantom* MASH reconstruídos por meio do método iterativo OSMAPOSL que considera correção de atenuação e normalização.

OSMAPOSL + AC + NORM		Anscombe / Wiener			Bartlett / Wiener			Freeman & Tukey / Wiener		
Frame	Tipo de Volume	TP (%)	FP (%)	MaxDist	TP (%)	FP (%)	MaxDist	TP (%)	FP (%)	MaxDist
1	Vol. c/ruído Poisson	89.05	4.74	5	89.05	4.74	5	89.05	4.74	5
	Vol. filtrado	93.42	1.95	3	91.89	3.47	3	92.97	2.47	3
2	Vol. c/ruído Poisson	88.74	4.82	5	88.74	4.82	5	88.74	4.82	5
	Vol. filtrado	95.63	1.02	3	91.78	3.51	3	92.68	2.74	1
3	Vol. c/ruído Poisson	89.16	4.67	5	89.16	4.67	5	89.16	4.67	5
	Vol. filtrado	93.94	1.83	3	91.88	2.89	3	92.49	2.69	3
4	Vol. c/ruído Poisson	90.28	4.23	5	90.28	4.23	5	90.28	4.23	5
	Vol. filtrado	94.94	1.81	2	92.78	3.95	3	94.58	2.44	3
5	Vol. c/ruído Poisson	90.94	3.79	5	90.94	3.79	5	90.94	3.79	5
	Vol. filtrado	94.74	1.49	2	92.58	3.45	3	94.74	2.78	3
6	Vol. c/ruído Poisson	90.93	3.83	4	90.93	3.83	4	90.93	3.83	4
	Vol. filtrado	94.84	2.78	2	93.01	3.11	3	95.51	2.16	2
7	Vol. c/ruído Poisson	92.15	3.51	4	92.15	3.51	4	92.15	3.51	4
	Vol. filtrado	93.79	3.04	2	92.74	3.42	2	95.59	2.23	2
8	Vol. c/ruído Poisson	92.35	3.49	4	92.35	3.49	4	92.35	3.49	4
	Vol. filtrado	94.68	2.37	2	92.94	3.09	2	95.67	2.18	2
Vol. c/ruído Poisson		<b>90.45 ± 1.39</b>	<b>4.14 ± 0.55</b>	<b>5</b>	<b>90.45 ± 1.39</b>	<b>4.14 ± 0.55</b>	<b>5</b>	<b>90.45 ± 1.39</b>	<b>4.14 ± 0.55</b>	<b>5</b>
Vol. filtrado		<b>94.50 ± 0.72</b>	<b>2.03 ± 0.66</b>	<b>3</b>	<b>92.45 ± 0.51</b>	<b>3.36 ± 0.33</b>	<b>3</b>	<b>94.28 ± 1.36</b>	<b>2.46 ± 0.26</b>	<b>3</b>

## 4.2 Estimação dos parâmetros metabólicos característicos do Modelo Compartimental que representa o coração do MASH

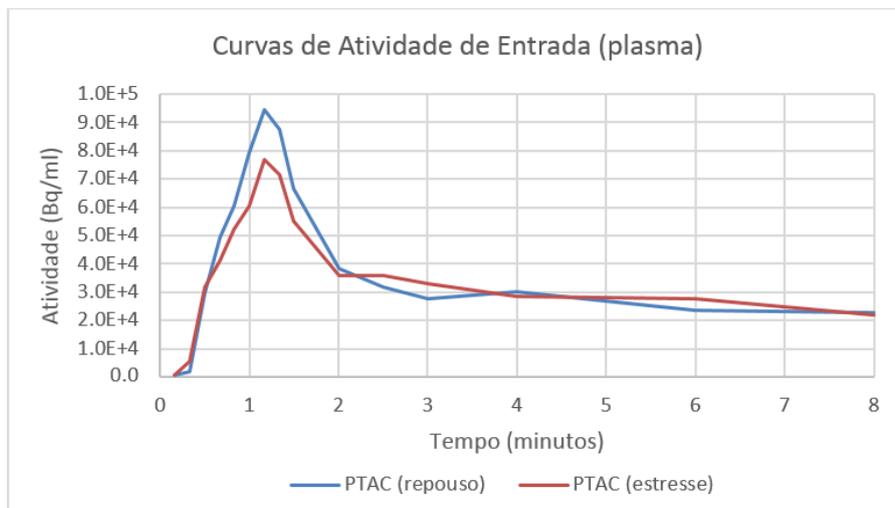
Foram analisadas as diferentes abordagens utilizadas ao longo do projeto, e calculados os parâmetros metabólicos característicos do Modelo que representa o coração do MASH. Na Tabela 9 são mostrados os valores das constantes  $K_i$  para cada procedimento realizado.

**TABELA 9 –** Parâmetros metabólicos característicos do Modelo Compartimental com três compartimentos baseados no *phantom* antropomórfico MASH para cada uma das abordagens utilizadas neste projeto. Adicionalmente na parte inferior estão os valores das constantes características que representam o modelo utilizado, calculadas teoricamente.

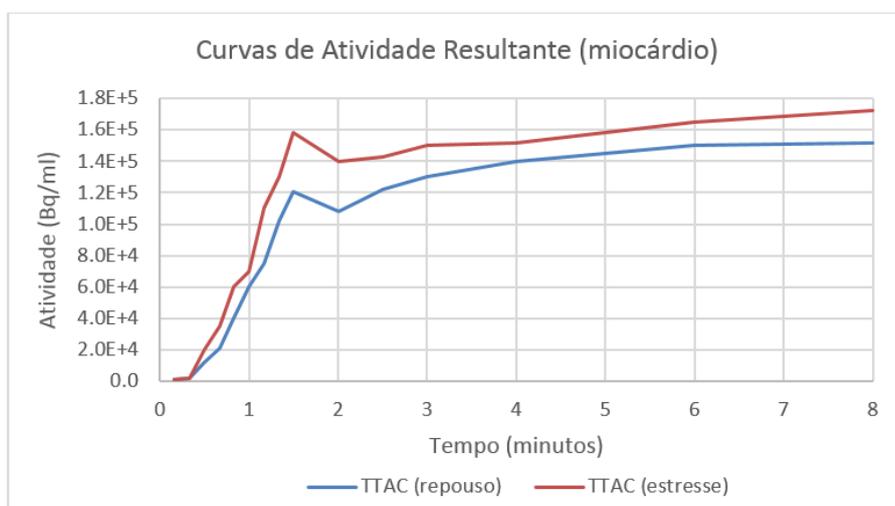
Procedimentos Aplicados	$K_1$	$k_2$	$k_3$	$k_4$
FBP2D	0.3318	0.2447	0.1929	0.0093
FBP2D + FILTRO A/W	0.2759	0.2255	0.1427	0.0082
FBP3D	0.1452	0.1327	0.1248	0.0086
FBP3D + FILTRO A/W	0.1259	0.0954	0.0852	0.0065
OSMAPOSL	0.1271	0.1186	0.1157	0.0076
OSMAPOSL + NORM + AC	0.1146	0.0627	0.0854	0.0055
OSMAPOSL + NORM + AC + FILTRO A/W	0.0934	0.0375	0.0432	0.0008
Valores referência (Convolução)	0.10	0.04	0.05	0.00

### 4.3 Estimação dos parâmetros metabólicos do Modelo Compartmental utilizando exames PET reais.

Como indicado, temos utilizado N=4 casos PET reais, cada um deles com dois estudos dinâmicos: um em repouso e outro em estresse. Nas Figuras 38 e 39 são mostrados a comparação da Curvas de Atividade no plasma e no miocárdio de um dos pacientes avaliados, respectivamente.



**FIGURA 38** – Comparação das Curvas de Atividade de Entrada no plasma tanto em repouso quanto em estresse, pertencentes a um dos casos do Estudo de Perfusão Miocárdica (Rb-82).



**FIGURA 39** – Comparação das Curvas de Atividade Resultante no miocárdio tanto em repouso quanto em estresse, pertencentes a um dos casos do Estudo de Perfusão Miocárdica (Rb-82).

Nas Tabelas 10 e 11, são apresentados os resultados dos parâmetros metabólicos  $K_1$ ,  $k_2$ ,  $k_3$  e  $k_4$  que representam o Modelo Compartimental (3 compartimentos) utilizado na avaliação das imagens PET pertencentes ao Estudo de Perfusão Miocárdica com Rb-82, calculados através do COMKAT e do GuiTotal, respectivamente.

Deve-se indicar que as curvas de atividade de entrada e resultante foram construídas considerando as mesmas regiões nos dois exames dinâmicos de cada paciente (repouso e estresse), a fim de compará-los.

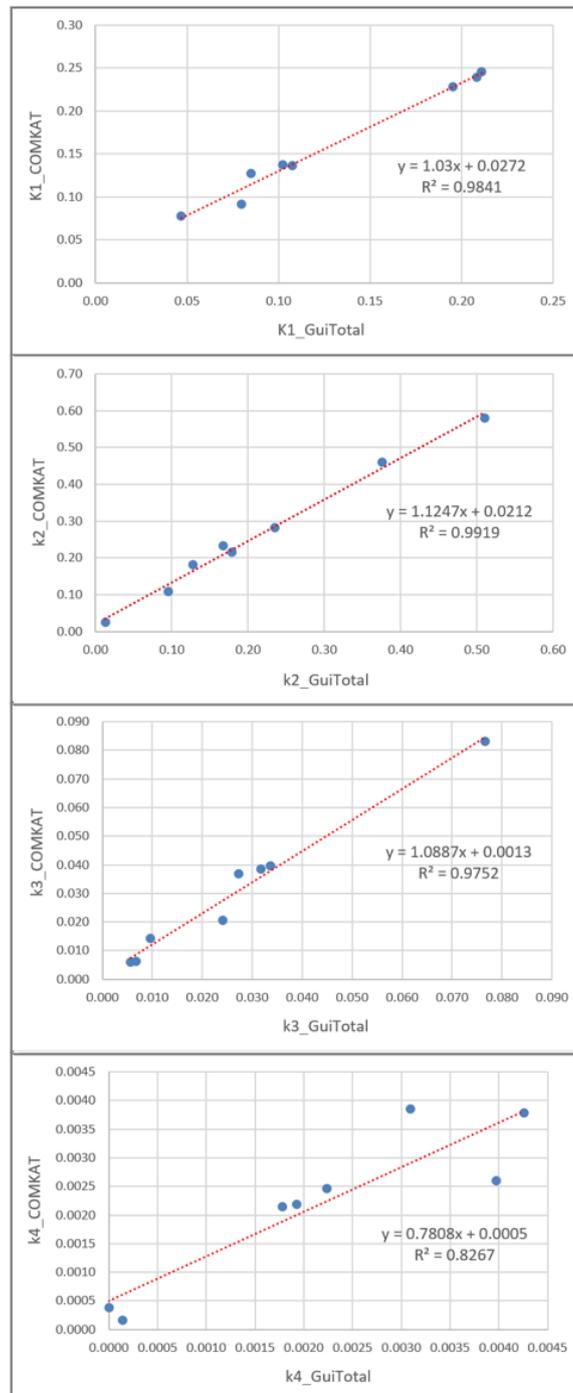
**TABELA 10** – Cálculo dos parâmetros metabólicos característicos do Modelo Compartimental com três compartimentos através do COMKAT aplicados sobre os casos PET reais.

No.	Paciente ID	Tipo do Estudo (código)	$K_1$	$k_2$	$k_3$	$k_4$	$\frac{K_1 * k_3}{k_2 + k_3}$
1	S53240	Repouso S25820	0.1364	0.1083	0.0060	0.0004	0.007154
		Estresse S24850	0.2287	0.2155	0.0062	0.0002	0.006364
2	S53910	Repouso S31900	0.2393	0.5797	0.0385	0.0021	0.015820
		Estresse S15270	0.2455	0.4595	0.0396	0.0022	0.021067
3	S54030	Repouso S24853	0.1378	0.2320	0.0370	0.0039	0.018933
		Estresse S193600	0.1274	0.1816	0.0205	0.0025	0.012924
4	S54900	Repouso S45180	0.0915	0.2824	0.0831	0.0038	0.026555
		Estresse S47130	0.0774	0.0252	0.0142	0.0026	0.039606

**TABELA 11** – Cálculo dos parâmetros metabólicos característicos do Modelo Compartimental com três compartimentos através do GuiTotal [Silva, 2010] aplicados sobre os casos PET reais.

No.	Paciente ID	Tipo do Estudo (código)	$K_1$	$k_2$	$k_3$	$k_4$	$\frac{K_1 * k_3}{k_2 + k_3}$
1	S53240	Repouso S25820	0.107548	0.095070	0.005648	0.000000	0.006031
		Estresse S24850	0.195143	0.178707	0.006685	0.000141	0.007037
2	S53910	Repouso S31900	0.208122	0.510410	0.031778	0.001778	0.012912
		Estresse S15270	0.210916	0.375827	0.033627	0.001929	0.018775
3	S54030	Repouso S24853	0.102434	0.166876	0.027277	0.003094	0.014391
		Estresse S193600	0.085074	0.127752	0.024139	0.002239	0.013520
4	S54900	Repouso S45180	0.079574	0.234993	0.076613	0.004262	0.025481
		Estresse S47130	0.046767	0.012705	0.009567	0.003974	0.026824

Uma vez calculados as constantes metabólicas utilizando tanto o COMKAT quanto o GuiTotal, os resultados foram comparados por meio do coeficiente de correlação de Pearson ( $r$ ), a fim de conhecer a relação linear entre os dois conjuntos de dados (Figura 40). Assim, considerando os quatro parâmetros metabólicos característicos foi obtida uma correlação média de  $r = 0.971$ .



**FIGURA 40** – Relação linear entre os parâmetros metabólicos calculados através do COMKAT e do GuiTotal por meio do coeficiente de correlação de Pearson ( $r = 0.971$ ).

# Capítulo 5

## DISCUSSÃO

### 5.1 Considerações relevantes

- a) O nosso primeiro obstáculo foi lidar com as longas horas de simulação (superiores a 3 dias). Por um lado, o computador inicialmente utilizado não dava conta da magnitude dos processos pelo que foi necessário migrar de PC e assim, foram reduzidos os tempos de processamento e foram executadas várias simulações em forma paralela através dos múltiplos núcleos do novo computador. Por outro lado, as quedas de energia elétrica foram outra causa que originaram a perda de várias simulações (inclusive usando um nobreak de 90 min de autonomia). Esses problemas promoveram a migração da execução das nossas simulações para o Laboratório de Computação Científica Avançada (LCCA) da Universidade de São Paulo, com o qual a continuidade das nossas simulações esteve garantida. Finalmente, foi utilizado até o final do projeto um computador de alto desempenho com as características indicadas na seção 3.1 adquirido pelo Laboratório de Engenharia Biomédica da EPUSP.
- b) Cada migração de computador originou trabalhos adicionais de instalação. Para aliviar esse processo, temos desenvolvido dois manuais com a sequência detalhada de passos para a instalação das versões mais recentes das plataformas GATE (v6.2) e STIR (v2.4).
- c) Outro obstáculo superado foi a magnitude dos dados processados e a conversão dos tipos de arquivos de saída/entrada em cada um dos blocos.
- d) Nosso projeto teve como prioridade a aproximação constante dos processos de simulação ao realismo dos exames PET. É assim que durante a execução do projeto temos experimentado uma notável evolução no âmbito desta finalidade. No que respeita aos *phantoms*, temos migrado de uma esfera representativa do órgão cardíaco (incluindo material do coração, coeficiente de atenuação, etc.), passando por modelos específicos como o caso do *phantom* de Hoffman (cérebro) e o *phantom* NCAT-4D (coração). Finalmente,

todos os *phantoms* anteriormente mencionados foram substituídos pelo *phantom* antropomórfico MASH, que sendo uma representação do corpo humano inteiro permite o estudo de qualquer órgão.

- e) Em conformidade com o acima exposto, na etapa de simulações foram tomadas em consideração doses clínicas reais conforme a procedimentos atualmente aplicados na modalidade PET e que são regulamentados pela Comissão Internacional de Medicina Nuclear.
- f) O efeito da aproximação para situações realistas significou também a degradação dos resultados obtidos por dois motivos: Primeiro, porque os valores das atividades (bolo injetado) consideradas nas simulações ficaram limitadas às quantidades reais utilizados nos procedimentos clínicos atualmente. Em segundo lugar, porque nas simulações foram usadas todas as estruturas que pertencem à região torácica do phantom MASH (onde fica localizado o coração) com o qual foram acrescentadas as atenuações como produto da quantidade de materiais existentes na simulação e, portanto, houve também redução do número de detecções coincidentes que irão resultar no volume reconstruído.
- g) A discrepância entre os parâmetros metabólicos calculados  $K_1 = 0.0934$ ,  $k_2 = 0.0375$ ,  $k_3 = 0.0432$  e  $k_4 = 0.0026$  (através do processo que considera a reconstrução iterativa com normalização e correção de atenuação, além de filtragem) e os valores de referência  $K_1 = 0.10$ ,  $k_2 = 0.04$ ,  $k_3 = 0.05$  e  $k_4 = 0.0$ , citados na secção 3.7.3 - equação (25), deve-se ao fato das imperfeições do processo de simulação. Por exemplo em todas as situações, os valores de intensidade obtidos nas imagens foram multiplicados pelo fator arbitrário pelo qual a atividade foi dividida (Tabela 5). Porém, esse procedimento que foi feito para permitir terminar as simulações em tempo hábil acarreta em distorções nas simulações, pois o fato de reduzir o número da média de distribuição Poisson altera a relação sinal/ruído, podendo ter influenciado na estimativa das atividades finais.
- h) Apesar das discrepâncias mencionadas no item anterior, nota-se que os valores estimados são consistentes com os valores atribuídos na simulação, mostrando que todo o processo desde a geração das emissões até a estimativa dos parâmetros da modelagem compartimental está essencialmente correto.

- i) Em relação aos casos PET 3D reais, este estudo apresenta algumas limitações: Primeiro, o reduzido número de casos clínicos PET reais utilizados (N=4), que foram selecionados aleatoriamente de um total de (N=750) pacientes com condições clínicas diversas. Cada um dos casos analisados está constituído por dois estudos dinâmicos (em repouso e em estresse). Em segundo lugar, todos os casos clínicos estão ligados a uma única instituição médica (InCor-HCFMUSP). Contudo, mostrou-se a aplicabilidade do nosso processo em casos reais.
- j) Foi possível diferenciar a condição de repouso e estresse nos casos clínicos obtidos, por meio das curvas de atividade geradas.
- k) Há uma alta correlação ( $r = 0.971$ ) entre os resultados das constantes metabólicas dos casos PET reais, calculados através do COMKAT e do entorno gráfico GuiTotal. Considera-se que a diferença das estimações pode ter sido causada por imprecisões e/ou erros no processo de fornecimento manual dos dados das curvas de atividade para a plataforma GuiTotal.
- l) Respeito dos casos PET reais utilizados, desde que os exames foram feitos como  $^{82}\text{Rb}$ , não foi possível calcular a taxa metabólica de consumo de glicose (MRGlc) desse grupo de pacientes. Além de que, para o cálculo da taxa metabólica de consumo de glicose (MRGlc) é indispensável conhecer os dados clínicos de glicemia. No entanto, foi possível aplicar a nossa metodologia na obtenção de parâmetros metabólicos em casos clínicos reais.
- m) Considerando-se que a dose de Rb-82 injetado nos pacientes é inferior da dose regular de FDG em um exame de rotina, e sabendo a grande diferença de meia-vida destes radiofármacos, seria ser uma boa maneira de aplicar o nosso processo de quantificação usando imagens PET dinâmicas com FDG. Os resultados deste processo deveriam ser confrontados com o quadro clínico dos pacientes, com o assessoramento do(s) médico(s) especialista(s).
- n) O COMKAT, a diferença de outros produtos comerciais utilizados para quantificar o metabolismo (e.g.  $\pi$ .pmod, CSI, Flow-Quant, entre outros), é gratuito e tem a particularidade de determinar os parâmetros metabólicos de uma região determinada manualmente pelo usuário. Isto pode ser visto como uma desvantagem, uma vez que outros softwares geram mapas polares com os parâmetros relativos da função metabólica de todas as regiões do órgão (e.g. coração). No entanto, com o COMKAT é possível fazer comparações

diretas entre diferentes áreas, por exemplo alguma região com presença tumoral versus outra área hígida.

m) Baseados no Diagrama de Blocos geral do projeto (Figura 12), deve-se ressaltar duas observações importantes:

- A etapa de Pos-Filtragem que pertence ao processamento de imagens 3D dinâmicas não tem sido utilizada devido a que os processos de Filtragem 3D e de Segmentação 3D alcançaram bons resultados.
- Além disso, não foi possível realizar a avaliação dos especialistas usando dados clínicos retrospectivos (processo definido no Bloco J do diagrama).

## 5.2 Importância e contribuições da pesquisa

- a) Tem-se bem definida, implementada e com total controle toda a plataforma desenhada para atingir os objetivos do projeto, desde o início até a quantificação metabólica dos volumes PET dinâmicos simulados e reais. Todos os softwares utilizados no desenvolvimento deste projeto são gratuitos e de fácil acesso.
- b) Foram avaliados diferentes métodos de reconstrução, filtragem e segmentação. Esta comparação permitiu selecionar as abordagens com melhores desempenhos, com base nos objetivos do projeto.
- c) Com o intuito de transmitir conhecimento, foi envolvido um aluno de Iniciação Científica para desenvolver uma parte deste projeto. Durante os 12 meses de duração do projeto IC foram transmitidos todos os *conhecimentos* possíveis no desenvolvimento do projeto e do aluno. O projeto foi bem conduzido, satisfatoriamente concluído e apresentado no 23º Simpósio Internacional de Iniciação Científica da USP (SIICUSP), realizado o passado mês de setembro de 2015 nas dependências da Escola Politécnica da Universidade de São Paulo (Poli/USP).
- d) É conhecida a efetividade do filtro de Anscombe/Wiener no tratamento de imagens de Medicina Nuclear. Entretanto, no presente trabalho foram testadas outras abordagens de estabilização de Variância, como a transformada de Bartlett e a transformada de Freeman&Turkey. Os testes realizados mostram a superioridade do filtro Anscombe/Wiener, considerando as particularidades do nosso projeto e assumindo que o ruído apresentado nas imagens reconstruídas segue uma distribuição de probabilidade Poisson. No entanto, precisa-se aplicar esta comparação em um número maior de casos para estabelecer estatisticamente essa supremacia.
- e) Comprovou-se que os algoritmos baseados na teoria Fuzzy Connectedness são bastante adequados para a Segmentação de imagens 3D, e facilmente escaláveis para n-dimensões. Além disso, a abordagem com pesos dinâmicos apresentou robustez e bons resultados quando testados nos diferentes volumes simulados. Deve-se enfatizar que o processo de segmentação desempenhou um papel relevante no processo de avaliação funcional da etapa de filtragem 3D. Também foi realizado um *upgrade* no

algoritmo usado com o intuito de segmentar mais de uma ROI em um mesmo processo de segmentação.

- f) Tanto a Curva de Atividade de Entrada no plasma quanto a Curva de Atividade Resultante no miocárdio, foram produzidas utilizando diretamente a informação contida nas imagens PET 3D. Este processo é muito mais propício, não invasivo e confiável, uma vez que toda a informação foi obtida diretamente dos valores dos pixels da série temporal das imagens dinâmicas.
- g) Conseguiu-se testar e incorporar novos modelos na ferramenta COMKAT, através da qual foi conduzido o processo de quantificação do metabolismo nas imagens PET 3D simuladas e reais. Cabe ressaltar que os softwares comerciais utilizados para quantificação do metabolismo (e.g.  $\pi$ .pmod, CSI, Flow-Quant, entre outros) exigem o pagamento de taxas de licenciamento, ao contrário da ferramenta COMKAT que é um software open source.
- h) Em geral, a abordagem apresentada neste projeto pode ser de particular interesse principalmente em dois âmbitos:
  - Do ponto de vista acadêmico, temos implementado uma plataforma integral para a quantificação do metabolismo constituída por um conjunto de processos devidamente inter-relacionados. Acreditamos que esta seja uma ferramenta interessante através da qual se consiga treinar alunos e profissionais que atuem na área de Medicina Nuclear, bem como propor, testar e avaliar novos algoritmos e/ou modelos como alternativas que visem melhorar esse processo.
  - Do ponto de vista clínico e de pesquisa, todas as ferramentas que temos utilizado ao longo do projeto são do tipo open-source e de fácil acesso. Nesse sentido, estão sendo disponibilizados os manuais com as etapas detalhadas de instalação para cada um dos softwares utilizados, assim como todos os detalhes, parâmetros e linhas de comando de cada etapa do projeto. Isso permitirá à comunidade científica e outros usuários que tenham interesse em testar e/ou continuar esta pesquisa, assim como conhecer e usar os parâmetros que temos usado em cada parte do projeto.

## Capítulo 6

# CONCLUSÕES

Temos desenhado uma plataforma integral para análise da dinâmica de transporte e quantificação do metabolismo de imagens 3D de Medicina Nuclear na modalidade PET. Todos os processos que fazem parte desta plataforma estão bem estruturados, documentados e devidamente inter-relacionados.

Finalmente, baseados nos resultados do teste de avaliação da sequência de processamento, realizado nos volumes do coração do MASH, podemos indicar que o filtro Anscombe/Wiener proporcionou os melhores resultados, considerando como critério funcional a qualidade da segmentação. Da mesma forma, ficou evidenciado que, tendo aplicado o processo de filtragem sobre os volumes reconstruídos, os resultados das constantes metabólicas estimadas através da nossa metodologia alcançaram uma proximidade com os valores metabólicos teóricos. Pode-se então concluir que a etapa de filtragem tridimensional utilizando o filtro Anscombe/Wiener é relevante e de alto impacto no processo de quantificação metabólica e do projeto em geral.

# **PUBLICAÇÕES**

## **Publicações em Periódicos**

1. FLOREZ PACHECO, E.; FURUIE, S. S. Processing of three-dimensional structures of Nuclear Medicine in PET modality. Paper published in the Brazilian Journal of Biomedical Engineering, Volume 29 / number 1, 2013.
2. FLOREZ PACHECO, E.; VIJAYAKUMAR V.; FURUIE, S. S. Analysis of the Dynamic and Metabolic Quantification of Nuclear Medicine images in the PET/CT modality. Artigo em processo de elaboração para ser publicado no Brazilian Journal of Biomedical Engineering (data estimada de publicação: setembro, 2016).

## **Publicações em Congressos**

1. FLOREZ PACHECO, E.; FURUIE, S. S. Effects of filtering in the segmentation process of three-dimensional PET Images. XXIII Brazilian Congress of Biomedical Engineering - CBEB 2012, Porto de Galinhas, Pernambuco - Brazil.
2. FLOREZ PACHECO, E.; FURUIE, S. S. Variance stabilizing transformations in the reduction of Poisson noise in 3D Nuclear Medicine images. Computing in Cardiology (CinC) 2014, Cambridge, Massachusetts – USA.
3. FLOREZ PACHECO, E.; FONSECA, H.; VIJAYAKUMAR V.; FURUIE, S. S. Framework to quantify the metabolic rate in the heart using Monte Carlo simulation and Compartmental Modeling. Computing in Cardiology (CinC) 2015, Nice – France.

# REFERÊNCIAS BIBLIOGRÁFICAS

- Allen D. Variance Stabilization Transformations. University of Kentucky. 2013.
- Almeida E. Medicina Nuclear: Importância e Indicações. LINCX Passaporte para medicina de primeiro mundo. Enlace: <http://www.lincx.com.br/index.php>, 2004.
- Arno P.K. Image-derived input functions for determination of MRGlu in cardiac 18F-FDG scans. The Journal of Nuclear Medicine. v.42, n.11, p.1622-1629, 2001.
- Alzimami K.S., Sassi S.A., Spyrou N.M. Advances in Electrical Engineering and Computational Science. Chapter A Comparison Between 3D OSEM and FBP Image Reconstruction Algorithms in SPECT. Volume 39, pp 195-206, 2009.
- Anderson D.H. Compartmental Modeling and Tracer Kinetics. Department of Mathematics, Souther Methodist University, Dallas, TX, USA. Springer-Verlag Berlin Heidelberg, 1983.
- Andrade M.A. Implementação de uma interface gráfica em JAVA para reconstrução tomográfica em Medicina Nuclear. Porto Alegre, 2003. 48 f. Trabalho de Conclusão de Curso (Física) Faculdade de Física, PUCRS.
- Anscombe F. J. The transformation of Poisson, Binomial and Negative Binomial Dat. Biometrika 1948; 15:246-254.
- Bai B., Li Q., Holdsworth C.H., Asma E., Tai Y.C., Chatzizoiannou A., Leahy R.M. Model-based normalization for iterative 3D PET image reconstruction. Phys. Med. Biol. 47 (2002) 2773–2784.
- Bartlett M.S. The square root transformation in analysis of variance. J R Statist Soc Suppl 3(1): 68-78. 1936.
- Bethea R.M.; Duran R.A. Statistical methods for engineers and scientists. Mc Hill Reckker, New York, NY, 1995.
- Blankenberg F.G.; Strauss H.W. Nuclear medicine applications in molecular imaging. Journal of Magnetic Resonance Imaging, Volume 16, Issue 4, October 2002, Pgs 352 – 361.
- Borges F.D.A. Segmentação, Reconstrução e Quantificação 3D de Estruturas em Imagens Médicas – Aplicação em Imagem Funcional e Metabólica. Tese de Mestrado em Engenharia Biomédica. Faculdade de Engenharia da Universidade de Porto. 2010.
- Bruce, R. Hybrid Systems, PET dominates nuclear medicine. Diagnostic Imaging On-Line, November 22, 2000.
- Burckhardt D. Cardiac Positron Emission Tomography: Overview of Myocardial Perfusion, Myocardial Blood Flow and Coronary Flow Reserve Imaging. SIEMENS Medical Solutions USA. Molecular Imaging 2009.

- Carrió I., Gonzáles P., Estorch M., Canessa J., Mitjavila M., Massardo T. Medicina Nuclear: Aplicaciones clínicas. Editorial Masson S.A., España, 2003.
- Carson E., Cobelli C. Modelling Methodology for Physiology and Medicine. Academic Press Series in Biomedical Engineering 2001, Chapter 1: 1-13, Chapter 7: 179-208.
- Carson RE. Tracer Kinetic Modeling in PET. Positron Emission Tomography - Basic sciences. London: Springer; 2005. p. 127-159.
- Carvalho B.M., Gau C.J., Herman G.T., Kong T.Y. Algorithms for fuzzy segmentation. Pattern Analysis Application, 2(1):73–81, April 1999.
- De Almeida J. P., Trindade M. V., Gomes D., Fernandes C.D., de Moraes W. O que é Medicina Nuclear? Portal dos profissionais de Radiologia. <http://www.rxinfo.com.br>, 2002.
- De Amorin D.P.M.L. Modelos Fenomenológicos de Distribuição Intercompartimental de Substâncias Radioactivas. Faculdade de Engenharia da Universidade de Porto. 2007.
- Fabey, F.H. State of the art in emission tomography equipment. Radiographics, v. 16, n. 2, p. 409-20, 1996.
- Falcão, A. X.; Stolfi, J.; Lotufo, R. A. The image foresting transform: theory, algorithms, and applications. Pattern Analysis and Machine Intelligence, IEEE Transactions on Volume 26, Issue 1, Page(s):19 – 29, 2004.
- Feng H.W. Models for Computer simulation Studies of Input Functions for Tracer Kinetic Modeling with Positron Emission Tomography. International Journal of Biomedical Computing, 32(2):95-110, 1993.
- Ferreira N.C. “Contribuição para a quantificação em tomografia por emissão de positrões no modo 3D”, tese de doutoramento, Universidade de Coimbra, 2001.
- Flórez Pacheco E. Quantificação da Dinâmica de Estruturas em Imagens de Medicina Nuclear na modalidade PET [dissertação]. São Paulo: Escola Politécnica da Universidade de São Paulo; 2012. 200p.
- Freeman M.F., Tukey J.W. Transformations Related to the Angular and the Square Root. Institute of Mathematical Statistics – Princeton University. 1972; Volume 21, Number 4 (1950), 607-611.
- Gispert J.D., Reig S., Pascau J., Molina V., Santos A., Desco M. Técnicas de cuantificación de imágenes PET (Tomografía por Emisión de Positrones): Aplicación al estudio de la esquizofrenia, 2010.
- Gonzales R. C., Woods R. E. Digital Image Processing. 3rd ed, New Jersey, Pearson Prentice Hall, 2008. Chap.9 – Morphological Image Processing.

Hicks R.J., Herman W.H., Kalff V., Molina E., Wolfe E.R., Hutchins G. Quantitative evolution of regional substrate metabolism in the human heart by positron emission tomography. *J am Coll Cardiol*, 18:101-11, 1991.

Hoffman E.J; Cutler P.D.; Digby W.M.; Mazziotta J.C. 3D Phantom to simulate cerebral blood flow and metabolic images for PET. *IEEE Transactions on Nuclear Science*, V.2, 1990.

Homen M. R. P., Mascarenhas N. D. A., Costa L. F., Preza C. Biological Image Restoration in Optical – Sectioning Microscopy Using Prototype Image Constraints. *Real-Time Imaging* 8, 475-490. 2002.

Homen M. R. P., Mascarenhas N. D. A., Costa L. F. Linear Filters for Deconvolution Microscopy. 6-th IEEE Southwest Symposium on Image Analysis and Interpretation Proceedings, Lake Tahoe.v.1 p. 142-146. 2004.

Huang S.C., Phelps M.E., Hoffman E.J., Sideris K., Selin C.J., Kuhl D. Noninvasive determination of local cerebral metabolic rate of glucose in man. Division of Nuclear Medicine, School of Medicine; and The Laboratory of Nuclear Medicine and Radiation Biology, University of California at Los Angeles, Los Angeles, California. 1980.

Huang S.C., Phelps M.E., Hoffman E.J., Sideris K., Selin C.J., Kuhl D.E. Noninvasive determination of local cerebral metabolic rate of glucose in man. *Am.J.Physiol.* 1980, 238:E69-E82.

Inouye T. Square Root Transform for the Analysis of Quantum Fluctuations in Spectrum Data. *Nuclear Instruments and Methods, Nuclear Instruments and Methods*, Volume 91, Issue 4, p. 581-584, 1971.

Jadvar H., Parker J.A. *Clinical PET and PET/CT: Principles and Applications*. Chapter 2. Springer, 2nd edition, 279p. 2010.

Jan S., Santin G., Strul D., Staelens S., Assié K., Autret D., Avner S., Barbier R., Bardiès M., Bloomfield P. M., Brasse D., Breton V., Bruyndonckx P., Buvat I., Chatziioannou A. F., Choi Y., Chung Y. H., Comtat C., Donnarieix D., Ferrer L., Glick S. J., Groiselle C. J., Guez D., Honore P. F., Kerhoas-Cavata S., Kirov A. S., Kohli V., Koole M., Krieguer M., van der Laan D. J., Lamare F., Largeron G., Lartizien C., Lazaro D., Maigne L., Mayet F., Melot F., Merheb C., Pennacchio E., Perez J., Pietrzyk U., Rannou F. R., Rey M., Schaart D. R., Schmidtlein C. R., Simon L., Song T. Y., Vieira JM., Visvikis D., Van de Walle R., Wieërs E. and Morel C. GATE: a simulation toolkit for PET and SPECT, 2004, *Phys. Med. Biol.* 49 4543.

Kamasak M.E.; Bouman C.A.; Morris E.D.; Sauer K.D. Parametric reconstruction of kinetic PET data with plasma function estimation. *Proc. SPIE 5674, Computational Imaging III*, 293, Vol. 5674, 2005.

Kass M., Witkin A., Terzopoulos D., "Snakes: Active Contour Models," *International Journal of Computer Vision*, pp. 321-331, 1988.

- Knapp Jr., FF.; Mirzadeh, S. The continuing important role of radionuclide generator systems for nuclear medicine. *European Journal of Nuclear Medicine*, v. 20, n. 12, p. 1151-1165, 1994.
- Kramer R, Cassola V F, Khoury H J, Vieira J W, de Melo Lima V J, Brown K R. 2010 FASH and MASH: female and male adult human phantoms based on polygon mesh surfaces: II. Dosimetric Calculations. *Phys. Med. Biol.* 55 163-169.
- King M.A., Doherty P.W., Schwinger R.B. A Wiener filter for Nuclear Medicine images. *Med Phys.* 1983; 10(6):876-80.
- Lage D. M. Visualização de artérias coronárias epicárdicas em imagens ecocardiográficas tridimensionais com contraste de microbolhas. Tese de doutorado, Faculdade de Medicina da Universidade de São Paulo. Programa de Cardiologia. 2010.
- Lee J. S. Digital image enhancement and noise filtering by use of local statistics. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-2, pp. 165-168, 1980. (Adaptive Wiener filter)
- López J. Técnicas avanzadas de reconstrucción de imagen nuclear PET, X-CT y SPECT. Trabajo de Maestría en Física Biomédica. Dpto. de Física Atómica, Molecular y Nuclear Facultad de Ciencias Físicas de la Universidad Complutense de Madrid. Madrid. 2008.
- Ljungberg M., Strand S.E., King M.A. Monte Carlo calculations in Nuclear Medicine: Applications in Diagnostic Imaging. Second Edition. Series in medical physics and biomedical engineering. 2013.
- Maisey M.N. Positron Emission Tomography in Clinical Medicine. In: BAILEY, D. (editor). *Positron Emission Tomography - Basic sciences*. London: Springer, 2005. p. 1-12.
- Mascarenhas N.D.A., Santos C.A.N., Cruvinel P.E. Transmission Tomography under Poisson noise using the Anscombe Transform and a Wiener Filter of the projections. *Nuclear Instruments and Methods in Physics Research A*. v. 423, p. 265-271. 1999.
- Matej S., Surti S., Jayanthi S., Daube-Witherspoon M.E., Lewitt R.M., Karp J.S. Efficient 3D TOF PET Reconstruction Using View-Grouped Histo-Images: DIRECT – Direct Image Reconstruction for TOF. *IEEE Transactions on Medical Imaging*, 2008.
- Matis J.H., Wehrly T.E., Gerald K.B. *Tracer Kinetics and Physiologic Modeling: Theory to practice*. Texas A&M Univeristy, College Station, TX. Springer-Verlag Berlin Heidelberg, 1983.
- Mettler F.A., Huda W., Yoshizumi T.T., Mahesh M. Effective doses in radiology and diagnostic nuclear medicine: a catalog. *Radiology*. 2008;248:254–63.
- McQuaid S, Hutton B. Sources of attenuation-correction artefacts in cardiac PET/CT and SPECT/CT. *Eur J Nucl Med Mol Imaging* 2008;35:1117-23.

Morita K., Katoh C., Yoshinaga K., Noriyasu K., Mabuchi M., Tsukamoto T., Kageyama H., Shiga T., Kuge Y., Tamaki N. Department of Nuclear Medicine, Hokkaido University Graduate School of Medicine. Quantitative analysis of myocardial glucose utilization in patients with left ventricular dysfunction by means of  $^{18}\text{F}$  – FDG dynamic positron tomography and three-compartment analysis. *European Journal of Nuclear Medicine and Molecular Imaging*, vol. 32, N° 7, 2005.

Muzic R., Cornelius S. COMKAT: Compartment Model Kinetic Analysis Tool. *The Journal of Nuclear Medicine*, vol 42, N° 4, 2001.

Noori-Asl M., Sadremomtaz A. Analytical image reconstruction methods in emission tomography. *J. Biomedical Science and Engineering*, 2013, 6, 100-107.

Nyúl, A. L. G; Falcão, A. X; Udupa, J. K. Fuzzy-connected 3D image segmentation at interactive speeds. *Elsevier on Graphical Models*, v.64, p. 259-281, 2002.

Oliveira R., Santos D., Ferreira D., Coelho P., Veiga F. Preparações radiofarmacêuticas e suas aplicações. *Brazilian Journal of Pharmaceutical Sciences*, vol. 2, n. 2, 2006.

Osher, S; Fedkiw, R. *Level Set Methods and Dynamic Implicit Surfaces*. Springer-Verlag, New York, 2002.

Otsu N. Threshold Selection Method from Gray-Level Histograms. *IEEE Transactions on Systems, Man and Cybernet*, 9, N° 1, pp. 62-66. 1979.

Pednekar, A. S.; Kakadiaris, I. A. Image Segmentation Based on Fuzzy Connectedness using Dynamic Weights. Department of Computer Science University of Houston Houston, TX, 77204, USA. Technical Report Number UH-CS-05-13, 2006.

Phelps M.E. *PET: Molecular Imaging and Its Biological*. Springer-Verlag New York, 2004.

Pozzo L. Tomografia por emissão de pósitrons com sistemas PET/SPECT: Um estudo da viabilidade de Quantificação. Tese de Doutorado do Instituto de Física da Universidade de São Paulo. 2005.

Ragheb M. *Nuclear Medicine Applications*, 2007.

Reivich M., Alavi, A., Wolf, A. et al., Glucose metabolic rate kinetic model parameter determination in humans: the lumped constant and rate constants for  $^{18}\text{F}$ fluorodeoxyglucose and  $^{11}\text{C}$ deoxyglucose. *J Cereb Blood Flow Metabol* 5: 179-192 1985.

Reivich M., Kuhl D., Wolf A. Measurement of local cerebral glucose metabolism in man with  $^{18}\text{F}$ -2- fluoro-2-deoxy-d-glucose. *Acta Neurol.Scand.Suppl.* 1977; 64:190-1:190-191.

Robilotta C.C. A tomografia por emissão de pósitrons: uma nova modalidade na medicina nuclear brasileira. *Rev Panam Salud Publica*. 2006; 20(2/3):134–42.

Rodrigues I., Sanches J., Bioucas-Dias J..Denoising of medical images Corrupted by poisson noise. Image Processing, 2008. ICIP 2008. 15th IEEE International Conference, páginas 1756–1759, 2008.

Romualdo L.C.S. Restauração de imagens mamográficas digitais utilizando o filtro de Wiener no domínio de Anscombe e o filtro inverso da MTF no domínio da frequência. Tese de Mestrado, Escola de Engenharia de São Carlos da Universidade de São Paulo, 2009.

Sabbatini R. The PET Scan: A New Window Into the Brain. Brain & Mind, Electronic Magazine on Neuroscience, 1997.

Sarmiento V.A.; Carvalho A.C.R; Luz I.M. Evaluation of photon noise on digitalized images.Revista da Faculdade de Odontologia, Porto Alegre, vol. 45, N°1, pg. 23-28, 2004.

Sergars W.P.; Tsui B.M.W. MCAT TO XCAT: The Evolution of 4-D Computerized Phantoms for Imaging Research. IEEE, Vol. 97, No. 12, 2009.

Shaw I. S., Simões M. G. Controle e Modelagem Fuzzy. Edgar Blücher e FAPESP. 1999.

Shepp L.A.; Vardi Y. “Maximum likelihood reconstruction for emission tomography”, IEEE Trans. Med. Imag., vol. MI-2, pp. 113–122, 1982.

Silva J.E.M.M.. Estimação de parâmetros de modelos compartimentais para tomografia por emissão de pósitrons. [dissertação]. São Paulo: Escola Politécnica da Universidade de São Paulo; 2010. 146p.

Silva N. A. I. R. On the use of image derived input function for quantitative PET imaging with a simultaneous measuring MR-BrainPET. Tese de mestrado integrado em Engenharia Biomédica e Biofísica, apresentada à Universidade de Lisboa, através da Faculdade de Ciências, 2012.

Simoncic U., Jeraj R. Cumulative input function method for linear compartmental models and spectral analysis in PET. Journal of Cerebral Blood Flow & Metabolism, n. 31, p. 750-56, 2011.

Sokoloff L. [1-14C]-2-deoxy-d-glucose method for measuring local cerebral glucose utilization. Mathematical analysis and determination of the "lumped" constants. Neurosci.Res.Program.Bull. 1976; 14:466-468.

Sokoloff L., Reivich M., Kennedy C. The [14C] deoxyglucose method for the measurement of local cerebral glucose utilization: theory, procedure, and normal values in the conscious and anesthetized albino rat. J.Neurochem. 1977; 28:897-916.

Souza S.; De Oliveira A. Alguns comentários sobre a Teoria Fuzzy. Exacta versão1. Matematica Aplicada IME – USP. 1992. p. 139 – 147.

Storne R.T. Cintilografia e PET/CT nas Neoplasias Malignas. Monografia apresentada ao Programa de Pós-graduação em Radiologia Odontológica e Imaginologia da Universidade Federal do Rio Grande do Sul como requisito à obtenção do título de Especialista. Porto Alegre, 2011.

Vieira MAC, Bakic PR, Maidment ADA, Mascarenhas NDA. Investigating Poisson Noise Filtering in Digital Breast Tomosynthesis. University of Kentucky. 2013.

Tarantola G., Zito-Gerundini P. “PET instrumentation and reconstruction algorithms in whole-body applications”, J. Nucl. Med, vol. 44, no 5, pp. 756–769, 2003.

Tinois E., Imagem Funcional – PET e fMRI. MultiCiência: Tecnologia para a Saúde #5, 2005.

Udupa, J. K.; Saha, P. K. Fuzzy Connectedness and Image Segmentation. IEEE, Proceedings of IEEE, Vol. 91, pp. 1649-1669, October 2003.

Udupa KJ, LaBlanc RV, Schmidt H. A Methodology for Evaluating Image Segmentation Algorithms 2002.

Wagner Jr. H.N. A brief history of positron emission tomography (PET). Seminars in Nuclear Medicine, v. 28, n°3, p. 213-20, 1998.

Wolbarst A.B. Physics of Radiology. Appleton & Lange, Norwalk, USA, 1993.

Xu, J. P. Generalized Gradient Vector Flow External Forces for Active Contours. Signal Processing 71, pp 131-139, 1998.

Zaidi H. Quantitative Analysis in Nuclear Medicine Imaging. Division of Nuclear Medicine, Geneva University Hospital Geneva. Springer Science & Business Media, Inc. 2006.

# APÊNDICE A

## MANUAL DE INSTALAÇÃO: GATE v6.2

A seguir indicamos os passos que são necessários para instalar e rodar o ambiente de simulação GATE na versão 6.2 (última versão), usando OS Linux Fedora v.15 como Sistema Operativo principal:

1. O primeiro passo é fazer algumas atualizações no nosso sistema e instalar alguns pacotes que serão necessários no processo de instalação dos softwares que conformam a plataforma de simulação GATE. Abrimos o terminal e executamos as seguintes linhas de comando. É importante ser Super Usuário (su) para completar este passo:

```
[eflorez@localhost ~]$ su
```

Senha:

```
[root@localhost eflorz]# sudo yum install freeglut freeglut-devel gtkglext-devel gtkglext-libs gcc  
gcc-gfortran gcc-c++ compat-libgfortran-41 libgfortran glibc-kernheaders glibc-headers glibc-devel  
glibc glibc-static openmotif openmotif-devel libXaw-devel libXaw libXpm-devel libXpm libxml2-devel  
libxml2 xerces-c-devel qt qt-devel qt-x11 binutils libX11-devel libXft-devel libXext-devel ncurses-  
devel pcre-devel mesa-libGL-devel mesa-libGL gtkglarea2 gtkglarea2-devel InventorXt InventorXt-  
devel lesstif lesstif-devel libfftw3-3 libfftw3-dev libfftw3-doc
```

Possivelmente os seguintes pacotes não serão instalados:

```
openmotif openmotif-devel libfftw3-3 libfftw3-dev libfftw3-doc
```

Os dois primeiros (**openmotif** e **openmotif-devel**) não existem dentro dos pacotes de programas disponíveis para ser instalados no Fedora 15, mas podem ser descarregados da rede e posteriormente instalados. Os links usados para esse fim são os seguintes:

Para o **openmotif**:

[http://rpm.pbone.net/index.php3/stat/4/idpl/20844304/dir/fedora\\_15/com/openmotif-2.3.3-1.fc14.i686.rpm.html](http://rpm.pbone.net/index.php3/stat/4/idpl/20844304/dir/fedora_15/com/openmotif-2.3.3-1.fc14.i686.rpm.html)

Para o **openmotif-devel**:

[http://rpm.pbone.net/index.php3/stat/4/idpl/20844519/dir/fedora\\_15/com/openmotif-devel-2.3.3-1.fc14.i686.rpm.html](http://rpm.pbone.net/index.php3/stat/4/idpl/20844519/dir/fedora_15/com/openmotif-devel-2.3.3-1.fc14.i686.rpm.html)

**Nota:** Ter em consideração o tipo de Linux que está se utilizando (32 ou 64 bits), os links acima indicados instalarão pacotes para Linux de 32 bits. Caso se tenha instalado Linux de 64 bits, deverá se procurar pacotes com a extensão **x86\_64**.

Já para os últimos três pacotes (**libfftw3-3**, **libfftw3-dev** e **libfftw3-doc**), podemos utilizar a lista de programas que é disponibilizada pelo Fedora, da seguinte maneira:

- Nos aplicativos, tem um ícone **Adicionar/remover programas**.

- Digitamos no buscador a palavra 'fftw'
- Seleccionamos os seguintes pacotes: **fftw-3.3-3.fc15 (i686 ou x86\_64)**, **fftw-devel-3.3-3.fc15 (i686 ou x86\_64)**, **fftw-doc-3.3-3.fc15 (i686 ou x86\_64)**.
- Fazemos clic em Aplicar para fazer as modificações indicadas, vai ser solicitada a senha de administrador para concluir as instalações.

2. O passo seguinte é descarregar e colocar todos os arquivos dos programas requeridos em uma pasta específica dentro do computador. Os seis (06) programas e as versões requeridas são: **CLHEP\_v2.1.1.0**, **Geant4\_9.5.p02**, **Gate\_v6.2**, **ROOT\_v5.34**, **ECAT7** e **LMF\_v3.0**. Todos estes pacotes a serem instalados podem ser descarregados da página ([www.opengatecollaboration.org/](http://www.opengatecollaboration.org/)) fazendo uso dos seguintes dados de acesso:

Username : **edward florez**  
 Password : **\*\*\*\*\***

No nosso sistema temos um mapa (organização) do tipo:

```
[eflorez@localhost ~]$ pwd
/home/eflorez
[eflorez@localhost ~]$ cd Gate6.2/
[eflorez@localhost Gate6.2]$ ls
CLHEP_2.1.1.0  ecat7  gate_v6.2  geant4.9.5.p02  lmf_v3.0  root_v5.34
```

3. Instalação do pacote **CLHEP**:

```
[eflorez@localhost Gate]$ cd CLHEP_2.1.1.0/CLHEP
[eflorez@localhost CLHEP]$ pwd
/home/eflorez/Gate/CLHEP_2.1.1.0/CLHEP
[eflorez@localhost CLHEP]$ ./configure --prefix=$PWD
[eflorez@localhost CLHEP]$ make -j2
[eflorez@localhost CLHEP]$ make install
[eflorez@localhost CLHEP]$
export PATH=$PATH:/home/eflorez/Gate6.2/CLHEP_2.1.1.0/CLHEP/bin
[eflorez@localhost CLHEP]$
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/home/eflorez/Gate6.2/CLHEP_2.1.1.0/CLHEP/lib
```

4. Instalação do pacote **ROOT**:

```
[eflorez@localhost root_v5.34]$ ./configure
[eflorez@localhost root_v5.34]$ make -j2
[eflorez@localhost root_v5.34]$ export ROOTSYS=$PWD
[eflorez@localhost root_v5.34]$ make install
[eflorez@localhost root_v5.34]$ cd bin
[eflorez@localhost bin]$ pwd
/home/eflorez/Gate/root_v5.34/bin
```

```
[eflorez@localhost bin]$ source thisroot.sh
```

## 5. Instalação do pacote **ECAT**:

```
[eflorez@localhost ecat7]$ cp Makefile.unix Makefile  
[eflorez@localhost ecat7]$ make
```

Criar a pasta 'include' e posteriormente copiar todos os arquivos \*.h nessa pasta:

```
[eflorez@localhost ecat7]$ mkdir include  
[eflorez@localhost ecat7]$ pwd  
/home/eflorez/Gate/ecat7  
[eflorez@localhost ecat7]$ cp *.h /home/eflorez/Gate6.2/ecat7/include
```

Verificar se o arquivo '**libecat.a**' encontra-se na pasta lib/Linux-g++, caso contrario fazer:

```
[eflorez@localhost ecat7]$ mkdir lib  
[eflorez@localhost ecat7]$ mkdir lib/Linux-g++  
[eflorez@localhost ecat7]$ cp libecat.a lib/Linux-g++/
```

## 6. Instalação do pacote **LMF**:

```
[eflorez@localhost Gate]$ cd lmf_v3.0/  
[eflorez@localhost lmf_v3.0]$ ls  
configure examples lib makefile README TAGS  
documentation includes license obj src testForFormat.c
```

No arquivo '**makefile**' modificar as seguintes linhas e gravar:

```
INCLUDE = -I./includes/ -I/home/eflorez/Gate6.2/lmf_v3.0/includes  
PATHLMFCOMMON = /home/eflorez/Gate6.2/lmf_v3.0  
[eflorez@localhost lmf_v3.0]$ ./configure  
[eflorez@localhost lmf_v3.0]$ make clean  
[eflorez@localhost lmf_v3.0]$ make  
[eflorez@localhost lmf_v3.0]$ mkdir lib/Linux-g++  
[eflorez@localhost lmf_v3.0]$ cp lib/libLMF.a lib/Linux-g++/
```

## 7. Instalação do pacote **Geant4**:

É necessário instalar os pacotes '**cmake**' e '**cmake-gui**' da seguinte forma:

```
[eflorez@localhost ~]$ su  
Senha:  
[root@localhost eflorez]# sudo yum install cmake  
[root@localhost eflorez]# sudo yum install cmake-gui  
[root@localhost eflorez]# exit
```

Dentro da pasta principal '**geant4.9.5.p02**' é necessário criar duas pastas '**geant4.9.5.p02-build**' e '**geant4.9.5.p02-install**' da seguinte maneira:

```
[eflorez@localhost ~]# cd Gate6.2/geant4.9.5.p02
[eflorez@localhost geant4.9.5.p02]# mkdir geant4.9.5.p02-build
[eflorez@localhost geant4.9.5.p02]# mkdir geant4.9.5.p02-install
```

Na sequência, entramos na pasta '**geant4.9.5.p02-build**' para gerar as configurações de instalação:

```
[eflorez@localhost geant4.9.5.p02]$ cd geant4.9.5.p02-build
[eflorez@localhost geant4.9.5.p02-build]$ cmake ../
```

Nesse momento vai aparecer uma janela onde precisaremos configurar algumas opções, pressionando a tecla '**c**' poderemos fazer a configuração tal como segue:



Caso não seja possível gerar a compilação, aparecerão umas linhas adicionais no começo da janela, solicitando os path das pastas "include" e "lib" do CLHEP, nesse caso é só preencher essas duas linhas da seguinte forma:

```
CLHEP_INCLUDE_DIR  /home/eflorez/Gate6.2/CLHEP_2.1.1.0/CLHEP/include
CLHEP_LIBRARY      /home/eflorez/Gate6.2/CLHEP_2.1.1.0/CLHEP/lib
```

Uma vez finalizada as modificações, pressionamos a tecla '**g**' para gerar a compilação de todos os arquivos do pacote geant4. (**NOTA:** Se a opção de geração não aparecer, aperte uma ou duas vezes a tecla '**c**' até a opção de geração aparecer).

Uma vez finalizada a compilação, voltará automaticamente para o terminal onde precisamos colocar o seguinte:

```
[eflorez@localhost geant4.9.5.p02-build]$ make -j2
[eflorez@localhost geant4.9.5.p02-build]$ make install
```

**NOTA:** Não instalar a versão 'geant4.9.5.p01' do Geant4 porque tem várias falhas de compilação.

## 8. Instalação do pacote **GATE**:

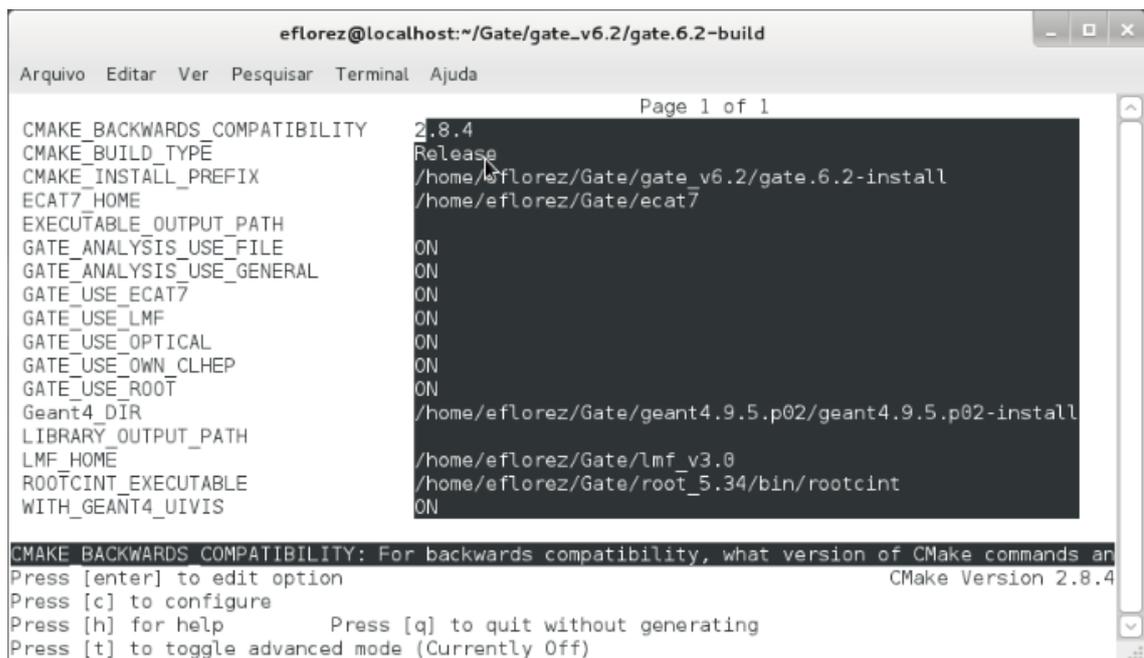
Dentro da pasta principal 'gate\_v6.2' é necessário criar duas pastas 'gate.6.2-build' e 'gate.6.2-install' da seguinte maneira:

```
[eflorez@localhost ~]# cd Gate/gate_v6.2
[eflorez@localhost gate_v6.2]# mkdir gate.6.2-build
[eflorez@localhost gate_v6.2]# mkdir gate.6.2-install
```

Logo após, nos posicionamos na pasta 'gate.6.2-build' para gerar as configurações de instalação através do cmake :

```
[eflorez@localhost gate_v6.2]$ cd gate.6.2-build
[eflorez@localhost gate.6.2-build]$ cmake ..
```

Nesse momento vai aparecer uma janela onde precisaremos configurar algumas opções, pressionando a tecla 'c' poderemos fazer a configuração tal como segue:



The screenshot shows a terminal window titled 'eflorez@localhost:~/Gate/gate\_v6.2/gate.6.2-build'. The window displays the output of the 'cmake ..' command, showing a list of configuration options and their values. The options are:

CMAKE_BACKWARDS_COMPATIBILITY	2.8.4
CMAKE_BUILD_TYPE	Release
CMAKE_INSTALL_PREFIX	/home/eflorez/Gate/gate_v6.2/gate.6.2-install
ECAT7_HOME	/home/eflorez/Gate/ecat7
EXECUTABLE_OUTPUT_PATH	
GATE_ANALYSIS_USE_FILE	ON
GATE_ANALYSIS_USE_GENERAL	ON
GATE_USE_ECAT7	ON
GATE_USE_LMF	ON
GATE_USE_OPTICAL	ON
GATE_USE_OWEN_CLHEP	ON
GATE_USE_ROOT	ON
Geant4_DIR	/home/eflorez/Gate/geant4.9.5.p02/geant4.9.5.p02-install
LIBRARY_OUTPUT_PATH	
LMF_HOME	/home/eflorez/Gate/lmf_v3.0
ROOTCINT_EXECUTABLE	/home/eflorez/Gate/root_5.34/bin/rootcint
WITH_GEANT4_UIVIS	ON

At the bottom of the window, there is a message: 'CMAKE BACKWARDS COMPATIBILITY: For backwards compatibility, what version of CMake commands are you using? (Default is 2.8.4)'. Below this message, there are instructions: 'Press [enter] to edit option', 'Press [c] to configure', 'Press [h] for help', 'Press [q] to quit without generating', and 'Press [t] to toggle advanced mode (Currently Off)'. The window also shows 'Page 1 of 1' and 'CMake Version 2.8.4'.

Uma vez finalizada as modificações, pressionamos a tecla 'g' para gerar a compilação de todos os arquivos do pacote gate\_v6.2. (**NOTA:** Se a opção de geração não aparecer, aperte uma ou duas vezes a tecla 'c' até a opção de geração aparecer).

Uma vez finalizada a compilação, voltará automaticamente para o terminal onde precisamos colocar o seguinte:

```
[eflorez@localhost gate.6.2-build]$ make -j2
```

No meu caso, após executar a linha de comando anterior tive as seguintes linhas de erro devido a não existência da livreria **'-lecat'**:

```
[100%] Building CXX object CMakeFiles/Gate.dir/source/general/src/GateMiscFunctions.cc.o
[100%] Building CXX object
CMakeFiles/Gate.dir/source/general/src/GateApplicationMgrMessenger.cc.o
[100%] Building CXX object
CMakeFiles/Gate.dir/source/general/src/GateObjectChildListMessenger.cc.o
Linking CXX executable Gate
/usr/bin/ld: cannot find -lecat
collect2: ld returned 1 exit status
make[2]: ** [Gate] Erro 1
make[1]: ** [CMakeFiles/Gate.dir/all] Erro 2
make: ** [all] Erro 2
```

Para resolver esses erros, foi necessário instalar o pacote [libecat7-1.4-1.i386.rpm](http://sourceforge.net/projects/amide/files/libecat/1.5/) da página <http://sourceforge.net/projects/amide/files/libecat/1.5/> (é bom aclarar que para Linux de 64 bits, tem que ser instalado o pacote **libecat7-1.5-1.x86\_64.rpm**). Depois de isso executar novamente o comando 'make -j2' e vamos poder compilar finalmente sem nenhum tipo de erro.

Logo após, consegui prosseguir com a instalação executando:

```
[eflorez@localhost gate.6.2-build]$ make install
```

Uma vez finalizada a instalação, voltamos para a raiz onde devemos incorporar no arquivo **'bashrc'** as seguintes linhas de comandos (endereços e fontes):

```
[eflorez@localhost gate.6.2-build]$ cd
[eflorez@localhost ~]# ls
```

Caso não esteja visível o arquivo **'bashrc'** dentro da nossa pasta raiz que nesse caso é /home/eflorez/, basta fazer **Ctrl+H** para visualizar os arquivos ocultos na pasta indicada. Abrimos o arquivo **'bashrc'** e incorporamos as seguintes linhas destacadas:

```
# .bashrc
source /home/eflorez/Gate/root_5.34/bin/thisroot.sh
export PATH=$PATH:/home/eflorez/Gate/CLHEP_2.1.1.0/CLHEP/bin
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/home/eflorez/Gate/CLHEP_2.1.1.0/CLHEP/lib
source /home/eflorez/Gate/geant4.9.5.p02/geant4.9.5.p02-install/bin/geant4.sh
export PATH=$PATH:/home/eflorez/Gate/gate_v6.2/gate.6.2-install/bin

# Source global definitions
if [ -f /etc/bashrc ]; then
```

```
./etc/bashrc
fi
# User specific aliases and functions
```

Uma vez salvo as modificações no arquivo '**.bashrc**', só falta mais um passo para poder abrir o entorno de simulação GATE. Fazemos:

```
[eflorez@localhost ~]# source /home/eflorez/.bashrc
[eflorez@localhost ~]# Gate
```

E finalmente obtemos o Gate pronto para ser usado, com um cabeçalho indicando alguns dados das versões que estão sendo utilizadas. Aqui o cabeçalho do Gate versão 6.2:

```
[G4]
[G4] *****
[G4] Geant4 version Name: geant4-09-05-patch-02 (22-October-2012)
[G4] Copyright : Geant4 Collaboration
[G4] Reference : NIM A 506 (2003), 250-303
[G4] WWW : http://cern.ch/geant4
[G4] *****
[Core-0] Initialization of geometry
[Core-0] Initialization of physics
[Core-0] Initialization of actors
[Core-0]
[Core-0] *****
[Core-0] GATE version name: gate_v6.2
[Core-0] Copyright : OpenGATE Collaboration
[Core-0] Reference : Phys. Med. Biol. 49 (2004) 4543-4561
[Core-0] Reference : Phys. Med. Biol. 56 (2011) 881-901
[Core-0] WWW : http://www.opengatecollaboration.org
[Core-0] *****
[Core-0]
Idle>
```

# APÊNDICE B

## MANUAL DE INSTALAÇÃO: STIR v2.4

A seguir indicamos os passos que são necessários para instalar e rodar o ambiente de reconstrução STIR na versão 2.4 (última versão), usando OS Linux Fedora v.15 como sistema operativo principal:

Para descarregar tanto os arquivos de instalação, quanto a documentação necessária do STIR é necessário ir à página <http://stir.sourceforge.net/> e na seção *Registered Users* temos que colocar os seguintes dados de acesso:

Username : **registered**

Password : **\*\*\*\*\***

Com isso, teremos acesso a fazer download dos arquivos para fazer a instalação.

Assim como foi utilizado a plataforma CMake na instalação do Geant4 e do Gate, para fazer a instalação do STIR também utilizaremos a mesma ferramenta. No nosso caso temos instalado a **v2.8.4** do CMake.

A instalação está composta por três passos, os quais serão apresentados na sequência com os detalhes necessários para que qualquer pessoa possa completar a instalação do STIR:

### I. Primeiro passo: Instalação de pacotes e softwares externos

Antes de começarmos temos que instalar os seguintes pacotes, através do terminal e utilizando a senha de super usuário (su):

```
[eflorez@Adm-LEB-06 ~]$ su
Senha:
[root@Adm-LEB-06 efllorez]# sudo yum install gcc gcc-c++ make cmake cmake-gui ncurses-devel
libX11-devel boost boost-devel tcsh python
```

Na sequência, se não for possível instalar através do terminal tanto o boost/ quando o python/ será necessário instalar eles por meio dos seguintes links:

- **Boost** (<http://www.boost.org>) que é uma biblioteca do C++ e que constitui a base para muitos novos recursos do C++.

- **Python** (<http://www.python.org>). STIR só usa esse softwares para algum SimSETscripts.

- **Ecat**. Neste caso seguiremos os seguintes passos:

```
[eflorez@Adm-LEB-06 ecat]$ cp Makefile.unix Makefile
```

```

[eflorez@Adm-LEB-06 ecat]$ make
[eflorez@Adm-LEB-06 ecat]$ mkdir include
[eflorez@Adm-LEB-06 ecat]$ pwd
/home/eflorez/STIR/lln/ecat
[eflorez@Adm-LEB-06 ecat]$ cp *.h /home/eflorez/STIR/lln/ecat/include
[eflorez@Adm-LEB-06 ecat]$ mkdir lib
[eflorez@Adm-LEB-06 ecat]$ cp libecat.a lib/

```

## II. Segundo passo: Configuração do STIR

Para fazer a configuração do STIR, é necessário criar a pasta **Release** onde ficaram armazenados todos os arquivos que fazem parte da configuração do ambiente de reconstrução. Executamos os seguintes passos:

```

[eflorez@Adm-LEB-06 STIR_2.4]$ mkdir STIR-bin
[eflorez@Adm-LEB-06 STIR_2.4]$ cd STIR-bin/
[eflorez@Adm-LEB-06 STIR-bin]$ mkdir Release
[eflorez@Adm-LEB-06 STIR-bin]$ cd Release
[eflorez@Adm-LEB-06 Release]$ ccmake ../../

```

Preenchemos a tabela com os seguintes dados:

```

eflorez@Adm-LEB-06:~/STIR/STIR_2.4/STIR-bin/Release
Arquivo  Editar  Ver  Pesquisar  Terminal  Ajuda
Page 1 of 2
AVW_ROOT_DIR
BOOST_ROOT
BUILD_SHARED_LIBS      OFF
BUILD_SWIG_OCTAVE     OFF
BUILD_SWIG_PYTHON     OFF
CMAKE_BUILD_TYPE      Release
CMAKE_INSTALL_PREFIX  /home/eflorez/STIR/STIR_2.4
CURSES_CURSES_H_PATH  /usr/include
CURSES_FORM_LIBRARY   /usr/lib/libform.so
CURSES_HAVE_CURSES_H  /usr/include/curses.h
DISABLE_AWV            OFF
DISABLE_LLN_MATRIX    OFF
DISABLE_RDF            OFF
DISABLE_STIR_LOCAL    OFF
GRAPHICS               X
LLN_INCLUDE_DIRS       /home/eflorez/STIR/lln/ecat
LLN_LIBRARIES          /home/eflorez/STIR/lln/ecat/lib/libecat.a
AVW_ROOT_DIR: root of AVW, normally found from the AVW environment variable
Press [enter] to edit option                               CMake Version 2.8.4
Press [c] to configure
Press [h] for help                                         Press [q] to quit without generating
Press [t] to toggle advanced mode (Currently Off)

```

```
eflorez@Adm-LEB-06:~/STIR/STIR_2.4/STIR-bin/Release
Arquivo  Editar  Ver  Pesquisar  Terminal  Ajuda
Page 2 of 2
RDF_INCLUDE_DIRS  RDF_INCLUDE_DIRS-NOTFOUND
RDF_LIBRARIES    RDF_LIBRARIES-NOTFOUND
STIR_LOCAL
STIR_MPI         OFF
STIR_OPENMP     OFF

STIR_LOCAL: Path to a file.
Press [enter] to edit option
Press [c] to configure
Press [h] for help
Press [q] to quit without generating
Press [t] to toggle advanced mode (Currently Off)
CMake Version 2.8.4
```

### III. Terceiro passo: Compilação do STIR

Como passo final executamos os seguintes comandos para concluir a instalação do STIR:

```
[eflorez@Adm-LEB-06 Release]$ make
[eflorez@Adm-LEB-06 Release]$ make test
[eflorez@Adm-LEB-06 Release]$ make install
```

#### **Passo adicional: Testes de funcionamento (opcional)**

```
[eflorez@Adm-LEB-06 Release]$ cd
[eflorez@Adm-LEB-06 ~]$ cd STIR/
[eflorez@Adm-LEB-06 STIR]$ cdrecon_test_pack/
[eflorez@Adm-LEB-06 recon_test_pack]$ ./run_tests.sh
[eflorez@Adm-LEB-06 recon_test_pack]$ ./run_ecat_tests.sh
[eflorez@Adm-LEB-06 recon_test_pack]$ ./run_scatter_tests.sh
[eflorez@Adm-LEB-06 recon_test_pack]$ ./run_test_simulate_and_recon.sh
```

Depois de todos esses passos, tudo deveria compilar sem problema nenhum e sem apresentar erros. Com isso, a instalação da plataforma de reconstrução STIR está concluída e pronta para ser utilizada.

# APÊNDICE C

## CONECTIVIDADE FUZZY (FUZZY CONNECTEDNESS - FC)

### 1. Definições iniciais da Teoria Fuzzy

O principal conceito da Teoria Fuzzy é o fato de dar flexibilidade na modelagem, permitindo desenvolver algoritmos semelhantes ao pensamento humano [Souza; De Oliveira, 1992].

A Teoria Fuzzy utiliza, como princípio fundamental de execução, uma função de associação ou de vínculo que envolve elementos específicos dentro de uma imagem [Furuie et al., 1996]. Baseados nestes vínculos, pode-se estabelecer uma classificação de relação entre os elementos que pertencem, ou não, a um objeto específico [Nyúl et al., 2002; Udupa; Samarasekera, 1996; Udupa; Saha, 2003].

### 2. Espaço Digital Fuzzy

Tendo o conjunto  $Z$  como a representação dos números inteiros, então o conjunto  $Z^n$  seria uma representação de um espaço digital  $n$ -dimensional, cujos elementos são denominados spels (Spatial Elements). No caso de uma imagem bidimensional (onde  $n=2$ ) passam a ser chamados pixels, já para imagens tridimensionais (onde  $n=3$ ), como é nosso caso, são chamados voxels [Gonzalez; Woods, 2008; Nyúl et al., 2002].

Uma relação fuzzy  $\rho$  é um subconjunto fuzzy de  $Z^n$ , definido por  $\rho = \{(x, y), \mu_\rho(x, y) \mid x, y \in Z^n\}$ , onde  $\mu_\rho(x, y): Z^n \times Z^n \rightarrow [0, 1]$  representa uma função de pertinência, cujo valor indica o grau de pertinência dos elementos  $(x, y)$  dentro do subconjunto  $\rho$  [Shaw; Simões, 1999; Gonzalez; Woods, 2008].

A relação  $\rho$  é reflexiva, se,  $\forall x \in Z^n, \mu_\rho(x, x) = 1$ ; simétrica, se,  $\forall x, y \in Z^n, \mu_\rho(x, y) = \mu_\rho(y, x)$ ; e transitiva se  $\forall x, z \in Z^n, \mu_\rho(x, z) = \max_{y \in Z^n} (\min[\mu_\rho(x, y), \mu_\rho(y, z)])$ . Caso as três propriedades tenham ocorrência simultânea,  $\rho$  é chamada de relação de semelhança. A vizinhança fuzzy cumpre as propriedades reflexiva e simétrica simultaneamente [Lage, 2010; Udupa; Samarasekera, 1996; Udupa; Saha, 2003; Gonzalez; Woods, 2008].

Considerando as definições anteriores, o par  $(Z^n, \alpha)$ , onde  $\alpha$  é uma relação de vizinhança fuzzy é chamado Espaço Digital Fuzzy.

### 3. Cena de Pertinência

Uma cena definida em  $(Z^n, \alpha)$  é um par  $C = (\mathcal{C}, f)$ , onde  $\mathcal{C} = \{c \mid -b_i \leq c_i \leq b_i, b \in Z_+^n\}$  é o domínio da cena,  $f$ , chamada intensidade da cena, é uma função definida como  $f: \mathcal{C} \rightarrow [L, H]$ , onde  $[L, H]$  é um conjunto de números, geralmente inteiros que representam os valores mínimo e máximo, dentro dos quais estão contidas as intensidades de cada um dos elementos [Nyúl et al., 2002]. Com essas premissas, denominamos cena de pertinência ao subconjunto fuzzy definido no domínio da imagem obtido como resultado de uma segmentação, cuja representação equivalente em cena é tal que a intensidade da cena age como uma função de pertinência fuzzy, com valores compreendidos entre  $[0, 1]$  [Lage, 2010].

### 4. Adjacência e Afinidade Fuzzy

A relação de adjacência fuzzy entre cada par de spels,  $(c, d)$ , definidos no domínio do espaço digital, indica a proximidade espacial entre os dois elementos comparados, atribuindo valores entre zero e um [Nyúl et al., 2002]. Se os elementos  $c$  e  $d$  fossem vizinhos seria uma condição suficiente para que tivessem adjacência Fuzzy unitária. Genericamente, a relação de adjacência é representada por  $\alpha$  e o grau de pertinência por  $\mu_\alpha(c, d)$  [Lage, 2010; Udupa; Saha, 2003; Udupa; Samarasekera, 1996; Nyúl et al., 2002].

Agora, uma relação de afinidade fuzzy, definida no domínio de uma cena  $C$ , para cada par de spels  $(c, d)$ , representa a similaridade local existente entre as intensidades dos spels comparados. Genericamente, a relação de afinidade é representada por  $k$  e o grau de pertinência dessa relação é definido por  $\mu_k(c, d)$  [Lage, 2010; Udupa; Saha, 2003; Udupa; Samarasekera, 1996; Nyúl et al., 2002]. O conceito de afinidade em um processo de segmentação é fundamental na obtenção de resultados confiáveis e eficientes [Lage, 2010].

### 5. Conectividade Fuzzy e Objeto Fuzzy

Partindo dos conceitos de adjacência e afinidade fuzzy abordados, é possível encontrar uma conectividade entre quaisquer dois spels,  $c$  e  $d$ , no domínio da cena  $C$ , a partir de uma série de afinidades locais entre o par de elementos [Nyúl et al., 2002]. Fica claro que existem vários caminhos passando somente por adjacências unitárias que consigam unir esses dois spels. A uma conectividade

global no domínio  $C$  dá-se o nome de Fuzzy Connectedness, representada por  $K$  e sua função de pertinência entre  $c$  e  $d$  por  $\mu_K(c, d)$ .

Assumindo que  $p_{c,d} = \langle s_1, s_2, \dots, s_N \rangle$  seja um caminho entre  $s_1 = c$  e  $s_N = d$ , onde  $s_i$  são  $N$  spels adjacentes em pares entre si, a afinidade desse caminho é dada por:

$$\mu_K(p_{c,d}) = \min_{1 < i < N} (\mu_k(s_{i-1}, s_i)) \quad (17)$$

Então, a equação (2.1) mostra cada caminho entre dois elementos  $(c, d)$  a qual estará definida, representada ou limitada pela afinidade mais fraca do conjunto de afinidades que esse caminho possua. Como foi dito, são inúmeros os caminhos entre dois spels e como é evidente em um caso ótimo, a conectividade global deverá considerá-los todos. Com tudo isso, a seleção da maior afinidade de caminho será a melhor representação da conectividade global. Portanto, assumindo que  $P_{c,d}$  seja um conjunto de todos os caminhos  $p$ , a conectividade global entre  $c$  e  $d$  é definida como:

$$\mu_K(c, d) = \max_{p \in P_{c,d}} (\min_{1 < i < N} (\mu_k(s_{i-1}, s_i))) \quad (18)$$

Dessa forma, a quantificação da função de pertinência de uma conectividade é totalmente dependente da definição de afinidade local entre os elementos que a compõem [Lage, 2010].

# APÊNDICE D

## USO DA MACRO NA OBTENÇÃO DOS *PHANTOMS* MASH

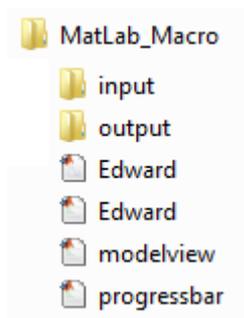
Como foi indicado no Capítulo 3 na secção 3.2.1, na simulação foi usado um *phantom* antropomórfico do corpo de um homem adulto o qual é gerado através de um programa feito em MatLab chamado MASH.

Nesse anexo são indicados com detalhe os passos necessários para conseguir obter o *phantom* que foi incorporado e executado dentro da plataforma GATE. A vantagem deste processo é que o usuário pode pegar a simulação inteira (corpo inteiro) o no defeito, selecionar uma parte dele para começar a simulação.

A macro e o conjunto de arquivos que dão origem a estes *phantoms* podem ser descarregados do seguinte endereço eletrônico:

<https://dl.dropboxusercontent.com/Edward.tar.gz>

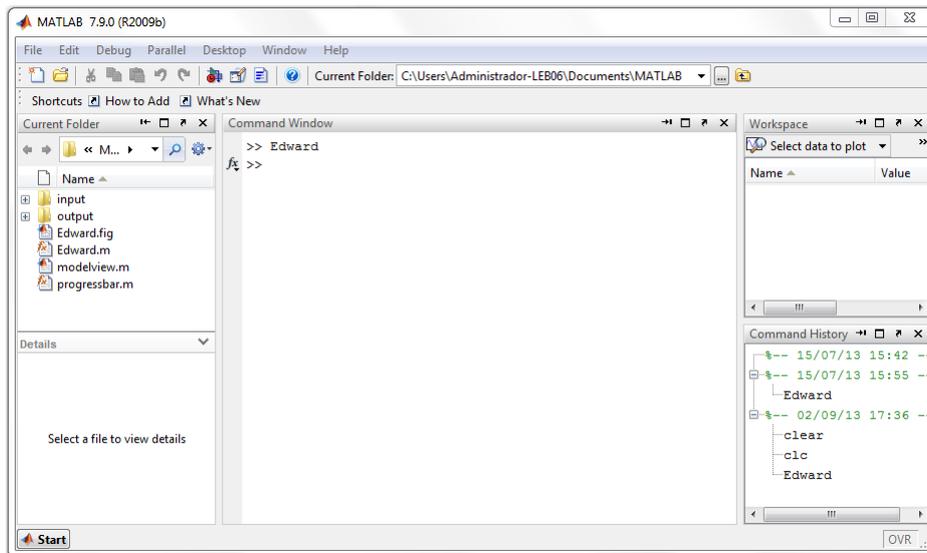
Dentro dessa pasta principal encontraremos os seguintes arquivos que foram utilizados no MatLab para obter os volumes antropomórficos simulados, os quais estão organizados da seguinte maneira:



Todos esses arquivos devem ser copiados e colados na carpeta do MatLab do nosso computador na seguinte rota: Bibliotecas>Documentos>MATLAB. Na sequência são indicados os passos necessários para o processo de obtenção dos volumes simulados:

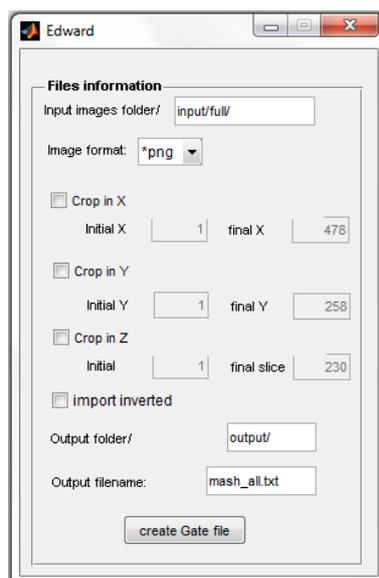
- Abrimos o MatLab na tela de comando executamos (Figura 41):

```
>> Edward
```



**FIGURA 41** – Tela principal do MatLab onde é executado o comando que permite gerar a simulação do *phantom* antropomórfico MASH.

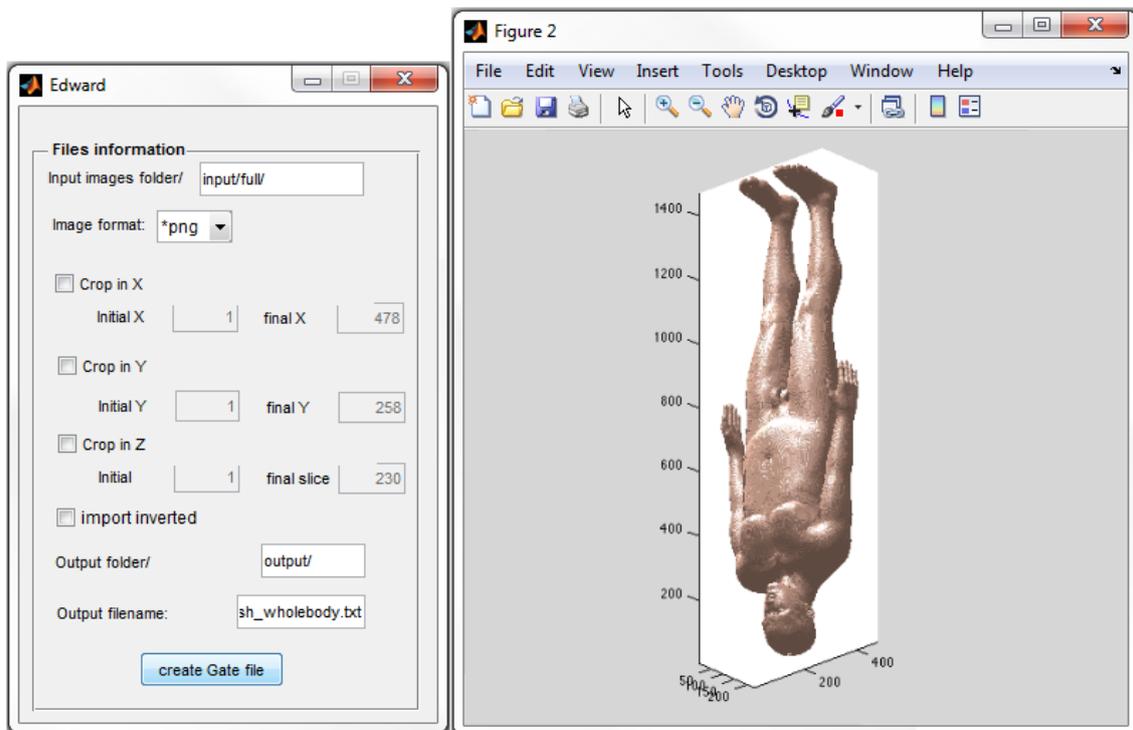
- Uma vez executado o arquivo principal, uma janela de interface será mostrada (Figura 42) para iniciar a geração do *phantom*. Nesta interface, o usuário pode fazer escolhas dependendo se quer obter a simulação do corpo inteiro ou somente uma parte dele, assim mesmo é possível definir algumas características do arquivo de saída, nome do arquivo de saída e a extensão que ele vai ter.



**FIGURA 42** – Janela de simulação onde o usuário pode fazer algumas escolhas para definir a abrangência da simulação resultante.

Como exemplo, na janela de simulação (Figura 43) não mudamos nada com o objetivo de exportar a simulação do corpo inteiro. Chamamos o arquivo de saída como `mash_wholebody.txt` o qual vai ficar guardado na pasta Output como é indicado. Fazemos clic em 'create Gate file' e a geração será iniciada.

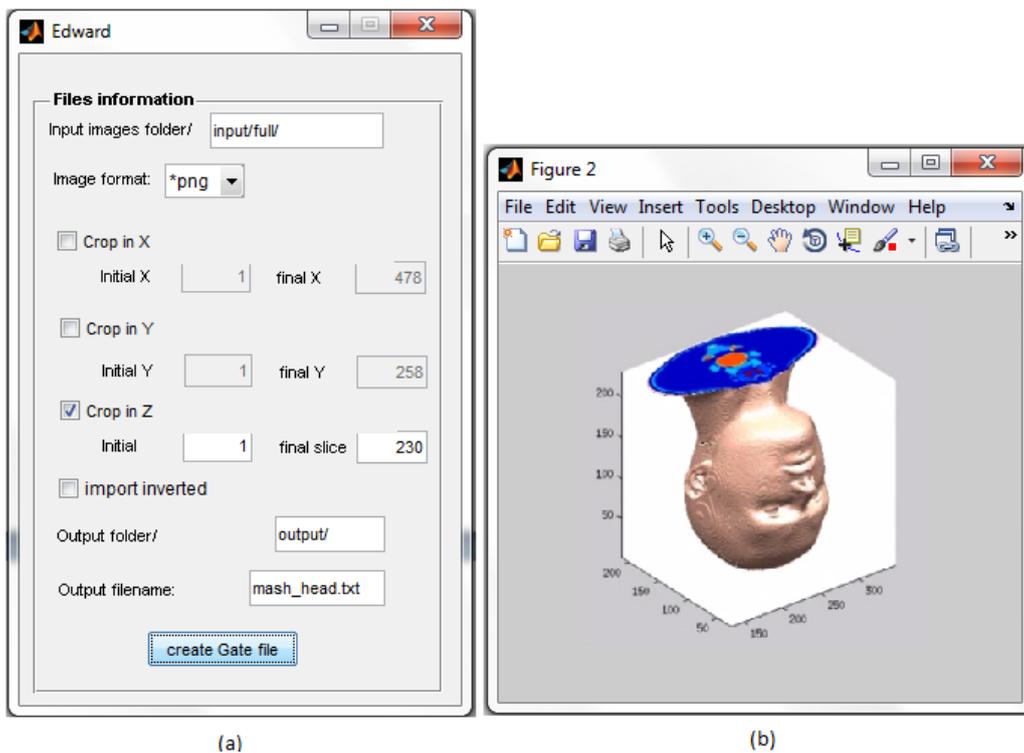
No processo de geração do volume simulada, duas janelas serão visualizadas: uma com o estado da informação de entrada que está sendo utilizada na simulação, e a outra é a reconstrução 3D propriamente dita do volume simulado. Uma vez que os dois processos atingiram o 100%, uma janela contendo a simulação requerida vai ser visualizada.



**FIGURA 43** – MASH: *Phantom* antropomórfico inteiro simulado, vista lateral do corpo simulado.

Como foi dito, a simulação pode simplesmente conter uma parte específica do corpo. Por exemplo, se queremos obter unicamente a parte da cabeça poderemos delimitar a simulação no eixo “z” para ser desde o ponto inicial 1 até o ponto 230 (eu sei que a cabeça está nessa faixa), tal como é mostrado na Figura 44 junto com a simulação resultante.

Da mesma forma como foi obtida a simulação da cabeça, podemos alterar as opções dos comandos da janela de simulação para obter qualquer outra parte do corpo. Deve-se ter em consideração que todos os arquivos pertencentes às simulações geradas são carregados na pasta OUTPUT.



**FIGURA 44** – MASH: Simulação parcial, (a) janela de comando da simulação, e (b) volume da cabeça obtido na simulado.

Como informação adicional, na Tabela 12 são mostrados a lista completa dos materiais que pertencem ao *phantom* antropomórfico MASH. Os números de ID são usados na simulação realística PET através do GATE, a fim de identificar uma região específica e associar essa região com uma quantidade específica de radiofármaco.

ID	Material	mat	rho	Tissue
1	Skin	27	-1.090	Skin
2	Adipose_subcutaneous	49	-0.950	Adipose_tissue
4	Eyes	34	-1.050	Eyes
5	Lens_of_eye	34	-1.050	Eyes
6	Muscle	29	-1.050	Muscle_tissue
7	Brain	32	-1.050	Brain
8	Spinal_chord	54	-1.038	Spinal_chord
9	Adrenals	43	-1.030	Adrenals
10	Lungs	50	-0.382	Lung_tissue_(compressed_lungs)
12	Pancreas	31	-1.050	Pancreas
13	Liver	30	-1.050	Liver
14	Kidneys	35	-1.050	Kidneys
15	Thymus	45	-1.030	Gallbladder_Pituitary_gland_T
16	Spleen	39	-1.040	Spleen
17	Nasal_passage	45	-1.030	Gallbladder_Pituitary_g
18	Tongue	29	-1.050	Muscle_tissue
19	Oral_cavity	45	-1.030	Gallbladder
20	Salivary_glands_submandibula	45	-1.030	Gallbladder_
21	Salivary_glands_paroti	45	-1.030	Gallbladder_Pituitary
22	Salivary_glands_sublingual	45	-1.030	Gallbladder_Ps
24	Farynx+_Larynx	45	-1.030	Gallbladder_Pituitary_
26	Oesophagus	44	-1.030	Oesophagus
27	Trachea	45	-1.030	Gallbladder_Pituitary_gland_Trac
28	Thyroid	40	-1.040	Thyroid
29	Bronchi	50	-0.382	Lung_tissue_(compressed_lungs)
30	Gall_bladder_contents	51	-1.040	Gastrointestinal_tract__contents
32	Stomach_contents	51	-1.040	Gastrointestinal_tract__contents
34	Small_intestine_contents	51	-1.040	Gastrointestinal_tract__contents
36	Colon_contents	51	-1.040	Gastrointestinal_tract__contents
38	Urinary_bladder_contents	52	-1.040	Urine
39	Prostate	46	-1.030	Prostate
40	Testicles	42	-1.040	Testes
41	Lymphatic_nodes	47	-1.030	Lymph
42	Soft_tissue	3	-1.050	soft tissue
43	Periarticular_tissue+_tendon	26	-1.100	Cartilage
46	Glandular_tissue	45	-1.030	Gallbladder_Pituita
48	Blood	28	-1.060	Blood
49	Penis	42	-1.040	Testes
50	Adipose_abdominal_(visceral)	49	-0.950	Adipose_tissue
55	Peritoneum	55	-1.040	Lower_arm_spongiosa
89	Gall_bladder_wall	45	-1.030	Gallbladder_Pituitary
91	Stomach_wall	36	-1.040	Stomach
93	Small_intestine_wall	37	-1.040	Small_intestine
95	Colon_wall	38	-1.040	Large_intestine
97	Urinary_bladder_wall	41	-1.040	Urinary_bladder
98	Adipose_breasts	49	-0.950	Adipose_tissue
99	Heart	33	-1.050	Heart
100	Teeth	1	-2.750	Teeth

ID	Material	mat	rho	Tissue
101	Mandible_compact_bone	2	-1.920	Mineral_bone
102	Cranium_compact_bone	2	-1.920	Mineral_bone
103	Sternum_compact_bone	2	-1.920	Mineral_bone
104	Ribs_compact_bone	2	-1.920	Mineral_bone
105	Clavicle_left_compact_bone	2	-1.920	Mineral_bone
106	Clavicle_right_compact_bone	2	-1.920	Mineral_bone
107	Scapula_left_compact_bone	2	-1.920	Mineral_bone
108	Scapula_right_compact_bone	2	-1.920	Mineral_bone
109	Cervical_spine_compact_bone	2	-1.920	Mineral_bone
110	Thorarcic_spine_compact_bor	2	-1.920	Mineral_bone
111	Lumbar_spine_compact_bone	2	-1.920	Mineral_bone
112	Sacrum_compact_bone	2	-1.920	Mineral_bone
113	Pelvis_compact_bone	2	-1.920	Mineral_bone
114	Humero_left_compact_bone	2	-1.920	Mineral_bone
115	Radius+Ulna_left_compact_bc	2	-1.920	Mineral_bone
116	Hand_left_compact_bone	2	-1.920	Mineral_bone
117	Humero_right_compact_bone	2	-1.920	Mineral_bone
118	Radius+Ulna_right_compact_t	2	-1.920	Mineral_bone
119	Hand_right_compact_bone	2	-1.920	Mineral_bone
120	Femur_left_compact_bone	2	-1.920	Mineral_bone
121	Patela_left_compact_bone	2	-1.920	Mineral_bone
122	Tibia+Fibula_left_compact_bo	2	-1.920	Mineral_bone
123	Foot_left_compact_bone	2	-1.920	Mineral_bone
124	Femur_right_compact_bone	2	-1.920	Mineral_bone
125	Patela_right_compact_bone	2	-1.920	Mineral_bone
126	Tibia+Fibula_right_compact_b	2	-1.920	Mineral_bone
127	Foot_right_compact_bone	2	-1.920	Mineral_bone
128	Hioide_compact_bone_	2	-1.920	Mineral_bone
131	Mandible_spongiosa	63	-1.143	Mandible_spongiosa
132	Cranium_spongiosa	59	-1.081	Cranium_spongiosa
133	Sternum_spongiosa	57	-1.055	ribcage_spongiosa
134	Ribs_spongiosa	57	-1.055	ribcage_spongiosa
135	Clavicle_left_spongiosa	57	-1.055	ribcage_spongiosa
136	Clavicle_right_spongiosa_	57	-1.055	ribcage_spongiosa
137	Scapula_left_spongiosa	57	-1.055	ribcage_spongiosa
138	Scapula_right_spongiosa	57	-1.055	ribcage_spongiosa
139	Cervical_spine_spongiosa	58	-1.339	spine_spongiosa
140	Thorarcic_spine_spongiosa	58	-1.339	spine_spongiosa
141	Lumbar_spine_spongiosa	58	-1.339	spine_spongiosa
142	Sacrum_spongiosa	60	-1.063	pelvis_spongiosa
143	Pelvis_spongiosa	60	-1.063	pelvis_spongiosa
144	Humero_left_spongiosa_uppe	56	-1.079	Upper_arm_bones_spongiosa
145	Radius+Ulna_left_spongiosa	55	-1.140	Radius_Ulna
146	Hand_left_spongiosa	55	-1.140	Lower_arm_bones_spongiosa
147	Humero_right_spongiosa_upp	56	-1.079	Upper_arm_bones_spongiosa
148	Radius+Ulna_right_spongiosa	55	-1.140	Lower_arm_bones_spongiosa
149	Hand_right_spongiosa	55	-1.140	Lower_arm_bones_spongiosa
150	Femur_left_spongiosa_upper_	61	-1.107	upper_leg_spongiosa

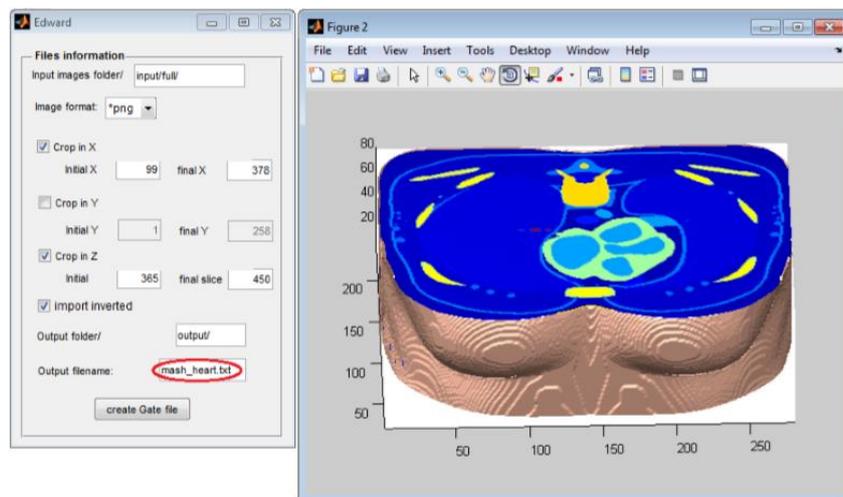
ID	Material	mat	rho	Tissue
151	Patela_left_spongiosa	62	-1.206	lower_leg_spongiosa
152	Tibia+Fibula_left_spongiosa	62	-1.206	lower_leg_spongiosa
153	Foot_left_spongiosa	62	-1.206	lower_leg_spongiosa
154	Femur_right_spongiosa_upper	61	-1.107	upper_leg_spongiosa
155	Patela_right_spongiosa	62	-1.206	lower_leg_spongiosa
156	Tibia+Fibula_right_spongiosa	62	-1.206	lower_leg_spongiosa
157	Foot_right_spongiosa	62	-1.206	lower_leg_spongiosa
161	Humero_left_medullary_yellow	22	-0.980	Humeri_and_femora_upper_half
162	Radius+Ulna_left_medullary_yell	22	-0.980	Lower_arm_bones_medullary_cav
163	Humero_right_medullary_yell	22	-0.980	Humeri_and_femora_upper_half
164	Radius+Ulna_right_medullary_yell	22	-0.980	Lower_arm_bones_medullary_cav
165	Femur_left_medullary_yellow	22	-0.980	Humeri_and_femora_upper_half
166	Tibia+Fibula_left_medullary_yell	22	-0.980	Lower_leg_bones_medullary_cav
167	Femur_right_medullary_yellow	22	-0.980	Humeri_and_femora_upper_half
168	Tibia+Fibula_right_medullary_yell	22	-0.980	Lower_leg_bones_medullary_cav
170	Spine_-_intervertebral_disks	26	-1.100	Cartilage
171	Trunk_-_cartilage_articular	26	-1.100	Cartilage
172	Neck_-_cartilage_thyroid_and	26	-1.100	Cartilage
173	Ams_-_cartilage_articular	26	-1.100	Cartilage
174	Ribs_-_cartilage_articular	26	-1.100	Cartilage
175	Head_-_cartilage_ears	26	-1.100	Cartilage
176	Head_-_cartilage_nose	26	-1.100	Cartilage
177	Legs_-_cartilage_articular	26	-1.100	Cartilage
180	Humero_left_-_spongiosa_low	62	-1.206	lower_leg_spongiosa
182	Humero_right_-_spongiosa_low	62	-1.206	lower_leg_spongiosa
184	Femur_left_-_spongiosa_low	62	-1.206	lower_leg_spongiosa
186	Femur_right_-_spongiosa_low	62	-1.206	lower_leg_spongiosa
200	Air_internal	53	-0.001	Air
255	Vacuum	53	-0.001	Vacuum

TABELA 12 – Lista dos materiais do *phantom* antropomórfico MASH completo.

# APÊNDICE E

## PHANTOMS 3D: FORMATO DE ENTRADA ASCII

No caso do *phantom* antropomórfico MASH, a macro de simulação gera como saída um arquivo ASCII (\*.txt / \*.dat) com os valores das matrizes de intensidades que conformam a parte do corpo ou o volume selecionado. Na Figura 45 pode-se apreciar que temos limitado a simulação unicamente à região do tórax, obtendo como saída o arquivo **mash\_heart.txt** com as matrizes de intensidades do volume apresentado.



**FIGURA 45** – Plataforma de simulação do *phantom* MASH. Note-se que o arquivo de saída já está no formato ASCII \*.txt.

No entanto, pode ser o caso, como o *phantom* gerado pelo NCAT-4D por exemplo, que precisa de um passo intermediário de conversão das suas imagens 2D para matrizes de valores de intensidades. Assim, através de algumas linhas de MatLab fizemos a extração dos valores das matrizes de cada uma das fatias de 1 dos 16 volumes do NCAT-4D. Cada volume está composto por um grupo de 20 fatias de 64 x 64 pixels. O código utilizado para esse fim é o seguinte:

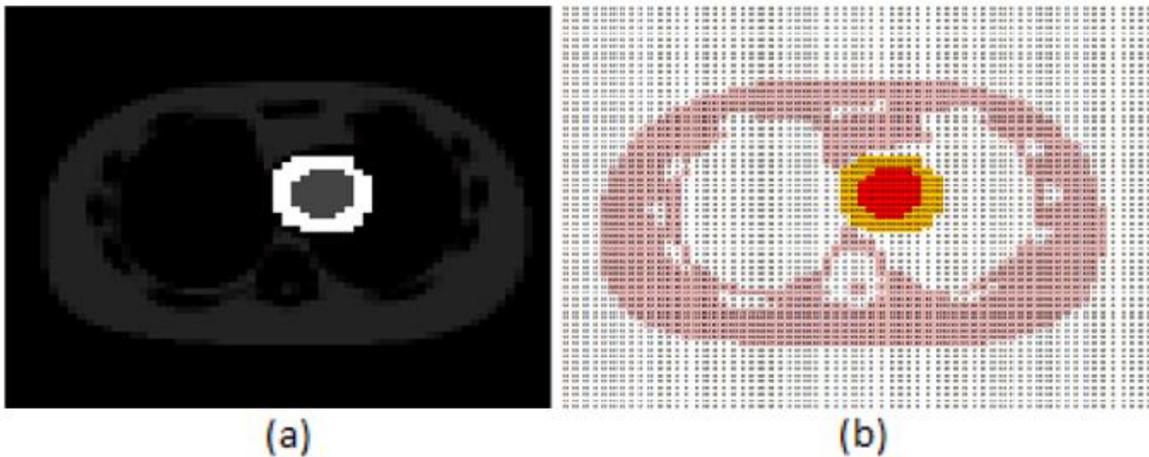
```
I = zeros(64,64,20);  
for fatia = 1:20  
    close all  
    Imagem = ['caso_',num2str(fatia),'.png']; %'Phantom.png'
```

```

Im = im2uint8(imread(Imagem));
figure
image(im2uint8(Im));colormap(gray(256))
title('Imagem Original');
I(:,:,fatia) = Im;
xlswrite('resultadosEd.xls', Im(1:64,1:64));
end

```

Na Figura 46(a) se mostra uma das fatias do *phantom* NCAT\_4D que conforma o ventrículo esquerdo do coração simulado. Depois de aplicado o algoritmo de extração das matrizes de intensidades anteriormente mencionado, foi possível obter uma matriz com os respectivos valores de intensidades, apresentada na Figura 46(b). Só para efeitos de visualização foram associadas algumas cores aos valores representativos da caixa torácica, do miocárdio e do sangue (parte interna do miocárdio).



**FIGURA 46** – Processo de conversão e geração das matrizes de valores: (a) Fatia 2D do volume do VE do coração, e (b) matriz de valores de intensidade da fatia 2D.

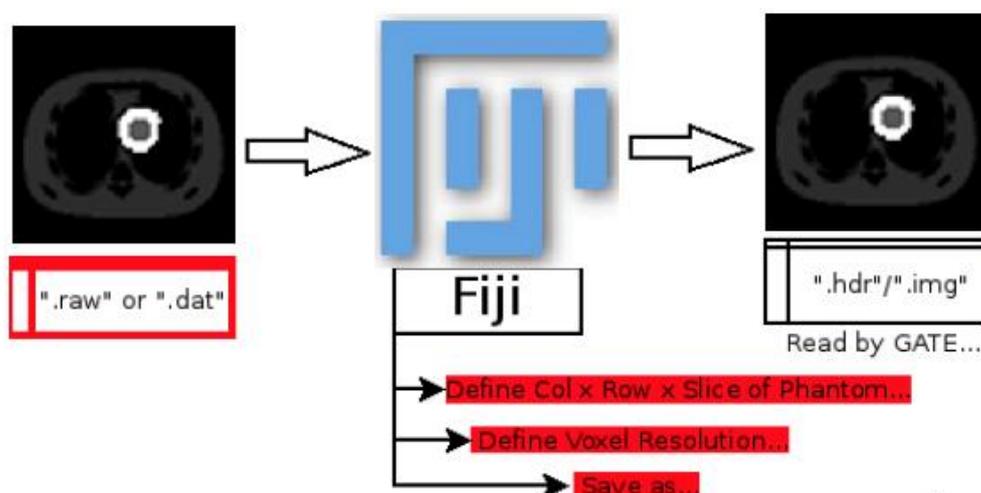
# APÊNDICE F

## PROCESSO PARA GERAR OS FORMATOS ANALYSE / INTERFILE

Este procedimento é referido em “*A user-friendly step by step tutorial for use of voxel phantoms in Gate v6.1: Applications in Dosimetry*”, publicado pelo Dr. Daniel A.B. Bonifácio. Como exemplo para representar este processo foi utilizado um dos 16 volumes do Ventrículo Esquerdo do NCAT-4D, mas serve para qualquer outro tipo de *phantom* antropomórfico como o MASH, entre outros. É preciso indicar que este procedimento pressupõe a utilização dos seguintes programas:

- (1) GATE Versão 6.2 (<http://www.opengatecollaboration.org/>);
- (2) Fiji ou ImageJ (<http://ij.sc/wiki/index.php/Downloads>);
- (3) XMedcon (<http://xmedcon.sourceforge.net/>).

A Figura 47 ilustra o diagrama do processo de conversão no Fiji (ImageJ) do formato '.raw' para o formato analyse ('.img/.hdr'), o qual é um dos formatos lidos pelo GATE.



**FIGURA 47** – Diagrama Geral: Conversão de ".raw" or ".dat" para formato analyse no Fiji.

Assim, foi importado o arquivo `ncat_lv60bpm_act_TOTAL.bin`, o qual contém os parâmetros do *phantom* antropomórfico do Ventrículo Esquerdo. Após disso, são definidas as dimensões tal como se apresenta na Figura 48. Posteriormente, foram checados os parâmetros e/ou dimensões do *phantom* que será acoplado, como se mostra na Figura 49 y com isso, foi gravado o arquivo em formato Analyse: \*.hdr / \*.img.

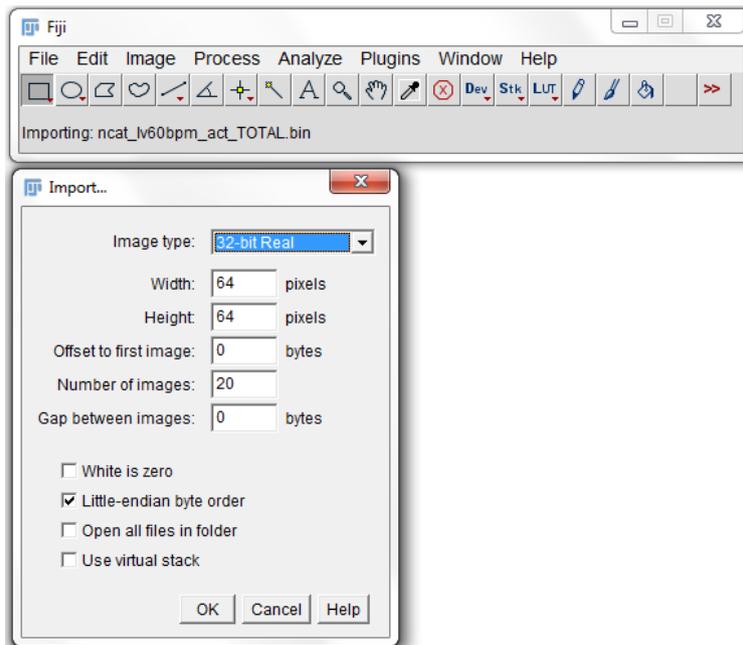


FIGURA 48 – Caminho para abrir e definir as dimensões do arquivo “.raw”.

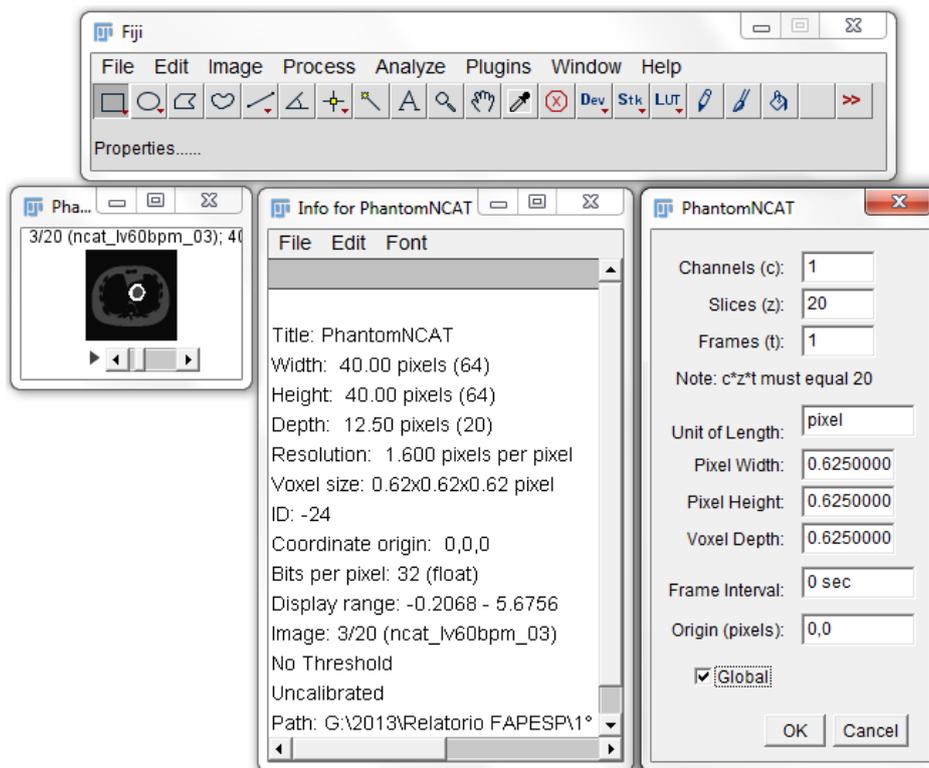


FIGURA 49 – Caminho para checar as dimensões dos voxels.

Na sequência, utilizando o Xmedcon é carregado o arquivo \*.hdr como é detalhado na Figura 50.

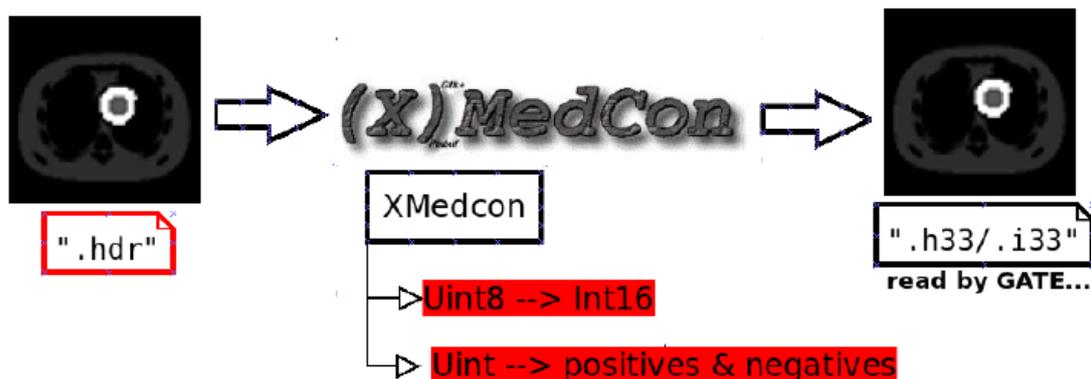


FIGURA 50 – Conversão de Analyse para o formato interfile no XMedCon.

Uma vez com a imagem aberta, tem-se que seleccionar as seguintes opções (esse processo está ilustrado na Figura 51):

- Na aba Pixels seleccionar:
  - Values - without quanti\_cation
  - Sign - positives only
  - Types - writing Uint8 pixels
  - Normalization - over all images
- Na aba File seleccionar:
  - Endian Type - writing LITTLE endian.

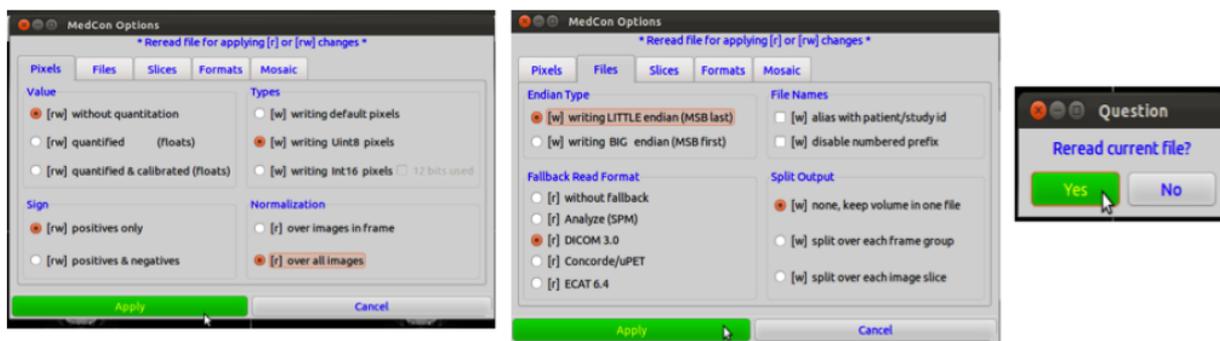


FIGURA 51 – Janela e abas do "XMedCon Options": parâmetros fundamentais para que o GATE possa ler os *phantoms*.

Finalmente, é salvo o arquivo no formato Interfile como apresentado na Figura 52:

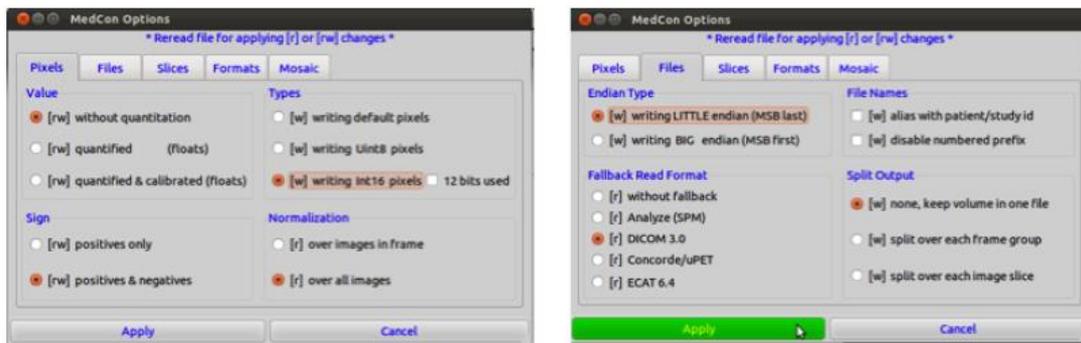


FIGURA 52 – Janela "Save as Interfile".

Uma vez obtido o arquivo na extensão interfile, podemos incorporar esse arquivo na simulação do *Phantom* NCAT-4D dentro da plataforma GATE através das seguintes linhas de comando e mostrado na Figura 53:

```
# VOXELIZED MATRIX NCAT-4D PHANTOM
/gate/world/daughters/name Heart_LV
/gate/world/daughters/insert regularMatrix
/gate/Heart_LV/geometry/insertReader interfile
/gate/Heart_LV/interfileReader/insertTranslator range
/gate/Heart_LV/interfileReader/rangeTranslator/readTable range_LV.dat
/gate/Heart_LV/interfileReader/rangeTranslator/describe 1
/gate/Heart_LV/interfileReader/readFile NCAT_LV.h33
/gate/Heart_LV/placement/setTranslation 0. 0. 0. mm
/gate/Heart_LV/placement/setRotationAxis 1 0 0
/gate/Heart_LV/placement/setRotationAngle 180 deg
/gate/Heart_LV/attachVoxelPhantomSD
```

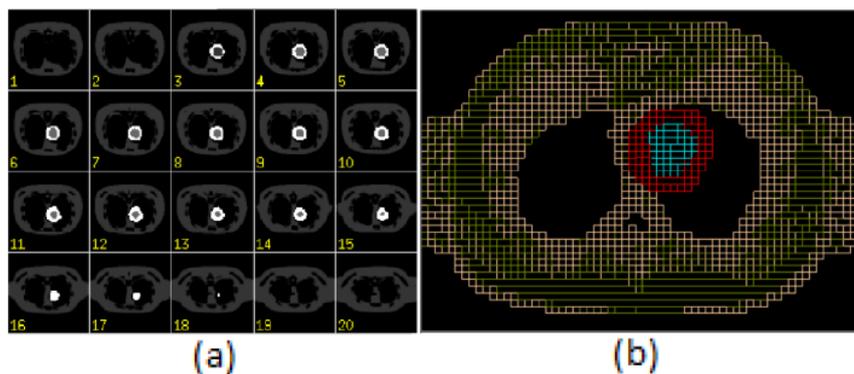


FIGURA 53 – *Phantom* simulado usando o software NCAT-4D. (a) projeções 2D do ventrículo esquerdo dentro da caixa torácica usando o software XMedCon, (b) volume gerado com a plataforma GATE.

# APÊNDICE G

## LINHAS DE COMANDO GATE: TOMÓGRAFO PET SIEMENS BIOGRAPH + PHANTOM ANTROPOMORFICO MASH (TÓRAX)

```
#-----oOOOOO00000O0000O-----#
#
# Simulation PET_Siemens_BiographTM + MASH (Heart) (versão 6.24)
# AUTOR: Edward Flórez Pacheco
# MENTOR: Prof. Dr. Sergio S. Furuie
# CO-MENTOR: Dr. Vani Vijayakumar
# School of Engineering of the University of Sao Paulo, Brazil
# Medical Image Processing Laboratory
#
# Made in the Department of Radiology
# University of Mississippi Medical Center (UMMC)
# Jackson Mississippi - USA
#
# OBJETIVE: Execute the acquisition of the projections of the heart using
#           a mathematical phantom (MASH) w/ all the others organs of the
#           toracic region (lungs, skeleton, etc), within the modelated
#           Siemens BiographTM scanner (Nuclear Medicine Center of SP).
#-----oOOOOO00000O0000O-----#

#=====
# VISUALISATION
#=====

/vis/open                OGLIX
#/vis/open               OGLSX
/vis/viewer/set/viewpointThetaPhi  0 0
/vis/viewer/zoom        6.0
/vis/viewer/set/style   surface
/vis/drawVolume
/vis/viewer/flush
/tracking/verbose       0
/tracking/storeTrajectory 1
/vis/scene/add/trajectories
/vis/scene/endOfEventAction accumulate -1
/vis/ogl/set/displayListLimit 10000000000
/vis/modeling/trajectories/create/drawByCharge

#=====
# VERBOSE
#=====

/gate/verbose Physic    5
/gate/verbose Cuts     0
/gate/verbose SD       0
/gate/verbose Actions  0
/gate/verbose Actor    0
/gate/verbose Step     0
/gate/verbose Error    0
/gate/verbose Warning  0
/gate/verbose Output   0
/gate/verbose Beam     0
/gate/verbose Volume   0
/gate/verbose Image    0
/gate/verbose Geometry 0
/gate/verbose Core     0
```

```
/run/verbose      0
/event/verbose    0
/tracking/verbose 0
```

```
#=====
# GEOMETRY
#=====
```

```
/gate/geometry/setMaterialDatabase data/GateMaterials.db
/gate/geometry/setMaterialDatabase data/MASHGateMaterials.db
```

```
/gate/geometry/setMaterialDatabase ../../GateMaterials.db
```

```
#
# WORLD
#
```

```
/gate/world/geometry/setXLength 400. cm
/gate/world/geometry/setYLength 400. cm
/gate/world/geometry/setZLength 400. cm
```

```
#-----oooooooooooooooooooooooo-----#
#
# DEFINITION AND DESCRIPTION
# OF YOUR PET DEVICE
#
#-----oooooooooooooooooooooooo-----#
```

```
#=====
# CAMERA PET: Siemens BiographTM (Nuclear Medicine Center - SP)
#=====
```

```
# ECAT
/gate/world/daughters/name      ecat
/gate/world/daughters/insert    cylinder
/gate/ecat/setMaterial          Air
/gate/ecat/geometry/setRmax     44.2 cm
/gate/ecat/geometry/setRmin     41.2 cm
/gate/ecat/geometry/setHeight   15.52 cm
/gate/ecat/vis/forceWireframe
```

```
# BLOCK
/gate/ecat/daughters/name      block
/gate/ecat/daughters/insert    box
/gate/block/placement/setTranslation 42.7 0.0 0.0 cm
/gate/block/geometry/setXLength 30.0 mm
/gate/block/geometry/setYLength 35.8594 mm
/gate/block/geometry/setZLength 38.7 mm
/gate/block/setMaterial        Air
/gate/block/vis/forceWireframe
```

```
# CRYSTAL
/gate/block/daughters/name     crystal
/gate/block/daughters/insert   box
/gate/crystal/geometry/setXLength 30.0 mm
/gate/crystal/geometry/setYLength 4.4 mm
/gate/crystal/geometry/setZLength 4.75 mm
/gate/crystal/setMaterial      BGO
/gate/crystal/vis/setColor     red
```

```
# REPEAT CRYSTAL
/gate/crystal/repeaters/insert cubicArray
```

```

/gate/crystal/cubicArray/setRepeatNumberX 1
/gate/crystal/cubicArray/setRepeatNumberY 8
/gate/crystal/cubicArray/setRepeatNumberZ 8
/gate/crystal/cubicArray/setRepeatVector 0. 4.4942 4.85 mm

```

```
# R E P E A T B L O C K
```

```

/gate/block/repeaters/insert linear
/gate/block/linear/setRepeatNumber 4
/gate/block/linear/setRepeatVector 0. 0. 38.8 mm
/gate/block/repeaters/insert ring
/gate/block/ring/setRepeatNumber 72

```

```
# T U N G S T E N S H I E L D
```

```

/gate/world/daughters/name carter
/gate/world/daughters/insert cylinder
/gate/carter/setMaterial Air
/gate/carter/geometry/setRmax 44. cm
/gate/carter/geometry/setRmin 28. cm
/gate/carter/geometry/setHeight 9. cm
/gate/carter/placement/setTranslation 0.0 0.0 12.5 cm
/gate/carter/vis/forceWireframe

```

```

/gate/carter/daughters/name carter1
/gate/carter/daughters/insert cylinder
/gate/carter1/setMaterial Tungsten
/gate/carter1/geometry/setRmax 30.0 cm
/gate/carter1/geometry/setRmin 29.0 cm
/gate/carter1/geometry/setHeight 3.0 cm
/gate/carter1/vis/setColor gray
/gate/carter1/placement/setTranslation 0.0 0.0 -3 cm

```

```

/gate/carter/daughters/name carter2
/gate/carter/daughters/insert cylinder
/gate/carter2/setMaterial Tungsten
/gate/carter2/geometry/setRmax 40.7 cm
/gate/carter2/geometry/setRmin 30.0 cm
/gate/carter2/geometry/setHeight 1.0 cm
/gate/carter2/vis/setColor gray
/gate/carter2/placement/setTranslation 0.0 0.0 -2 cm

```

```

/gate/carter/daughters/name carter3
/gate/carter/daughters/insert cylinder
/gate/carter3/setMaterial Tungsten
/gate/carter3/geometry/setRmax 40.7 cm
/gate/carter3/geometry/setRmin 39.7 cm
/gate/carter3/geometry/setHeight 3.0 cm
/gate/carter3/vis/setColor gray
/gate/carter3/placement/setTranslation 0.0 0.0 0 cm

```

```

/gate/carter/daughters/name carter4
/gate/carter/daughters/insert cylinder
/gate/carter4/setMaterial Tungsten
/gate/carter4/geometry/setRmax 43.7 cm
/gate/carter4/geometry/setRmin 40.7 cm
/gate/carter4/geometry/setHeight 1.0 cm
/gate/carter4/vis/setColor gray
/gate/carter4/placement/setTranslation 0.0 0.0 1 cm

```

```
# A T T A C H S Y S T E M
```

```

/gate/systems/ecat/block/attach block
/gate/systems/ecat/crystal/attach crystal

```

```

# ATTACH CRYSTAL SD
/gate/crystal/attachCrystalSD

#=====
# Global Box (For the MASH PHANTOM)
#=====

/gate/world/daughters/name          mainbox
/gate/world/daughters/insert        box
/gate/mainbox/geometry/setXLength   40 cm
/gate/mainbox/geometry/setYLength   40 cm
/gate/mainbox/geometry/setZLength   40 cm
#/gate/mainbox/placement/setTranslation 0.0 0.0 +20.1 cm
/gate/mainbox/placement/setTranslation 0.0 0.0 0.0 cm
/gate/mainbox/setMaterial            Air
#/gate/mainbox/vis/setVisible        1
/gate/mainbox/vis/setVisible        1
/gate/mainbox/vis/setVisible        0
/gate/mainbox/vis/setColor          magenta
/gate/mainbox/vis/forceWireframe

#-----oooooooooooooooooooooooo-----#
#
# DEFINITION AND DESCRIPTION
# OF YOUR PHANTOM
#
#-----oooooooooooooooooooooooo-----#

#=====
# VOXEL PHANTOM: MASH Phantom (Heart)
#=====

/gate/world/daughters/name          mash_heart
/gate/world/daughters/insert        regularMatrix

/gate/mash_heart/geometry/insertReader          interfile
/gate/mash_heart/interfileReader/insertTranslator range
/gate/mash_heart/interfileReader/rangeTranslator/readTable range_atten_mash_heart.dat
/gate/mash_heart/interfileReader/rangeTranslator/describe 1
/gate/mash_heart/interfileReader/readFile        Heart_study.h33
/gate/mash_heart/placement/setTranslation        0. 0. 0. mm
/gate/mash_heart/placement/setRotationAxis      1 0 0
/gate/mash_heart/placement/setRotationAngle    180 deg
/gate/mash_heart/attachVoxelPhantomSD

#/gate/phantom/setSkipEqualMaterials            0
#/gate/phantom/setFictitiousEnergy              111. keV
#/gate/phantom/setGammaDiscardEnergy            0.1 keV

#=====
# PHYSICS
#=====

/gate/physics/addProcess PhotoElectric
/gate/physics/processes/PhotoElectric/setModel StandardModel

/gate/physics/addProcess Compton
/gate/physics/processes/Compton/setModel      StandardModel

/gate/physics/addProcess RayleighScattering
/gate/physics/processes/RayleighScattering/setModel PenelopeModel

```

```
/gate/physics/addProcess ElectronIonisation
/gate/physics/processes/ElectronIonisation/setModel StandardModel e-
/gate/physics/processes/ElectronIonisation/setModel StandardModel e+
```

```
/gate/physics/addProcess Bremsstrahlung
/gate/physics/processes/Bremsstrahlung/setModel StandardModel e-
/gate/physics/processes/Bremsstrahlung/setModel StandardModel e+
```

```
/gate/physics/addProcess PositronAnnihilation
```

```
/gate/physics/addProcess MultipleScattering e+
/gate/physics/addProcess MultipleScattering e-
```

```
/gate/physics/processList Enabled
/gate/physics/processList Initialized
```

```
#####
# C U T S
#####
```

```
/gate/physics/Gamma/SetCutInRegion crystal 1.0 cm
/gate/physics/Electron/SetCutInRegion crystal 1.0 cm
/gate/physics/Positron/SetCutInRegion crystal 1.0 cm
```

```
/gate/physics/Gamma/SetCutInRegion mash_heart 0.1 mm
/gate/physics/Electron/SetCutInRegion mash_heart 0.1 mm
/gate/physics/Positron/SetCutInRegion mash_heart 0.1 mm
```

```
/gate/physics/SetMaxStepSizeInRegion mash_heart 0.01 mm
```

```
#=====
# INITIALISATION
#=====
```

```
/gate/run/initialize
```

```
#-----oOOOOO00000O0000O0000o-----#
#
# DEFINITION OF YOUR ACQUISITION
# DIGITIZER & COINCIDENCE SHORTER
#
#-----oOOOOO00000O0000O0000o-----#
```

```
#####
# DIGITIZER
#####
```

```
# A D D E R
/gate/digitizer/Singles/insert adder
```

```
# R E A D O U T
/gate/digitizer/Singles/insert readout
/gate/digitizer/Singles/readout/setDepth 1
```

```
# E N E R G Y B L U R R I N G
/gate/digitizer/Singles/insert blurring
/gate/digitizer/Singles/blurring/setResolution 0.26
/gate/digitizer/Singles/blurring/setEnergyOfReference 511. keV
```

```
# E N E R G Y C U T
/gate/digitizer/Singles/insert thresholder
```

```

/gate/digitizer/Singles/threshold/setThreshold 250. keV
/gate/digitizer/Singles/insert upholder
/gate/digitizer/Singles/upholder/setUphold 750. keV

#   D E A D   T I M E
/gate/digitizer/Singles/insert deadtime
/gate/digitizer/Singles/deadtime/setDeadTime 3000000. ps
/gate/digitizer/Singles/deadtime/setMode paralysable
/gate/digitizer/Singles/deadtime/chooseDTVolume block

```

```

#####
#   C O I N C I D E N C E   S O R T E R
#####

```

```

#/gate/digitizer/Coincidences/setWindow 10. ns
#/gate/digitizer/Coincidences/MultiplesPolicy takeWinnerOfGoods

```

```

#/gate/digitizer/name delay
#/gate/digitizer/insert coincidenceSorter
#/gate/digitizer/delay/setWindow 10. ns
#/gate/digitizer/delay/setOffset 500. ns
#/gate/digitizer/delay/MultiplesPolicy takeWinnerOfGoods

```

```

/gate/digitizer/Coincidences/setWindow 10. ns
/gate/digitizer/Coincidences/setOffset 0. ns
/gate/digitizer/Coincidences/describe

```

```

/gate/digitizer/name delay
/gate/digitizer/insert coincidenceSorter
/gate/digitizer/delay/setWindow 10. ns
/gate/digitizer/delay/setOffset 500. ns
/gate/digitizer/delay/describe

```

```

/gate/digitizer/name finalCoinc
/gate/digitizer/insert coincidenceChain
/gate/digitizer/finalCoinc/addInputName delay
/gate/digitizer/finalCoinc/addInputName Coincidences
/gate/digitizer/finalCoinc/usePriority true
/gate/digitizer/finalCoinc/describe

```

```

#-----ooooooooOOOOO00000OOOOOooooo-----#
#
#           D E F I N I T I O N   O F
#           Y O U R   O U T P U T   F I L E
#
#-----ooooooooOOOOO00000OOOOOooooo-----#

```

```

# R A N D O M

```

```

# JamesRandom Ranlux64 MersenneTwister
/gate/random/setEngineName JamesRandom
#/gate/random/setEngineSeed auto
#/gate/random/setEngineSeed 123456789
/gate/random/setEngineSeed default
/gate/random/verbose 1

```

```

#####
#
#           R O O T   O U T P U T
#
#####

```

```

#/gate/output/root/enable
#/gate/output/root/setFileName          Output/PET
#/gate/output/root/setRootSinglesAdderFlag  1
#/gate/output/root/setRootSinglesReadoutFlag 1
#/gate/output/root/setRootHitFlag          1
#/gate/output/root/setRootSinglesFlag      1
#/gate/output/root/setRootCoincidencesFlag 1

#####
#
#   SINOGRAM OUTPUT
#
#####
/gate/output/sinogram/enable
/gate/output/sinogram/setFileName          Output/PET
/gate/output/sinogram/setTangCrystalBlurring 1.8 mm
/gate/output/sinogram/setAxialCrystalBlurring 1.8 mm
/gate/output/sinogram/verbose              2
/gate/output/sinogram/RawOutputEnable
/gate/output/sinogram/RadialBins           256
/gate/output/sinogram/TruesOnly            true
/gate/output/sinogram/StoreDelayedS
/gate/output/sinogram/StoreScatters
/gate/output/sinogram/setInputDataName     finalCoinc
/gate/output/sinogram/describe

#####
#
#   ECAT7 OUTPUT
#
#####

/gate/output/ecat7/enable
/gate/output/ecat7/verbose                  2
/gate/output/ecat7/setFileName              Output/PET
/gate/output/ecat7/describe
/gate/output/ecat7/mashing                  2
/gate/output/ecat7/span                     9
/gate/output/ecat7/maxringdiff              22
/gate/output/ecat7/system                   962
/gate/output/ecat7/IsotopeCode              F-18
/gate/output/ecat7/IsotopeHalfLife          6586.2 second
/gate/output/ecat7/IsotopeBranchingFraction 1.0

#-----oOOOOO00000O0000Oo-----#
#
#   DEFINITION OF
#   YOUR VERBOSITY LEVEL
#
#-----oOOOOO00000O0000Oo-----#

/gate/verbose Physic    5
/gate/verbose Cuts      0
/gate/verbose SD        0
/gate/verbose Actions   0
/gate/verbose Actor     0
/gate/verbose Step      0
/gate/verbose Error     0
/gate/verbose Warning   0
/gate/verbose Output    0

```

```

/gate/verbose Beam      0
/gate/verbose Volume   0
/gate/verbose Image    0
/gate/verbose Geometry 0
/gate/verbose Core     0

/run/verbose           0
/event/verbose         0
/tracking/verbose     0

#-----oooooooooooooooooooooooo-----#
#
#       DEFINITION OF
#       YOUR SOURCES
#
#-----oooooooooooooooooooooooo-----#

#=====
# VOXELIZED SOURCE
#=====

/gate/source/addSource voxel_heart voxel

/gate/source/voxel_heart/reader/insert          interfile
/gate/source/voxel_heart/interfileReader/translator/insert      range
/gate/source/voxel_heart/interfileReader/rangeTranslator/readTable activity_range_mash_heart.dat
/gate/source/voxel_heart/interfileReader/rangeTranslator/describe 1
/gate/source/voxel_heart/interfileReader/readFile                Heart_study.h33
/gate/source/voxel_heart/setPosition                  -285.8 -155.8 -55.6 mm

/gate/source/voxel_heart/setType                    backtoback
/gate/source/voxel_heart/gps/particle                gamma
/gate/source/voxel_heart/setForcedUnstableFlag       true
/gate/source/voxel_heart/setForcedHalfLife           6586.2 s
/gate/source/voxel_heart/gps/energytype              Mono
/gate/source/voxel_heart/gps/monoenergy              0.511 MeV
/gate/source/voxel_heart/gps/confine                 NULL
/gate/source/voxel_heart/gps/angtype                 iso
/gate/source/voxel_heart/dump                        1

/gate/source/list

#-----oooooooooooooooooooooooo-----#
#
#       START ACQUISITION
#       TIME PARAMETERS
#
#-----oooooooooooooooooooooooo-----#

/gate/application/setTimeSlice      228. s
/gate/application/setTimeStart      0. s
/gate/application/setTimeStop       228. s

#
# LET'S RUN THE SIMULATION!
#
/gate/application/startDAQ

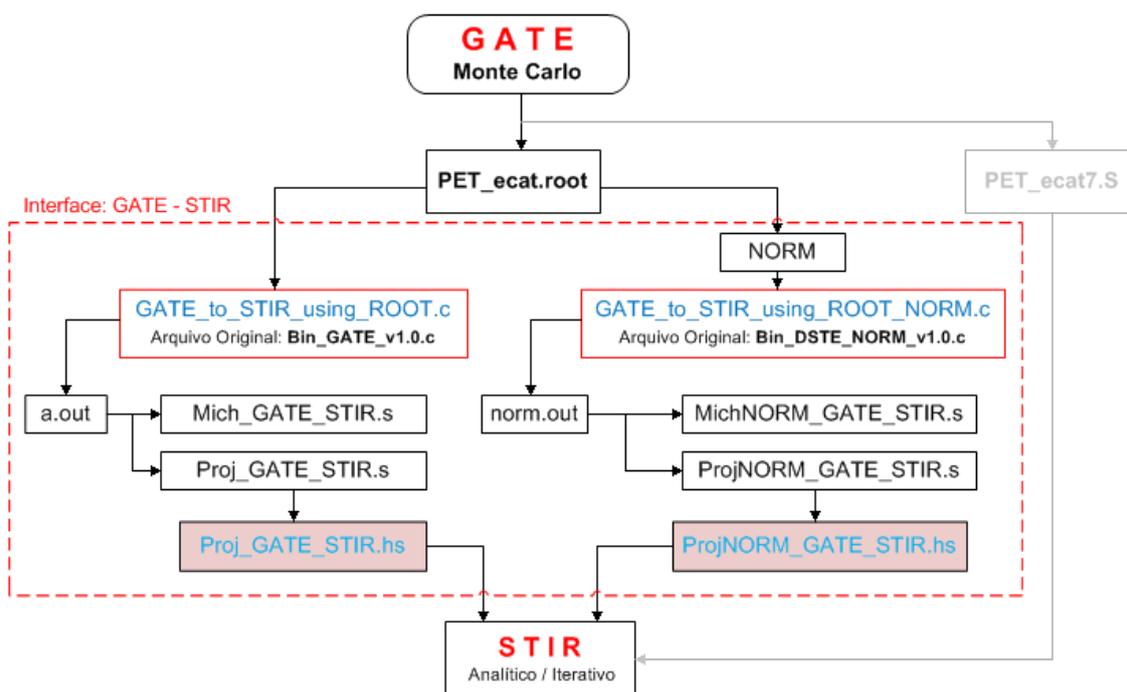
```

# ANEXO H

## INTERFACE GATE / STIR

Uma vez obtidos os arquivos de simulação resultantes através da plataforma GATE, e antes de proceder com a execução da reconstrução das projeções obtidas, é preciso realizar um passo intermediário que serve como interface entre o arquivo de saída ROOT (\*.root) gerado pelo GATE e o arquivo de entrada (\*.hs) que é solicitado pela plataforma STIR para realizar a reconstrução da região avaliada ou do objeto analisado.

No diagrama da Figura 54 é mostrado a sequência de passos e o fluxo dos caminhos que foram adotados desde a saída do GATE até a entrada do STIR com o objetivo de ter todas as condições para realizar a reconstrução dos volumes analisados através do STIR.



**FIGURA 54** – Fluxo que exemplifica a interface entre a plataforma de simulação GATE e a plataforma de reconstrução STIR (demarcado pelas linhas vermelhas descontinuas).

Como se pode apreciar no diagrama anterior, existem dois algoritmos de interface entre o GATE e o STIR que foram modificados e/ou adaptados para serem utilizados no nosso projeto: (a) GATE\_to\_STIR\_using\_ROOT.c e (b) GATE\_to\_STIR\_using\_ROOT\_NORM.c, as versões originais dos algoritmos foram cedidas pelo Sr. C. Ross Schmidlein (autor original), através de uma

solicitação direta que fiz com ele por meio de e-mail dando a conhecer o nosso projeto e os objetivos do mesmo.

O arquivo **Bin\_Gate\_v1.0.c**, está também disponível para download na página principal do projeto GATE (<http://www.opengatecollaboration.org/STIR/>).

Tal como foi indicado, estes dois algoritmos têm sido modificados parcialmente de acordo às necessidades do nosso projeto. Assim, foram alterados principalmente os parâmetros da arquitetura do Tomógrafo Siemens Biograph-TM que estamos utilizando e o qual tem sido modelado na plataforma GATE como apresentado na secção 3.3.1, assim como também outros parâmetros específicos que fazem parte do nosso projeto têm sido considerados no processo de adequação destes dois algoritmos.

Na sequência são apresentados os dois algoritmos de Interface que temos adequado e utilizado no projeto. Ambos geram como saída dois tipos de arquivos: Michelograma (**Mich\_GATE\_STIR.s**) e Projeções (**Proj\_GATE\_STIR.s**).

Cabe indicar que a diferença principal entre os dois algoritmos de interface utilizados é o processo de normalização que é feita só no segundo deles. Este procedimento de normalização consiste em fazer uma redistribuição equitativa do número total de eventos contabilizados durante o processo de simulação no número total de detectores que o Tomógrafo PET simulado possui.

Algoritmos de interface: GATE – STIR

Algorithm N° 1: Bin_GATE_ECAT962_v2.c
---------------------------------------

```
#####  
## Authors : Sadek A. Nehmeh, CR Schmidlein  
##  
## Modified by: Edward Florez (with the Mr. Ross Schmidlein's permission)  
## We modified the code in order to represent the ECAT 962 architecture  
## Looking at the camera.mac file it looks like to have three objects:  
## - 0 x gantry  
## - 4 x 72 x block  
## - 8 x 8 x crystal  
## These parameters above modified the identification of the rings and  
## the crystals in the code.  
## Furthermore, the root file with its branches need to be changed.  
##  
## Program : Bin_GATE_ECAT962_v2.c (15-MAR-2015)  
##  
## Objective : To read the coincidences TTree from the .root file, and generates the  
## corresponding Michelogram and Projection files.  
##
```

```

## Input   : Monte Carlo data from GATE (PET_ecat.root)
##
## Output  : 1 Michelogram files according to various binning definitions
##          : 2 Projection files according to various binning definitions
##
## HOW TO COMPILE:
## 1) Compile using this command line in the terminal:
##    g++ (name of file) `root-config --cflags --libs` → a.out
##
## HOW TO RUN:
## 1) After compiling the code type the following command line:
##    ./a.out 'directory name of root files' 'output file name'
##
#####

#include <iostream>
#include <sstream>
#include <fstream>
#include <string>
#include <vector>
#include <math.h>
#include <stdlib.h>
#include <stdio.h>
#include "TROOT.h"
#include "TSystem.h"
#include "TChain.h"
#include "TH2D.h"
#include "TDirectory.h"
#include "TList.h"
#include "Rtypes.h"
#include "TChainElement.h"
#include "TTree.h"
#include "TFile.h"
#include "TStyle.h"
#include "TH2.h"
#include "TH2F.h"
#include "TCanvas.h"
#include "TRandom.h"

using namespace std ;

// ECAT 962 Parameters
#define N_RINGS    32 // ECAT: Number of rings
#define N_SEG      63 // ECAT: Number of segments [2*N_RINGS-1]--> 63
#define N_DET      576 // ECAT: Detectors per ring
#define S_WIDTH    256 // ECAT: Number of radial sinogram bins (288 -> 256). Must be: 256
#define N_CASSET  72 // ECAT: Number of resector: 0xGANTRY --> 72 cassettes
#define N_CASS_BUCK 3 // ECAT: Number of resector: GANTRY 72 ctes / 3 ctes per bucket = 24
modul
#define N_MODULES  24 // ECAT: Number of modules (3x4)
#define N_BLO_y    72 // ECAT: Number of tangential modules (transaxial) : BLOCK [0,(4x72)-1] in
Y-Z
#define N_BLO_z    4 // ECAT: Number of axial modules
#define N_CRYSTAL 64 // ECAT: Number of crystals (8x8): CRISTAL [0, 63] in Y-Z
#define N_CRY_y    8 // ECAT: Number of axial crystals (array 8x8)
#define N_CRY_z    8 // ECAT: Number of axial crystals (array 8x8)
#define MAX_D_RING 31 // ECAT: Maximum ring difference [N_RINGS-1] (31 -> 22). Acc.paper: 22
#define N_PLANES  1024 // ECAT: Total number of sinograms [N_RINGS^2]

```

```

#define FOV          155 // ECAT: Width of the FOV (mm) (Axial:155 mm / Transaxial: 583 mm)
#define SLICES_PER_FOV 63 // ECAT: Number of slices per FOV
#define USE_OFFSET    0 // ECAT: On/Off use of offset
#define OFFSET        499 // ECAT: Sets initial sinogram angle (499 -> 0)

unsigned short ans, ans1;
unsigned short ans_SC, ans_RC, ans_SRC, ans_S, ans_R;
/*
  Current data constructions available for output.
  These are: Michelograms: [ring1][ring2][phi][u]

  total counts, trues, scatter, and randoms can be independently collected
*/
Float_t Mich_r1r2fu[N_RINGS][N_RINGS][N_DET/2][S_WIDTH]={0};

int main(int argc, char** argv)
{
  //-----
  // the first argument (argv[1]) is the sub-directory of the input file
  // the second argument (argv[2]) is the name of the output file
  //-----
  if(argc<2) {
    std::cout<<" Right number of input argument please !! "<<std::endl ;
    return 1;
  }

  Double_t PI = acos(-1.0);

  string filedir, inputfilename ;
  string filename, Moutputfilename, Poutputfilename ;

  filedir = argv[1] ;
  inputfilename = filedir + "*.root" ;

  cout << "Input file name is " << inputfilename << endl;
  TChain *Coincidences = new TChain("Coincidences");
  Coincidences->Add(inputfilename.c_str());

  filename = argv[2] ;
  Moutputfilename = "Mich_" + filename + ".s" ;
  cout << "Michelogram file name is " << Moutputfilename << endl ;
  Poutputfilename = "Proj_" + filename + ".s" ;
  cout << "Projection file name is " << Poutputfilename << endl ;

  FILE *Mich_r1r2fuFile, *Proj_File;
  Mich_r1r2fuFile = fopen(Moutputfilename.c_str(),"wb");
  Proj_File      = fopen(Poutputfilename.c_str(),"wb");

  //#####
  //#      Loop over the .root file in the directory "PATH"      #
  //#####
  Int_t Trues = 0, Scatters = 0, Randoms = 0;
  Int_t nbytes = 0;

  //#####
  //#      Declaration of leaves types - TTree Coincidences      #
  //#####

```

```

Float_t    axialPos, rotationAngle, sinogramS, sinogramTheta;
Char_t     comptVolName1[40], comptVolName2[40];
Int_t      compton1, compton2;
Int_t      runID, sourceID1, sourceID2, eventID1, eventID2;
//Int_t    layerID1, layerID2, crystalID1, crystalID2;
//Int_t    submoduleID1, submoduleID2, moduleID1, moduleID2, rsectorID1, rsectorID2;
//////////
// Added by EFlorez
Char_t     RayleighVolName1[40], RayleighVolName2[40];
Int_t      RayleighCrystal1, RayleighCrystal2, RayleighPhantom1, RayleighPhantom2;
Int_t      crystalID1, crystalID2;
Int_t      blockID1, blockID2, gantryID1, gantryID2;
//////////
Int_t      comptonPhantom1, comptonPhantom2;
Float_t    energy1, energy2;
Float_t    globalPosX1, globalPosX2, globalPosY1, globalPosY2, globalPosZ1, globalPosZ2;
Float_t    sourcePosX1, sourcePosX2, sourcePosY1, sourcePosY2, sourcePosZ1, sourcePosZ2;
Double_t   time1, time2;

#####
//#           Set branch addresses - TTree Coincidences           #
#####

Coincidences->SetBranchStatus("*",0);
Coincidences->SetBranchAddress("axialPos",&axialPos);
Coincidences->SetBranchAddress("comptVolName1",&comptVolName1);
Coincidences->SetBranchAddress("comptVolName2",&comptVolName2);
Coincidences->SetBranchAddress("comptonCrystal1",&compton1);
Coincidences->SetBranchAddress("comptonCrystal2",&compton2);
Coincidences->SetBranchAddress("crystalID1",&crystalID1);
Coincidences->SetBranchAddress("crystalID2",&crystalID2);
Coincidences->SetBranchAddress("comptonPhantom1",&comptonPhantom1);
Coincidences->SetBranchAddress("comptonPhantom2",&comptonPhantom2);
Coincidences->SetBranchAddress("energy1",&energy1);
Coincidences->SetBranchAddress("energy2",&energy2);
Coincidences->SetBranchAddress("eventID1",&eventID1);
Coincidences->SetBranchAddress("eventID2",&eventID2);
Coincidences->SetBranchAddress("globalPosX1",&globalPosX1);
Coincidences->SetBranchAddress("globalPosX2",&globalPosX2);
Coincidences->SetBranchAddress("globalPosY1",&globalPosY1);
Coincidences->SetBranchAddress("globalPosY2",&globalPosY2);
Coincidences->SetBranchAddress("globalPosZ1",&globalPosZ1);
Coincidences->SetBranchAddress("globalPosZ2",&globalPosZ2);
//Coincidences->SetBranchAddress("layerID1",&layerID1); Do not exist
//Coincidences->SetBranchAddress("layerID2",&layerID2); Do not exist
//////////
//Added by EFlorez
Coincidences->SetBranchAddress("RayleighPhantom1",&RayleighPhantom1);
Coincidences->SetBranchAddress("RayleighPhantom2",&RayleighPhantom2);
Coincidences->SetBranchAddress("RayleighCrystal1",&RayleighCrystal1);
Coincidences->SetBranchAddress("RayleighCrystal2",&RayleighCrystal2);
Coincidences->SetBranchAddress("RayleighVolName1",&RayleighVolName1);
Coincidences->SetBranchAddress("RayleighVolName2",&RayleighVolName2);
Coincidences->SetBranchAddress("blockID1",&blockID1);
Coincidences->SetBranchAddress("blockID2",&blockID2);
Coincidences->SetBranchAddress("gantryID1",&gantryID1);
Coincidences->SetBranchAddress("gantryID2",&gantryID2);
//////////

```

```

//Coincidences->SetBranchAddress("moduleID1",&moduleID1); To change -> blockID1
//Coincidences->SetBranchAddress("moduleID2",&moduleID2); To change -> blockID2
Coincidences->SetBranchAddress("rotationAngle",&rotationAngle);
//Coincidences->SetBranchAddress("rsectorID1",&rsectorID1); To change -> gantryID1
//Coincidences->SetBranchAddress("rsectorID2",&rsectorID2); To change -> gantryID2
Coincidences->SetBranchAddress("runID",&runID);
Coincidences->SetBranchAddress("sinogramS",&sinogramS);
Coincidences->SetBranchAddress("sinogramTheta",&sinogramTheta);
Coincidences->SetBranchAddress("sourceID1",&sourceID1);
Coincidences->SetBranchAddress("sourceID2",&sourceID2);
Coincidences->SetBranchAddress("sourcePosX1",&sourcePosX1);
Coincidences->SetBranchAddress("sourcePosX2",&sourcePosX2);
Coincidences->SetBranchAddress("sourcePosY1",&sourcePosY1);
Coincidences->SetBranchAddress("sourcePosY2",&sourcePosY2);
Coincidences->SetBranchAddress("sourcePosZ1",&sourcePosZ1);
Coincidences->SetBranchAddress("sourcePosZ2",&sourcePosZ2);
//Coincidences->SetBranchAddress("submoduleID1",&submoduleID1); Do not exist
//Coincidences->SetBranchAddress("submoduleID2",&submoduleID2); Do not exist
Coincidences->SetBranchAddress("time1",&time1);
Coincidences->SetBranchAddress("time2",&time2);

Int_t nentries = (Int_t)(Coincidences->GetEntries());

#####
##          SINOGRAMS AND PROJECTION PLANES BINNING          #
#####
Int_t  ring1, ring2, crystal1, crystal2;
Int_t  phi, u;
Int_t  Counts = 0;
int flip, swap, zi, c1, c2;

printf("Total Number of Coincidence Events:= %d \n",nentries );

for (Int_t i = 0 ; i < nentries ; i++)
{

    if ((i%250000) == 0 && i!=0) printf("... %d ",i);
    if ((i%1000000) == 0 && i!=0) printf("... %d\n",i);

    nbytes += Coincidences->GetEntry(i);

    // Update the number of Trues and Randoms...
    //-----

    if (eventID1 == eventID2)
    {
        if (comptonPhantom1 == 0 && comptonPhantom2 == 0) Trues++;
        else Scatters++;
    }
    else Randoms++;

    //-----
    // Identify the ring#...
    //-----
    //ring1 = (Int_t)(crystalID1/N_CRY_xy)
    //+ (Int_t)(submoduleID1/N_SMOD_xy)*N_CRY_z
    //+ (Int_t)(moduleID1/N_MOD_xy)*N_SMOD_z*N_CRY_z;
    //ring2 = (Int_t)(crystalID2/N_CRY_xy)
    //+ (Int_t)(submoduleID2/N_SMOD_xy)*N_CRY_z

```

```

//+ (Int_t)(moduleID2/N_MOD_xy)*N_SMOD_z*N_CRY_z;

//ring1 = (Int_t)(crystalID1/N_CRY_xy)
//+ ((Int_t)(blockID1/N_BLO_xy) * N_CRY_xy);
//ring2 = (Int_t)(crystalID2/N_CRY_xy)
//+ ((Int_t)(blockID2/N_BLO_xy) * N_CRY_xy);

// Modified by EFlorez (according the scanner architecture)
ring1 = (Int_t)((blockID1/N_BLO_y) * N_CRY_z)
+ (Int_t)(crystalID1/N_CRY_z);

ring2 = (Int_t)((blockID2/N_BLO_y) * N_CRY_z)
+ (Int_t)(crystalID2/N_CRY_z);

if ( abs(ring1 - ring2) > MAX_D_RING ) continue;

//-----
// Identify the crystal#...
//-----
//crystal1 = rsectorID1 * N_MOD_xy * N_SMOD_xy * N_CRY_xy
//+ (moduleID1%N_MOD_xy) * N_SMOD_xy * N_CRY_xy
//+ (submoduleID1%N_SMOD_xy) * N_CRY_xy
//+ (crystalID1%N_CRY_xy);
//crystal2 = rsectorID2 * N_MOD_xy * N_SMOD_xy * N_CRY_xy
//+ (moduleID2%N_MOD_xy) * N_SMOD_xy * N_CRY_xy
//+ (submoduleID2%N_SMOD_xy) * N_CRY_xy
//+ (crystalID2%N_CRY_xy);

//crystal1 = (gantryID1/N_RSEC * N_BLO_xy * N_CRY_xy)
//+ ((blockID1 % N_BLO_xy) * N_CRY_xy)
//+ (crystalID1 % N_CRY_xy);
//crystal2 = (gantryID2/N_RSEC * N_BLO_xy * N_CRY_xy)
//+ ((blockID2 % N_BLO_xy) * N_CRY_xy)
//+ (crystalID2 % N_CRY_xy);

// Modified by EFlorez (according the scanner architecture)
crystal1 = ((blockID1 % N_BLO_y) * N_CRY_y)
+ (crystalID1 % N_CRY_y);

crystal2 = ((blockID2 % N_BLO_y) * N_CRY_y)
+ (crystalID2 % N_CRY_y);

//-----
// Rotate the image correctly#...
//-----
if (USE_OFFSET == 1)
{
crystal1 = crystal1 + OFFSET;
crystal2 = crystal2 + OFFSET;
if (crystal1 >= N_DET) crystal1 = crystal1 - N_DET;
if (crystal2 >= N_DET) crystal2 = crystal2 - N_DET;
}

//-----
// Bin the crystal ring pairs into Michelograms
// u - radial sinogram component
// phi - azimuthal sinogram component
// ring pairs are sorted according to c1 < c2 else flip

```

```

// where c1 and c2 are crystals at phi(u = S_WIDTH/2)
//-----
phi = ((crystal1 + crystal2 + N_DET/2)%N_DET)/2;

if (((crystal1 + crystal2) < (3*N_DET/2)) && ((crystal1 + crystal2) >= (N_DET/2)))
u = abs(crystal1 - crystal2) - N_DET/2 + S_WIDTH/2;
else u = -abs(crystal1 - crystal2) + N_DET/2 + S_WIDTH/2;

if ( u >= S_WIDTH || u < 0 ) continue;

if (u%2 == 0)
{
zi = (N_DET/2 - (crystal1 - crystal2) - 1)/2;
if (zi >= N_DET/4) zi = zi - N_DET/2 + 1;
if (zi <= -N_DET/4) zi = zi + N_DET/2 - 1;
}
else
{
zi = (N_DET/2 - (crystal1 - crystal2))/2;
if (zi >= N_DET/4) zi = zi - N_DET/2;
if (zi <= -N_DET/4) zi = zi + N_DET/2;
}

c1 = crystal1 + zi;
c2 = crystal2 - zi;
if (c1 >= N_DET) c1 = c1 - N_DET;
if (c1 < 0) c1 = c1 + N_DET;
if (c2 >= N_DET) c2 = c2 - N_DET;
if (c2 < 0) c2 = c2 + N_DET;

if (c1 < c2) flip = 0;
else flip = 1;

if (flip)
{
swap = ring1;
ring1 = ring2;
ring2 = swap;
}

// Update the different arrays...
//-----
/**ALL EVENTS
Mich_r1r2fu[ring2][ring1][phi][u] += 1.;

if (eventID1 == eventID2)
{
/**true+scatter
if (comptonPhantom1 == 0 && comptonPhantom1 == 0)
{
/**true
}
}
else
{
/**scatter
}
}
else
{

```

```

    /***random
    }
    Counts++;

}
printf("\n");

// Write the data to disk, and then close Michelogram file...
//-----
ans = fwrite(Mich_r1r2fu,4,(N_RINGS*N_RINGS*N_DET/2*S_WIDTH),Mich_r1r2fuFile);
fclose(Mich_r1r2fuFile);

// Generate projection files...
//-----
// From Segment number -MAX_D_RING to +MAX_D_RING
// After phi
// After z
// After r
Int_t S_NUM;
for (int i = 0 ; i < 2*MAX_D_RING + 1 ; i++)
{
    if (i <= MAX_D_RING) S_NUM = N_RINGS - MAX_D_RING + i;
    else                S_NUM = N_RINGS + MAX_D_RING - i;

    for (int j = 0 ; j < N_DET/2 ; j++)
    {
        float Proj[S_NUM][S_WIDTH];
        for (int k = 0 ; k < S_NUM ; k++)
        {
            if (i <= MAX_D_RING) ring1 = k;
            else                ring2 = k;

            if (i <= MAX_D_RING) ring2 = ring1 + MAX_D_RING - i;
            else                ring1 = ring2 - MAX_D_RING + i;

            for (int l = 0 ; l < S_WIDTH ; l++)
                Proj[k][l] = Mich_r1r2fu[ring2][ring1][j][l];
        }
        ans = fwrite(Proj,4,(S_NUM*S_WIDTH),Proj_File);
    }
}

fclose(Proj_File);

return(0);
}

```

O procedimento para gerar os arquivos contendo as projeções ou michelograma é o seguinte:

```

[eflorez@localhost BinningCode]$ g++ Bin_GATE_EC962_v2.c `root-config --cflags --
libs`
[eflorez@localhost BinningCode]$ ./a.out /home/eflorez/PETnorma/BinningCode/ EC962
Input file name is /home/eflorez/PETnorma/BinningCode/*.*root
Michelogram file name is = Mich_EC962.s

```

Projection file name is = Proj\_ECAT962.s  
 Total Number of Coincidence Events:= 9996629  
 ... 250000 ... 500000 ... 750000 ... 1000000 ... 1000000  
 ... 1250000 ... 1500000 ... 1750000 ... 2000000 ... 2000000  
 ... 2250000 ... 2500000 ... 2750000 ... 3000000 ... 3000000  
 ... 3250000 ... 3500000 ... 3750000 ... 4000000 ... 4000000  
 ... 4250000 ... 4500000 ... 4750000 ... 5000000 ... 5000000  
 ... 5250000 ... 5500000 ... 5750000 ... 6000000 ... 6000000  
 ... 6250000 ... 6500000 ... 6750000 ... 7000000 ... 7000000  
 ... 7250000 ... 7500000 ... 7750000 ... 8000000 ... 8000000  
 ... 8250000 ... 8500000 ... 8750000 ... 9000000 ... 9000000  
 ... 9250000 ... 9500000 ... 9750000

Algorithm N° 2: GATE_to_STIR_using_ROOT_NORM.c
--

```
#####
## Authors   : Sadek A. Nehmeh, CR Schmidlein
##
## Modified by: Edward Florez (with the Mr. Ross Schmidlein's permission)
##           We modified the code in order to represent the ECAT 962 architecture
##           Looking at the camera.mac file it looks like to have 3 objects:
##           - 0 x gantry
##           - 4 x 72 x block
##           - 8 x 8 x crystal
##           These parameters above modified the identification of the rings and
##           the crystals in the code.
##           Furthermore, the root file with its branches need to be changed.
##
## Program   : Bin_ECAT962_NORM_v2.c 10-MAY-2015
##
## Objective : To read the coincidences TTree from the .root file, and generates the
##             corresponding Michelogram and Projection files.
##
## Input     : Monte Carlo data from GATE (PET_ecat.root)
##
## Output    : 1 Michelogram files according to various binning definitions
##            : 2 Projection files according to various binning definitions
##
## HOW TO COMPILE:
## 1) Compile using this command line in the terminal:
##    g++ (name of file) `root-config --cflags --libs` → a.out
##
## HOW TO RUN:
## 1) After compiling the code type the following command line:
##    ./a.out 'directory name of root files' 'output file name'
#####

#include <iostream>
#include <sstream>
#include <fstream>
#include <string>
#include <vector>
#include <math.h>
```

```

#include <stdlib.h>
#include <stdio.h>
#include "TROOT.h"
#include "TSystem.h"
#include "TChain.h"
#include "TH2D.h"
#include "TDirectory.h"
#include "TList.h"
#include "Rtypes.h"
#include "TChainElement.h"
#include "TTree.h"
#include "TFile.h"
#include "TStyle.h"
#include "TH2.h"
#include "TH2F.h"
#include "TCanvas.h"
#include "TRandom.h"

using namespace std ;

// ECAT 962 Parameters
#define N_RINGS      32 // ECAT: Number of rings
#define N_SEG        63 // ECAT: Number of segments [2*N_RINGS-1]
#define N_DET        576 // ECAT: Detectors per ring
#define S_WIDTH      256 // ECAT: Number of radial sinogram bins (288 -> 256). Must be: 256
#define N_CASSET     72 // ECAT: Number of resector: 0xGANTRY --> 72 cassettes
#define N_CASS_BUCK  3 // ECAT: Number of resector: GANTRY> 72 cts/3 cts per bucket =
24modules
#define N_MODULES    24 // ECAT: Number of modules (3x4)
#define N_BLO_y      72 // ECAT: Number of tangential modules (transaxial): BLOCK [0,(4x72)-1] in
YZ
#define N_BLO_z      4 // ECAT: Number of axial modules
#define N_CRYSTAL    64 // ECAT: Number of crystals (8x8): CRISTAL [0, 63] in Y-Z
#define N_CRY_y      8 // ECAT: Number of tangential crystals for avoid issues with non-sym blocks
#define N_CRY_z      8 // ECAT: Number of axial crystals (array 8x8) --> note in GATE its "z" not
"y"
#define MAX_D_RING  31 // ECAT: Maximum ring difference [N_RINGS-1] (31 -> 22). Acc. Paper:
22
#define N_PLANES     1024 // ECAT: Total number of sinograms [N_RINGS^2]
#define FOV          155 // ECAT: Width of the FOV (mm) (Axial:155 mm / Transaxial: 583 mm)
#define SLICES_PER_FOV 63 // ECAT: Number of slices per FOV
#define USE_OFFSET   0 // ECAT: On/Off use of offset
#define OFFSET       499 // ECAT: Sets initial sinogram angle (499 -> 0)
#define USE_FWHM     0 //

unsigned short ans,ans1;
unsigned short ans_SC, ans_RC, ans_SRC, ans_S, ans_R;
/*
Current data constructions available for output.
These are: Michelograms: [ring1][ring2][phi][u]

total counts, trues, scatter, and randoms can be independently collected
*/
Float_t Mich_r1r2fu[N_RINGS][N_RINGS][N_DET/2][S_WIDTH]={0};
//Float_t NORM_SRC[MAX_D_RING+1][N_DET/70][S_WIDTH]={0};
// Modified by EFlorez
Float_t NORM_SRC[MAX_D_RING+1][N_DET/72][S_WIDTH]={0};

int main(int argc, char** argv)

```

```

{
//-----
// the first argument (argv[1]) is the sub-directory of the input file
// the second argument (argv[2]) is the name of the output file
//-----
if(argc<2) {
    std::cout<<" Right number of input argument please !! "<<std::endl ;
    return 1;
}

Double_t PI = acos(-1.0);

// Normalization Constants and Blur Variables
Double_t FWHM_theta      = 0.33; // Full Width Half Max of crystal blurring
Double_t FWHM_z          = 0.33; // Full Width Half Max of crystal blurring
Double_t sigma_theta     = FWHM_theta/(2.0*sqrt(2.0*log(2.0)));
Double_t sigma_z         = FWHM_z/(2.0*sqrt(2.0*log(2.0)));
Double_t C_theta         = 0.405;
Double_t C_z             = 0.439;

//Float_t Ring_dia = 889.4;
//Modified by EFlorez
Float_t Ring_dia = 824.0;
Float_t Norm_dia = 800.0;
Float_t cord[S_WIDTH]={0},min_cord = 10000000;
Float_t ring_diff[N_RINGS];

for (Int_t i = 0; i < S_WIDTH ; i++)
{
    Int_t j = i - (S_WIDTH-1)/2; //Be careful w/this formula because S_WIDTH is even (256)
    //Int_t j = i - (S_WIDTH)/2;
    cord[i] = 2.0*pow(pow(Norm_dia/2.0,2)-
        pow(Ring_dia/2.0*sin(PI*(j+0.5)/(Float_t)N_DET),2),0.5);
    if (min_cord > cord[i]) min_cord = cord[i];
}

for (Int_t i = 0; i < S_WIDTH ; i++)
    cord[i] = cord[i]/min_cord;

for (Int_t i = 0; i < N_RINGS ; i++)
    ring_diff[i] = (Float_t)i*10*C_z*Norm_dia/Ring_dia/min_cord;

string filedir, inputfilename ;
string filename, Moutputfilename, Poutputfilename ;

filedir = argv[1] ;
inputfilename = filedir + "/*.root" ;

cout << "Input file name is " << inputfilename << endl;
TChain *Coincidences = new TChain("Coincidences") ;
Coincidences->Add(inputfilename.c_str()) ;

filename = argv[2] ;
Moutputfilename = "Mich_norm_" + filename + ".s" ;
cout << "Michelogram file name is " << Moutputfilename << endl ;
Poutputfilename = "Proj_norm_" + filename + ".s" ;
cout << "Projection file name is " << Poutputfilename << endl ;

FILE *Mich_r1r2fuFile, *Proj_File;

```

```

Mich_r1r2fuFile = fopen(Moutputfilename.c_str(),"wb");
Proj_File      = fopen(Poutputfilename.c_str(),"wb");

#####
//#      Loop over the .root file in the directory "PATH"  #
#####
Int_t  Trues = 0, Scatters = 0, Randoms = 0;
Int_t  nbytes = 0;

#####
//#      Declaration of leaves types - TTree Coincidences  #
#####

Float_t  axialPos, rotationAngle, sinogramS, sinogramTheta;
Char_t   comptVolName1[40], comptVolName2[40];
Int_t    compton1, compton2;
Int_t    runID, sourceID1, sourceID2, eventID1, eventID2;
//Int_t  layerID1, layerID2, crystalID1, crystalID2;
//Int_t  submoduleID1, submoduleID2, moduleID1, moduleID2, rsectorID1, rsectorID2;
////////////////////////////////////
// Added by EFlorez
Char_t   RayleighVolName1[40], RayleighVolName2[40];
Int_t    RayleighCrystal1, RayleighCrystal2, RayleighPhantom1, RayleighPhantom2;
Int_t    crystalID1, crystalID2;
Int_t    blockID1, blockID2, gantryID1, gantryID2;
////////////////////////////////////
Int_t    comptonPhantom1, comptonPhantom2;
Float_t  energy1, energy2;
Float_t  globalPosX1, globalPosX2, globalPosY1, globalPosY2, globalPosZ1, globalPosZ2;
Float_t  sourcePosX1, sourcePosX2, sourcePosY1, sourcePosY2, sourcePosZ1, sourcePosZ2;
Double_t  time1, time2;

#####
//#      Set branch addresses - TTree Coincidences  #
#####

Coincidences->SetBranchStatus("*",0);
Coincidences->SetBranchAddress("axialPos",&axialPos);
Coincidences->SetBranchAddress("comptVolName1",&comptVolName1);
Coincidences->SetBranchAddress("comptVolName2",&comptVolName2);
Coincidences->SetBranchAddress("comptonCrystal1",&compton1);
Coincidences->SetBranchAddress("comptonCrystal2",&compton2);
Coincidences->SetBranchAddress("crystalID1",&crystalID1);
Coincidences->SetBranchAddress("crystalID2",&crystalID2);
Coincidences->SetBranchAddress("comptonPhantom1",&comptonPhantom1);
Coincidences->SetBranchAddress("comptonPhantom2",&comptonPhantom2);
Coincidences->SetBranchAddress("energy1",&energy1);
Coincidences->SetBranchAddress("energy2",&energy2);
Coincidences->SetBranchAddress("eventID1",&eventID1);
Coincidences->SetBranchAddress("eventID2",&eventID2);
Coincidences->SetBranchAddress("globalPosX1",&globalPosX1);
Coincidences->SetBranchAddress("globalPosX2",&globalPosX2);
Coincidences->SetBranchAddress("globalPosY1",&globalPosY1);
Coincidences->SetBranchAddress("globalPosY2",&globalPosY2);
Coincidences->SetBranchAddress("globalPosZ1",&globalPosZ1);
Coincidences->SetBranchAddress("globalPosZ2",&globalPosZ2);
//Coincidences->SetBranchAddress("layerID1",&layerID1); Do not exist
//Coincidences->SetBranchAddress("layerID2",&layerID2); Do not exist

```

```

////////////////////////////////////
//Added by EFlorez
Coincidences->SetBranchAddress("RayleighPhantom1",&RayleighPhantom1);
Coincidences->SetBranchAddress("RayleighPhantom2",&RayleighPhantom2);
Coincidences->SetBranchAddress("RayleighCrystal1",&RayleighCrystal1);
Coincidences->SetBranchAddress("RayleighCrystal2",&RayleighCrystal2);
Coincidences->SetBranchAddress("RayleighVolName1",&RayleighVolName1);
Coincidences->SetBranchAddress("RayleighVolName2",&RayleighVolName2);
Coincidences->SetBranchAddress("blockID1",&blockID1);
Coincidences->SetBranchAddress("blockID2",&blockID2);
Coincidences->SetBranchAddress("gantryID1",&gantryID1);
Coincidences->SetBranchAddress("gantryID2",&gantryID2);
////////////////////////////////////
//Coincidences->SetBranchAddress("moduleID1",&moduleID1); To change -> blockID1
//Coincidences->SetBranchAddress("moduleID2",&moduleID2); To change -> blockID2
Coincidences->SetBranchAddress("rotationAngle",&rotationAngle);
//Coincidences->SetBranchAddress("rsectorID1",&rsectorID1); To change -> gantryID1
//Coincidences->SetBranchAddress("rsectorID2",&rsectorID2); To change -> gantryID2
Coincidences->SetBranchAddress("runID",&runID);
Coincidences->SetBranchAddress("sinogramS",&sinogramS);
Coincidences->SetBranchAddress("sinogramTheta",&sinogramTheta);
Coincidences->SetBranchAddress("sourceID1",&sourceID1);
Coincidences->SetBranchAddress("sourceID2",&sourceID2);
Coincidences->SetBranchAddress("sourcePosX1",&sourcePosX1);
Coincidences->SetBranchAddress("sourcePosX2",&sourcePosX2);
Coincidences->SetBranchAddress("sourcePosY1",&sourcePosY1);
Coincidences->SetBranchAddress("sourcePosY2",&sourcePosY2);
Coincidences->SetBranchAddress("sourcePosZ1",&sourcePosZ1);
Coincidences->SetBranchAddress("sourcePosZ2",&sourcePosZ2);
//Coincidences->SetBranchAddress("submoduleID1",&submoduleID1); Do not exist
//Coincidences->SetBranchAddress("submoduleID2",&submoduleID2); Do not exist
Coincidences->SetBranchAddress("time1",&time1);
Coincidences->SetBranchAddress("time2",&time2);

Int_t nentries = (Int_t)(Coincidences->GetEntries());

#####
//#          SINOGRAMS AND PROJECTION PLANES BINNING          #
#####
Int_t  ring1, ring2, crystal1, crystal2;
Int_t  phi, u;
Int_t  Counts = 0, Counts_SRC = 0, Counts_total = 0;
int flip, swap, zi, c1, c2;
Double_t posXYZ1, posXYZ2;
printf("Total Number of Coincidence Events:= %d \n",nentries );

for (Int_t i = 0 ; i < nentries ; i++)
{
    if ((i%250000) == 0 && i!=0) printf("... %d ",i);
    if ((i%1000000) == 0 && i!=0) printf("... %d\n",i);

    nbytes += Coincidences->GetEntry(i);

    // Update the number of Trues and Randoms...
    //-----
    posXYZ1 = sourcePosX1*sourcePosX1 + sourcePosY1*sourcePosY1 + sourcePosZ1*sourcePosZ1;
    posXYZ1 = pow(posXYZ1,0.5);
    posXYZ2 = sourcePosX2*sourcePosX2 + sourcePosY2*sourcePosY2 + sourcePosZ2*sourcePosZ2;

```

```

posXYZ2 = pow(posXYZ2,0.5);
if (posXYZ1 > Norm_dia/2. || posXYZ2 > Norm_dia/2.)
continue;

if (abs(sinogramS) > Norm_dia/2)
continue;

if ((eventID1 == eventID2) && (comptonPhantom1 == 0 && comptonPhantom2 == 0))
{
//-----
// Identify the ring#...
//-----
//ring1 = (Int_t)(crystalID1/N_CRY_xy)
//+ (Int_t)(submoduleID1/N_SMOD_xy)*N_CRY_z
//+ (Int_t)(moduleID1/N_MOD_xy)*N_SMOD_z*N_CRY_z;
//ring2 = (Int_t)(crystalID2/N_CRY_xy)
//+ (Int_t)(submoduleID2/N_SMOD_xy)*N_CRY_z
//+ (Int_t)(moduleID2/N_MOD_xy)*N_SMOD_z*N_CRY_z;

// Modified by EFlorez (according the scanner architecture)
ring1 = (Int_t)((blockID1/N_BLO_y) * N_CRY_z)
+ (Int_t)(crystalID1/N_CRY_z);

ring2 = (Int_t)((blockID2/N_BLO_y) * N_CRY_z)
+ (Int_t)(crystalID2/N_CRY_z);

if ( abs(ring1 - ring2) > MAX_D_RING ) continue;
//-----
// Identify the crystal#...
//-----
//crystal1 = rsectorID1 * N_MOD_xy * N_SMOD_xy * N_CRY_xy
//+ (moduleID1%N_MOD_xy) * N_SMOD_xy * N_CRY_xy
//+ (submoduleID1%N_SMOD_xy) * N_CRY_xy
//+ (crystalID1%N_CRY_xy);
//crystal2 = rsectorID2 * N_MOD_xy * N_SMOD_xy * N_CRY_xy
//+ (moduleID2%N_MOD_xy) * N_SMOD_xy * N_CRY_xy
//+ (submoduleID2%N_SMOD_xy) * N_CRY_xy
//+ (crystalID2%N_CRY_xy);

// Modified by EFlorez (according the scanner architecture)

crystal1 = ((blockID1 % N_BLO_y) * N_CRY_y)
+ (crystalID1 % N_CRY_y);

crystal2 = ((blockID2 % N_BLO_y) * N_CRY_y)
+ (crystalID2 % N_CRY_y);

//-----
// Adding crystal blurring for light smearing#...
//-----
/* Crystal Blurring
1) generate random number with a Gaussian distribution
2) determine distance to new location in crystal numbers
Delta_crystal = sign(RNDM) * (2 * | RNDM/Crystal_width | + 1)/2
note integer truncation
3) if shifted ensure valid crystal number and still in the block
and within the GE light Shielded area within a block
4) if not go back to step 1

```

```

5) repeat process for second crystal and in the axial direction
*/
if (USE_FWHM == 1)
{
  Int_t flag;
  do
  {
    flag = 0;
    Double_t rand_theta1 = gRandom->Gaus(0.0, sigma_theta);
    if (rand_theta1 != 0)
    {
      Int_t delta_theta1 = (Int_t)((rand_theta1/fabs(rand_theta1))
        *(2*fabs(rand_theta1/C_theta)+1)/2);
      if (delta_theta1 != 0)
      {
        Int_t crystal1_tmp = crystal1 + delta_theta1;
        if (crystal1_tmp < N_DET && crystal1_tmp >= 0)
        {
          //if (crystal1_tmp/N_CRY_xy == crystal1/N_CRY_xy)
          //Modified by EFlorez
          if (crystal1_tmp/N_CRY_y == crystal1/N_CRY_y)
          {
            //if ((crystal1%N_CRY_xy != 0 || crystal1%N_CRY_xy != N_CRY_xy-1)
            //  && (crystal1_tmp%N_CRY_xy != 0 || crystal1_tmp%N_CRY_xy != N_CRY_xy-1))
            if ((crystal1%N_CRY_y != 0 || crystal1%N_CRY_y != N_CRY_y-1)
              && (crystal1_tmp%N_CRY_y != 0 || crystal1_tmp%N_CRY_y != N_CRY_y-1))
              crystal1 = crystal1_tmp;
            else flag = 1;
          }
          else flag = 1;
        }
        else flag = 1;
      }
      else flag = 1;
    }
  }while (flag != 0);
  do
  {
    flag = 0;
    Double_t rand_theta2 = gRandom->Gaus(0.0, sigma_theta);
    if (rand_theta2 != 0)
    {
      Int_t delta_theta2 = (Int_t)((rand_theta2/fabs(rand_theta2))
        *(2*fabs(rand_theta2/C_theta)+1)/2);
      if (delta_theta2 != 0)
      {
        Int_t crystal2_tmp = crystal2 + delta_theta2;
        if (crystal2_tmp < N_DET && crystal2_tmp >= 0)
        {
          //if (crystal2_tmp/N_CRY_xy == crystal2/N_CRY_xy)
          //Modified by EFlorez
          if (crystal2_tmp/N_CRY_y == crystal2/N_CRY_y)
          {
            //if ((crystal2%N_CRY_xy != 0 || crystal2%N_CRY_xy != N_CRY_xy-1)
            //  && (crystal2_tmp%N_CRY_xy != 0 || crystal2_tmp%N_CRY_xy != N_CRY_xy-1))
            if ((crystal2%N_CRY_y != 0 || crystal2%N_CRY_y != N_CRY_y-1)
              && (crystal2_tmp%N_CRY_y != 0 || crystal2_tmp%N_CRY_y != N_CRY_y-1))
              crystal2 = crystal2_tmp;
            else flag = 1;
          }
        }
      }
    }
  }
}

```

```

        else flag = 1;
    }
    else flag = 1;
}
}
}while (flag != 0);
do
{
flag = 0;
Double_t rand_z1 = gRandom->Gaus(0.0, sigma_z);
if (rand_z1 != 0)
{
    Int_t delta_z1 = (Int_t)((rand_z1/fabs(rand_z1))
        *(2*fabs(rand_z1/C_z)+1)/2);
    if (delta_z1 != 0)
    {
        Int_t ring1_tmp = ring1 + delta_z1;
        if (ring1_tmp < N_RINGS && ring1_tmp >= 0)
        {
            if (ring1_tmp/N_CRY_z == ring1/N_CRY_z)
            {
                if ((ring1%N_CRY_z != 0 || ring1%N_CRY_z != N_CRY_z-1)
                    && (ring1_tmp%N_CRY_z != 0 || ring1_tmp%N_CRY_z != N_CRY_z-1))
                    ring1 = ring1_tmp;
                else flag = 1;
            }
            else flag = 1;
        }
        else flag = 1;
    }
}
}while (flag != 0);
do
{
flag = 0;
Double_t rand_z2 = gRandom->Gaus(0.0, sigma_z);
if (rand_z2 != 0)
{
    Int_t delta_z2 = (Int_t)((rand_z2/fabs(rand_z2))
        *(2*fabs(rand_z2/C_z)+1)/2);
    if (delta_z2 != 0)
    {
        Int_t ring2_tmp = ring2 + delta_z2;
        if (ring2_tmp < N_RINGS && ring2_tmp >= 0)
        {
            if (ring2_tmp/N_CRY_z == ring2/N_CRY_z)
            {
                if ((ring2%N_CRY_z != 0 || ring2%N_CRY_z != N_CRY_z-1)
                    && (ring2_tmp%N_CRY_z != 0 || ring2_tmp%N_CRY_z != N_CRY_z-1))
                    ring2 = ring2_tmp;
                else flag = 1;
            }
            else flag = 1;
        }
        else flag = 1;
    }
}
}while (flag != 0);
}
}

```

```

//-----
// Rotate the image correctly#...
//-----
if (USE_OFFSET == 1)
{
    crystal1 = crystal1 + OFFSET;
    crystal2 = crystal2 + OFFSET;
    if (crystal1 >= N_DET) crystal1 = crystal1 - N_DET;
    if (crystal2 >= N_DET) crystal2 = crystal2 - N_DET;
}
//-----
// Bin the crystal ring pairs into Michelograms
// u - radial sinogram component
// phi - azimuthal sinogram component
// ring pairs are sorted according to c1 < c2 else flip
// where c1 and c2 are crystals at phi(u = S_WIDTH/2)
//-----
phi = ((crystal1 + crystal2 + N_DET/2)%N_DET)/2;

if (((crystal1 + crystal2) < (3*N_DET/2)) && ((crystal1 + crystal2) >= (N_DET/2)))
    u = abs(crystal1 - crystal2) - N_DET/2 + S_WIDTH/2;
else u = -abs(crystal1 - crystal2) + N_DET/2 + S_WIDTH/2;

if ( u >= S_WIDTH || u < 0 ) continue;

if (u%2 == 0)
{
    zi = (N_DET/2 - (crystal1 - crystal2) - 1)/2;
    if (zi >= N_DET/4) zi = zi - N_DET/2 + 1;
    if (zi <= -N_DET/4) zi = zi + N_DET/2 - 1;
}
else
{
    zi = (N_DET/2 - (crystal1 - crystal2))/2;
    if (zi >= N_DET/4) zi = zi - N_DET/2;
    if (zi <= -N_DET/4) zi = zi + N_DET/2;
}

c1 = crystal1 + zi;
c2 = crystal2 - zi;
if (c1 >= N_DET) c1 = c1 - N_DET;
if (c1 < 0) c1 = c1 + N_DET;
if (c2 >= N_DET) c2 = c2 - N_DET;
if (c2 < 0) c2 = c2 + N_DET;

if (c1 < c2) flip = 0;
else flip = 1;

if (flip)
{
    swap = ring1;
    ring1 = ring2;
    ring2 = swap;
}

##### NEW
// Bins the seperate symetries
// (|dr|[N_RINGS-1],phi[N_DET/N_MOD],u[(S_WIDTH-1)/2])

```

```

// because counts are reused every reuse requires a rescaling
// factor of 1/2 and the number of sinograms per segment
Int_t theta = ring1 - ring2;
//Int_t phi_lim = phi%N_CRY_xy;
//Modified by EFlorez
Int_t phi_lim = phi%N_CRY_y;
Float_t corr;
corr = pow(ring_diff[abs(theta)],2) + pow(cord[u],2);
corr = cord[(S_WIDTH-1)/2]/pow(corr,0.5);
// printf("%d %d %f %f %f \n", theta, u, ring_diff[abs(theta)], cord[u], corr);
if (ring1 == ring2)
    NORM_SRC[abs(theta)][phi_lim][u] += 2*corr;
else
{
    NORM_SRC[abs(theta)][phi_lim][u]
    += 0.5*N_RINGS*corr / (N_RINGS - abs(theta));
    NORM_SRC[abs(theta)][phi_lim][S_WIDTH-1-u]
    += 0.5*N_RINGS*corr / (N_RINGS - abs(theta));
}
Counts_SRC++;
// ##### Binned in (|dr|,phi,u) #####
}
}
printf(" : Counts kept = %d\n",Counts_SRC);
Counts_total = Counts_total + Counts_SRC;
Counts_SRC = 0;

printf("\n");
printf("Total Counts kept = %d\n",Counts_total);

Double_t Counts_SINO = 0;
for (int i = 0 ; i < N_RINGS; i++)
for (int j = 0 ; j < N_RINGS; j++)
{
    if (abs(i-j) > MAX_D_RING) continue;
    for (int k = 0 ; k < N_DET/2; k++)
    for (int l = 0 ; l < S_WIDTH; l++)
        //Counts_SINO = Counts_SINO + (Double_t)(NORM_SRC[abs(i-j)][k%N_CRY_xy][l]);
        //Modified by EFlorez
        Counts_SINO = Counts_SINO + (Double_t)(NORM_SRC[abs(i-j)][k%N_CRY_y][l]);
}
printf("Total Counts reported = %14.4f\n",Counts_SINO);
printf("Average Counts per LOR = %14.4f\n",Counts_SINO/(N_DET/2*S_WIDTH*N_PLANES));

// Data Inversion
// 0) Find path length corrections
// 1) Find min. counts != 0
// 2) Assign minimum counts to zero count areas
// 3) Find average counts
// 4) Invert counts

Float_t Min_Val = 1000000;
Float_t Max_Val = 0;
Float_t Avg_Val = 0;
Int_t Scounts = 0, n_zero = 0;
for (int i = 0 ; i <= MAX_D_RING; i++)
//for (int k = 0 ; k < N_CRY_xy; k++)
//Modified by EFlorez
for (int k = 0 ; k < N_CRY_y; k++)

```

```

    for (int l = 0 ; l < S_WIDTH; l++)
    {
        if (NORM_SRC[i][k][l] > 0)
        {
            if (NORM_SRC[i][k][l] < Min_Val) Min_Val = NORM_SRC[i][k][l];
            if (NORM_SRC[i][k][l] > Max_Val) Max_Val = NORM_SRC[i][k][l];
            n_zero++;
        }
        Counts++;
    }

for (int i = 0 ; i <= MAX_D_RING ; i++)
//for (int k = 0 ; k < N_CRY_xy; k++)
//Modified by EFlorez
for (int k = 0 ; k < N_CRY_y; k++)
for (int l = 0 ; l < S_WIDTH; l++)
{
    if (NORM_SRC[i][k][l] == 0) NORM_SRC[i][k][l] = Min_Val;
    Avg_Val += NORM_SRC[i][k][l];
}
Avg_Val = Avg_Val/(Float_t)Counts;
printf("Average number of counts (Number of zeros %d)= %f\n",Counts-n_zero,Avg_Val);
printf("Maximum and Minimum number of counts = (%f,%f)\n",Max_Val,Min_Val);

for (int i = 0 ; i <= MAX_D_RING ; i++)
//for (int k = 0 ; k < N_CRY_xy; k++)
//Modified by EFlorez
for (int k = 0 ; k < N_CRY_y; k++)
for (int l = 0 ; l < S_WIDTH; l++)
NORM_SRC[i][k][l] = Avg_Val/NORM_SRC[i][k][l];

for (int i = 0 ; i < N_RINGS ; i++)
for (int j = 0 ; j < N_RINGS ; j++)
for (int k = 0 ; k < N_DET/2; k++)
for (int l = 0 ; l < S_WIDTH; l++)
//Mich_r1r2fu[i][j][k][l] = NORM_SRC[abs(i-j)][k%N_CRY_xy][l];
//Modified by EFlorez
Mich_r1r2fu[i][j][k][l] = NORM_SRC[abs(i-j)][k%N_CRY_y][l];

// Write the data to disk, and then close Michelogram file...
//-----
ans = fwrite(Mich_r1r2fu,4,(N_RINGS*N_RINGS*N_DET/2*S_WIDTH),Mich_r1r2fuFile);
fclose(Mich_r1r2fuFile);

// Generate projection files...
//-----
// From Segment number -MAX_D_RING to +MAX_D_RING
// After phi
// After z
// After r
Int_t S_NUM;
for (int i = 0 ; i < 2*MAX_D_RING + 1 ; i++)
{
    if (i <= MAX_D_RING) S_NUM = N_RINGS - MAX_D_RING + i;
    else S_NUM = N_RINGS + MAX_D_RING - i;

    for (int j = 0 ; j < N_DET/2 ; j++)
    {
        float Proj[S_NUM][S_WIDTH];

```

```

for (int k = 0 ; k < S_NUM ; k++)
{
  if (i <= MAX_D_RING) ring1 = k;
  else          ring2 = k;

  if (i <= MAX_D_RING) ring2 = ring1 + MAX_D_RING - i;
  else          ring1 = ring2 - MAX_D_RING + i;

  for (int l = 0 ; l < S_WIDTH ; l++)
  Proj[k][l] = Mich_r1r2fu[ring2][ring1][j][l];
}
ans = fwrite(Proj,4,(S_NUM*S_WIDTH),Proj_File);
}
}

fclose(Proj_File);

//-----

return(0);
}

```

O procedimento para gerar os arquivos com as projeções ou michelograma do processo de normalização é como segue:

```
[eflorez@localhost Normalization]$ g++ Bin_ECAT962_NORM_v2.c `root-config --cflags --libs`
```

```
[eflorez@localhost Normalization]$ ./a.out
```

```
/home/eflorez/PETnorma/BinningCode/Normalization/ ECAT962
```

```
Input file name is /home/eflorez/PETnorma/BinningCode/Normalization/*.*root
```

```
Michelogram file name is = Mich_norm_ECAT962.s
```

```
Projection file name is = Proj_norm_ECAT962.s
```

```
Total Number of Coincidence Events:= 9996629
```

```
... 250000 ... 500000 ... 750000 ... 1000000 ... 1000000
```

```
... 1250000 ... 1500000 ... 1750000 ... 2000000 ... 2000000
```

```
... 2250000 ... 2500000 ... 2750000 ... 3000000 ... 3000000
```

```
... 3250000 ... 3500000 ... 3750000 ... 4000000 ... 4000000
```

```
... 4250000 ... 4500000 ... 4750000 ... 5000000 ... 5000000
```

```
... 5250000 ... 5500000 ... 5750000 ... 6000000 ... 6000000
```

```
... 6250000 ... 6500000 ... 6750000 ... 7000000 ... 7000000
```

```
... 7250000 ... 7500000 ... 7750000 ... 8000000 ... 8000000
```

```
... 8250000 ... 8500000 ... 8750000 ... 9000000 ... 9000000
```

```
... 9250000 ... 9500000 ... 9750000 : Counts kept = 6063532
```

```
Total Counts kept = 6063532
```

```
Total Counts reported = 15256854555.0697
```

```
Average Counts per LOR = 202.0843
```

```
Average number of counts (Number of zeros 0)= 199.030853
```

```
Maximum and Minimum number of counts = (365.859070,31.999098)
```

```
[eflorez@localhost Normalization]$
```

```

!INTERFILE :=
!imaging modality := PT
name of data file := Proj_ECAT962.s
originating system := ECAT 962
!GENERAL DATA :=
!GENERAL IMAGE DATA :=
!type of data := PET
patient orientation := head_in
patient rotation := supine
imagedata byte order := LITTLEENDIAN
!PET STUDY (General) :=
!PET data type := Emission
applied corrections := {None}
!number format := float
!number of bytes per pixel := 4
number of dimensions := 4
matrix axis label [4] := segment
!matrix size [4] := 5
matrix axis label [3] := view
!matrix size [3] := 144
matrix axis label [2] := axial coordinate
!matrix size [2] := { 35,53,63,53,35}
matrix axis label [1] := tangential coordinate
!matrix size [1] := 256
minimum ring difference per segment := { -22,-13,-4,5,14}
maximum ring difference per segment := { -14,-5,4,13,22}
Scanner parameters:=
Scanner type := ECAT 962
Number of rings := 32
Number of detectors per ring := 576
Inner ring diameter (cm) := 82.4
Average depth of interaction (cm) := 0.7
Distance between rings (cm) := 0.485
Default bin size (cm) := 0.225
View offset (degrees) := 0
Maximum number of non-arc-corrected bins := 288
Default number of arc-corrected bins := 288
Number of blocks per bucket in transaxial direction := 3
Number of blocks per bucket in axial direction := 4
Number of crystals per block in axial direction := 8
Number of crystals per block in transaxial direction := 8
Number of detector layers := 1
Number of crystals per singles unit in axial direction := 8
Number of crystals per singles unit in transaxial direction := 24
end scanner parameters:=
effective central bin size (cm) := 0.228528
number of time frames := 1
image duration (sec)[1] := 120
image relative start time (sec)[1] := 0
!END OF INTERFILE :=

```

# APÊNDICE I

## STIR: COMANDOS DE RECONSTRUÇÃO 3D

Entre os pacotes que fazem parte da plataforma de reconstrução STIR, encontra-se uma série de comandos que permitem ao usuário exibir, manusear, converter e reconstruir um conjunto de imagens, projeções geradas pelo GATE.

A seguir são indicados cada um dos blocos de comandos e os parâmetros necessários para conseguir executá-los.

### (i) FBP2D

```
[eflorez@localhost FBP2D]$ ./FBP2D
fbp2dparameters :=
output filename prefix := FBP2D
output file format type := Interfile
byte order := Littleendian
number format := float
number of bytes per pixel := 4
scale to write data := 0
post-filter type := None
input file := PET.S
maximum absolute segment number to process := -1
zoom := 1
xy output image size (in pixels) := 180
z output image size (in pixels) := -1
z offset (in mm) := 0
num segments to combine with ssrb := 1
alpha parameter for ramp filter := 1
cut-off for ramp filter (in cycles) := 0.5
transaxial extension for fft := 1
display level := 0
back projector type := Interpolation
back projector using interpolation parameters :=
use piecewise linear interpolation := 1
use exact jacobian := 0
end back projector using interpolation parameters :=
end :=
INFO: Determined voxel size by dividing default_bin_size (2.25) by zoom
[eflorez@localhost FBP2D]$
```

### (ii) FBP3DRP

```
[eflorez@localhost FBP3DRP]$ ./FBP3DRP
fbp3dparameters :=
output filename prefix := FBP3D
output file format type := Interfile
byte order := LITTLEENDIAN
```

```

number format := float
number of bytes per pixel := 4
scale to write data := 0
post-filter type := None
input file := PET.S
maximum absolute segment number to process := -1
zoom := 1
xy output image size (in pixels) := 180
z output image size (in pixels) := -1
z offset (in mm) := 0
image to be used for reprojection :=
num segments to combine with ssrb := -1
alpha parameter for ramp filter := 1
cut-off for ramp filter (in cycles) := 0.5
transaxial extension for fft := 1
axial extension for fft := 1
alpha parameter for colsher filter in axial direction := 1
cut-off for colsher filter in axial direction (in cycles) := 0.5
alpha parameter for colsher filter in planar direction := 1
cut-off for colsher filter in planar direction (in cycles) := 0.5
stretch factor for colsher filter definition in axial direction := 2
stretch factor for colsher filter definition in planar direction := 2
back projector type := Interpolation
use piecewise linear interpolation := 1
use exact jacobian := 0
forward projector type :=
save intermediate images := 0
display level := 0
WARNING:
Reading frame 1, gate 1, data 0, bed 0 from file PET.S
INFO: Determined voxel size by dividing default_bin_size (2.25) by zoom
INFO: Determined voxel size by dividing default_bin_size (2.25) by zoom
[eflorez@localhost FBP3DRP]$

```

### (iii) OSMAPOSL

```

[eflorez@localhost OSMAPOLS]$ ./OSMAPOLS
OSMAPOSLParameters :=
objective function type:= PoissonLogLikelihoodWithLinearModelForMeanAndProjData
PoissonLogLikelihoodWithLinearModelForMeanAndProjData Parameters:=

input file := /home/eflorez/STIR/STIR_new/STIR-
bin/Release/src/iterative/OSMAPOSL/PET.S

; if disabled, defaults to maximum segment number in the file
maximum absolute segment number to process := -1
; see User's Guide to see when you need this
zero end planes of segment 0:= 1
; if next is set to 1, sensitivity will be recomputed
; and also written to file (if "sensitivity filename" or "subset sensitivity
filenames" is set)
; if it is zero, the sensitivity will be read from file

```

```

recompute sensitivity := 1
use subset sensitivities:=1 ; recommended
; specify additive projection data to handle randoms or so
; see User's Guide for more info
additive sinogram := 0

; choose size of pixels
; x,y pixel size = bin_size/zoom
zoom := 1
; number of x,y pixels (-1 takes default value to cover the FOV)
xy output image size (in pixels) := -1

projector pair type := Matrix
  Projector Pair Using Matrix Parameters :=
    Matrix type := Ray Tracing
    Ray tracing matrix parameters :=
      ; use multiple (almost) parallel LORs for every bin in the sinogram
      ; to avoid discretisation artefacts
      number of rays in tangential direction to trace for each bin:= 10
      ; you could disable some symmetries if you have enough memory
      ; this would for instance allow you to increase the number of subsets
      ; do_symmetry_90degrees_min_phi:=0
    End Ray tracing matrix parameters :=
  End Projector Pair Using Matrix Parameters :=

Bin Normalisation type := Chained
  Chained Bin Normalisation Parameters:=
    Bin Normalisation to apply first:= from ProjData
      Bin Normalisation From ProjData :=
        normalisation projdata filename:= /home/eflorez/STIR/STIR_new/STIR-
bin/Release/src/iterative/OSMAPOSL/final/Proj_norm_EC962_seg2.hs
      End Bin Normalisation From ProjData:=
    Bin Normalisation to apply second:= from Attenuation Image
      Bin Normalisation From Attenuation Image :=
        attenuation_image_filename := /home/eflorez/STIR/STIR_new/STIR-
bin/Release/src/iterative/OSMAPOSL/final/mu_Map_Heart_256.hv
      forward projector type := Matrix
      Forward Projector Using Matrix Parameters :=
        Matrix type := Ray Tracing
        Ray tracing matrix parameters :=
          number of rays in tangential direction to trace for each bin:=10
          End Ray tracing matrix parameters :=
        End Forward Projector Using Matrix Parameters :=
      End Bin Normalisation From Attenuation Image:=
    END Chained Bin Normalisation Parameters:=

end PoissonLogLikelihoodWithLinearModelForMeanAndProjData Parameters:=

initial estimate:= 1
;initial estimate:= some_image
output filename prefix := /home/eflorez/STIR/STIR_new/STIR-
bin/Release/src/iterative/OSMAPOSL/OSMAPOSL

```

```
; Number of subsets should be a divisor of num_views/4
number of subsets:= 12
; Use for starting the numbering from something else than 1
;start at subiteration number:=1
; Use if you want to start from another subset than 0 (but why?)
;start at subset:= 0
number of subiterations:= 144
save estimates at subiteration intervals:= 12

; initial estimate := some_image
; enable this when you read an initial estimate with negative data
enforce initial positivity condition:=0

inter-update filter subiteration interval:= 0
inter-update filter type := None

inter-iteration filter subiteration interval:= 0
inter-iteration filter type := None

post-filter type := None

END :=

INFO: Determined voxel size by dividing default_bin_size (2.25) by zoom
[eflorez@localhost OSMAPOLS]$
```

# APÊNDICE J

## ALGORITMO DE CORREÇÃO DE ATENUAÇÃO (PETac)

Este programa dimensiona uma imagem de CT e converte-o para um mapa de atenuação ( $\mu$  map).

Local de descarga: <https://code.google.com/p/petac/>

O PETac foi escrito pelo Prof. Robbie Barnett ([rbar9508@uni.sydney.edu.au](mailto:rbar9508@uni.sydney.edu.au)) e está composto pelos seguintes arquivos:

- a) CMakeLists.txt
- b) DicomSeriesReadImageWrite2.cxx
- c) petac.cxx
- d) petac.h
- e) scanner.txt
- f) readme.txt (contém os requisitos mínimos do processo de instalação)

Requisitos de instalação:

- Cmake 2.6 ou versão mais recente (<http://www.cmake.org>)
- ITK 3.16 ou versão mais recente (<http://www.itk.org>)

Nota importante: Tive alguns problemas no processo de instalação. Os problemas são devido a uma incompatibilidade entre o PETac e a versão 4.xx do ITK (última versão). Finalmente, eu consegui instalar o PETac usando o **CMake v.3.0.2** e o **ITK v.4.6.1**.

Estes problemas foram resolvidos principalmente com duas modificações no código do PETac que detalhamos a seguir:

1. Para alterar (em todos os arquivos): itkOrientedImage.h → **itkImage.h**

2. Para alterar no arquivo CMakeLists.txt:

```
target_link_libraries(myexecutable → target_link_libraries(myexecutable
  ITKCommon                               ${ITK_LIBRARIES})
  ITKAlgorithms
  ITKIO
)
```

Portanto, o arquivo **CMakeLists.txt** deve ser:

```
cmake_minimum_required (VERSION 2.6)

PROJECT(PETAC)
FIND_PACKAGE(ITK)
IF(ITK_FOUND)
INCLUDE(${ITK_USE_FILE})
ELSE(ITK_FOUND)
MESSAGE(FATAL_ERROR
"ITK not found. Please set ITK_DIR.")
ENDIF(ITK_FOUND)
ADD_EXECUTABLE(petac petac.cxx )
TARGET_LINK_LIBRARIES(petac ${ITK_LIBRARIES})

ADD_EXECUTABLE(DicomSeriesReadImageWrite2 DicomSeriesReadImageWrite2.cxx )
TARGET_LINK_LIBRARIES(DicomSeriesReadImageWrite2 ${ITK_LIBRARIES})
```

Para o PETac a sequência de passos são:

1. Aplicar um filtro gaussiano que seja igual ao tamanho do cristal
2. Reamostrar a imagem para corresponder às dimensões do PET
3. Aplicar escalamento bilinear
4. Girar o mapa de atenuação se necessário
5. Usar um cabeçalho para o arquivo Interfile, o qual é entendido pelo STIR

A definição do scanner PET é especificada no arquivo de texto scanner.txt. Os dados desse arquivo, segundo o scanner SIEMENS BiographTM que temos utilizado, contém sete linhas com os seguintes valores:

<Number of bins>	256
<Number of rings>	32
<Bin size (mm)>	2.25
<Distance between rings (mm)>	4.85
<Flip x axis (0 or 1)>	0
<Flip y axis (0 or 1)>	0
<Flip z axis (0 or 1)>	0

Muitas vezes, pode ser que o CT tenha um sistema de coordenadas diferente para a geometria do scanner. Se os pontos do mapa de atenuação ficaram na direção incorreta, pode-se aplicar inversão desses pontos ou a rotação de 180 graus. Tais modificações podem ser realizadas, alterando o arquivo da geometria do scanner indicado anteriormente. Aqui estão alguns exemplos possíveis de alterar o sistema de coordenadas utilizando os três últimos valores no arquivo de definição do scanner como segue:

No flipping:	Flip along scanner axis:	Rotate 180 degrees:
0	0	1
0	0	0
0	1	1

Adicionalmente, pode-se usar três argumentos adicionais se a imagem CT abrange mais do tamanho da câmara PET ou não está centrado no eixo do PET/CT com o intuito de se obter o mapa de atenuação na posição correta.

## PROCEDIMENTO DE USO

### Primeiro passo:

A imagem CT pode estar em vários formatos de arquivos que são suportados pelo ITK. Recomenda-se usar a imagem CT em formato DICOM. A maneira mais fácil de ler uma série DICOM é usar o programa **DicomSeriesReadImageWrite2** que foi fornecido como parte do pacote PETac.

Forma geral:

```
./DicomSeriesReadImageWrite2 DicomDirectory outputFileName [seriesName]
```

Exemplo de uso (no nosso caso):

```
[eflorez@localhost petac]$ ./DicomSeriesReadImageWrite2 ~/petac/mu_Map_Heart_288/
~/petac/mu_Map_Heart_288/AC_CT_Map_288.mhd
The directory:
/home/eflorez/petac/mu_Map_Heart_288/

Contains the following DICOM Series:
777.777.0.0.0.1428003178.3347468616.220986933.01.200000e00258478

Now reading series:
777.777.0.0.0.1428003178.3347468616.220986933.01.200000e00258478

Writing the image as
/home/eflorez/petac/mu_Map_Heart_288/AC_CT_Map_288.mhd
[eflorez@localhost petac]$
```

## **Segundo passo:**

O DicomSeriesReadImageWrite2 irá converter o CT em um único arquivo para que ele possa ser lido pelo PETac.

Forma geral:

```
./petac <scanner filename> <input filename> <output filename> [ <x-offset> <y-offset>  
<starting-slice> ]
```

Exemplo de uso (no nosso caso):

```
[eflorez@localhost petac]$ ./petac ~/petac/mu_Map_Heart_288/scanner.txt  
~/petac/mu_Map_Heart_288/AC_CT_Map_288.mhd  
~/petac/mu_Map_Heart_288/mu_Map_Heart_288  
Scanner definition:  
Number of bins: 288, Number of rings: 32  
Bin size: 2.25mm, Distance between rings: 4.85mm  
Flip axis: 0, 0, 0  
CT origin is 0mm, 0mm, 0mm  
CT size is 478, 258, 90  
CT spacing is 1.2mm, 1.2mm, 1.2mm  
MuMap X size too small: 254.933  
MuMap X size expanded to: 289  
MuMap Y size too small: 137.6  
MuMap Y size expanded to: 289  
MuMap origin is -38.325mm, -170.325mm, 0mm  
MuMap size is 289, 289, 63  
MuMap spacing: 2.25mm, 2.25mm, 2.425mm  
Writing the MuMap to file /home/eflorez/petac/mu_Map_Heart_288/mu_Map_Heart_288.mhd  
Writing the MuMap to interfile (scanner coordinates)  
/home/eflorez/petac/mu_Map_Heart_288/mu_Map_Heart_288.hv  
[eflorez@localhost petac]$
```

Finalmente, 3 arquivos foram gerados como parte do processo (mu\_Map\_Heart\_288.hv, mu\_Map\_Heart\_288.mhd e mu\_Map\_Heart\_288.raw). Os primeiros dois arquivos trabalham com cabeçalhos dos dados brutos que estão contidos no arquivo mu\_Map\_Heart\_288.raw. Este último será utilizado como o mapa de atenuação ( $\mu$  map) no processo de reconstrução 3D com correção de atenuação.

# APÊNDICE K

## NORMALIZAÇÃO

```
#-----oOoOoOoOoOoOoOoOoOoOo-----#
# Simulation PET_Siemens_BiographTM + Cylindrical Source (versao 5.15)
# AUTOR: Edward Flórez Pacheco
# DATE: 1/may/2015
# Department of Radiology
# University of Mississippi Medical Center (UMMC)
#
# OBJETIVO: Executar a adquisicao das projecoes de um phantom simples
#           (Cilindro) utilizando como radiofarmaco o FDG,
#           no tomografo simulado: Siemens - BiographTM (CMN).
#           - Adquisicao em um instante de tempo só!
#           - 100000 bq (emissão inicial)
#           - Phantom deve cobrir todo o diametro do scanner.
#           - O cilindro nao deve ter material associado (vacuum)
#-----oOoOoOoOoOoOoOoOoOoOo-----#
```

```
#=====
# VISUALISATION
#=====
```

```
/vis/open                                OGLSX
/vis/ogl/set/displayListLimit            10000000
/vis/viewer/set/viewpointThetaPhi        30 0
/vis/viewer/zoom                          6.0
/vis/drawVolume
/tracking/storeTrajectory                 1
/vis/scene/add/trajectories
/vis/scene/endOfEventAction               accumulate
```

```
/gate/verbose Physic      5
/gate/verbose Cuts        0
/gate/verbose SD          0
/gate/verbose Actions     0
/gate/verbose Actor       0
/gate/verbose Step        0
/gate/verbose Error       0
/gate/verbose Warning     0
/gate/verbose Output      0
/gate/verbose Beam        0
/gate/verbose Volume      0
/gate/verbose Image       0
/gate/verbose Geometry    0
/gate/verbose Core        0
/gate/verbose Geometry    0
/run/verbose               0
/event/verbose             0
/tracking/verbose         0
```

```
#=====
# GEOMETRY
#=====
```

```
/gate/geometry/setMaterialDatabase ../GateMaterials.db
```

```
# WORLD
/gate/world/geometry/setXLength 400. cm
```

```

/gate/world/geometry/setYLength 400. cm
/gate/world/geometry/setZLength 400. cm

#-----oooooooooooooooooooooooooooooooo-----#
#  DEFINITION AND DESCRIPTION
#    OF YOUR PET DEVICE
#-----oooooooooooooooooooooooooooooooo-----#

#  E C A T
/gate/world/daughters/name      ecat
/gate/world/daughters/insert    cylinder
/gate/ecat/setMaterial          Air
/gate/ecat/geometry/setRmax     44.2 cm
/gate/ecat/geometry/setRmin     41.2 cm
/gate/ecat/geometry/setHeight   15.52 cm
/gate/ecat/vis/forceWireframe

#  B L O C K
/gate/ecat/daughters/name      block
/gate/ecat/daughters/insert    box
/gate/block/placement/setTranslation 42.7 0.0 0.0 cm
/gate/block/geometry/setXLength 30.0 mm
/gate/block/geometry/setYLength 32.4 mm
/gate/block/geometry/setZLength 35.12 mm
/gate/block/setMaterial        Air
/gate/block/vis/forceWireframe

#  C R Y S T A L
/gate/block/daughters/name     crystal
/gate/block/daughters/insert   box
/gate/crystal/geometry/setXLength 30.0 mm
/gate/crystal/geometry/setYLength 4.05 mm
/gate/crystal/geometry/setZLength 4.39 mm
/gate/crystal/setMaterial      BGO
/gate/crystal/vis/setColor     red

#  R E P E A T  C R Y S T A L
/gate/crystal/repeaters/insert  cubicArray
/gate/crystal/cubicArray/setRepeatNumberX 1
/gate/crystal/cubicArray/setRepeatNumberY 8
/gate/crystal/cubicArray/setRepeatNumberZ 8
/gate/crystal/cubicArray/setRepeatVector 0. 4.4942 4.85 mm

#  R E P E A T  B L O C K
/gate/block/repeaters/insert    linear
/gate/block/linear/setRepeatNumber 4
/gate/block/linear/setRepeatVector 0. 0. 38.8 mm
/gate/block/repeaters/insert    ring
/gate/block/ring/setRepeatNumber 72

#  T U N G S T E N  S H I E L D
/gate/world/daughters/name     carter
/gate/world/daughters/insert   cylinder
/gate/carter/setMaterial       Air
/gate/carter/geometry/setRmax  44. cm
/gate/carter/geometry/setRmin  28. cm
/gate/carter/geometry/setHeight 9. cm
/gate/carter/placement/setTranslation 0.0 0.0 12.5 cm
/gate/carter/vis/forceWireframe

/gate/carter/daughters/name     carter1
/gate/carter/daughters/insert   cylinder

```

```

/gate/carter1/setMaterial      Tungsten
/gate/carter1/geometry/setRmax 30.0 cm
/gate/carter1/geometry/setRmin 29.0 cm
/gate/carter1/geometry/setHeight 3.0 cm
/gate/carter1/vis/setColor    white
/gate/carter1/placement/setTranslation 0.0 0.0 -3 cm

```

```

/gate/carter/daughters/name    carter2
/gate/carter/daughters/insert  cylinder
/gate/carter2/setMaterial      Tungsten
/gate/carter2/geometry/setRmax 40.7 cm
/gate/carter2/geometry/setRmin 30.0 cm
/gate/carter2/geometry/setHeight 1.0 cm
/gate/carter2/vis/setColor    white
/gate/carter2/placement/setTranslation 0.0 0.0 -2 cm

```

```

/gate/carter/daughters/name    carter3
/gate/carter/daughters/insert  cylinder
/gate/carter3/setMaterial      Tungsten
/gate/carter3/geometry/setRmax 40.7 cm
/gate/carter3/geometry/setRmin 39.7 cm
/gate/carter3/geometry/setHeight 3.0 cm
/gate/carter3/vis/setColor    white
/gate/carter3/placement/setTranslation 0.0 0.0 0 cm
/gate/carter/daughters/name    carter4
/gate/carter/daughters/insert  cylinder
/gate/carter4/setMaterial      Tungsten
/gate/carter4/geometry/setRmax 43.7 cm
/gate/carter4/geometry/setRmin 40.7 cm
/gate/carter4/geometry/setHeight 1.0 cm
/gate/carter4/vis/setColor    white
/gate/carter4/placement/setTranslation 0.0 0.0 1 cm

```

```

# ATTACH SYSTEM
/gate/systems/ecat/block/attach block
/gate/systems/ecat/crystal/attach crystal

```

```

# ATTACH CRYSTAL SD
/gate/crystal/attachCrystalSD

```

```

#-----ooooooooOOOOO00000OOOOOooooo-----#
# DEFINITION AND DESCRIPTION
# OF YOUR PHANTOM
#-----ooooooooOOOOO00000OOOOOooooo-----#

```

```

/gate/world/daughters/name    phantom
/gate/world/daughters/insert  cylinder
/gate/phantom/setMaterial      Vacuum
/gate/phantom/vis/forceSolid
/gate/phantom/vis/setColor    gray
/gate/phantom/geometry/setRmax 41.2 cm
/gate/phantom/geometry/setRmin 0.0 cm
/gate/phantom/geometry/setHeight 15.52 cm
/gate/phantom/placement/setTranslation 0.0 0.0 0.0 cm

```

```

/gate/phantom/attachPhantomSD

```

```

#=====
# PHYSICS
#=====
/gate/physics/addProcess PhotoElectric
/gate/physics/processes/PhotoElectric/setModel StandardModel

```

```

/gate/physics/addProcess Compton
/gate/physics/processes/Compton/setModel StandardModel

/gate/physics/addProcess RayleighScattering
/gate/physics/processes/RayleighScattering/setModel PenelopeModel

/gate/physics/addProcess ElectronIonisation
/gate/physics/processes/ElectronIonisation/setModel StandardModel e-
/gate/physics/processes/ElectronIonisation/setModel StandardModel e+

/gate/physics/addProcess Bremsstrahlung
/gate/physics/processes/Bremsstrahlung/setModel StandardModel e-
/gate/physics/processes/Bremsstrahlung/setModel StandardModel e+

/gate/physics/addProcess PositronAnnihilation

/gate/physics/addProcess MultipleScattering e+
/gate/physics/addProcess MultipleScattering e-

/gate/physics/processList Enabled
/gate/physics/processList Initialized

#=====
# CUTS
#=====

/gate/physics/Gamma/SetCutInRegion    crystal 1.0 cm
/gate/physics/Electron/SetCutInRegion crystal 1.0 cm
/gate/physics/Positron/SetCutInRegion  crystal 1.0 cm

/gate/physics/Gamma/SetCutInRegion    phantom 0.1 mm
/gate/physics/Electron/SetCutInRegion  phantom 0.1 mm
/gate/physics/Positron/SetCutInRegion   phantom 0.1 mm

/gate/physics/SetMaxStepSizeInRegion  phantom 0.01 mm

#=====
# INITIALISATION
#=====

/gate/run/initialize

#-----oooooooooooooooooooooooo-----#
# DEFINITION OF YOUR ACQUISITION
# DIGITIZER & COINCIDENCE SHORTER
#-----oooooooooooooooooooooooo-----#

#   A D D E R
/gate/digitizer/Singles/insert adder

#   R E A D O U T
/gate/digitizer/Singles/insert readout
/gate/digitizer/Singles/readout/setDepth 1

#   E N E R G Y   C R Y T A L   B L U R R I N G
/gate/digitizer/Singles/insert crystalblurring
/gate/digitizer/Singles/crystalblurring/setCrystalResolutionMin 0.1699
/gate/digitizer/Singles/crystalblurring/setCrystalResolutionMax 0.1701
/gate/digitizer/Singles/crystalblurring/setCrystalQE          1
/gate/digitizer/Singles/crystalblurring/setCrystalEnergyOfReference 511. keV

```

```

# ENERGY CUT
/gate/digitizer/Singles/insert thresholder
/gate/digitizer/Singles/thresholder/setThreshold 425. keV
/gate/digitizer/Singles/insert upholder
/gate/digitizer/Singles/upholder/setUphold 650. keV

## DEAD TIME
#/gate/digitizer/Singles/insert deadtime
#/gate/digitizer/Singles/deadtime/setDeadTime 3000000. ps
#/gate/digitizer/Singles/deadtime/setMode paralyzable
#/gate/digitizer/Singles/deadtime/chooseDTVVolume block

# COINC SORTER
/gate/digitizer/Coincidences/setInputName Singles
/gate/digitizer/Coincidences/setWindow 4.875 ns
/gate/digitizer/Coincidences/allPulseOpenCoincGate true
/gate/digitizer/Coincidences/MultiplesPolicy keepIfAllAreGoods
/gate/digitizer/Coincidences/minSectorDifference 15

#-----oooooooooooooooooooooooooooooooo-----#
# DEFINITION OF
# YOUR OUTPUT FILE
#-----oooooooooooooooooooooooooooooooo-----#

#####
# SINOGRAM OUTPUT
#####

/gate/output/sinogram/enable
/gate/output/sinogram/setFileName output/PET_Sinogram
/gate/output/sinogram/setTangCrystalBlurring 1.8 mm
/gate/output/sinogram/setAxialCrystalBlurring 1.8 mm
/gate/output/sinogram/verbose 2
/gate/output/sinogram/RawOutputEnable

/gate/output/sinogram/RadialBins 256
/gate/output/sinogram/TruesOnly true

/gate/output/sinogram/StoreDelayed
/gate/output/sinogram/StoreScatters
/gate/output/sinogram/setInputDataName finalCoinc
/gate/output/sinogram/describe

#####
# ECAT7 OUTPUT
#####

/gate/output/ecat7/enable
/gate/output/ecat7/verbose 2
/gate/output/ecat7/setFileName output/PET_ecat7
/gate/output/ecat7/describe
/gate/output/ecat7/mashing 2
/gate/output/ecat7/span 9
/gate/output/ecat7/maxringdiff 22
/gate/output/ecat7/system 962
/gate/output/ecat7/IsotopeCode F-18
/gate/output/ecat7/IsotopeHalfLife 6586.2 second
/gate/output/ecat7/IsotopeBranchingFraction 1.0

#####
# RANDOM

```

```

#####

/gate/random/setEngineName JamesRandom
/gate/random/setEngineSeed default
/gate/random/verbose 1

#-----oooooooooooooooooooooooooooooooo-----#
#           DEFINITION OF
#           YOUR VERBOSITY LEVEL
#-----oooooooooooooooooooooooooooooooo-----#

/gate/verbose Physic      5
/gate/verbose Cuts        0
/gate/verbose SD          0
/gate/verbose Actions     0
/gate/verbose Actor       0
/gate/verbose Step        0
/gate/verbose Error       0
/gate/verbose Warning    0
/gate/verbose Output      0
/gate/verbose Beam        0
/gate/verbose Volume      0
/gate/verbose Image       0
/gate/verbose Geometry    0
/gate/verbose Core        0
/run/verbose              0
/event/verbose            0
/tracking/verbose        0

#-----oooooooooooooooooooooooooooooooo-----#
#           DEFINITION OF
#           YOUR SOURCES
#-----oooooooooooooooooooooooooooooooo-----#
/gate/source/addSource      Norm
/gate/source/Norm/setActivity 1000000. becquerel
/gate/source/Norm/setType   backtoback
/gate/source/Norm/gps/particle gamma
/gate/source/Norm/gps/energytype Mono
/gate/source/Norm/gps/monoenergy 0.511 MeV
/gate/source/Norm/gps/type   Volume
/gate/source/Norm/gps/shape   Cylinder
/gate/source/Norm/gps/angtype iso
/gate/source/Norm/gps/radius  41.2 cm
/gate/source/Norm/gps/halfz   8.5 cm
/gate/source/Norm/gps/centre  0.0 0.0 0.0 cm
/gate/source/Norm/gps/confine NULL
/gate/source/Norm/dump
/gate/source/list

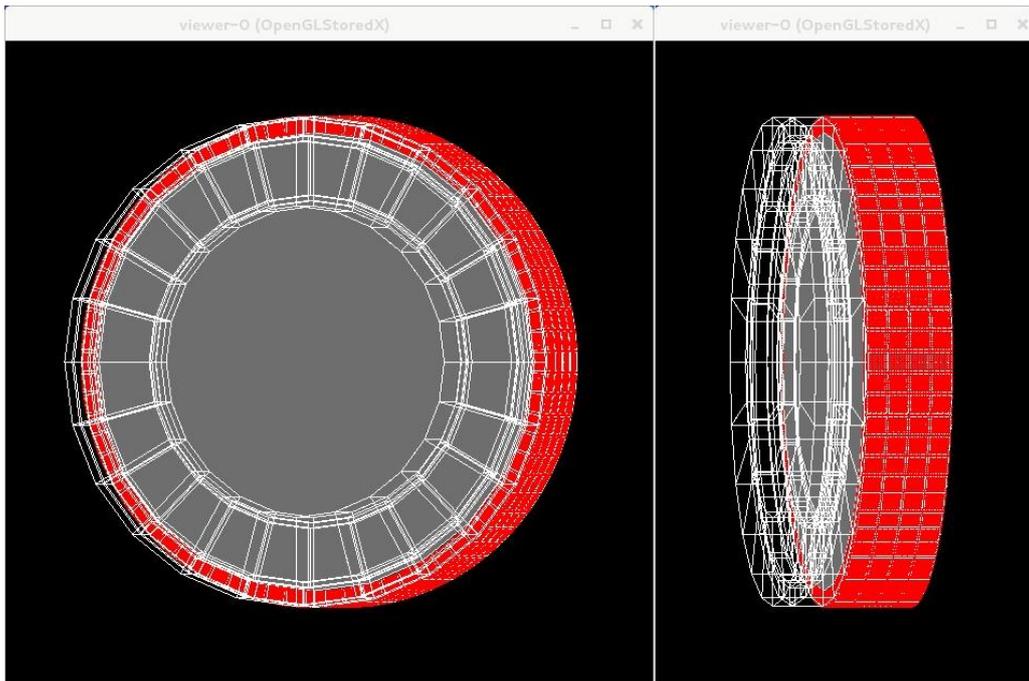
#-----oooooooooooooooooooooooooooooooo-----#
#           START ACQUISITION
#           TIME PARAMETERS
#-----oooooooooooooooooooooooooooooooo-----#

# EXPERIMENT
/gate/application/setTimeSlice 360. s
/gate/application/setTimeStart 0. s
/gate/application/setTimeStop 360. s

# LET'S RUN THE SIMULATION!
/gate/application/startDAQ

```

Na Figura 55, pode-se apreciar o posicionamento de um grande cilindro no interior do tomógrafo PET modelado. Esse elemento não tem nenhum elemento associado a ele, apenas vazio. O processo de aquisição foi de 360 segundos com uma alta emissão de fótons (1GBq). O arquivo resultante desta simulação foi incluído no processo de normalização na reconstrução PET 3D.



**FIGURA 55** – A normalização dos dados foi obtida através da simulação no GATE utilizando uma fonte cilíndrica (sem material de atenuação, apenas vazio) que cobre todo o diâmetro e a largura da câmara PET.

# APÊNDICE L

## COMKAT: MODELO PARA O ESTUDO DO METABOLISMO EM IMAGENS PET REAIS DE PERFUSÃO MIOCÁRDICA COM RB-82

Foi utilizada a versão 3.2 do COMKAT (<http://comkat.case.edu/>), um software de código aberto que oferece modelos existentes prontos para serem aplicados. Além disso, permite ao usuário criar modelos próprios com detalhes específicos para aplicações específicas tais como em nosso caso, onde foram considerados três compartimentos e 4 constantes paramétricas ( $K_1 - k_4$ ), a fim de estudar, analisar e estimar a taxa metabólica dos exames PET reais que fazem parte do estudo de Perfusão Miocárdica com  $^{82}\text{Rb}$ .

As linhas de comando estão listadas a seguir:

```
function [cm modelname]= modelRB82_USPUMMC()
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Myocardical Perfusion Study (USP): Compartments model(3) using Rb-82
%
% Compartmental Modeling Tool (version 14.1)
% AUTOR: Edward Flórez Pacheco
% MENTOR: Prof. Dr. Sergio Furuie
% SUPERVISOR: Dr. Vani Vijayakumar
%
% University of Mississippi Medical Center (UMMC)
% Department of Radiology
% Division of Nuclear Medicine
% Jackson, MS
%
% OBJETIVE: To model the human region by the use of compartments
%           (three compartments) in order to quantify the glucose
%           uptake of the element marker (Rb-84) in a Myocardical
%           Perfusion study (rest and stress).
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

cm = compartmentModel;
cm = addCompartment(cm, 'C1', '');
cm = addCompartment(cm, 'C2', '');
cm = addCompartment(cm, 'J', '');
% fengInput parameters:
a = [851.1 21.88 20.81];
lambda = [-4.134 -0.1191 -0.01043];
%"P"-type input
parmFeng = [2 0 a lambda];
cm = addInput(cm, 'Cp', 'sa', 'dk1', 'fengInput', parmFeng);
%"W"-type input
parmFeng = [2 0 a lambda];
cm = addInput(cm, 'Ca', 'sa', 'dk2', 'fengInput', parmFeng);
% connect compartments and inputs
cm = addLink(cm, 'L', 'Cp', 'C1', 'K1');
cm = addLink(cm, 'K', 'C1', 'J', 'k2');
cm = addLink(cm, 'K', 'C1', 'C2', 'k3');
cm = addLink(cm, 'K', 'C2', 'C1', 'k4');
```

```

% Provide the information of compartment positions to plot
cm = set(cm, 'CompartmentXData', [0.5600    0.6350;    0.8400    0.9150;
    0.5600    0.6350]);
cm = set(cm, 'CompartmentYData', [0.9200    0.8375;    0.9200    0.8375;
    0.7    0.6175]);
cm = set(cm, 'InputXData', [0.28    0.3550]);
cm = set(cm, 'InputYData', [0.8375    0.92]);
cm = set(cm, 'LinkXData', [0.355  0.56;    0.5975  0.5975; 0.6350  0.84;
    0.84  0.6350]);
cm = set(cm, 'LinkYData', [0.8788 0.8788; 0.8375 0.7;    0.8925
    0.8925; 0.865 0.865]);
cm = set(cm, 'LinkLabelPosition', [0.45 0.86; 0.585 0.7488; 0.73 0.88 ;
    0.73 0.835]);

% Default rate constants taken from Phelps M.E. (1979) Ann Neurol.
cm = addParameter(cm, 'K1', 0.1020); % ml/g/min
cm = addParameter(cm, 'k2', 0.1300); % 1/min
cm = addParameter(cm, 'k3', 0.0620); % 1/min
cm = addParameter(cm, 'k4', 0.0068); % 1/min
cm = addParameter(cm, 'sa', 1); % uci/pmol specific activity of
    injection
cm = addParameter(cm, 'dk1', log(2)/1.27); % radioactive decay const
    % for Rb-82 (76 sec)
cm = addParameter(cm, 'dk2', 0); % radioactive decay constant
    % for Rb-82
cm = addParameter(cm, 'PV', 1); % Partial vol
cm = addParameter(cm, 'Fv', 0.05); % Vascular fraction

Wlist = {'C1', 'PV';
    'C2', 'PV'};
Xlist = {'Ca', 'Fv'}; % Vascular contribution
cm = addOutput(cm, 'PET', Wlist, Xlist);

% define the scan frames (according to Nuclear Medicine - InCor)
ttt = [ones(9,1)*10/60; % 9 frames, 10 sec each
    ones(3,1)*30/60; % 3 frames, 30 sec each
    ones(1,1)*1; % 1 frames, 1 min each
    ones(2,1)*2; % 2 frames, 2 min each

% scant(:,1) frame start times; scant(:,2) frame end times
scant = [[0; cumsum(ttt(1:(length(ttt)-1)))] ...
    cumsum(ttt)];

cm = set(cm, 'ScanTime', scant);

% set the bounds of parameters (for estimation)
cm=set(cm, 'ParameterMaximum', [ones(1,9)]);
cm=set(cm, 'ParameterMinimum', [zeros(1,9)]);

% set the default sensitivity
cm=addSensitivity(cm, 'K1', 'k2', 'k3', 'k4', 'PV', 'Fv');

if nargin ==2
    modelname = ' Myocardical Perfusion Study (USP):Compartments
        model(3) using Rb-82';
end
return

```

Uma vez associado o código completo do nosso modelo no COMKAT, podemos usá-lo por meio dos seguintes passos dentro da plataforma COMKAT:

```
>> comkat
```

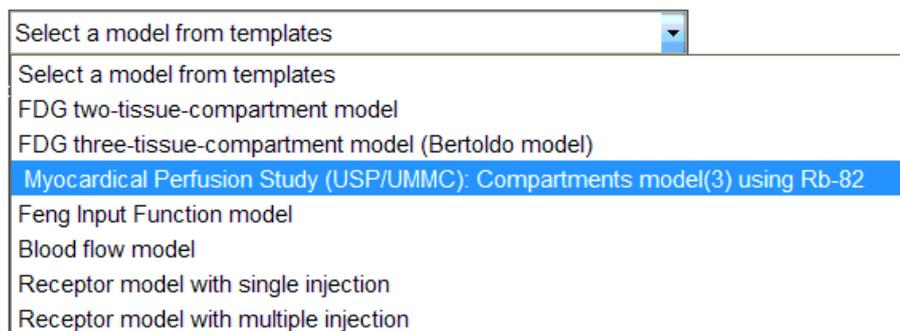
```
=====
```

```
Welcome to use COMKAT GUI!
```

```
GUI initializing...
```

```
=====
```

Deve-se escolher o modelo criado exibindo a lista de modelos na aba "Selecione um modelo a partir de modelos", como mostrado na Figura 56.



**FIGURA 56** – Lista com todos os modelos disponíveis no COMKAT incluindo o modelo criado para o projeto usando Rubídeo-82 com três compartimentos.

## GERAÇÃO DA CURVA DE ATIVIDADE DE ENTRADA USANDO CÓDIGO

Como indicado, se o estudo dinâmico não tiver nenhuma região com presença do sangue (isto é, casos clínicos de estudo cerebral), é adequado utilizar um algoritmo para gerar a Curva de Atividade de Entrada.

Por meio do algoritmo do Feng [Feng, 1993], pode-se representar a concentração de radiofármaco no sangue num período de tempo “t”. Esta curva é representando pela seguinte equação:

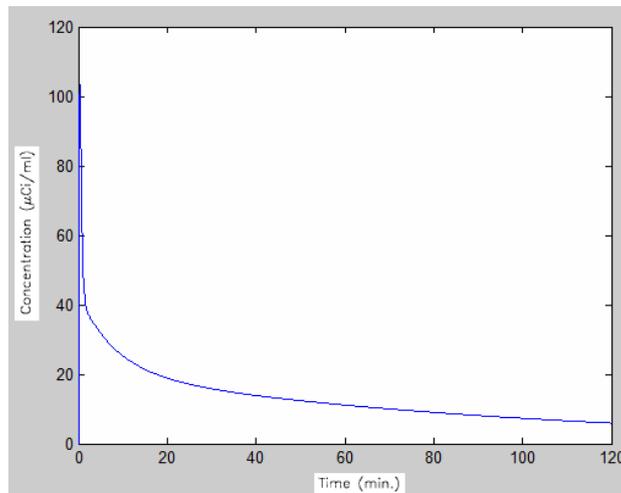
$$y(t) = [A_1(t) - A_2 - A_3]. \exp^{\lambda_1(t)} + A_2. \exp^{\lambda_2(t)} + A_3. \exp^{\lambda_3(t)} \quad (27)$$

onde  $\lambda_1$ ,  $\lambda_2$  e  $\lambda_3$  (em  $\text{min}^{-1}$ ) são os valores próprios do modelo e  $A_1$  (em  $\text{Ci/ml/min}$ ),  $A_2$  e  $A_3$  (in  $\text{Ci/ml}$ ) são os coeficientes do modelo.

Esta representação considera o uso de  $^{18}\text{F} - \text{FDG}$  como radiofármaco, com um tempo de meia-vida de 110 minutos aproximadamente, em um período de tempo de 120 minutos. A curva de entrada considerando estes parâmetros é mostrado na Figura 57 através do código a seguir:

```
p = [2 0 851.1 21.88 20.81 -4.134 -0.1191 -0.01043];
t = [0:1:120];
cm = addParameter(cm, 'sa', 100); % KBq/pmol specific activity of
                                % for F-18
cm = addParameter(cm, 'dk1', log(2)/109.8); % radioactive decay constant
                                            % for F-18
cm = addParameter(cm, 'dk2', 0); % radioactive decay constant
                                  % for F-18
cm = addInput(cm, 'sa', 'dk1', 'fengInput', p, t);

plot(cm)
```



**FIGURA 57** – Gráfico da Curva de Atividade de Entrada utilizado quando o estudo dinâmico não tem nenhuma representação de sangue. Os dados foram obtidos a partir de um estudo humano real.