Universidade Estadual de Campinas
Instituto de Física “Gleb Wataghin”

Estudo Comparativo entre os Algoritmos de Reconstrução Tomográfica por Retro - projeção Filtrada e Iterativos em SPECTs

Armando Rinaldi Neto

Orientador: Prof. Dr. Sergio Querino Brunetto

Co-orientador : Prof. Dr. Julio César Hadler Neto

Banca examinadora

Prof. Dr. Sergio Querino Brunetto

Prof. Dr. Alípio Luis Dias Neto

Prof. Dr. Roberto José Maria Covolan

Julho de 2002

Tese apresentada no Instituto de Física “Gleb Wataghin” como requisito para a obtenção do Título de Mestre em Ciências.

FICHA CATALOGRÁFICA ELABORADA PELA

BIBLIOTECA DO IFGW - UNICAMP

Rinaldi Neto, Armando
R47e Estudo comparativo entre os algoritmos de reconstrução tomográfica por retro-projeção filtrada e interativos em SPECTs / Armando Rinaldi Neto. -- Campinas, SP : [s.n.], 2002.

Orientadores: Sergio Querino Brunetto e Júlio César Hadler Neto.
Dissertação (mestrado) - Universidade Estadual de Campinas, Instituto de Física "Gleb Wataghin".

1. Física médica. 2. Medicina nuclear. 3. Tomografia computadorizada. I. Brunetto, Sergio Querino. II. Hadler Neto, Júlio César. III. Universidade Estadual de Campinas. Instituto de Física "Gleb Wataghin". IV. Título.

UNICAMP

PÓS GRADUAÇÃO

Instituto de Física Gleb Wataghin

C.P. 6165
CEP: 13083-970
Tel. (19) 788-5305
e-mail: secpos@ifi.unicamp.br

MEMBROS DA COMISSÃO JULGADORA DA TESE DE MESTRADO DE **ARMANDO RINALDI NETO – RA 940443** APRESENTADA E APROVADA AO INSTITUTO DE FÍSICA "GLEB WATAGHIN", DA UNIVERSIDADE ESTADUAL DE CAMPINAS, EM 29/07/2002.

COMISSÃO JULGADORA:

Prof. Dr. Sérgio Querino Brunetto (Orientador do Candidato) – CEB/UNICAMP

Prof. Dr. Alipio Luiz Dias Neto – CMN/DR/FM/USP

Prof. Dr. Roberto José Maria Covolan – DRCC/IFGW/UNICAMP

Agradecimentos

Seria impossível agradecer por ordem de importância as pessoas que direta ou indiretamente contribuíram para a realização deste trabalho. Por isso o farei em ordem cronológica de minha vida.

Aos meus pais, Roberto e Regina, que me deram a vida e a formação pessoal e moral, agradeço pelo amor incondicional , apoio e torcida.

Ao meu irmão Archimedes que tanto me ensinou sobre a vida e como ser feliz e, mesmo hoje, ainda me ensina.

A minha esposa Lenir que me fez uma pessoa melhor, completa e muito feliz, me ensinando todos os dias a não desistir.

Ao meu filho Pedro que, em tão pouco tempo de vida, me fez rever todos os meus valores, me dando o privilégio de ser seu pai.

Ao meu colega e irmão de profissão Israel M. de Oliveira, sem o qual jamais teria iniciado este trabalho.

Ao meu orientador Sergio Q. Brunetto, que mesmo frente a tantas adversidades na vida pessoal, manteve seus esforços para comigo e me guiou com mão de mestre que é na área de Medicina Nuclear.

Aos colegas do CEB-Unicamp e Serviço de Medicina Nuclear do HC Unicamp, que me apoiaram incondicionalmente, me recebendo de braços abertos.

Aos Drs. Bárbara, João Paulo, Rodrigo, Marco, Patrícia e Marcelo, pela colaboração nas análises das imagens clínicas, meu muito obrigado.

Aos amigos e colegas que torceram pela minha vitória.

A todos minha eterna gratidão.

Em memória de Archimedes Rinaldi

Homem, Músico e Irmão.

"(...) e o futuro é uma astronave que
tentamos pilotar. Não tem tempo,
nem piedade nem tem hora de chegar.
Sem pedir licença muda nossa vida e
depois convida a rir ou chorar (...)"

Toquinho

"La vita è adesso nel vecchio
albergo della terra (...)"

Cláudio Baglioni

Resumo

A Medicina Nuclear se apresenta como uma área de grande importância dentre as especialidades médicas atualmente praticadas, utilizando equipamentos de tecnologia de ponta para o diagnóstico através de imagens e informações clínicas singulares.

Em um ambiente de alta tecnologia como este se faz necessária a formação de uma equipe multidisciplinar, onde a presença de um profissional responsável pelo elo entre a tecnologia e as necessidades médicas torna–se indispensável. Neste contexto surge o papel do físico e da Física Médica gerando e executando metodologias e procedimentos para o controle da Qualidade dos equipamentos e processos de aquisição, auxiliando médicos e técnicos na busca de soluções para a obtenção de melhores imagens e serviços associados, contribuindo para a melhoria continua dos serviços prestados.

Dentre os trabalhos desenvolvidos pelo físico na área de Medicina Nuclear destacam-se a elaboração, estudo e otimização de algoritmos de reconstrução de cortes tomográficos na modalidade SPECT, buscando soluções para a visibilização da real distribuição do radiofármaco no órgão de interesse com o menor gasto computacional possível.

Neste trabalho buscou-se a construção de um texto que não só dissertasse sobre os algoritmos de reconstrução tomográfica em Medicina Nuclear mas também fosse uma referência para o estudo da modalidade SPECT, abrangendo aspectos de equipamentos, controle da Qualidade e métodos de avaliação de imagens, dando uma noção da importância do trabalho do físico dentro da área.

Como objetivo principal, buscou-se determinar as diferenças de performances entre os algoritmos de reconstrução *Filtered Back Projection* (FBP) e Iterativos,

utilizando–se conceitos do cotidiano da área de Física em Medicina Nuclear, tais como o cálculo de uniformidade tomográfica, estudo do contraste de objetos e estruturas das imagens de *phantoms* matemáticos, *phantoms* físicos e de estudos clínicos.

Dentre os resultados mais importantes alcançados destaca–se a construção de uma função $r(\sigma)$ variável com Δp (diferença entre a projeção medida e a calculada) como coeficiente de relaxação para os algoritmos Iterativos, onde σ é um parâmetro de controle de uma função heurística. Através do emprego da função $r(\sigma)$ os algoritmos Iterativos tornam-se menos susceptíveis a ruídos, convergindo mais rapidamente para a distribuição do corte.

Ao final dos testes e avaliações empregadas sobre os algoritmos implementados observa-se um padrão de comportamento para a reconstrução de cortes tomográficos de estudos de baixa densidade de informação : os algoritmos Iterativos , mesmo sem a otimização de filtros suavizadores pós processamento apresentam uma discreta superioridade de performance na reconstrução de objetos e estruturas quentes em relação aos reconstruídos com o algoritmo FBP, ocorrendo uma inversão da performance destes quando da reconstrução de objetos frios com freqüências próximas ou igual a de Nyquist.

Abstract

Among the medical specialties in current practice the Nuclear Medicine arises as an important area that uses high technology equipments which enables through unique images and clinical information diagnoses.

At such high technology atmosphere demands a highly skilled professional responsible for linking technology an medical needs. That is when the physicist and medical physics appears creating and carrying out methodologies and procedures for equipment's acquisition process and Quality control. This technician also helps physicians and operators find solutions in order to obtain the best images and other correlated issues therefore contributing for continuous improvement of final results.

The reconstruction algorithms of the tomographic slices in SPECT stands outs as one of many important physicist's tasks in the Nuclear Medicine area, searching solutions to make visible the real tracer distribution in the organ of interest with the lowest computational time possible.

In this work not only the construction of a text the lectured about the tomographic reconstruction algorithms in Nuclear Medicine was looked up for but, it also it is meant to be a reference to the study of the SPECT modality, embracing aspects of equipments, Quality control and evaluation of images methods, as well as a briefing on the importance of the physicist's work in this area.

As main objective, the determination of the differences of performances between the reconstruction algorithms Filtered Back Projection (FBP) and Iteratives was looked up for , using daily concepts of the Physics area in Nuclear Medicine such as : calculation of tomographic uniformity, study of the objects and structures contrasts for mathematical and real phantoms images and clinical studies.

Among the main obtained results the construction of a function *r*(σ) variable with Δp (difference between projection and calculated projection) stands out as one relaxation coefficient for the Iteratives algorithms, where σ is a control parameter of an heuristic function. Through the employment of *r*(σ) the Iteratives algorithms becomes less susceptive to noises, converging more quickly to the slice distribution.

At the end of the tests and evaluations employed on the implemented algorithms, a pattern of behavior for the slices tomographic reconstruction of studies of bw density of information was observed : the Iteratives algorithms, even without the optimization of smoothing filters after processing they presented a discreet performance superiority on the reconstruction of hot objects and structures in relation to those reconstructed with the algorithm FBP, whereas for cold objects with close frequencies or equal the one of Nyquist an investment of performance of these algorithms, takes place.

Sumário

Lista de Figuras

Lista de Tabelas

Motivação

A Medicina Nuclear é uma modalidade médica que se utiliza dos compostos chamados radiofármacos (ou traçadores) para fins de diagnóstico e terapia, os quais são constituídos de macromoléculas orgânicas marcadas com um elemento radioativo.

As imagens diagnósticas são a representação da distribuição do radiofármaco no órgão de interesse, o qual é normalmente administrado através de injeção intra - venosa no paciente.

Dentro dos possíveis procedimentos diagnósticos em Medicina Nuclear destaca-se o ECT (Emission Computed Tomography), técnica que permite, sem invasão e em contraponto com outras técnicas de radiodiagnóstico, a observação da funcionalidade de um órgão através da visibilização da distribuição do radiofármaco em um corte seccional deste.

O processo de obtenção de cortes tomográficos (ECT) possui uma série de fatores com características que introduzem erros e artefatos na imagem : a detecção pelas câmeras cintiladoras dos γ´s emitidos, as características físicas (obesidade por exemplo) e movimentos involuntários de pacientes (como a respiração) e a manipulação matemática (algoritmos) aplicada a reconstrução dos mesmos, podendo levar o especialista a ter dificuldades para identificar lesões ou doenças associadas ao padrão mostrado no estudo.

De todos os fatores que influenciam na qualidade das imagens tomográficas, dedicaremos neste trabalho atenção aos algoritmos de reconstrução, já que dependendo do algoritmo adotado e do tipo de procedimento clínico realizado, podemos alterar a sensibilidade e especificidade do método diagnóstico. Por essas razões, consideramos oportuno investigar o desempenho dos algoritmos : Retro - projeção Filtrada (FBP) e Iterativos.

O objetivo deste trabalho é comparar o desempenho destes algoritmos do ponto de vista das aplicações clínicas, buscando responder se as técnicas Iterativas, embora às expensas de maiores recursos computacionais, apresentam melhores resultados que o FBP. Para tanto, utilizou-se aquisições tomográficas cardíacas, MIBI-^{99m}TC, aquisições tomográficas de tórax e cabeça com ^{67}Ga , aquisições com ^{18}F (FluorDesoxiGlucose – FDG) e aquisições com simuladores tomográficos (*phantoms*) com ^{99m}Tc. A escolha desses protocolos clínicos se fez pela baixa densidade de informação das projeções (aquisições primárias que serão manipuladas matematicamente para reconstruir o corte tomográfico), decorrentes das características próprias desses exames. Concluímos nossa análise com aquisições de simuladores tomográficos, os quais permitem uma avaliação objetiva e inequívoca do desempenho dos algoritmos de reconstrução tomográfica.

As aquisições clínicas foram escolhidas dentre as realizadas rotineiramente junto ao Serviço de Medicina Nuclear do HC/Unicamp, contando sempre com a supervisão de sua equipe clínica e, as que continham simuladores, utilizaram as câmaras de cintilação Elscint Helix, Elscint APEX SP4 e Elscint APEX SP6 do mesmo Serviço e os simuladores tomográficos específicos para uso em Medicina Nuclear do Grupo de Medicina Nuclear da Área de Física Médica do CEB/Unicamp. Os algoritmos foram implementados segundo as sugestões apresentadas pela literatura, conforme bibliografia citada neste trabalho, utilizando uma plataforma AMD Duron 750 MHz com 256 MB de RAM e placa de vídeo com 16MBb , com sistema operacional Linux (Mandrake 7.2) e compilador ANSI C da GNU.

Capítulo 1 – Introdução

1.1 – Histórico

Iniciadas em tempos e por motivações diferentes e confundindo-se na história da física médica, o desenvolvimento da Medicina Nuclear e da Tomografia vêm se encontrar apenas no início dos anos 80, gerando as modalidades de diagnóstico conhecidas como SPECT (*Single Photon Emission Computed Tomography*) e PET (*Positron Emission Tomography*).

Os primeiros estudos realizados em Medicina Nuclear referem-se a diagnósticos e tratamento de doenças relacionadas a tireóide utilizando ^{131}I, no qual o principal problema de diagnóstico era verificar a presença de nódulos deste órgão, freqüentemente malignos. O equipamento usado para detectar e gravar as contagens da atividade do ^{131}I administrado ao paciente possuía posicionamento manual sobre as áreas de interesse do órgão, o que tornava o processo lento e impreciso.

Em 1950 Benedict Cassen introduziu o primeiro *scanner* linear [(1)] com um detector de controles de movimentos automatizados. A imagem formada num papel disposto em uma impressora com movimento conjugado ao detector indicava o local e a atividade do ponto emissor do paciente. Apesar do avanço, o equipamento ainda era bastante lento, pois a imagem era formada seqüencialmente.

A primeira câmera capaz de registrar todos os pontos de uma área em uma única vez foi descrita por Hal Anger em 1953 [(2)]. O sistema era composto por um colimador *pinhole* que projetava a imagem da distribuição dos γ's incidentes sobre um cristal cintilador de NaI (Tl) (tela) e uma chapa com filme. A imagem era formada por cintilações geradas pelo cristal. Infelizmente, o sistema - principalmente o filme - era ineficiente, necessitando de longas horas de exposição e doses de radiação bastante altas administradas aos pacientes para se obter um resultado apenas satisfatório na imagem.

Ao final dos anos 50 Anger trocou o conjunto tela – filme por um cristal único de grande área e um conjunto de tubos fotomultiplicadores (PMT´s), aumentando em muito a eficiência do sistema. Desde então a câmera cintiladora de Anger sofreu inúmeras melhorias, principalmente refinamentos de engenharia (*hardware* e *software*), sendo aceita e utilizada em grande escala após a metade dos anos 60.

Tal atraso na evolução das câmeras cintiladoras pode ser explicado por alguns fatores : as câmeras de Anger possuíam uma resolução espacial muito pobre e a energia de 364 keV do ^{131}I, o radioisótopo predominante até então, era um empecilho inerente a performance da câmera pois, para esta energia, a eficiência de cristais com espessura de 1/2´´ é de apenas 21% [(3)]. O sucesso veio apenas com a evolução dos equipamentos, seus componentes e também com a introdução de novos radiofármacos.

A evolução da eletrônica associada (computadores, *hardwares* e *softwares* mais rápidos e eficientes após 1960), a melhoria da tecnologia e dos materiais empregados na construção dos PMT´s e a redução de seu diâmetro (possibilitando um aumento de sua eficiência) melhoraram em muito a resolução energética. Isso tudo somado ao aprimoramento das técnicas de construção de colimadores (construídos com furos hexagonais, mais eficientes que os circulares) propiciou também uma melhorara da resolução espacial, da linearidade e da uniformidade das aquisições.

Além disso, a introdução em 1964 por Haper do ^{99m}Tc como radioisótopo para a marcação de traçadores possibilitou uma melhora significativa na estatística das aquisições e consequentemente na qualidade das imagens obtidas. Este filho metaestável do ^{99}Mo, obtido através de um processo químico chamado eluição, possuí duas linhas de energia próximas a 140 keV (140.5 e 141.0), valor de energia para o qual a eficiência da câmera é máxima [(4)] .

Em paralelo ao desenvolvimento dos instrumentos utilizados em Medicina Nuclear, ocorreu o desenvolvimento dos algoritmos de reconstrução tomográfica, os quais buscam obter, a partir de imagens planas (projeções), imagens de cortes seccionais de um corpo. Em Medicina Nuclear a imagem de um corte seccional do órgão estudado representa a distribuição volumétrica do radiofármaco administrado ao paciente .

Um primeiro método de tomografia chamado de clássico (ou convencional) foi introduzido com imagens de transmissão de raios X [5]. Uma ilustração do sistema utilizado para a aquisição das imagens é apresentado na figura 1.1. Enquanto raios X atravessam o corpo, a fonte e o detector movem-se conjugadamente numa trajetória tal que apenas a fatia de interesse permanece em foco, enquanto as demais são borradas .

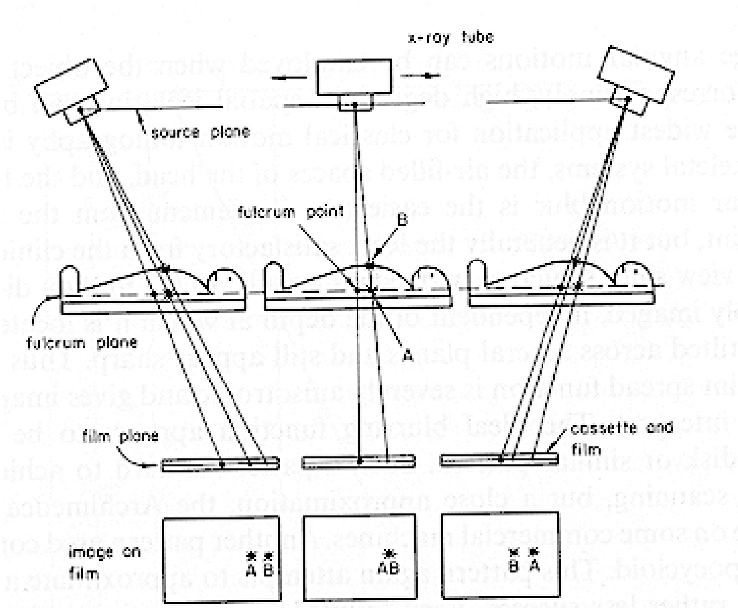


Figura 1.1 - Tomografia clássica - plano focal . Apenas o plano de interesse permanece em foco , os demais são borrados [6] .

Nota-se duas desvantagens no método. A primeira é a presença de estruturas borradas e contagens provenientes de planos que não o de interesse. Isto contribui para degradar a imagem do plano focal e aumentar o ruído. A segunda desvantagem é a limitação mecânica pois, com este método, é possível coletar projeções apenas num intervalo limitado de ângulos, menor que um arco de 180 º, dando um número insuficiente de projeções para uma correta reconstrução do corte [7].

Assim, surge a necessidade da elaboração de técnicas que reconstruam os cortes tomográficos sem a interferência dos planos adjacentes ao de interesse - a reconstrução tomográfica. A partir de imagens bi – dimensionais (projeções), coletadas um número suficiente e num intervalo angular adequado, pode-se reconstruir a distribuição do radiofármaco no plano de interesse sem a interferência dos demais [(7).]

O primeiro estudo efetuado nesta área foi feito pelo matemático tcheco J. Radon em 1917 [(8)], o qual propôs a solução em termos de uma equação integral. Na prática, seu método é de difícil implementação, pois o conhecimento das condições de contorno, isto é, do perfil do corte, é essencial para obter-se a imagem ou distribuição do radiofármaco no volume sob observação.

A primeira aplicação bem sucedida de técnicas de reconstrução tomográfica foi obtida na radioastronomia por Bracewell [(9)] em 1956. Com o objetivo de determinar a posição de fontes de radiação de microondas no interior do Sol, Bracewell lançou mão de técnicas tomográficas, obtendo, levando em consideração as limitações de suas antenas em focar as fontes, bons resultados para a época. A dificuldade encontrada foi que as antenas de microondas não focavam pontos, mas apenas finas tiras que atravessam a superfície solar. Assim, a emissão total de uma tira poderia ser medida e com o conjunto destas medidas, um mapa da atividade solar seria reconstruído.

Problema semelhante surgiu no estudo de moléculas complexas através da microscopia eletrônica. Novamente, técnicas de reconstrução tomográfica foram desenvolvidas e utilizadas (DeRosier e Klug (1968)[(10)]) independentemente dos primeiros trabalhos em radioastronomia.

As aplicações médicas da reconstrução tomografia surgiram no ano de 1972/1973 com lançamento do CT Scanner da EMI Ltda [(11)], o qual teve enorme impacto na radiologia diagnóstica. De lá para cá, o avanço foi gigantesco, sendo principalmente impulsionado pelo rápido desenvolvimento do *hardware* que permitiu o uso clínico de algoritmos mais sofisticados e complexos. Atualmente, utilizando a enorme velocidade e capacidade de processamento dos computadores, buscam-se métodos que, além de reconstruírem a distribuição volumétrica do radiofármaco no interior do paciente, agreguem técnicas de correção da atenuação e espalhamento [12 – 14].

Desse modo, alimentado também pelo desenvolvimento de novos radiofármacos, principalmente após o início dos anos 80, tais como HIPDM [15] e IPM[16], para a imagem de cérebros, a modalidade de diagnóstico SPECT alcançou maturidade tecnológica tal que tornou-se ferramenta indispensável no diagnóstico de doenças em todos os grandes centros de saúde.

1.2 - Medicina Nuclear - Princípios e Características

Os estudos em Medicina Nuclear baseiam-se na utilização de radioisótopos emissores de radiação γ. Os radioisótopos emissores γ´s são utilizados devido a maior capacidade de penetração destes na matéria que a de partículas β- ou α.

Os γ´s emitidos isotropicamente de cada ponto do paciente (alguns espalhados, outros absorvidos e outros emitidos sem qualquer interação) são detectados em câmeras cintiladoras (figura 1.2) para a formação da imagem. Para colimadores de baixa energia e propósitos gerais, apenas 1 a cada 5.000 [17] γ´s é detectado, o que reduz em muito a estatística do processo.

Os γ´s que chegam ao detector são selecionados dentro de certo ângulo sólido por colimadores. Estes γ´s por sua vez geram cintilações no cristal de NaI (Tl) , as quais são coletadas em um conjunto de PMT´s, responsável pela transformação de luz em corrente elétrica mensurável. Com estes pulsos se identifica a energia e a posição de incidência do γ incidente. Estes sinais são então processados, digitalizados e armazenados.

As informações adquiridas permitem conhecer a distribuição do radiofármaco no corpo do paciente, para tanto, associa-se um nível de cinza ou de cor ao número de iterações, ocorridas por posição, obtendo-se uma imagem da distribuição. O processo de detecção dos γ´s e os componentes da câmera de cintilação serão discutidos em detalhe no Capítulo 2.

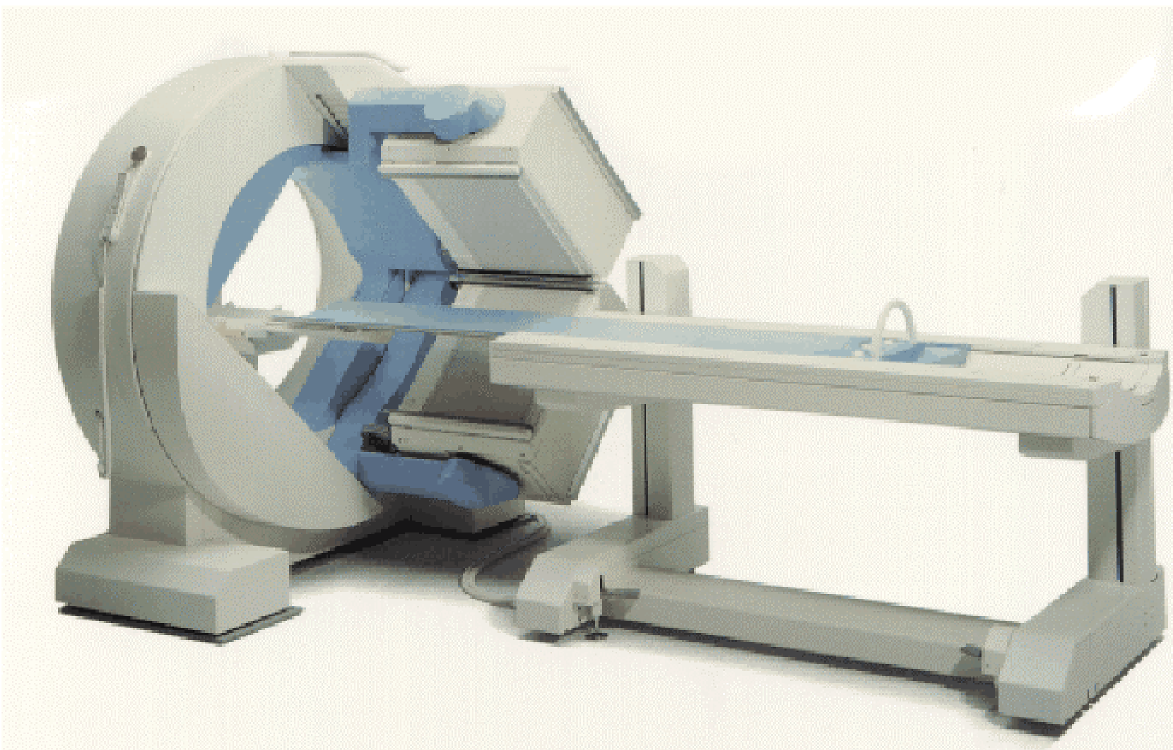


Figura 1.2 - Exemplo de câmera cintiladora de dois detectores - Siemens.

Dependendo do radioisótopo utilizado classifica-se o método de aquisição como SPECT , tendo como traçadores mais comuns ^{99m}Tc , ^{201}Tl , ^{67}Ga , ^{131}I e ^{123}I, ou PET, sendo o ^{11}C e o ^{18}F seus radiofármacos mais utilizados.

Em geral os radioisótopos são produzidos em reatores nucleares como produtos da fissão nuclear do ^{235}U (exemplos: ^{99}Mo, ^{131}I e o ^{137}Cs) ou através de aceleradores de partículas (exemplos: ^{67}Ga, ^{123}I, ^{57}Co, ^{18}F e ^{201}Tl).

Os materiais produzidos podem ser monoenergéticos ou não, com energias que variam de 80 à 500 keV para os emissores γ´s ou 511 keV, quando utiliza-se de fótons provenientes de aniquilações ocorridas em pósitrons emissores (PET). Dentre os materiais emissores β+, destaca–se o ^{18}F, utilizado para estudo em câmeras colimadas ou de coincidência.

Os radiofármacos são macromoléculas orgânicas marcadas com radioisótopos que permitem estudar características *in vivo* e *in vitro*, tais como: transporte, distribuição, metabolismo e processos de eliminação do corpo humano. Dessa forma podemos observar sem invasões, riscos de reações alérgicas ou efeitos colaterais e/ou administração de elevadas doses muitos dos sistemas e órgãos do corpo humano, através de estudos do tipo : aquisições estáticas (figura 1.3), dinâmicas (fluxo) e os de maior interesse neste trabalho, os exames tomográficos (figura 1.4) .

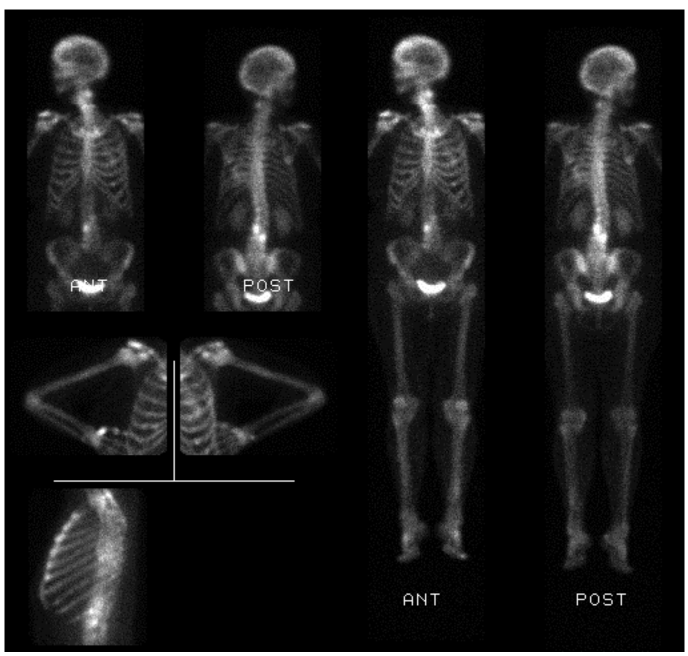
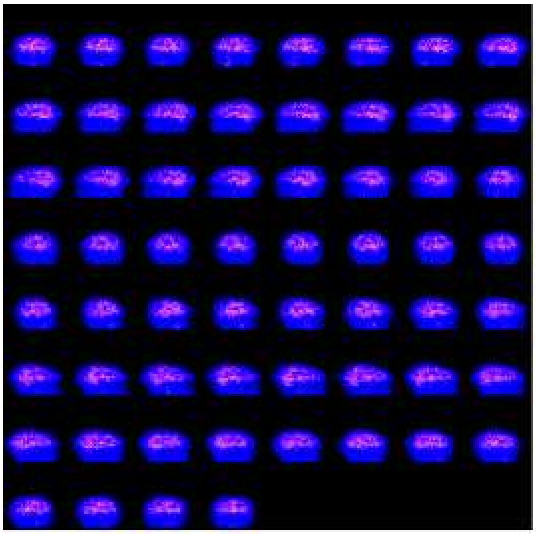
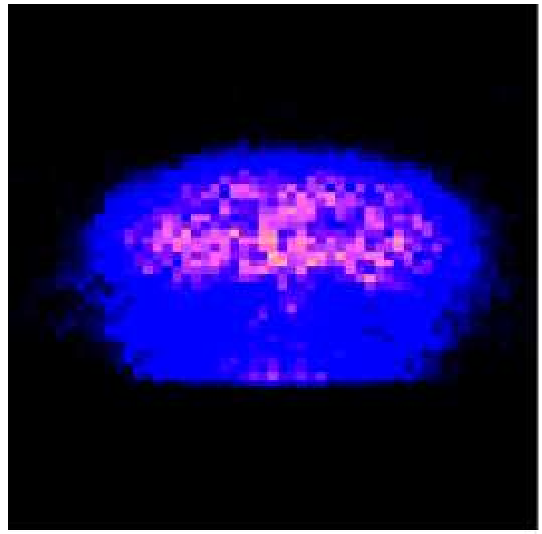


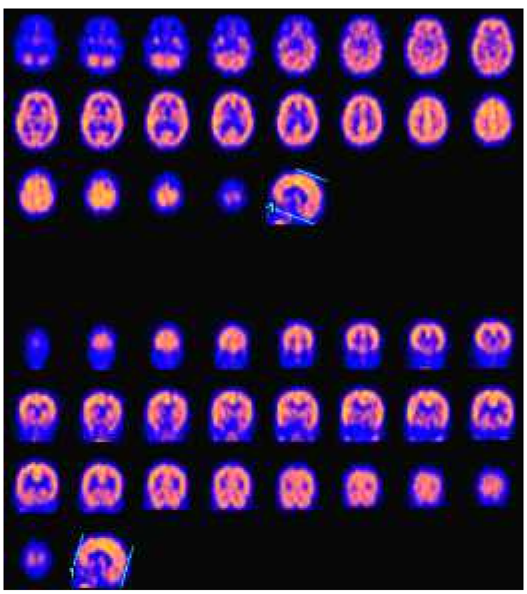
Figura 1.3 – Exemplo de exame estático . Cintilografia óssea (varredura de corpo inteiro e detalhes) . Realizado com traçador MDP (***Methilene Diphosphonate*** **- enzima marcada com ^{99m}Tc para estudo do sistema esquelético [18]).**



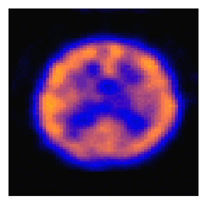
1.4(a)



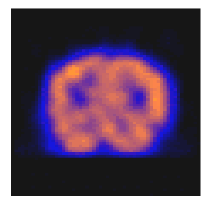
1.4(b)



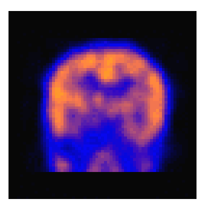
1.4 (c)



1.4 (d)



1.4 (f)



1.4 (e)

Figura 1.4 – Exemplo de estudo tomográfico do cérebro. (a) 60 projeções de um SPECT cerebral . (b) Projeção do ângulo 0º em destaque . (c) Vários cortes tomográficos (SPECT cerebral) reconstruído a partir de 1.4 (a) . (d) Corte transaxial nº 12 em 1.4 (c) - detalhe . (e) Corte coronal nº 13 em 1.4 (c) - detalhe . (f) Corte sagital nº 5 em 1.4 (c) - detalhe .

Capítulo 2 – Câmeras Cintiladoras

A figura 2.1 ilustra esquematicamente os principais componentes de uma câmera de cintilação.

A imagem de raios γ´s é projetada pelo colimador no cristal de NaI (Tl) (figura 2.1), criando neste um padrão de cintilações que representa a distribuição da radioatividade a frente do colimador.

A Radiação γ apresenta refração e reflexão desprezíveis, sendo necessária a utilização de colimadores para a formação das imagens. Os colimadores são construídos para absorver os γ´s emitidos do paciente fora de determinado ângulo sólido. Os γ´s detectados pelo cristal / PMT são contabilizados e os eventos transformados em informação, processada e armazenada pela eletrônica associada, formando a imagem.

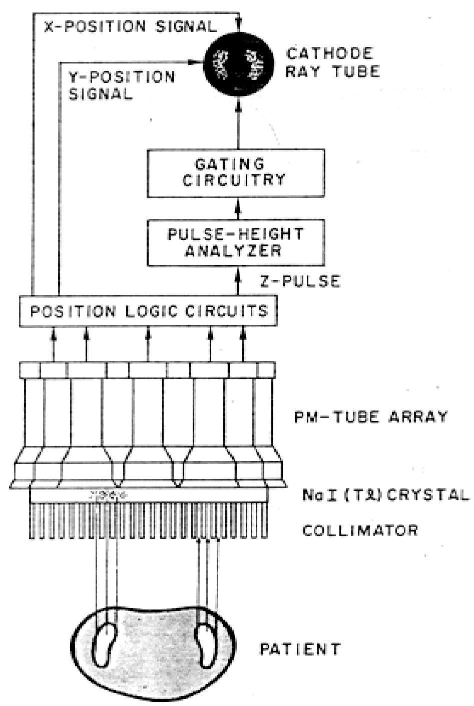


Figura 2.1 – Principais componentes da câmera cintiladora [19].

Para se entender todos os fatores que influenciam os estudos tomográficos, deve-se começar pelo funcionamento, construção e características das câmeras cintiladoras, iniciando o estudo destas por seus componentes e características de performance.

Antes, porém, define-se três características de performance, a saber [20] :

- a resolução espacial descreve a capacidade de resolver duas fontes de radiação puntiforme ou linear em objetos distintos. Convencionalmente é quantificada como a largura a meia altura do máximo (*full width at half maximum* – FWHM) da resposta de uma fonte linear ao longo do eixo da fonte, ou a mínima distância que possibilite a visualização distinta de duas fontes puntiformes;
- à habilidade de distinguir fótons de diferentes energias dá-se o nome de resolução energética. Quantificada em % de FWHM do espectro de energia, é expressa em % de FWHM / fotopico[1] do espectro;
- defini-se como eficiência, em %, a razão entre o número de γ´s que chegam ao detector e o número de γ´s que realmente são detectados.

Com estas definições, iniciam-se os estudos sobre câmeras cintiladoras pelo seu principal componente (o detector) e em seguida pela própria câmera e por características que influenciam na qualidade das imagens dos estudos adquiridos.

2.1 – Detectores Cintiladores

Uma das mais antigas maneiras de se detectar radiação ionizante é através de detectores cintiladores, os quais têm como principal componente um cristal cintilador e um ou mais tubos fotomultiplicadores, apresentados na figura 2.2.

[1] Fotopico – valor mais provável da energia de uma emissão mono - energética numa distribuição de energia observada em um espectro.

Idealmente um material cintilador deve possuir as seguintes propriedades [21] :

- Converter a energia dos γ´s incidentes em luz com alta eficiência de cintilação;
- A conversão deve ser linear, ou seja, a produção de luz deve ser proporcional a energia depositada pelo γ;
- O meio deve ser transparente para o comprimento de onda da luz cintilada, possibilitando uma boa coleta desta;
- O tempo de decaimento da luminescência induzida deve ser suficientemente curto e mais rápido que os pulso de sinais que serão gerados a partir destes;
- O índice de refração do material deve ser próximo a 1.5, permitindo maior eficiência no acoplamento com o PMT.

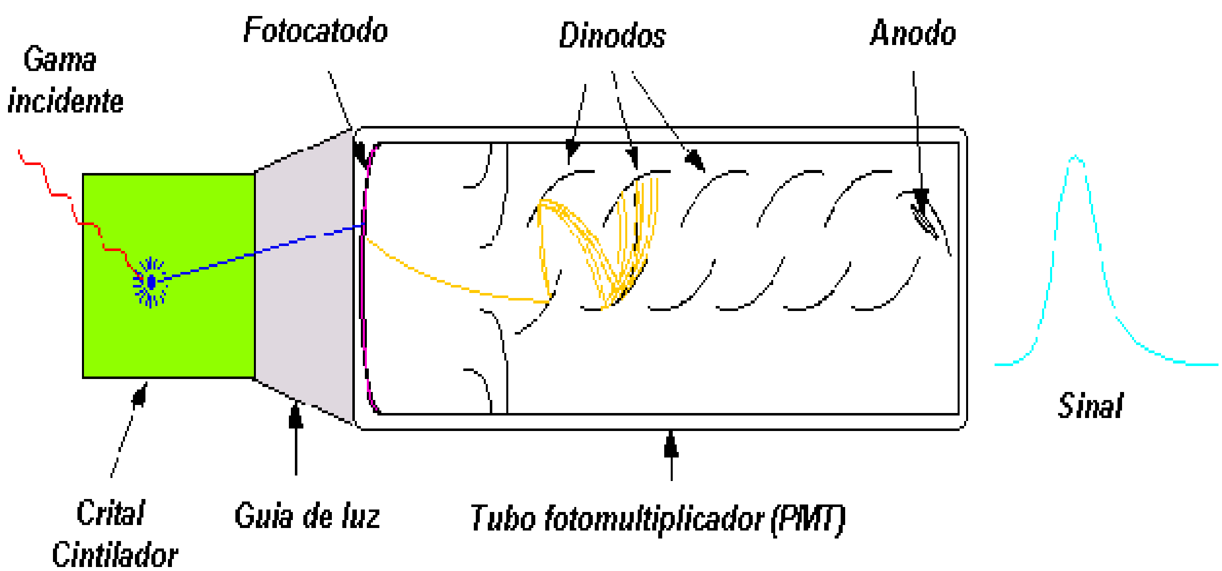


Figura 2.2 - Detector de cintilação e seus componentes : cristal e PMT.

Existe uma vasta quantidade de materiais cintiladores, onde as aplicações mais comuns incluem a utilização de cristais inorgânicos ou com bases orgânicas (líquidos ou plásticos). A escolha em particular para o uso dependerá da necessidade de velocidade e eficiência na produção de luz destes materiais. No caso da Medicina Nuclear, o mais comum dos cintiladores usado é o de cristal inorgânico de NaI (Tl), escolhido pela alta eficiência de cintilação e linearidade (21) .

O mecanismo de cintilação em cristais inorgânicos depende dos estados de energia determinados pela rede cristalina do material. Os elétrons da rede encontram-se em sua maioria na banda de valência (figura 2.3), havendo um *gap* entre esta e a banda de condução, chamado de banda proibida. Quando elétrons da banda de valência absorvem energia, estes são promovidos para a banda de condução, deixando um buraco. Para retornarem a banda de valência, ocorre a emissão de um fóton com energia igual a diferença energética entre as bandas que o elétron transita (cintilação). Em cristais puros e em baixas temperaturas, este processo é ineficiente (21).

Para otimizar este processo, são adicionadas impurezas aos cristais, criando centros ativadores (figura 2.3), que são estados energéticos intermediários (fundamental e excitados) dentro da banda proibida, criando-se *gaps* menores, possibilitando um aumento do número de cintilações, no tempo típico de 10^{-7} s (21).

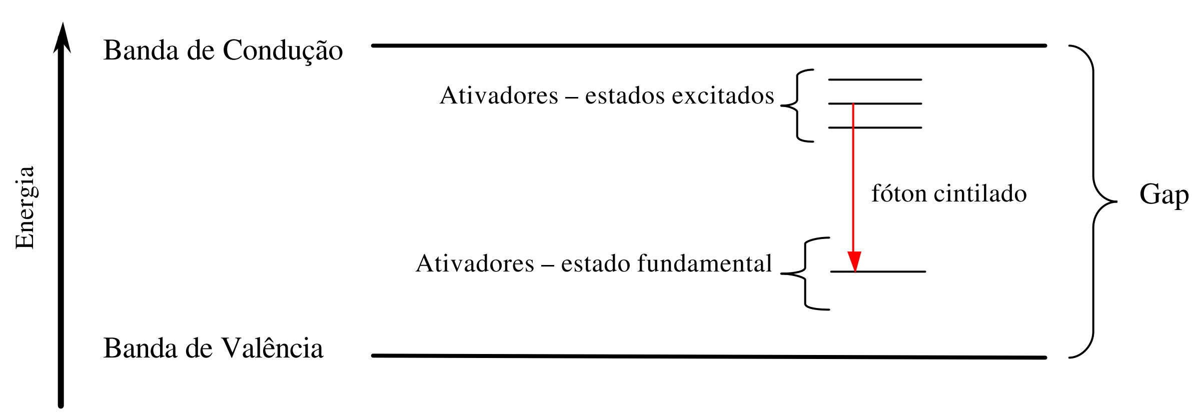


Figura 2.3 - Estrutura de bandas de energia com ativadores em cristais cintiladores .

Quando um γ incide sobre o cristal, este ejeta um elétron da banda de valência para a de condução por efeito Fotoelétrico ou Compton, que por sua vez atravessa o material excitando o resto da rede, criando vários pares elétron/buraco. Os elétrons que foram transferidos para estados excitados, retornam em seguida para o estado fundamental passando por centros ativadores criados com a introdução de impurezas, ocorrendo as cintilações.

Em 1948, Robert Hofstadter [22] produziu o primeiro cristal de iodeto de sódio com impurezas de Tálio (10^{-3} mol/mol de NaI) adicionadas durante um processo de fusão, iniciando a era dos cintiladores modernos. As principais características deste material são: possibilidade de crescimento com áreas bastante grandes, material higroscópico (se deteriora quando absorve água ou exposto a atmosfera por longo período), frágil (quebra facilmente com choques mecânicos ou térmicos), excelente produtor de luz com linearidade sobre um *range* significativo de energia, alcançando eficiência máxima na produção de cintilações na faixa de operação entre 20 - 40 ºC e, tendo um tempo de decaimento de cintilações em torno de 230 ηs, tem sua eficiência de contagem reduzida para altas taxas de contagem [21]. Outros dados físicos do material, como o comprimento de onda máximo emitido, encontram–se na Tabela 2.1.

O segundo componente, o tubo fotomultiplicador (figura 2.2), tem como função transformar os pulsos de luz gerados no cristal (cintilações), em pulsos de corrente elétrica de valores mensuráveis, proporcional a energia do γ incidente.

O processo de conversão de luz em corrente elétrica inicia-se pela passagem das cintilações por guias de luz, que fazem o acoplamento entre o cristal cintilador e o PMT (figura 2.2). Normalmente uma graxa de silicone com índice de refração próximo a 1.5 é utilizada para a tarefa.

Peso Específico	***Comprimento de onda máximo emitido (ηm) λ_{max}***	***Índice de refração a λ_{max}***	***Principal constante de decaimento (μs)***	***Número de photons produzidos/ MeV***	***Eficiência absoluta de Cintilação para elétrons rápidos***
3,67	415	1,85	0,23	38000	11,3 %

Tabela 2.1 – Características físicas do cristal de NaI (Tl) [21] .

Ao incidir no foto-catodo, o fóton cintilado transfere sua energia ($h\nu$) para um elétron do material foto-sensível. Este elétron então migra para a superfície, escapando, ocorrendo a emissão de um elétron primário de baixa velocidade (figura 2.4). Após a emissão do elétron primário, ocorre a multiplicação eletrônica, onde o processo de emissão de elétrons pelos eletrodos denominados dinodos (figura 2.4) é o mesmo ocorrido no foto-catodo, porém, sendo este iniciado pelos elétrons gerados no dinodo anterior.

Os elétrons que deixam os dinodos são acelerados em direção ao próximo dinodo por uma diferença de potencial gerada por um divisor de tensão resistivo e por uma fonte de alta tensão (figura 2.4). A diferença de potencial entre o foto-catodo e o primeiro dinodo é maior que a encontrada entre os outros dinodos, objetivando um aumento na eficiência da coleta dos elétrons primários.

A energia cinética típica de um elétron que escapa do dinodo é de 1 eV, sendo então sua energia cinética determinada, na prática, pela diferença de potencial entre os dinodos. A produção de elétrons secundários é função desta diferença de potencial. Pequenas energias produzem poucos elétrons excitados, porém próximos a superfície do material, com grande probabilidade de escape. Energias maiores criarão mais elétrons excitados, no entanto numa profundidade média maior, diminuindo a probabilidade de escape dos elétrons do material e, conseqüentemente, a produção de elétrons secundários.

Além disso, a direção do movimento destes elétrons é totalmente randômica, assim, muitos elétrons não atingem a superfície dos dinodos e outros não têm energia suficiente para ultrapassar a energia potencial da superfície, restando uma pequena fração dos elétrons excitados com condições de gerar elétrons secundários.

Estas características do processo de multiplicação eletrônica tem uma razão de produção de elétrons secundários tipicamente igual a 5, o que, para 10 dinodos daria um fator multiplicativo de 5^{10}, ou seja , 10^7.

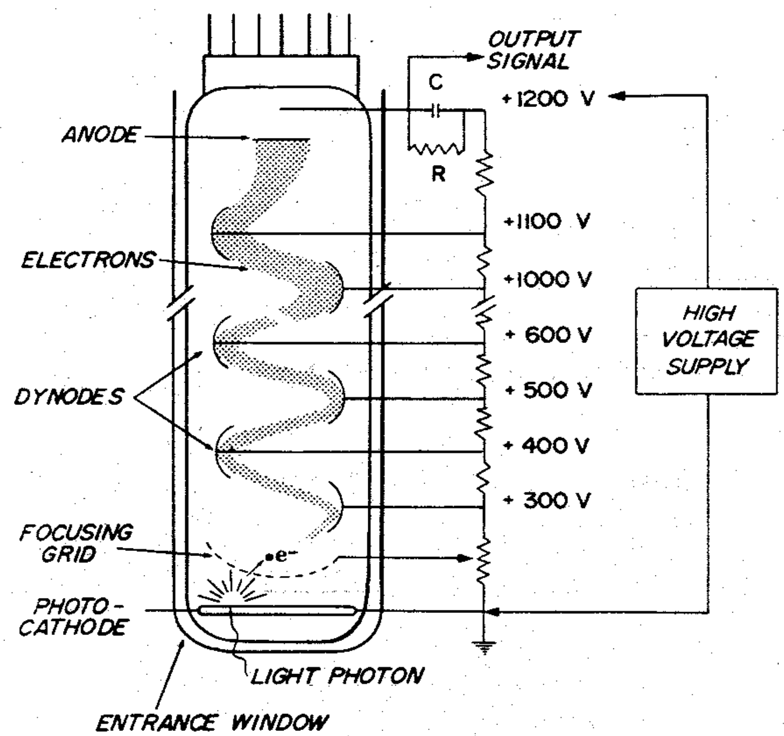


Figura 2.4 – Tubo foto - multiplicador (PMT) e seus componentes [23].

Após a geração da cascata de elétrons no processo de multiplicação eletrônica, ocorre a coleta desta no foto-anodo (figura 2.4), originando um pulso de corrente elétrica mensurável. Conhecendo-se o fator multiplicativo do PMT e, de posse da leitura do pulso elétrico formado, determina–se a energia do γ incidente no cristal.

Nas câmeras cintiladoras de Medicina Nuclear, os PMT´s normalmente são dispostos em arranjos hexagonais de 61 elementos. Este número é otimizado e tem influência direta nas resoluções espacial e energética. Uma quantidade menor implicaria no empobrecimento destas características, enquanto que um aumento traz complicações na calibração dos valores de alta tensão, gerando não uniformidades no sistema.

O pequeno sinal de saída dos PMT´s exige a utilização de pré – amplificadores, já acoplados nos PMT´s para se evitar ruídos. Estes têm duas funções: prover o ganho do sinal e fazer o casamento de impedância na linha, evitando-se distorções.

O sistema de detecção apresentado possuí limitações intrínsecas, que afetam diretamente a qualidade do processo de contagem dos γ´s incidentes e, conseqüentemente, das imagens. No caso do cristal, há uma competição entre eficiência de detecção e as resoluções energética e espacial – quanto mais espesso o cristal maior será sua eficiência porém, mais pobre serão as resoluções devido ao espalhamento interno da luz cintilada. No caso inverso, cristais mais finos têm melhores valores de resoluções, com detrimento da eficiência, devido a redução da secção de choque.

Com relação aos PMT´s, podemos citar quatro principais fatores que limitam a qualidade da conversão de luz em corrente elétrica e, conseqüentemente, nas resoluções energética e espacial e na eficiência (20):

- Definindo QE como o número de elétrons emitidos / número de γ´s incidentes * 100 %, este valor seria de 100% para um PMT ideal. Mas, devido às perdas de energia que os elétrons sofrem quando tentam escapar do foto - catodo (colisões entre elétrons / elétrons no processo de migração para a superfície), a QE apresenta-se em torno de 20–30 %;

- Foto-catodos, especialmente os com grande diâmetro, não apresentam uniformidade na emissão sobre toda a extensão de sua área, apresentando depreciação da performance nas bordas. Aliado a este problema, há dificuldades em se conseguir uma uniformidade na coleta do foto-elétron primário no primeiro dinodo para toda a área do foto-catodo. A combinação destes efeitos pode levar principalmente detectores com cristais finos a uma variação do pulso elétrico observado no anodo de 30 – 40 % sobre a posição na área de incidência no foto-catodo. Isto ocorre porque, devido ao ângulo sólido entre cristal e PMT, há uma tendência de espalhamento da luz cintilada sobre o foto-catodo, aumentando a densidade de cintilações sobre a borda do mesmo. Esta não uniformidade cria variações nas respostas do PMT, tendendo a empobrecer a resolução energética do sistema;
- O fator multiplicativo de conversão entre a intensidade dos pulsos de luz cintilados e a corrente elétrica gerada deve ser constante, caracterizando a linearidade da transformação. Porém, características de construção dos PMT´s (entre elas a distância entre o último dinodo e o foto – anodo) e variações nos valores de tensão sobre os dinodos durante o curso do pulso, geram desvios na trajetória e perdas, causando não linearidade na conversão dos sinais, provocando variações nas resoluções energéticas e espaciais;
- Ruídos e pulsos espúrios também são gerados no processo de conversão, sendo o resultado principalmente de elétrons termiônicos. Dependendo da razão de contagem, este efeito pode reduzir em muito as resoluções energética e espacial.

As características limitantes dos detectores cintiladores tornariam impossíveis imagens clínicas com qualidade e confiabilidade em Medicina Nuclear.

Para tanto, se faz necessária à utilização de uma eletrônica associada eficiente e *softwares* de correção para tornar as imagens viáveis.

2.2 – Princípios Básicos e Componentes

Além do detector constituído por um cristal de NaI (Tl) e por um conjunto de PMT´s, a câmera cintiladora também é composta pelos componentes apresentados na figura 2.1.

Construídos com uma liga de Pb/Tg/Cd, os colimadores têm a espessura do septo (parede entre furos) determinada pela energia do γ incidente. A geometria e o diâmetro dos furos dos colimadores (normalmente com formato hexagonal) determina a resolução espacial e a sensibilidade (eficiência) dos mesmos, sendo o primeiro melhorado apenas quando se deteriora o segundo – figura 2.5. Estas características, aliadas a espessura do septo, dão a classificação dos colimadores por : energia (baixa / média / alta) e resolução espacial (baixa / uso gera / alta), tendo sua utilização vinculada ao tipo de estudo e radioisótopo utilizado. Devido a essas características, o modelo de colimador implica diretamente na qualidade final da imagem.

Dependendo do estudo clínico, utilizam-se colimadores com geometrias diferenciadas: *pinhole* (para a magnificação de pequenas áreas do paciente, ocorrendo a inversão do sentido da imagem em relação ao objeto), paralelo (não interfere na imagem), divergente (para imagens reduzidas de grandes áreas dos pacientes) ou convergente (para a magnificação de pequenas áreas do paciente) - figura 2.6.

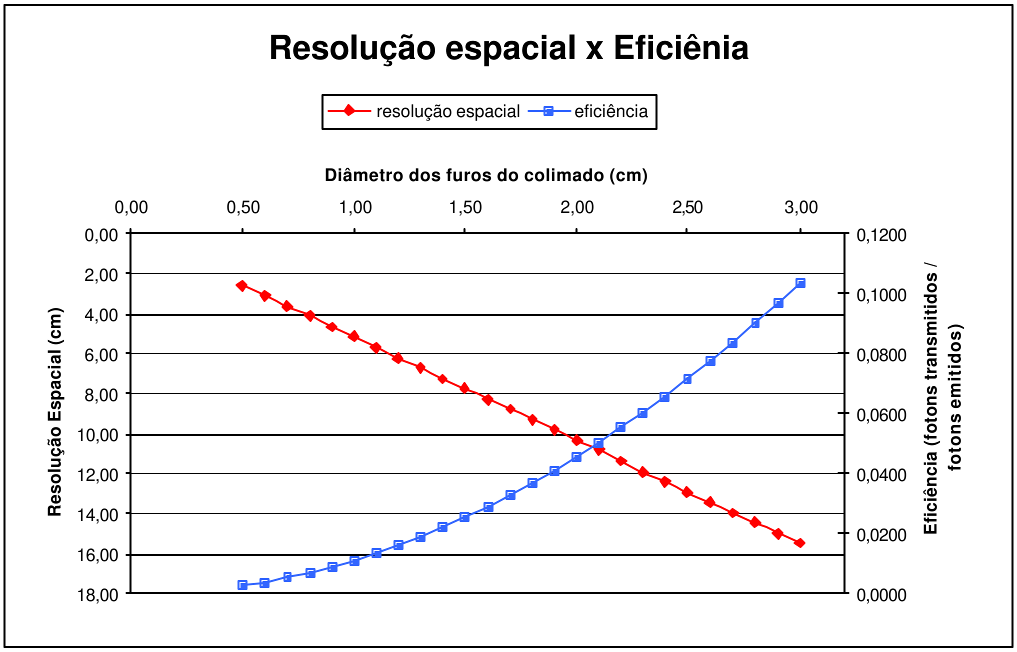


Figura 2.5 – Resolução espacial x eficiência de um colimador hipotético em função do diâmetro dos furos. Dados: energia de 150 keV , espessura de septo = 0,03 cm , distância da fonte ao colimador = 10 cm, furos hexagonais e espessura do colimador de 2,5cm.

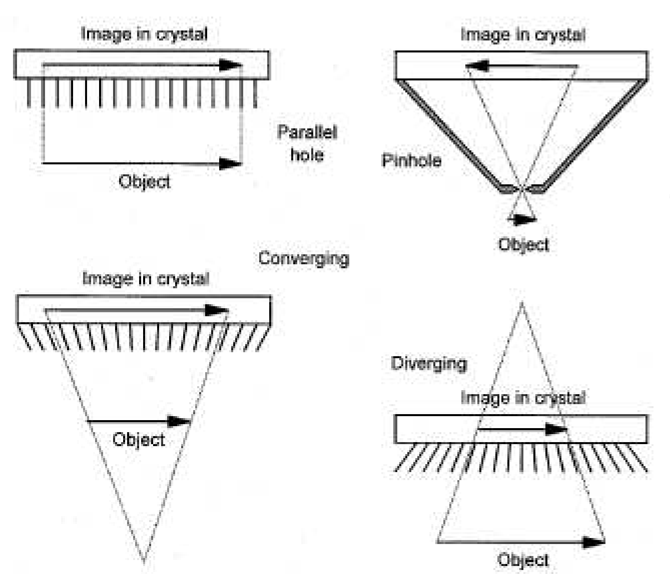


Figura 2.6 - Os quatro tipos de colimadores e suas relações de magnificação de Imagem / Objeto [24]. (a) Furos parelos . (b) *Pinhole* . (c) Convergente . (d) Divergente.

Após a restrição imposta pelo colimador e detecção dos γ´s pelo cristal, a partir da coleta e soma dos valores dos pulsos de corrente elétrica gerados pelo conjunto de PMT´ s, chega-se ao valor da energia e posição do γ incidente. O valor de energia encontrado é analisado pelo circuito PHA (*Pulse Height Analiser* - figura 2.1), o qual discrimina os γ´s detectados. Desse modo, γ´s espalhados com valores de energia fora dos limites de janelas de aceitação são descartados e não contabilizados.

Para a determinação da posição de incidência do γ sobre o detector, utiliza–se um circuito de matriz somativa (figuras 2.1 e 2.7). O sinal de cada PMT é somado em quatro diferentes circuitos, indicando o peso da direção e sentido do sinal (X^+, X^-, Y^+ e Y^-) relativamente no detector. A coordenada de incidência do γ é então calculada pela equação 2.1, onde : (X,Y) é a coordenada do γ incidente; X^+ e X^- a soma dos valores de corrente dos PMT´s na direção X ; Y^+ e Y^- , identico a X , porém na direção Y; k uma contante de proporcionalidade e Z o valor da energia do γ incidente.

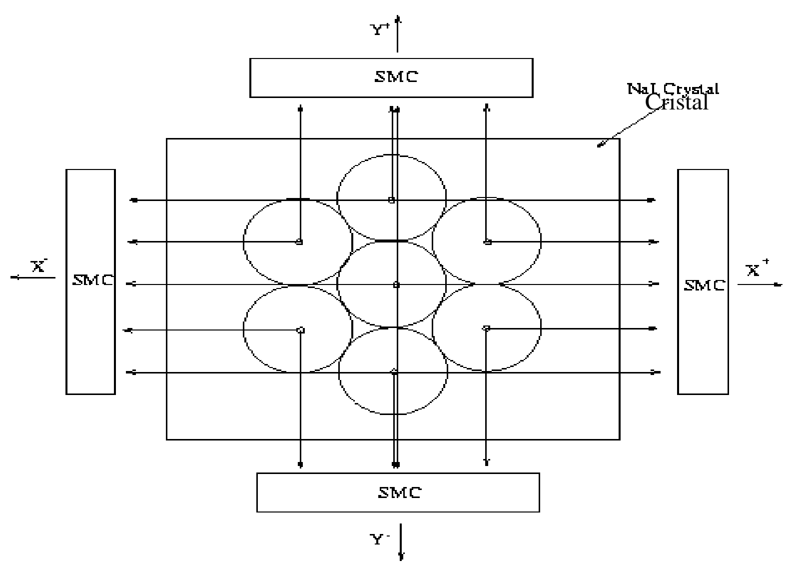


Figura 2.7 - Circuito lógico de posição. Com o pincípio somativo, os sinais dos PMT´s são somados nas quatro direções e, com a equação 2.1, defini-se uma coordenada para o γ incidente.

$$X = k(X^{+} - X^{-}) / Z$$
$$Y = k(Y^{+} - Y^{-}) / Z \qquad \textbf{(2.1)}$$

O sistema para detecção e formação de imagens descrito possui limitações. Mesmo expondo-o a um campo uniforme, não haverá linearidade, manifestada sob a forma dos efeitos *pincushion* e barril, nem uniformidade na imagem (figuras 2.8 e 2.9). Tais distorções se devem as curvas de resposta dos PMT´s e a diferença de resposta entre estas, proveniente da não exata homogeneidade da calibração da alta tensão, como descrito no item 2.1 deste capítulo. Para compensar estas limitações, tabelas (mapas) de correção de linearidade e energia são construídas, as quais atuam sobre as distorções de posição e de espectro de energia dos PMT´s, respectivamente. Estas tabelas compensam as contagens por pixel (elemento de imagem) segundo uma estatística previamente adquirida. Aplicadas em conjunto com um gerador de números aleatórios para manter a característica randômica da aquisição, este artifício de *software* reduz em muito as distorções descritas (figura 2.10).

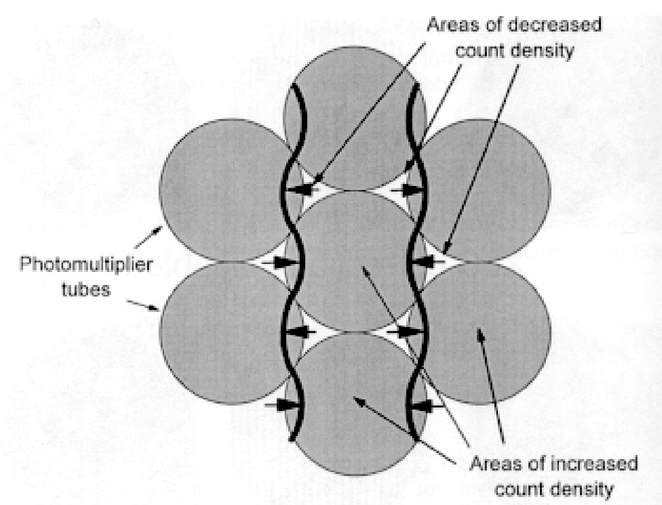


Figura 2.8 - Distorções na linearidade e uniformidade da imagem. As duas linha sinuosas representam duas linhas retas no objeto que foram distorcidas na aquisição. Tais distorções são causadas pela não linearidade das curvas de resposta dos PMT´s e pela diferença de resposta entre estes (25) .

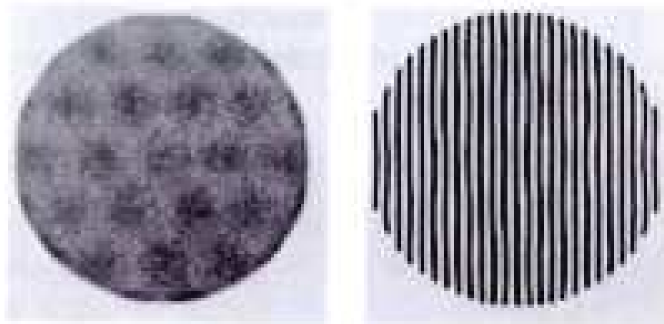


Figura 2.9 - Aquisição de (a) um campo uniforme, (b) phantom de barras com a correção dos mapas desabilitada [26].

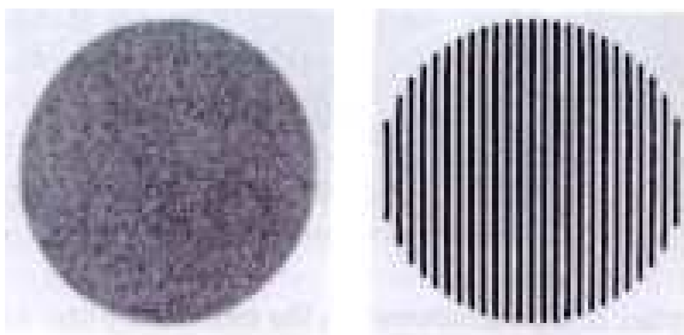


Figura 2.10 - Aquisições apresentadas na figura 2.9 com a aplicação dos mapas de correção [26].

Uma vez identificadas as energias e posições do γ´s incidentes usando-se sinais analógicos, utiliza-se um conversor A/D (Analógico /Digital) para se armazenar em matrizes o número de eventos validados pelo circuito PHA. Os índices dos elementos desta matriz identificam a posição de incidência relativa ao detector e o valor do elemento a atividade. A partir dos elementos da matriz, cria-se a imagem digital, associando-se a estes um nível de cinza ou de cor de uma escala para exprimir tal quantidade visualmente. Os tamanhos para as matrizes mais comumente utilizados são de 64x64, 128x128, 256x256 e 512x512, nas quais pode-se utilizar para armazenar o número de contagens por pixel um byte (256 valores) – chamado de modo *byte* – ou dois (65.536 valores) – chamado de modo *word.* Estes valores serão então armazenados por técnicas comuns em computadores, podendo ser manipulados e processados.

2.3 – Qualidade das Imagens e Fatores Limitantes

A qualidade das imagens em Medicina Nuclear é limitada por vários fatores, alguns dos quais já discutimos no ítem 2.2 deste capítulo. Para caracterizar ou medir a qualidade de uma imagem, utiliza-se a medida de três grandezas : resolução espacial (capacidade de visualizar detalhes ou formas), contraste (diferença na densidade ou intensidade da imagem entre áreas de objetos desta contendo diferentes concentrações de radioisótopo) e a relação sinal/ruído (ruídos devido às flutuações randômicas no decaimento radioativo e no processo de detecção). Todos eles sujeitos a fatores relacionados a equipamento, radiofármacos ou *softwares*.

A resolução espacial é limitada principalmente pela resolução do colimador. Colimadores com furos maiores têm melhor eficiência, porém pior resolução espacial (figura 2.6), ocorrendo um borramento das imagens para colimadores com furos muito grandes. Outros fatores de degradação da resolução espacial são a distância entre paciente e detector, devido a função resposta do sistema ser proporcional a 1/r, onde r é a distância da fonte ao detector, e a energia do γ incidente (γ´s com energias maiores tem taxas de contagem menores, pois, a secção de choque da interação com o cristal diminui – figura 2.11). Além destes fatores externos, o próprio detector tem sua resolução espacial limitada pela espessura do cristal e número de PMT´s , variações estas discutidas nos ítens 2.1 e 2.2 deste capítulo.

Parte dos problemas de contraste de uma imagem estão relacionados com a escolha do radiofármaco utilizado, sendo em geral desejados radiofármacos com alto contraste de lesão. Entre os fatores que degradam o contraste de um objeto, destacam-se a presença de radiação de fundo[2] (BG) e de radiação espalhada ou de penetração pelo septo (dependentes diretamente da energia do γ incidente), fator determinado pelo protocolo de aquisição escolhido. Estes γ´s detectados indesejavelmente, não descartados pelo circuito PHA , degradam a capacidade de visualizar objetos de baixa freqüência espacial com baixo contraste e objetos com alta freqüência espacial. A equação 2.2 define matematicamente o contraste C, onde R_i é a contagem sobre uma área com grande concentração de radiofármaco (denominada área quente) e R_o a contagem sobre uma área com pequena concentração de radiofármaco (denominada área fria).

$$C = (R_i - R_o)/(R_i + R_o) \qquad \textbf{(2.2)}$$

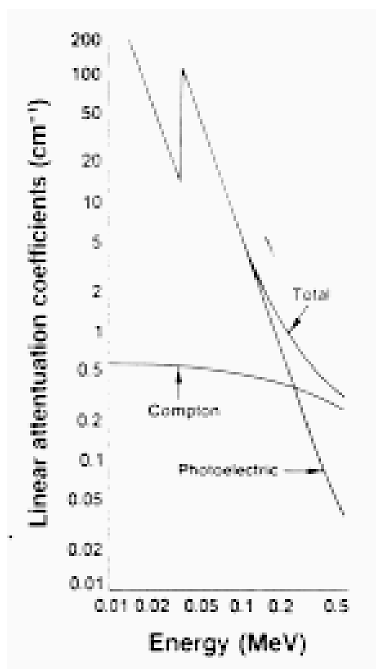


Figura 2.11 – Coeficiente linear de atenuação (cm^{-1}) do cristal de NaI (Tl), onde evidencia-se a predominância do efeito fotoelétrico para as energias utilizadas em Medicina Nuclear [27] .

---

[2] Em Medicina Nuclear entende-se por radiação de fundo ou *back ground* (BG) a radiação proveniente de materiais ou pacientes que não relacionados ao procedimento clínico em andamento.

O ruído está diretamente relacionado com o número de contagens registradas, ou seja, com a densidade de informação da imagem (ID), definida como o número de contagens registradas por unidade de área (contagens / cm^2). Esta grandeza depende da taxa de contagem em cada elemento da imagem e do tempo de aquisição. Pode-se incrementar a ID aumentando-se a taxa de contagem, a atividade, a eficiência do colimador ou ao tempo de aquisição.

A densidade de informação tem um efeito importante no mínimo tamanho e contraste detectável de uma lesão. Para uma lesão de área A, onde a densidade de informação da imagem é ID, o número de contagens registradas N nesta área é de :

$$N = A \times \mathrm{ID} \qquad \textbf{(2.3)}$$

Assim, o desvio padrão das contagens registradas é dado segundo uma distribuição de Poisson [28], por :

$$\sigma = \sqrt{A \cdot ID} \qquad \textbf{(2.4)}$$

ou

$$V_n(\%) = (1/\sqrt{A.ID}).100\% \qquad \textbf{(2.5)}$$

V_n (equação 2.5) é a porcentagem do desvio padrão das contagens registradas ou o contraste do ruído. Assim, para a detecção de uma lesão de área A, deve-se distinguir a flutuação do ruído na imagem (equações 2.4 e 2.5). Normalmente o contraste do ruído mínimo requerido para a detecção de uma lesão é igual a 4 [28]. Portanto, o contraste mínimo é dado por

$$C_{\min} = (4/\sqrt{A.ID}).100\% \qquad \textbf{(2.6)}$$

o que implica que, para lesões menores necessita-se de um contraste de ruído maior – figura 2.12.

Em Medicina Nuclear a densidade de informação (ID) é limitada, pois se usam fontes de relativa baixa atividade (pacientes injetados) e sistemas de detecção com baixa eficiência, principalmente devido a necessidade do uso de colimadores. A ID e, conseqüentemente a eficiência, podem ser aumentadas usando-se colimadores com furos maiores, resultando porém na perda de resolução espacial.

A detecção de objetos pequenos requer longo tempo de exposição, aumentando o número de contagens (ID), diminuindo o V_n e conseqüentemente aumentando o contraste, isto para imagens e sistemas com a mesma resolução espacial. Colimadores de alta resolução provém melhores contrastes e visibilidade na imagem, contudo, a baixa eficiência destes não permite a visualização de detalhes finos.

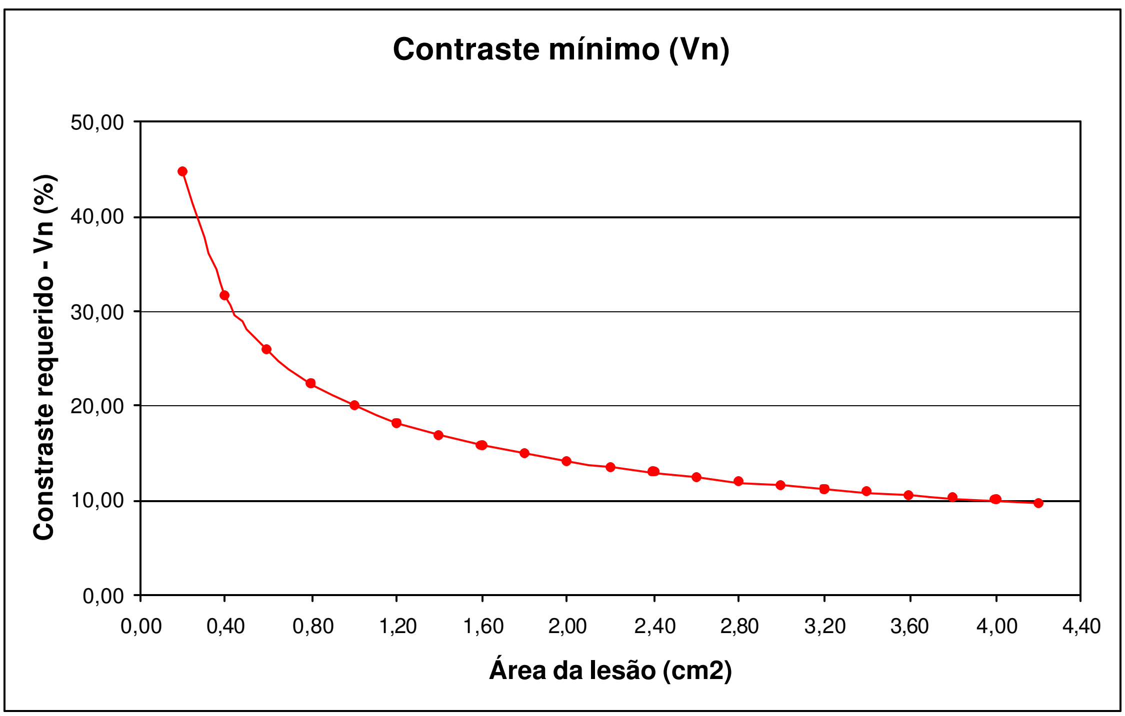


Figura 2.12 – Contraste mínimo requerido (sinal/ruído) para a visibilização de lesões. Dados : ID = 400 cts/cm2 e Cmin = 4.

O que se verifica é que os três fatores limitantes da qualidade de uma imagem estão relacionados, onde a melhora de um deles leva a degradação dos outros. Tal característica impossibilita melhorias com variações significativas na qualidade das imagens apenas em termos de protocolos clínicos ou calibrações de equipamento, principalmente nos estudos com radiofármacos com γ´s de média / alta energia ou estudos com meios espalhadores muito heterogêneos (por exemplo o tórax), objetos de estudo do próximo capítulo.

2.4 – Controle da Qualidade

O controle da Qualidade das câmeras cintiladoras consiste em obter imagens e dados sobre a uniformidade de imagens, resolução espacial e linearidade. Além disso, estas imagens permitem avaliar a sensibilidade do sistema, a presença de elementos que originam artefatos e a performance da eletrônica associada, tais como computadores e interfaces. Os testes podem ser efetuados com duas variáveis : testes intrínsecos (detector sem colimador) e extrínsecos (detector com colimador).

Dentre os testes sugeridos pela IAEA (*International Atomic Energy Agency*), NEMA (*National Electrical Manufacturer´s Association*) e CNEN (Comissão Nacional de Energia Nuclear), entidades que sugerem testes para a avaliação da performance de câmeras cintiladoras, destacam -se :

- *Uniformidade integral intrínseca com várias larguras de janelas (5% , 10% , 15% e 20% do valor da energia do γ incidente)*. O teste avalia a resposta da câmera cintiladora para janelas de aceitação do circuito de PHA com diferentes larguras , verificando a qualidade da calibração da alta tensão nos PMT´s e a eficiência das tabelas de correção – Linearidade / Energia - através da uniformidade apresentada pelo detector quando exposto a uma fonte puntiforme de ^{99m}Tc dista de 5 vezes ou mais o diâmetro do cristal. Espera-se encontrar valores de Uniformidade plana menores que 4% [20]. Exemplos de imagens coletadas no SMN – HC Unicamp para este tipo de avaliação encontram-se na figura 2.13, sendo a uniformidade integral (UI) calculada pela equação 2.7, onde Cmax é o maior valor de contagem encontrado na imagem e Cmin o menor. A imagem é adquirida em matriz [64] e filtrada com um filtro Gaussiano;

$$UI = \frac{(C\max - C\min)}{(C\max + C\min)} \cdot 100\% \qquad \textbf{(2.7)}$$

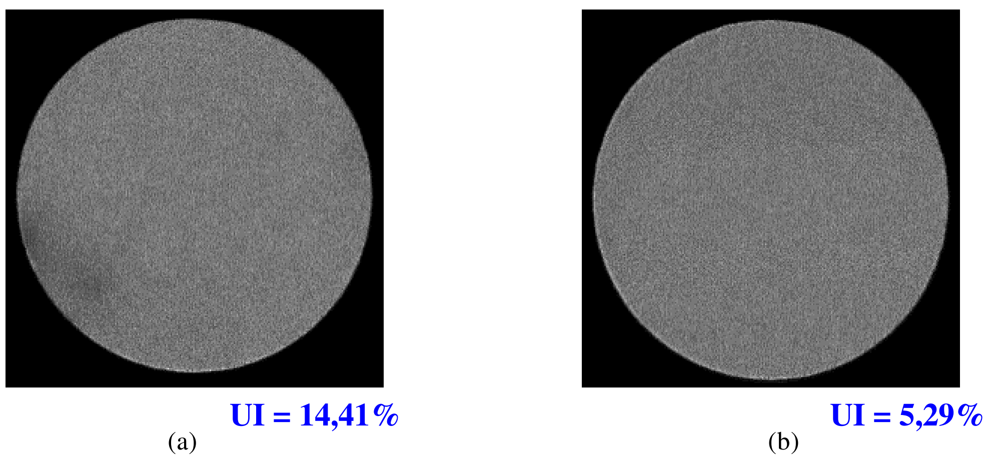
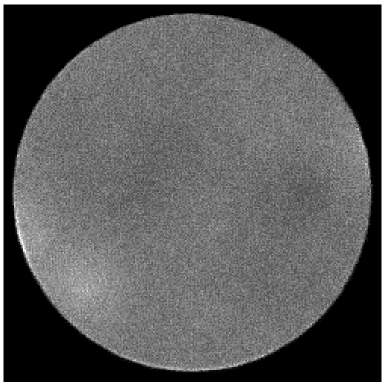


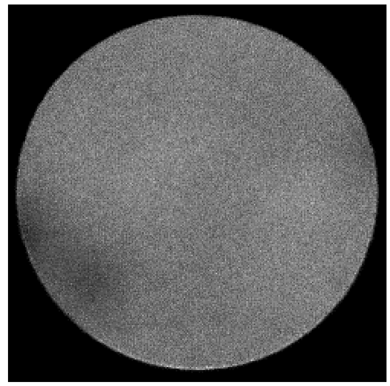
Figura 2.13 – (a) Imagem adquirida por equipamento Elscint APEX SP4 no SMN – HC / Unicamp para teste de uniformidade integral com janela simétrica de 5%. (b) Imagem adquirida por equipamento Elscint APEX SP4 no SMN – HC / Unicamp para teste de uniformidade integral com janela simétrica de 15%.

- *Uniformidade integral intrínseca com janela assimétrica (0 – 10% , 10% - 0).* O teste avalia a resposta da câmera cintiladora para janelas de aceitação assimétricas, verificando-se a qualidade e homogeneidade da calibração da alta tensão das PMT´s através da uniformidade apresentada pelo detector quando exposto a uma fonte puntiforme de ^{99m}Tc dista de 5 vezes ou mais o diâmetro do mesmo. Espera-se alcançar valores de uniformidade próximos entre estes. Exemplos de imagens coletadas no SMN – HC / Unicamp para este tipo de avaliação encontram-se na figura 2.14, sendo o valor de UI calculado pela equação 2.7. Novamente a imagem é adquirida em matriz [64] e filtrada com um filtro Gaussiano;



UI = 24,9%

(a)



UI = 30,47%

(b)

Figura 2.14 – (a) Imagem adquirida por equipamento Elscint APEX SP4 no SMN – HC / Unicamp para teste de uniformidade integral com janela assimétrica de 0 - 10%. (b) Imagem adquirida por equipamento Elscint APEX SP4 no SMN – HC / Unicamp para teste de uniformidade integral com janela assimétrica de 10 - 0%.

- *Resolução espacial do sistema (detector e colimador).* Objetiva determinar a resolução espacial da câmera cintiladora em termos de FWHM para cada colimador disponível. Utilizando-se uma fonte linear (LSP – *line spread func*tion – figura 2.15), adquiri-se imagens com o colimador de interesse com uma matriz 512 x 512 pixels nas direções X e Y a distâncias de 0 e 10 cm entre fonte e detector. Obtém-se então a FWHM da LSP em termos de pixels e, através da relação conhecida de cm/pixel do equipamento, chega-se a resolução espacial do sistema . Exemplos de imagens coletadas no SMN – HC / Unicamp para este tipo de avaliação encontram-se na figura 2.16;

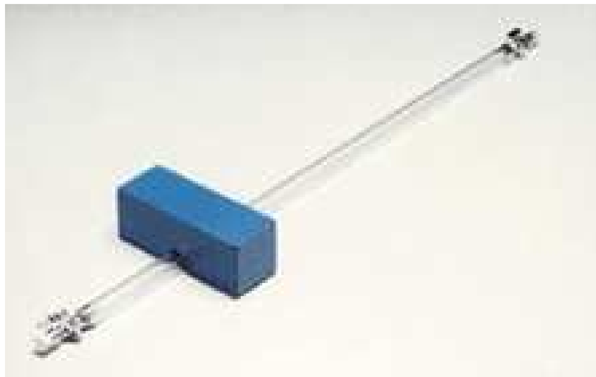


Figura 2.15 – Exemplo de fonte linear com suporte de fixação.

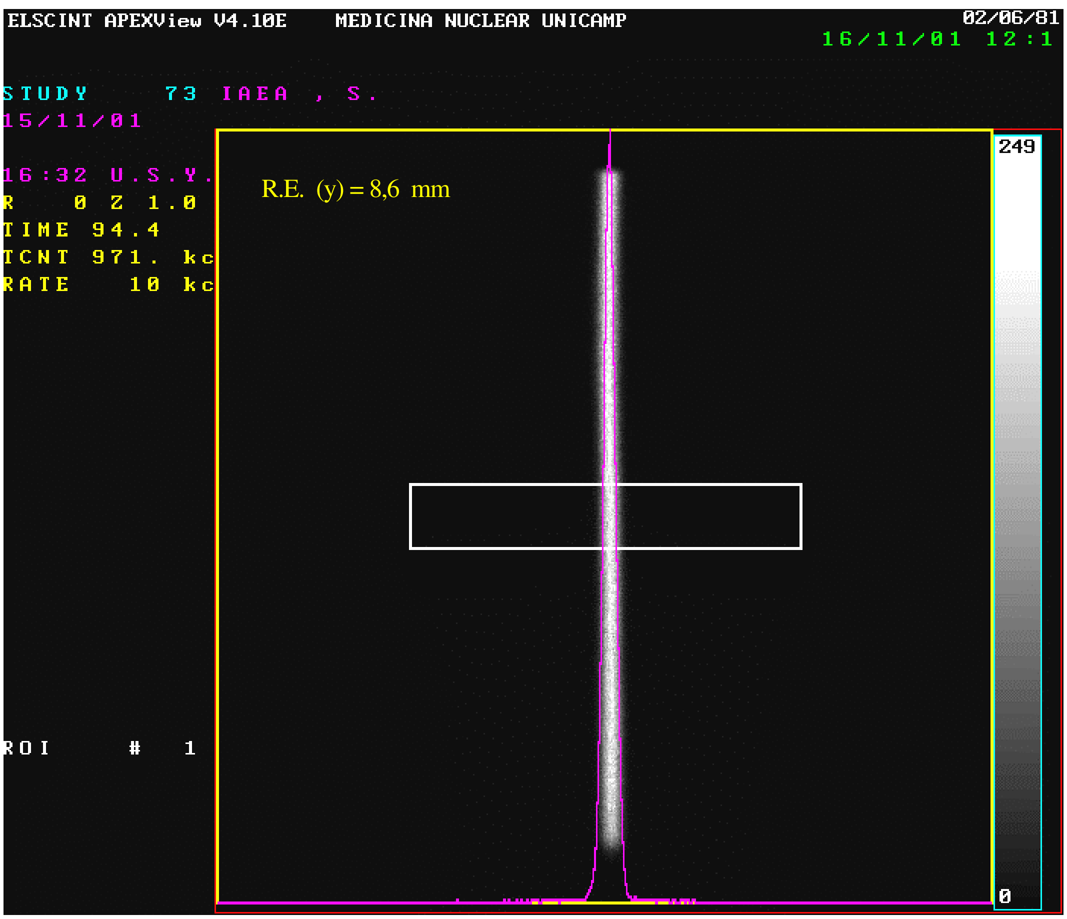
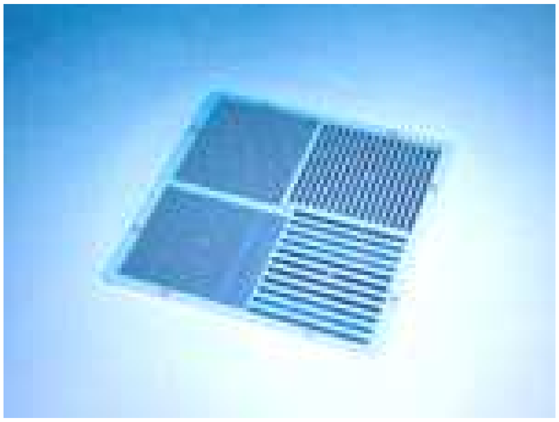


Figura 2.16 – Exemplo de cálculo da resolução espacial na direção Y do equipamento Elscint APEX SP4 com fonte linear – colimador LEHR (*low energy – high resolution*) , distancia fonte / detector =10cm .

- *Resolução espacial do sistema (detector e colimador) e linearidade com phantom de barras (figura 2.17.a) e flood (figura 2.17.b) com uma solução de* ^{99m}Tc . Objetiva determinar a resolução espacial da câmera cintiladora em termos da visualização e distinção de objetos com variadas freqüências espaciais (barras separadas por distâncias iguais a suas larguras) para cada colimador disponível. Utilizando-se um *flood* e um *phantom* de barras , adquiri-se imagens deste conjunto com o colimador de interesse com uma matriz 512 x 512 pixels a 0 cm de distância entre o conjunto e o detector. Determina-se então visualmente qual o objeto de menor freqüência (barras/espaços) possível de se distinguir sem que se forme um borrão. Devido ao desenho do *phantom*, o teste é repetido mais 3 vezes girando-se o mesmo afim de se cobrir as direções X e Y dos 4 quadrantes do detector. Um exemplo de uma imagem adquirida do phantom de barras, para o controle da resolução espacial no SMN –HC Unicamp encontra-se na figura 2.18;



(a)

(b)

Figura 2.17 – (a) Exemplo de *phantom* de barras. (b) Exemplo de *flood*.

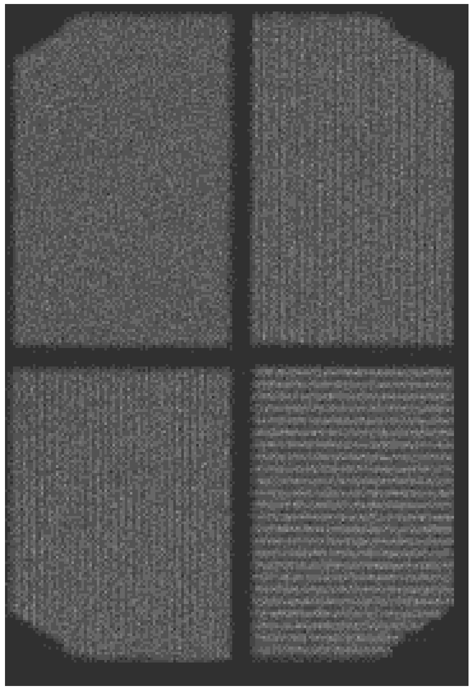


Figura 2.18 – Exemplo de aquisição do *phantom* de barras com *flood* para controle da resolução espacial – equipamento Elscint APEX SP4 .

- *Uniformidade tomográfica do sistema.* O teste avalia a uniformidade integral sobre um corte tomográfico de uma distribuição homogênea de atividade em um volume . Adquire-se uma imagem de um *phantom* de Jaszczak (figura 2.19) e, após a reconstrução dos cortes da região sem *inserts* (figura 2.19), calcula-se a uniformidade tomográfica. O método de cálculo da uniformidade tomográfica não é tão simples quanto a plana (equação 2.7), sendo introduzido no Capítulo 6;

Figura 2.19 – Exemplo de *phantom* Jaszczak com disco de *inserts* quentes e esferas frias.

- *Resolução tomográfica (no ar e com meio espalhador).* O teste avalia a resolução tomográfica da câmera cintiladora no ar e em condições clínicas (com espalhamento – *phantom* com água). Adquire-se um estudo tomográfico de um *phantom* com 3 fontes lineares (figura 2.20) distribuídas no espaço e calcula-se a FWHM destas fontes em alguns cortes . A resolução será a média destas;

- *Performance total.* Avalia-se a câmera cintiladora em condições próximas a de uso clínico, estimando o contraste de objetos de tamanho conhecido. Adquirindo-se imagens e reconstruindo cortes de *phantoms* de Jaszczak com *inserts* frios e quentes (figura 2.19) com todos os recursos disponíveis na câmera (filtros de reconstrução e correção de atenuação) , inspecionando-se o contraste dos objetos e a presença de artefatos – figura 2.21.

Figura 2.20 – Exemplo de *phantom* com 3 fontes lineares.

Figura 2.21 - Exemplo de aquisição e cortes tomográficos reconstruídos do *phantom* Jaszczak para controle da performance total – equipamento Elscint APEX SP4 .

Capítulo 3 – Técnicas Diagnósticas de Baixa Estatística

3.1 – Motivação

Conforme apresentado no início deste trabalho, de todos os fatores que influenciam a qualidade das imagens tomográficas, dedicaremos atenção especial aos algoritmos de reconstrução, comparando o desempenho destes do ponto de vista das aplicações clínicas. Para tanto, utilizou-se aquisições tomográficas cardíacas, MIBI - ^{99m}TC , tomográficas de tórax e cabeça com ^{67}Ga e aquisições de estudos oncológicos de tórax e pélvis com ^{18}F (FlúorDesoxiGlucose – FDG).

A escolha desses protocolos clínicos se fez pela baixa densidade de contagens (ID) das projeções, decorrentes das características próprias dos protocolos. Na prática, as características dos protocolos de aquisição em SPECT´s (tipo de colimador, tempo de exposição, dose administrada ao paciente e radiofármaco) são determinadas por uma otimização entre dose absorvida pelo paciente, limitações de equipamentos e os detalhes e estruturas que se deseja observar nos estudos.

No caso do protocolo cardíaco o baixo número de projeções (30), o meio heterogêneo (tórax), a baixa atividade no órgão alvo, o movimento involuntário do tórax devido a respiração e o baixo tempo de aquisição (~30 a 60 s por projeção), dão origem a projeções com muito ruído que exigem o emprego de filtros suavizadores com baixa freqüência de corte durante a reconstrução. Em virtude disso, só regiões maiores, áreas isquêmicas ou enfartadas, são visualizáveis. Além disso, dependendo dos parâmetros de aquisição e equipamento, podemos ter a presença de artefatos nas imagens reconstruídas. Problemas semelhantes enfrentam os protocolos com ^{67}Ga e

FDG , sendo nesse caso agravados pelo uso de colimadores de média/alta energia que contribuem ainda mais para a degradação da resolução espacial e contraste das estruturas visualizadas, devido as baixas taxas de contagem. Neste capítulo apresentam-se as particularidades clínicas destes protocolos.

3.2 – ^{99m}Tc – MIBI

Os estudos efetuados com o radiofármaco ^{99m}Tc – sestamibi (MIBI) possibilitam visualizar cortes tomográficos do coração. Nas imagens procura–se diagnosticar doenças coronárias, identificando-se lesões e isquemias, investigando a viabilidade miocárdica, podendo também avaliar intervenções terapêuticas como a angioplastia.

O ^{99m}Tc - MIBI é um agente de perfusão, e seu mecanismo de interação com o corpo está relacionado com o fluxo sanguíneo.

O protocolo mais usual utiliza a injeção intravenosa de 20 mCi (740 MBq) à 30 mCi (1110 MBq). O colimador utilizado é o de baixa energia / alta resolução (baixa sensibilidade) para uma E_γ = 140 keV. Durante o estudo são coletadas 30 projeções, em oposição as normalmente 60, num arco de 180º com *step and shoot*[1] de 6º, cada um com tempo de exposição de 30 à 45 segundos. A taxa de contagem para este tipo de protocolo alcança os valores de 5 – 7 k/s, dependendo da sensibilidade do equipamento.

[1] Técnica de aquisição das projeções onde o detector adquire apenas quando posicionado no ângulo de visada. Durante a movimentação do mesmo para incremento angular a aquisição é interrompida.

3.3 – ^{18}F – FDG

O transporte de glicose para dentro das células é mediado por uma família específica de proteínas de transporte. Uma das características bioquímicas de células cancerígenas é a acentuada taxa de metabolismo de glicose devido ao crescimento das proteínas de transporte nestas. O FDG entra nas células pelo mesmo mecanismo que a glicose, porém não sendo metabolizado, acumulando-se nas células, o que identifica, pela intensidade, as células cancerígenas. Infelizmente, o FDG não é um agente específico de câncer, podendo identificar também tuberculose, lesões inflamatórias, infecções e abscessos cerebrais.

As imagens com FDG são feitas com administração intra – venosa de 10 mCi (370 MBq) à 20 mCi (740 Mbq) aos pacientes. A aquisição das 60 projeções é feita em um arco de 360º com *step and shoot* de 6º e seu tempo de duração é aproximadamente de uma hora. O colimador utilizado é o de alta energia / propósito geral para um E_γ = 511 keV , sendo observada uma taxa de contagem de 4 à 7 k/s, dependendo do equipamento.

Por ser feito com um emissor $\beta+$, os estudos com FDG podem ser realizados em câmeras de coincidência ou colimadas, possuindo a primeira uma arquitetura bastante diferente da até aqui descrita. A principal diferença entre estes equipamentos reside no fato de que as câmeras de coincidência não utilizam colimadores, utilizando uma eletrônica associada otimizada para detectar e contabilizar os pares de γ´s provenientes da aniquilação dos pares elétron - positron. No SMN - HC Unicamp os estudos com FDG são realizados em câmeras colimadas.

Este protocolo utiliza também a administração conjunta de ^{99m}Tc, gerando uma imagem multisotópica, onde a câmera se utiliza da aquisição simultânea com dois radiofármacos diferentes para gerar imagens distintas. O objetivo é gerar imagens de referência anatômica com este segundo radioisótopo.

3.4 – ^{67}Ga

Os protocolos executados com este radioisótopo normalmente visam identificar nos pacientes doenças em órgãos localizados no abdômen e tórax, tais como : infecções ou inflamações, febres de origem desconhecida (com duração de mais de 3 semanas) e doenças pulmonares, tendo uma enorme aplicação em pacientes portadores do vírus HIV.

As imagens são adquiridas 48 horas após a administração intra – venosa de 6 (102 MBq) à 10 mCi (370 MBq). Devido as energias do γ´s emitidos ($E_{\gamma 1}$ = 93 keV , $E_{\gamma 2}$ = 185 keV e $E_{\gamma 3}$ = 300 keV), utiliza-se o colimador de média ou alta energia / média sensibilidade. São adquiridas 60 projeções num arco de 360º com *step and shoot* de 6º em aproximadamente 45 minutos com uma taxa de contagem que varia de 1 à 4 k/s. Normalmente junto com o estudo tomográfico, realizam-se imagens estáticas e PCI´s (Pesquisas de Corpo Inteiro).

O mecanismo de interação com o organismo se dá pelo transporte em proteínas com Fe^{+3} (o íon Ga^{+3} substituí o íon Fe^{+3} nestas proteínas). Estas são depositadas em vários tecidos e em vários níveis : córtex renal, fígado, glândulas salivares e lacrimais, pulmões, ossos e nódulos linfáticos.

Estudos abdominais com ^{67}Ga apresentam muitos problemas, pois o radiofármaco deposita-se no bolo fecal que pode apresentar-se como área anormal na imagem. Normalmente ocorre a repetição dos estudos estáticos 72 horas após a administração para a verificação da movimentação do bolo, descartando-se então a possibilidade de lesões ou doenças.

Capítulo 4 – Algoritmos de Reconstrução Tomográfica

4.1 – Projeções e Retro - Projeção

A partir da manipulação matemática de imagens planas adquiridas pelas câmeras de cintilação – as projeções, obtém-se os cortes tomográficos. A solução mais simples para o problema **de,** a partir de imagens planas se obter secções de corte de um volume, passa pela projeção e retro - projeção da distribuição da densidade de radiação do objeto – figura 4.1. A densidade de cada ponto da distribuição é somada sobre uma linha para cada ângulo **de visada**, gerando as projeções . Em seguida, as mesmas são retro – projetadas sobre o plano, reconstruindo a distribuição do corte seccional.

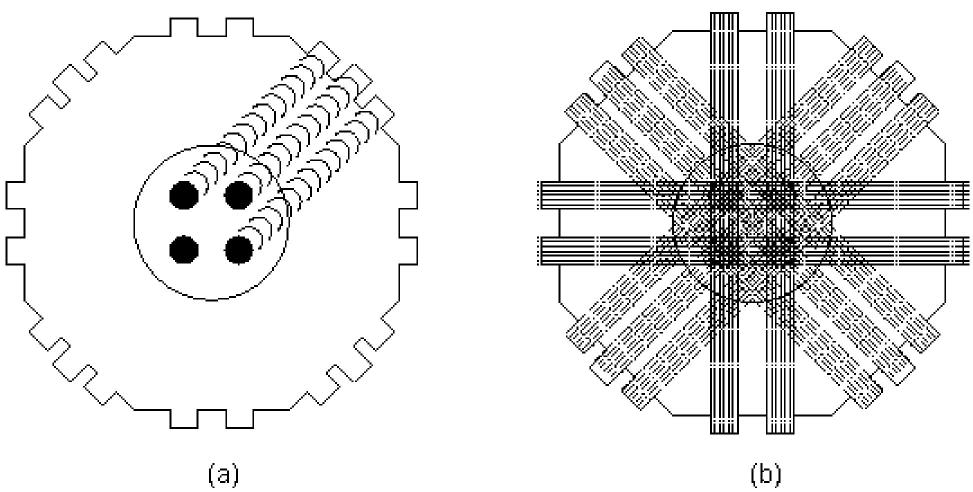


Figura 4.1 – (a) Projeção dos valores de densidade de radiação em vários ângulos de visada (0 / 45 / 90 / 135 180 / 225 / 270 / 305 / 360º) , construindo os perfis. (b) Retro – projeção dos perfis , reconstruindo a distribuição de radiação.

A distribuição da atividade do radiofármaco se encontra em um sistema inercial de coordenadas. Porém, o detector utilizado para as aquisições das projeções gira em torno do paciente, sendo, portanto, um sistema não inercial – figura 4.2. Cada fatia do volume dá origem a uma projeção, que corresponde a uma linha do detector.

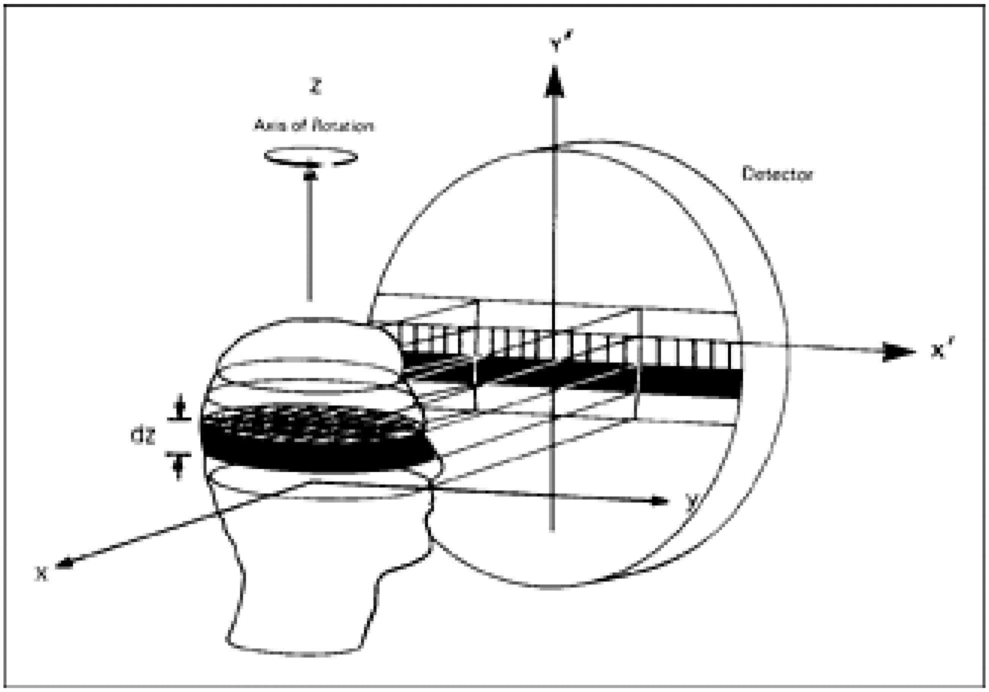


Figura 4.2 - Sistema de coordenadas para a detecção de γ 's em exames ECT (*Emission Computed Tomography*[29])

Em 1917 o matemático tcheco Radon definiu pela primeira vez matematicamente a projeção [8]. Dada uma função f (x,y) que representa uma densidade de radiação no plano (figura 4.3), chama-se de raio a linha que corre sobre a mesma e de projeção $P(\theta_i,t)$ (ou $P_{\theta i}(t)$) ao conjunto de integrais de linha de f (x,y) sobre cada raio, sendo o valor da projeção dado por:

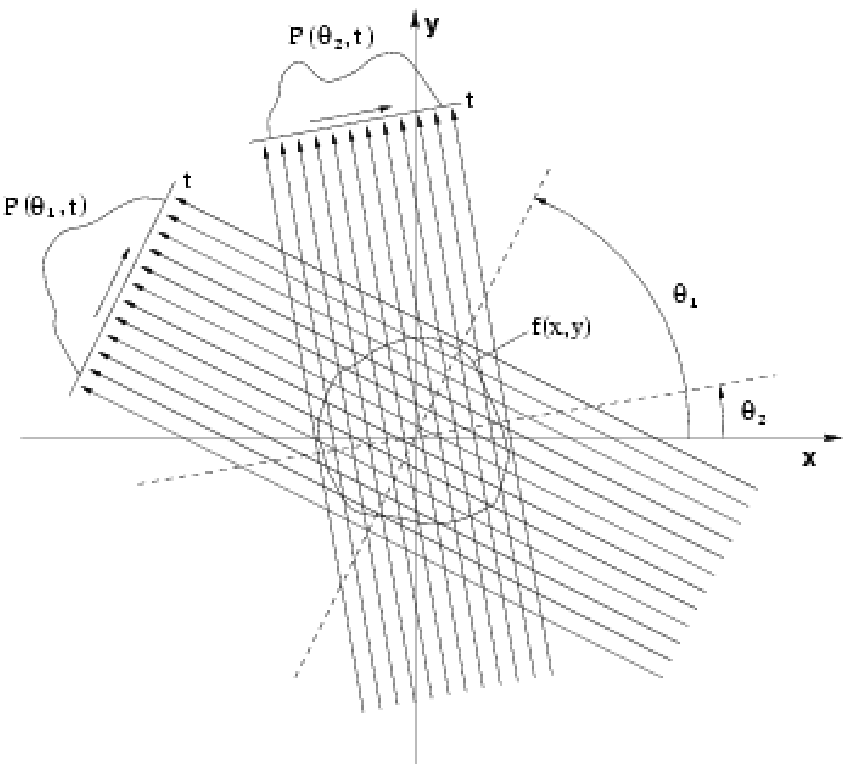


Figura 4.3 – Definição dos parâmetros geométricos da projeção.

$$P_{\theta_i}(t) = \int_{AB} f(x, y)ds = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x, y)\delta(xCos\ \theta_i + ySin\ \theta_i - t)dxdy \qquad \textbf{(4.1)}$$

Onde a transformação de coordenadas é dada por :

$$xCos\ \theta + ySin\ \theta = t \qquad \textbf{(4.2)}$$

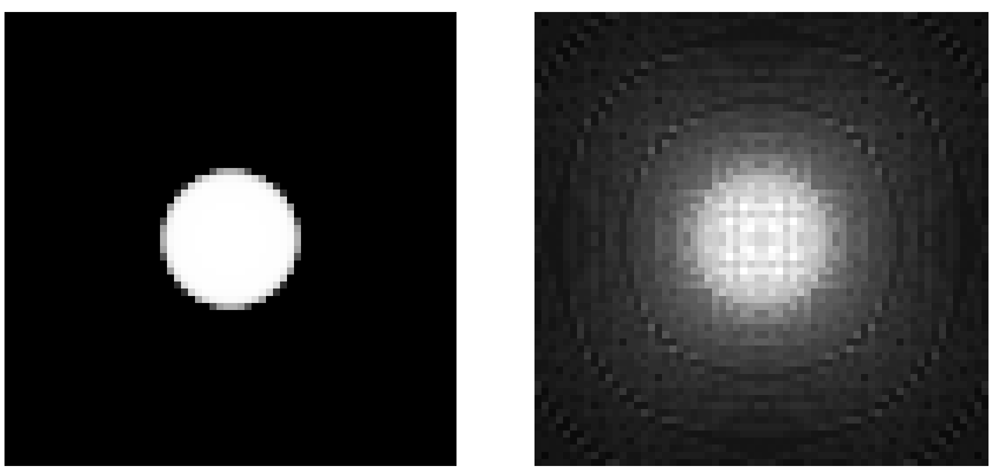
Mas f (x,y) é uma função contínua e requer uma quantidade infinita de projeções para ser reconstruída. Na prática , f (x,y) é calculada com um número finito de pontos de um número finito de projeções (M), ou seja, ocorre uma discretização da função f(x,y).

A discretização de f (x,y) exige uma interpolação dos valores retro – projetados, sendo a resolução da imagem dada pelo número de elementos (pixels) utilizados. Para se realizar a operação de interpolação com uma velocidade adequada durante as retro - projeções, constroem–se matrizes cujos os índices dos elementos (n,m,j) indicam sobre qual *pixel* (n,m) o raio (j) deve ser interpolado. O valor do elemento define o peso da interpolação onde, num caso mais simples, o peso igual a 1 indicaria que o raio retro – projetado passa pelo pixel e igual a 0 (zero) não. Estas matrizes e sua construção serão discutidas em detalhes no Capítulo 5.

Discretizando a função f(x,y) e reconstruindo-na a partir de um número finito de projeções (M), obtém–se a transformada de Radon inversa discretizada (equação 4.3), a qual obviamente não retorna f(x,y), mas sim uma função reconstruída f_R (x,y).

$$f_R(x,y) = \sum_{i=1}^{M} P_{\theta_i}(xCos\,\theta_i + ySin\,\theta_i, \theta_i)\Delta\theta \qquad \textbf{(4.3)}$$

A retro-projeção descrita pela equação 4.3 gera um artefato na imagem, conhecido como efeito estrela (borrões raiados em torno do objeto reconstruído – observados na figura 4.1) . A presença destes borrões pode ser melhor compreendida em termos da teoria de sistemas lineares.



(a) (b)

Figura 4.4 - (a) *Phantom* matemático - corte de um cilindro preenchido com uma distribuição homogênea de atividade . (b) Imagem 4.4.a reconstruída com retro – projeção, onde se evidencia o efeito estrela.

A imagem observada (I) na distribuição reconstruída é produto da convolução entre a imagem real da distribuição (TI) e a função resposta do sistema de aquisição *(Apêndice A)*. A função resposta do sistema de detecção é uma função de ponto espalhado (PSF – *Point Spread Function*), proporcional a 1/r, onde r é a distância entre a fonte e o detector. Portanto :

$$I = TI * 1 / r \qquad \textbf{(4.4)}$$

Para se eliminar os artefatos do tipo estrela , deve-se aplicar um filtro nos perfis antes da retro-projeção. Para tanto, deve-se tomar as transformadas de Fourier ($\mathfrak{F}$) *(Apêndice B)* das funções da equação 4.4 e manipula-las algebricamente :

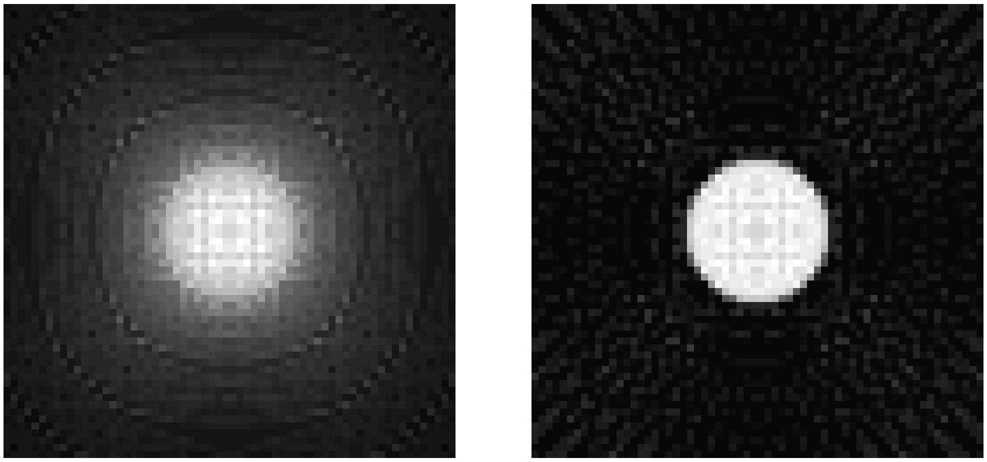
$$\mathfrak{F}\{I\} = \mathfrak{F}\{TI * 1/r\}$$
$$\mathfrak{F}\{I\} = \mathfrak{F}\{TI\}.\, \mathfrak{F}\{1/r\}$$
$$\mathfrak{F}\{TI\} = \mathfrak{F}\{I\} / \mathfrak{F}\{1/r\}$$

mas : $\mathfrak{F}\{1/r\} = 1/\nu \quad \Rightarrow \quad \mathfrak{F}\{TI\} = \mathfrak{F}\{I\}.\, \nu$

$\therefore\; TI = I * g$, onde : $g = \mathfrak{F}^{-1}\{\nu\}$

A função ν no espaço das freqüências dá-se o nome de filtro rampa, pois seu valor é incrementado linearmente com a freqüência.

A esta técnica de manipulação para redução de artefatos tipo estrela se dá o nome de Retro-Projeção Filtrada (*Filtered Back Projection* – FBP), a qual será detalhada nos próximos itens deste capítulo. Um exemplo do efeito da aplicação do filtro rampa sobre a imagem 4.4.b será mostrado na figura 4.5.



(a) (b)

Figura 4.5 - (a) *Phantom* matemático - corte de um cilindro preenchido com uma distribuição homogênea de atividade reconstruído com retro – projeção (Imagem 4.4.b) . (b) *Phantom* matemático - corte de um cilindro preenchido com uma distribuição homogênea de atividade reconstruído com retro – projeção filtrada – filtro rampa.

4.2 – Teorema do Corte Central (*Fourier Slice Theorem*)

Para a implementação do algoritmo de FBP, se faz necessária antes a demonstração do teorema do corte central, o qual relaciona a $\mathfrak{I}^{1}\{P_{\theta i}(t)\}$ com $\mathfrak{I}^{2}$?$\{f(x,y)\}$, demonstrando a possibilidade de se reconstruir f(x,y) através de $P_{\theta i}(t)$.

Partindo-se das definições das transformadas de Fourier de f(x,y) (equação 4.5) e das projeções adquiridas $P_\theta(t)$ (equação 4.6), calcula–se F(u,0), ou seja, a transformada de Fourier de f(x,y) ($\mathfrak{I}^{2}\{f(x,y)\}$) sobre uma linha paralela ao eixo u (v = 0) (equação 4.7).

$$F(u,v) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y)e^{-2i\pi(ux+vy)}dxdy \qquad \textbf{(4.5)}$$

$$S_{?}(w) = \int_{-\infty}^{\infty} P_\theta(t)e^{-2i\pi tw}dt \qquad \textbf{(4.6)}$$

$$F(u,0) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y)e^{-2i\pi ux}dxdy = \int_{-\infty}^{\infty}[\int_{-\infty}^{\infty} f(x,y)dy]e^{-2\pi iux}dx \quad \textbf{(4.7)}$$

Mas para θ = 0, ou seja v = 0, x = t nos planos das coordenadas :

$$\Rightarrow P_\theta(t) = \int_{-\infty}^{\infty} f(x,y)dy \quad \textbf{(4.8.a)}$$

$$\therefore F(u,0) = \int_{-\infty}^{\infty} P_\theta(t)e^{-2\pi iwt}dt = S_\theta(w) \quad \textbf{(4.8.b)}$$

A equação 4.8.b mostra que F (u,0) pode ser obtida através de $S_\theta(w)$. Pode-se generalizar este resultado para que F(w,θ) denote o valor de F(u,v) ao longo de uma linha com ângulo θ qualquer com o eixo u (F (w,θ) = S_θ (w)), onde w e θ são as coordenadas polares no plano das freqüências. Para tanto, faz-se f(t,s) = f (x,y) rodada de θ (vide figura 4.3), onde se tem a transformada de coordenadas dadas a seguir, substituindo-se as coordenadas (x,y) por (t,s) em (4.8.a) e em (4.8.b):

$$\begin{pmatrix} t \\ s \end{pmatrix} = \begin{pmatrix} \text{Cos }\theta & \text{Sin }\theta \\ -\text{Sin }\theta & \text{Cos }\theta \end{pmatrix} . \begin{pmatrix} x \\ y \end{pmatrix}$$

$$\Rightarrow P_\theta(t) = \int_{-\infty}^{\infty} f(t,s)ds \quad \textbf{(4.9.a)}$$

$$\therefore S_\theta(w) = \int_{-\infty}^{\infty} P_\theta(t)e^{-2\pi iwt}dt = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(t,s)e^{-2\pi iwt}dsdt \quad \textbf{(4.9.b)}$$

Tomando-se t $\Rightarrow$ x e s $\Rightarrow$ y, obtém -se o teorema do corte central (4.10):

$$S_\theta(w) = \int_{\infty}^{\infty}\int_{\infty}^{\infty} f(x,y)e^{-2\pi iw(xCos\theta + ySin\theta)}dxdy = F(u,v) \qquad \textbf{(4.10)}$$

Onde : $u = wCos\,\theta$
$v = wSin\,\theta$

A equação 4.10 diz que a $\mathfrak{F}^{2}$\{f(x,y)\}= F(u,v) sobre uma linha radial no plano das freqüências é idêntica a $\mathfrak{F}^{1}$\{P_θ(t)\} = S_θ(w) no ângulo de visada θ.

Tomando-se as transformadas de Fourier das projeções ($S_{\theta i}$(w)) para um número infinito de ângulos, pode-se determinar F(u,v) sobre um número infinito de linhas radiais, o que implica no conhecimento de todos os pontos do plano *u-v* , possibilitando a obtenção direta de f(x,y) (equação 4.11).

$$\mathfrak{F}^{-2}\ \{\ F(u,v)\ \} = f(x,y) \qquad \textbf{(4.11)}$$

Em termos computacionais, se f (x,y) é limitada em - A/2 < x < A/2 e - A/2 < y < A/2 e um número finito de componentes da transformada de Fourier são conhecidos , podemos tomar f_R (x,y) como :

$$f_R(x,y) = \frac{1}{A^2}\sum_{m=-N/2}^{N/2}\sum_{n=-N/2}^{N/2} F\left(\frac{m}{A},\frac{n}{A}\right)\exp\left[2\pi i\left(\frac{m}{A}x + \frac{n}{A}y\right)\right] \qquad \textbf{(4.12)}$$

para x e y definidos no intervalos descritos acima .

Na equação 4.12, N é um número inteiro o qual define a resolução espacial da imagem reconstruída, podendo esta ser implementada usando-se um algoritmo de DFT (*Discrete Fourier Transform*).

Além disso, a intenção de se manipular as projeções no espaço das freqüências (equações 4.9 e 4.10) se deve a facilidade computacional. A operação de convolução entre funções no espaço das coordenadas torna-se uma simples multiplicação no espaço das freqüências, muito mais veloz em termos computacionais.

Deve-se notar também que os valores de $f_R(x,y)$ devolvidos pela equação 4.12 ainda devem ser interpolados sobre o plano da imagem.

4.3 - Algoritmos Analíticos - FBP para Aquisições com Colimadores Paralelos

4.3.1 - Teoria

Partindo das definições :

- w e θ são as coordenadas polares no plano *u-v* dadas pela equação 4.10;
- a função da distribuição f (x,y) pode ser reconstruída conhecendo-se todos os pontos do plano *u-v* dada pela equação 4.11, ou seja, a transformada inversa de Fourier $\mathfrak{F}^{-2}\{F(u, v)\}$ e
- pelo teorema do corte central, F (u,v) pode ser obtida através das transformadas de Fourier das projeções, ou seja, $F(u,v) = S_\theta(w) = \mathfrak{F}^{1}\{P_\theta(t)\}$ dadas pelas equações 4.10 e 4.9 , pode–se escrever a equação 4.11 em coordenadas polares, onde, automaticamente um termo $|w|$ (filtro rampa) multiplicando $F(w,\theta)$ surge da transformação das coordenadas – equação 4.13.

$$f(x,y) = \int_{0}^{2\pi}\int_{-\infty}^{\infty} F(w,\theta)\, e^{2\pi i w (xCos\ \theta + ySin\ \theta)}\, w\, dw\, d\theta \qquad \textbf{(4.13)}$$

Abrindo–se o limite de integração em θ, de 0 a π na equação 4.13, têm -se:

$$f(x,y) = \int_{0}^{\pi}\int_{-\infty}^{\infty} F(w,\theta)\, e^{2\pi i w (xCos\ \theta + ySin\ \theta)}\, w\, dw\, d\theta + \int_{0}^{\pi}\int_{-\infty}^{\infty} F(w,\theta+\pi)\, e^{2\pi i w (xCos\ (\theta+\pi) + ySin\ (\theta+\pi))}\, w\, dw\, d? \qquad \textbf{(4.14)}$$

Sendo F (w , θ + π) = F (- w , θ) e utilizando a equação 4.10:

$$f(x,y) = \int_{0}^{\pi}\Big[\int_{-\infty}^{\infty} F(w,\theta)\,|w|\, e^{2\pi i w (xCos\ \theta + ySin\ \theta)}\, dw\Big] d\theta = \\ = \int_{0}^{\pi}\Big[\int_{-\infty}^{\infty} S_\theta(w)\,|w|\, e^{2\pi i w (xCos\ \theta + ySin\ \theta)}\, dw\Big] d\theta \qquad \textbf{(4.15)}$$

Fazendo :

$$Q_\theta(t) = Q_\theta(xCos\ \theta + ySin\ \theta) = \int_{-\infty}^{\infty} S_\theta(w)\,|w|\, e^{2\pi i w (xCos\ \theta + ySin\ \theta)}\, dw \qquad \textbf{(4.16)}$$

e portanto :

$$f(x,y) = \int_{0}^{\pi} Q_\theta(xCos\ \theta + ySin\ \theta)\, d\theta \qquad \textbf{(4.17)}$$

A equação 4.15 (Q_θ (t)) nada mais é que uma operação de filtragem sobre as projeções P_θ (t) no espaço das freqüências, na qual as componentes de alta freqüência são incrementadas em proporção ao seu valor devido a presença do filtro rampa, produzindo uma função bifásica com valor médio igual a zero[30], atenuando o artefato tipo estrela, porém, aumentando os problemas de ruído da imagem .

Para tornar a equação 4.17 passível de implementação em computadores, devemos discretizar todos os elementos envolvidos, para tal, tomando algumas aproximações :

- tendo um objeto o período τ, a partir do teorema da amostragem ($\tau = 1 / 2.f_N$), define-se a freqüência de Nyquist (f_N) como sendo a menor freqüência necessária para a amostragem deste objeto;
- $W = f_N$;
- considera-se S_θ (w) como banda limitada, ou seja, $S_\theta (w) = S_\theta (w) . b(w)$, onde:

$$b_w (w) = 1 \qquad |w| \leq W$$
$$= 0 \qquad \text{qualquer outro valor;}$$

- N = nº inteiro / nº de pixels , definindo a resolução espacial da imagem;
- Têm-se um número finito de amostras (N + 1) , k = - N / 2 ... 0 ... N / 2 ;
- discretiza-se $w = 2 W m / N$, m = - N / 2 ... 0 ... N / 2 .

Usando as afirmações acima , reescreve-se a equação 4.6 de forma discretizada:

$$S_\theta \left(m \frac{2W}{N}\right) = \frac{1}{2W} \sum_{k=-N/2}^{N/2} P_\theta \left(\frac{k}{2W}\right) e^{-2\pi i (mk/N)} \qquad \textbf{(4.18)}$$

A equação 4.18 representa a DFT de P_θ (t), podendo ser calculada com um algoritmo de FFT. Estes algoritmos escrevem uma DFT de N elementos como a soma de duas DFT´s de N/2 elementos. Um dos vetores de N/2 elementos é composto pelos elementos pares do vetor original de N elementos e o outro, pelos elementos ímpares, desde que N seja uma potência de 2. Utilizando-se a idéia da reversão de bits [31], processa-se a transformada, gerando uma economia de tempo proporcional ao número de operações : N^2 para a DFT e $N.\log_2(N)$ para a FFT.

A partir de 4.18, reescreve-se a equação das projeções filtradas Q_θ (t) (equação 4.16) para o caso discreto, utilizando-se as mesmas aproximações de 4.18:

$$Q_\theta\left(\frac{k}{2W}\right) = \frac{2W}{N} \sum_{m=-N/2}^{N/2} S_\theta\left(\frac{k}{2W}\right) \left| m \frac{2W}{N} \right| e^{2\pi i (m k/N)} \quad \textbf{(4.19.a)}$$

desde que S_θ (w) seja banda limitada.

Do ponto de vista de ruídos, uma imagem de melhor qualidade em termos de uniformidade pode ser obtida utilizando-se uma janela tipo *Hanning* ou *Butterworth*:

$$Q_\theta\left(\frac{k}{2W}\right) = \frac{2W}{N} \sum_{m=-N/2}^{N/2} S_\theta\left(\frac{k}{2W}\right) \left| m \frac{2W}{N} \right| G\left(\frac{m}{N} 2W\right) e^{2\pi i (m k/N)} \quad \textbf{(4.19.b)}$$

onde : G(2Wm/N) é uma função janela. O propósito das janelas é de atenuar as altas freqüências, reforçadas pelo filtro rampa. Os ruídos das imagens apresentam altas freqüências.

$$\therefore \; f_R(x, y) = \frac{\pi}{K} \sum_{i=1}^{K} Q_{\theta_i}(xCos\;\theta_i + ySin\;\theta_i) \qquad \textbf{(4.20)}$$

onde : K = número de projeções.

Os valores de Q_θ na equação 4.20 não são os mesmos de 4.19, mas sim valores interpolados para a retro-projeção. Para todo ponto (x,y) no plano da imagem há um correspondente valor de t (x Cosθ + y Sinθ) para dado valor de θ. A contribuição de $Q_{\theta i}$ (t) para a reconstrução do ponto (x,y) depende de t (figura 4.6). As imagens são reconstruídas sobre planos (matrizes quadradas), o que implica que o valor atribuído ao pixel na posição (x,y) recebe um valor aproximado de Q_θ (t), utilizando-se uma interpolação linear.

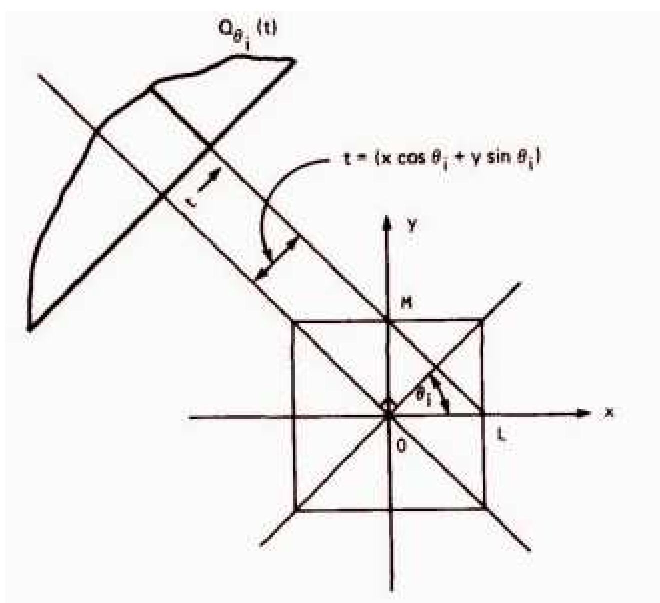


Figura 4.6 – Na reconstrução, a projeção filtrada Q_θ (t) tem a mesma contribuição para todos os pontos (*pixels*) sobre a linha LM no plano x-y [32]. O valor atribuído (Q_θ (t)) ao *pixel* é aproximado por interpolação.

4.3.2 - Aproximações e suas Implicações no Método FBP

A teoria apresentada é extremamente atraente do ponto de vista de implementação, bastando realizar as DFT´s das projeções $P_\theta(t)$, multiplica-las pelo filtro rampa ($|w|$) no espaço das freqüências, realizando a operação de convolução e realizar a DFT´s inversas, obtendo-se $Q_\theta(t)$. Contudo, a equação 4.19 tem suas limitações:

- considera-se S_θ (w) banda limitada, o que na realidade não acontece e
- w tem variação finita , ($|w| < W$), o que implica em ordem finita, condição necessária para realizar a DFT.

Estas duas condições devem ser verdadeiras e ocorrer simultaneamente para possibilitar a operação da convolução periódica-circular no espaço das freqüências em 4.19. Caso as duas aproximações não sejam satisfeitas simultaneamente, são criados artefatos de interperíodos [30], criados quando uma convolução não periódica é implementada como periódica .

A primeira afirmação significa que o teorema da amostragem deve ser satisfeito : $W \geq 1 / 2\tau = 2B$ (figura 4.7). Caso não o seja, acontecem os artefatos tipo *Aliasing*, onde há uma sobreposição das réplicas de $S_\theta(w)$ na equação 4.19. No exemplo da figura 4.7 a freqüência de amostragem W é menor que a necessária B, por isso ocorre o artefato *Aliasing*.

Da segunda afirmação ($|w| < W$) , tem-se que $|w|$ não possui uma $\mathfrak{F}^{-1}$, o que implica em uma função não periódica, convoluindo com uma função periódica ($S_\theta(w)$) , resultando em uma função não periódica. Além disso, o método utilizado para a discretização da freqüência w, hipótese utilizada para a construção da equação 4.18, implica que toda a informação vizinha de w = 0 contida no domínio contínuo da freqüência é anulada (caracterizado pelo valor de m = 0), que neste caso ocorreria apenas em w = 0.

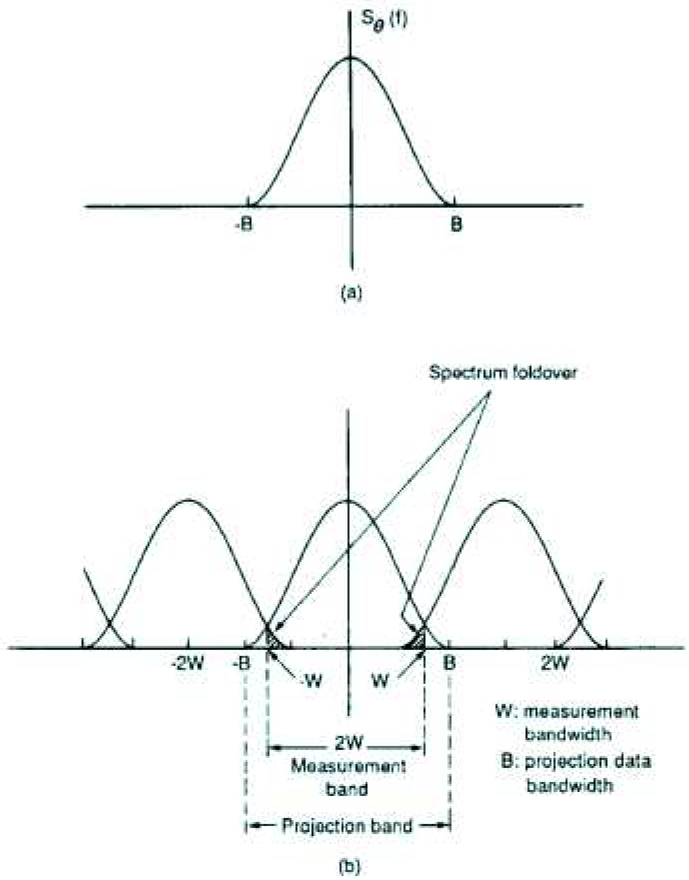


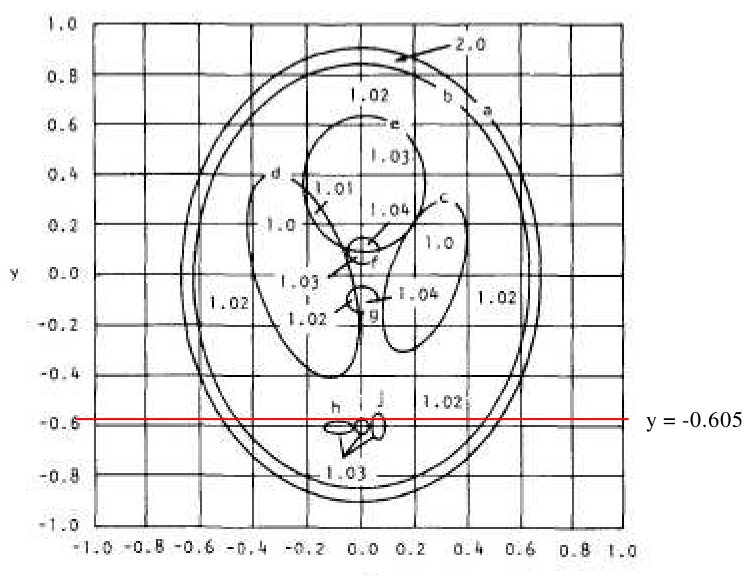
Figura 4.7 – (a) Projeção com banda limita, com freqüência B. (b) Se a projeção é amostrada (W) com freqüência menor que a necessária (B), ocorrem interferências entre as réplicas utilizadas na equação 4.19 (*Aliasing*) [(33)].

Estes artefatos são minimizados utilizando-se duas técnicas : *Zero Padding* [(34)] (ZP) e a modificação da função do filtro rampa [(30)], operando-se então duas funções periódicas na convolução.

Utilizando-se a equação 4.19 para reconstruir um *phantom* matemático de cabeça (Sheep e Logan), pode-se comparar o perfil da linha em y = -0.605 do objeto e da imagem reconstruída (figura 4.8). Percebe-se neste a presença de um artefato que curva o perfil reconstruído e reduz os valores do mesmo em relação ao perfil do objeto. Estes artefatos são provenientes parcialmente da convolução da equação 4.18 e parcialmente das informações anuladas em m = 0.

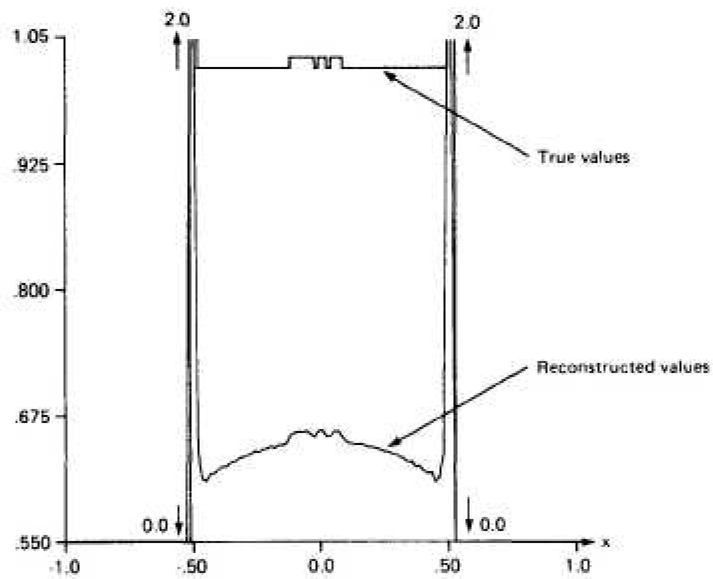
A adição de zeros nos vetores de $P_\theta(t)$ e do filtro rampa | w |, minimizam os efeitos provocados pelo *Aliasing*. P_θ (t) é normalmente amostrado com um número N potência de 2 (64,128 etc.) e | w | possui N+1 elementos (m = -N/2 ... 0 ... N/2). Assim, para a execução da DFT e da convolução no espaço das freqüências, deve-se adicionar N zeros ao vetor de P_θ (t) e N –1 zeros em | w |, onde o vetor produto da convolução terá 2N elementos [34]. Isto ocorre para que se possa realizar uma convolução periódica–circular, onde as funções iniciais têm o mesmo número de elementos da função resultante. Assim, os valores de interesse na função de saída encontram-se entre os elementos de posição N/2 à 3/2 N . O perfil da reconstrução do *phantom* da figura 4.8 com a adição de ZP é mostrado na figura 4.9, onde o abaulamento presente na figura anterior praticamente desaparece, porém ocorre ainda a redução dos valores do perfil reconstruído em relação ao perfil do objeto.

(a)



(b)

Legenda na pg. 57



(c)

Figura 4.8 – (a) – *Phantom* Sheep e Logan[35]. (b) Esquema de construção do *phantom* Sheep e Logan [35]. (c) Perfis da linha y = - 0.605 – valores reais e reconstruídos com a equação 4.19, apresentando abaulamento e deslocamento em relação ao real. Dados : matriz 128x128 e 110 projeções [36].

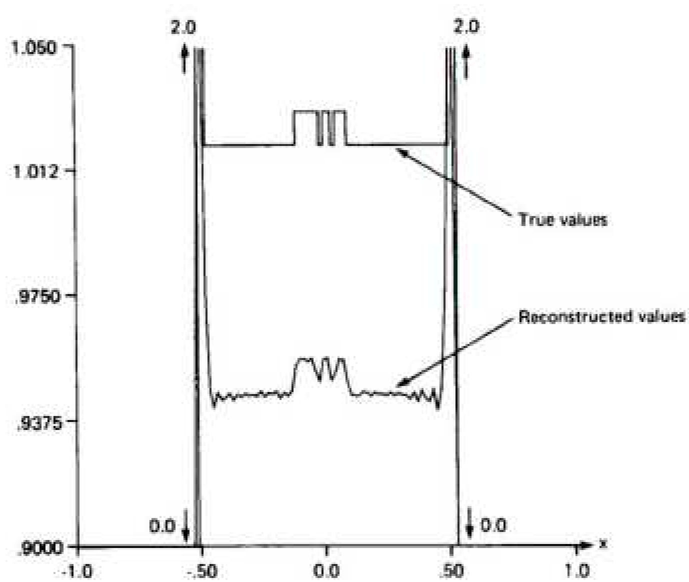


Figura 4.9 – Comparação dos perfis real e reconstruído do *phantom* Sheep e Logan da figura 4.8 com adição de ZP [36] .

Os artefatos introduzidos pela aproximação utilizada na equação 4.18 (discretização com o elemento de vetor m = 0), podem ser eliminados se esta não for utilizada, mas sim outro caminho for utilizado para discretizar a mesma.

Quando a mais alta freqüência nas projeções P_θ (t) for finita (W =1/2 τ, onde τ = intervalo de amostragem no espaço das coordenadas, ou seja, W = freqüência de Nyquist), a equação 4.16 pode ser escrita como(30) :

$$Q_\theta(t) = \int_{-\infty}^{\infty} S_\theta(w) H(w) e^{2\pi iwt}\, dw \qquad \textbf{(4.21)}$$

onde : H (w) = | w | . b_w (w) e b_w (w) definida como para a equação 4.18.

H (w) é a função (figura 4.10) de transferência, ou seja, a transformada de Fourier da função resposta de um sistema, no caso o filtro com a qual as projeções serão convoluídas . A função resposta h (t) é dada por (30):

$$h(t) = \int_{-\infty}^{\infty} H(w) e^{2\pi iwt} dw = \frac{1}{2\cdot\tau}\cdot\frac{Sin(2\pi t/2\tau)}{2\pi t/2\tau} - \frac{1}{4\cdot\tau^2}\cdot\left(\frac{Sin(2\pi t/2\tau)}{2\pi t/2\tau}\right)^2 \qquad \textbf{(4.22)}$$

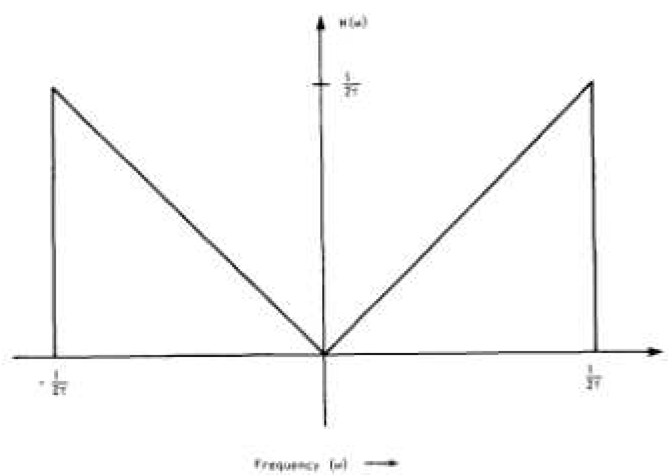


Figura 4.10 – Função de transferência H (w) (37).

Desde que as projeções P_θ (t) são medidas com um intervalo de amostragem τ, para o processamento digital de h(t) basta conhecer a mesma no intervalo de amostragem . As amostras, h ($n\tau$), de h (t) são dadas por :

$$h(n\tau) = \begin{cases} 1/4\tau^2, & n = 0 \\ 0, & n = \text{par} \\ -1/n^2\pi^2\tau^2, & n = \text{ímpar (figura 4.11)} \end{cases}$$

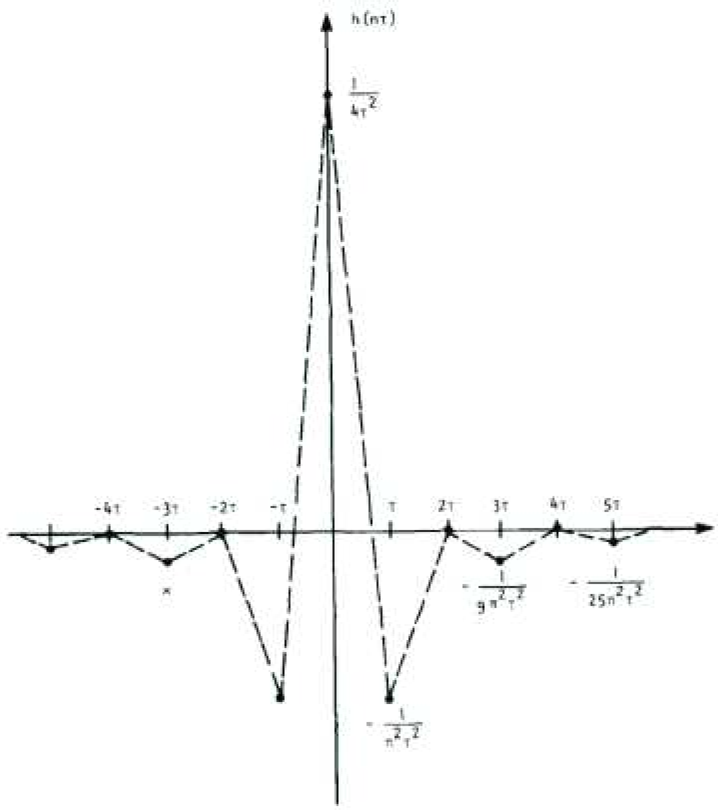


Figura 4.11 – Função resposta h ($n\tau$) (38).

A função h(nτ), diferentemente de | w | não irá a zero quando m = 0 (figura 4.12), o que resolve o problema descrito anteriormente.

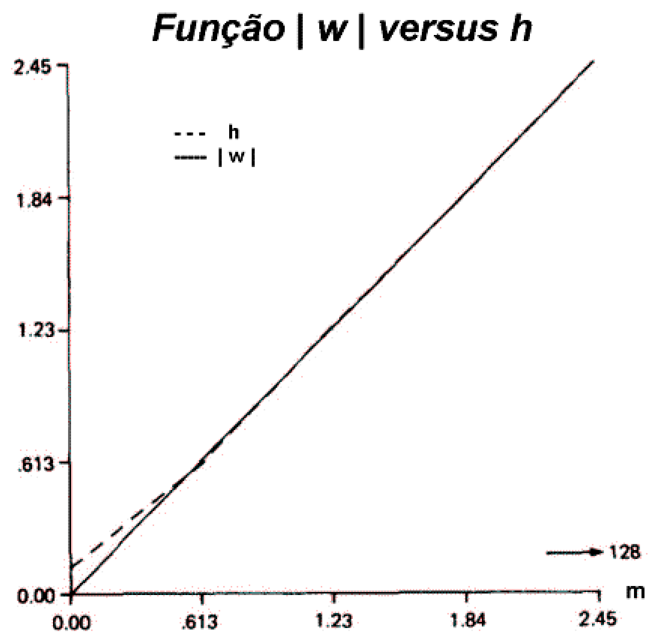


Figura 4.12 – Comparação entre as funções | w | e h (nτ) [39].

Além disso, desde que tanto $P_\theta(t)$ e h(t) são agora funções banda–limitada, estas podem ser escritas na forma discreta :

$$P_\theta(t) = \sum_{k=-\infty}^{\infty} P_\theta(k\tau) \cdot \frac{Sin(2\pi W(t-k\tau))}{2\pi W(t-k\tau)} \qquad \textbf{(4.23)}$$

$$h(t) = \sum_{k=-\infty}^{\infty} h(k\tau) \cdot \frac{Sin(2\pi W(t-k\tau))}{2\pi W(t-k\tau)} \qquad \textbf{(4.24)}$$

a convolução no espaço das coordenadas pode então ser escrita :

$$Q_\theta(t) = \int_{-\infty}^{\infty} P_\theta(t')\,h(t-t')\,dt \qquad \textbf{(4.25)}$$

Substituindo as equações 4.23 e 4.24 em 4.25 tem-se a convolução das projeções com a função h(nτ) no espaço das coordenadas :

$$Q_\theta(n\tau) = \tau \cdot \sum_{k=-\infty}^{\infty} P_\theta(k\tau)\,h(n\tau - k\tau) \qquad \textbf{(4.26)}$$

Na prática, cada projeção tem uma extensão finita. Fazendo o índice k = - (N-1) ... 0 ... N-1, têm-se :

$$Q_\theta(n\tau) = \tau \cdot \sum_{k=-(N-1)}^{N-1} P_\theta(n\tau - k\tau)\,h(k\tau) \qquad \textbf{(4.27)}$$

onde : n = 0,1,2 ... N-1.

A equação 4.27 poderia ser implementada diretamente, mas para fins computacionais trabalhar no espaço das freqüências, como dito anteriormente, é muito mais vantajoso em termos de velocidade. Utilizando um algoritmo de FFT e a técnica de ZP, implementa-se a equação 4.27 no espaço das freqüências como descrito na equação 4.28.

$$Q_\theta(n\tau) = \tau \cdot IFFT\ \{FFT\ \{P_\theta(n\tau)comZP\ \} \cdot FFT\ \{h(n\tau)comZP\ \}\} \qquad \textbf{(4.28.a)}$$

ou

$$Q_\theta(n\tau) = \tau \cdot IFFT\{FFT\{P_\theta(n\tau)comZP\} \cdot FFT\{h(n\tau)comZP\} \cdot janela\} \qquad \textbf{(4.28.b)}$$

Aplicando–se a equação 4.28 para se obter as projeções filtradas ($Q(n\tau)$) , pode-se reconstruir novamente os cortes tomográficos através da equação 4.20. No exemplo utilizado nas figuras 4.8 e 4.9 (*phantom* Sheep e Logan), o efeito da utilização da equação 4.28 pode ser percebido na figura 4.13, onde o abaulamento inexiste e a redução nos valores também, restando apenas o ruído matemático da reconstrução.

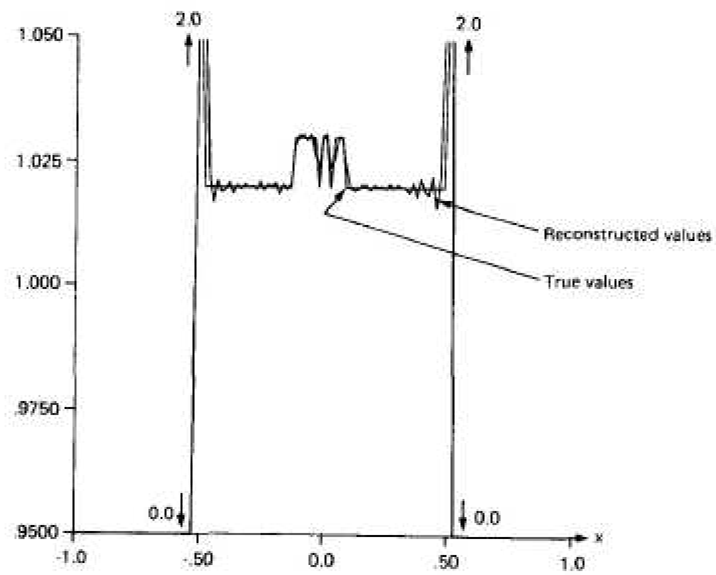


Figura 4.13 – Perfis do *phantom* Sheep e Logan (real) e reconstruído (equação 4.28) [40].

Desse modo, o algoritmo de FBP neste trabalho será implementado seguindo as equações 4.28 e 4.20.

4.4 - Algoritmos Algébricos (Iterativos)

Diferentemente do método analítico (FBP), os métodos iterativos de reconstrução de imagens estão baseados na solução de um sistema de equações. O problema resume-se em resolver a equação 4.3 para a função densidade da distribuição de radiação.

Impondo-se uma matriz quadrada sobre a distribuição de densidade de radiação f (x,y) (figura 4.14), considera-se o valor de cada célula constante. Têm-se que:

- f_i denota o valor da i - ésima célula no j – ésimo raio;
- N é o número total de células e
- w_{ij} é o peso (pode ser a área de intersecção, distância do centro do raio ao centro da célula ou comprimento do raio que passa pela célula) da i - ésima célula no j - ésimo raio para a projeção p_j .

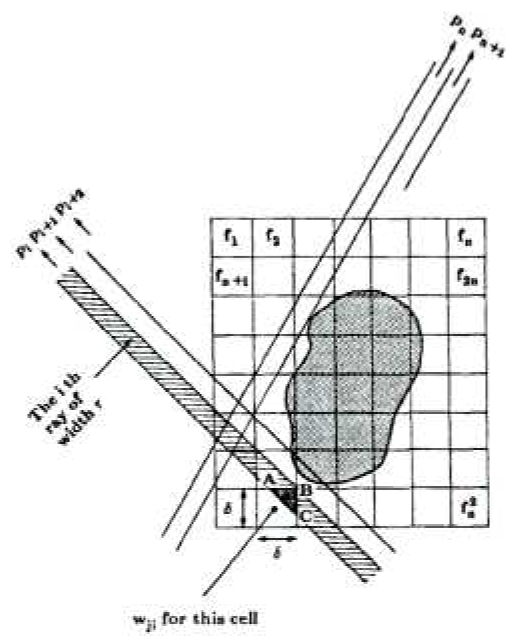


Figura 4.14 – Distribuição de densidade de radiação f(x,y) com a imposição da matriz [41].

Um raio agora não é apenas uma linha, tem largura também, normalmente a mesma largura do lado do quadrado da matriz imposta, pois agora o raio também define a resolução espacial da imagem.

A j – ésima projeção (p_j) pode ser obtida como mostrado na figura 4.14. A relação entre as projeções e os valores das células é dada por:

$$p_j = \sum_{i=1}^{N} w_{ij} f_i \qquad \textbf{(4.29)}$$

onde : j = 1,2 ... M , sendo M o número total de raios em todas as projeções.

Neste trabalho adotar-se-á w_{ij} como sendo o comprimento do raio j que passa dentro da célula i. Nota-se que muitos destes valores são iguais a zero (vide figura 4.14), havendo apenas um pequeno número de células que contribuem para o valor de p_j.

A imagem da distribuição de radiação f (x,y) pode ser obtida solucionando o sistema sugerido pela equação 4.29, invertendo-se as matrizes :

$$f_i = \sum_{j=1}^{M} w_{ij}^{-1} p_j \qquad \textbf{(4.30)}$$

Porém, este método tem algumas restrições que limitam o seu uso:

- se M < N o sistema terá várias soluções ;
- devido a ruídos e outros artefatos, os dados das projeções podem ser inconsistentes, o que levaria a uma não convergência da solução;
- a matriz W que tem como elementos valores de w_{ij}, é muito grande: para imagens de 64x64 pixels (4096 pixels) implicaria em uma

matriz de $17x10^6$ elementos, o que dificultaria bastante a manipulação do sistema.

Para transpor as limitações do método da inversão das matrizes, aplica-se o método iterativo. O procedimento consiste em se iniciando com um conjunto de valores f_i escolhidos, repete-se os cálculos de p_j até que estes convirjam (truncamento da precisão requerida) para os valores das projeções medidas, aplicando-se sucessivas correções.

Como métodos iterativos a serem estudados, optou-se neste trabalho pelas técnicas ART (*Algebric Reconstruction Technique*) e SIRT (*Simultaneous Iterative Reconstrtuction Technique*), descritas a seguir.

A correção aplicada as células a cada iteração é dada por [42] :

$$f_i^{\,l} = f_i^{\,l-1} + \sum_{j=1}^{M} \Delta f_{ij}^{\,l} \qquad \textbf{(4.31)}$$

Onde :

- $f_i^{\,l-1}$ é a densidade da i - ésima célula na interação anterior (l -1) a interação l e

- $\Delta f_{ij}^{\,l}$ é a correção aplicada a i - ésima célula do j - ésimo raio para a interação l .

Os tipos de correção (Δf_{ij}) utilizadas são a aditiva (4.32.a) e a multiplicativa (4.32.b), definidas como [42] :

$$\Delta f_{ij}^{\,l} = w_{ij} \Delta p_j \Big/ \sum_{i=1}^{N} w_{ij}^{\,2} \qquad \textbf{(4.32.a)}$$

$$\Delta f_{ij}^{\,l} = f_i^{\,l-1} \Delta p_j \Big/ p_j^{\,c} \qquad \textbf{(4.32.b)}$$

onde defini-se:

$$\Delta p_j = p_j - p_j^c$$

p_j = projeção medida ;

p_j^c = projeção calculada .

4.4.1 - Algoritmos Iterativos - ART

Também chamada de correção raio-a-raio, a técnica de reconstrução ART segue as seguintes etapas para a reconstrução do corte tomográfico:

1) Partindo-se de uma matriz com valores iniciais f_i, calcula-se as projeções (p_j^c) para dado ângulo de visada com a equação 4.29;
2) Com os valores de p_j^c calculados em (1), calcula-se as correções para as células da imagem em cada raio j com a equação 4.32, encontrando-se os valores de Δp_j;
3) corrigi-se os valores das células f_i da imagem utilizando-se a equação 4.31;
4) repete-se o procedimento para as projeções em outros ângulos de visada com os valores de f_i calculados nas projeções anteriores, ou seja, no ângulo anterior, até se realizar as correções para todos os ângulos de visada, repetindo-se o processo até que se alcance a precisão requerida.

4.4.2 - Algoritmos Iterativos - SIRT

Também chamada de correção ponto-a-ponto, a técnica de reconstrução SIRT segue as seguintes etapas para a reconstrução dos corte tomográficos :

1) Partindo de uma matriz com valores iniciais de f_i, calcula-se os p_j^c da célula i com a equação 4.29 ;
2) com um valor ponderado de $\Delta f_{ij}^{\ 1}$ calculado pelas equações 4.33.a (correção aditiva) ou 4.33.b (correção multiplicativa) , corrigi-se os valores de f_i (equação 4.31) , retornando-se ao cálculo de p_j^c até que os M raios sejam feitos.

$$f_i^{\,l} = f_i^{\,l-1} + \frac{\sum_{j=1}^{M} p_j \cdot w_{ij}}{\sum_{j=1}^{M} L_j \cdot w_{ij}} + \frac{\sum_{j=1}^{M} p_j^c \cdot w_{ij}}{\sum_{j=1}^{M} N_j \cdot w_{ij}} \qquad \textbf{(4.33.a)}$$

$$f_i^{\,l} = f_i^{\,l-1} \cdot \left(\frac{\sum_{j=1}^{M} p_j \cdot w_{ij}}{\sum_{j=1}^{M} L_j \cdot w_{ij}} \right) \cdot \left(\frac{\sum_{j=1}^{M} N_j \cdot w_{ij}}{\sum_{j=1}^{M} p_j^c \cdot w_{ij}} \right) \qquad \textbf{(4.33.b)}$$

onde :

- L_j = o comprimento do raio j no ângulo de visada para a célula i calculada no momento em unidades de d^2/W (figura 4.15);
- N_j = número de pixels contidos no raio j para o ângulo de visada .

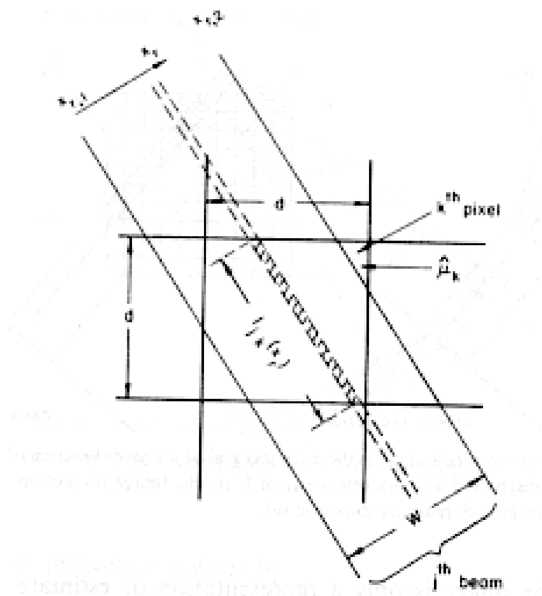


Figura 4.15 – Definição das quantidades usadas nas equações 4.33 [43].

4.4.3 - Algoritmos Iterativos - Considerações

Para a implementação dos algoritmos iterativos, deve-se destacar alguns aspectos:

1) *Valores iniciais* – Os pixels f_i da matriz da imagem a ser reconstruída, deve ter um valor inicial para que se possa realizar as iterações. Um dos esquemas mais utilizados para valores iniciais de f_i é o de $f_i^0 = \overline{f}$ [43], onde $\overline{f}$ é dado por:

$$\overline{f} = \frac{1}{K \cdot N} \cdot \sum_{j=1}^{M} p_j \qquad \textbf{(4.34)}$$

onde : K = número de projeções e N = número de elementos da imagem.

2) *Critério de convergência* – Para se definir o momento de parar as iterações, além do número de iterações a executar, pode-se utilizar critérios de convergência. Entre estes o de medida da Entropia (S) [43] (equação 4.35.a) e Desvio Padrão (SD) (equação 4.36.a). A Entropia S da imagem é uma medida das alterações observadas na imagem a cada iteração. Quando as alterações nos valores f_i´s da imagem ocorridos de uma iteração (l -1) para a outra (l) são menores que as determinadas por α (parâmetro de controle – equação 4.35.b), então a imagem convergiu para a distribuição f(x,y). Já o desvio padrão (SD) verifica esta quantidade entre as projeções calculadas e as adquiridas . O valor de SD utilizado para verificar a convergência é o maior deste em todos os cálculos das projeções (onde se define-se **sig** na equação 4.36.b), garantindo-se a convergência desta grandeza, controlada também por um parâmetro de controle α entre as iterações l e l-1 (equação 4.36.c).

$$S^{l} = -\frac{1}{\ln N} \cdot \sum_{i=1}^{N} \frac{f_i^{\,l}}{\overline{\overline{f}}} \cdot \ln\left(\frac{f_i^{\,l}}{\overline{\overline{f}}}\right) \qquad \textbf{(4.35.a)}$$

$$| S^{l+1} - S^{l} | < \alpha \cdot S^{l} \qquad \textbf{(4.35.b)}$$

$$SD = \sqrt{\frac{\left(p_j - p_j^{c}\right)^2}{K}} \qquad \textbf{(4.36.a)}$$

$$sig^{l} = \max\{ SD^{l} \} \qquad \textbf{(4.36.b)}$$

$$| sig^{l+1} - sig^{l} | < \alpha \cdot sig^{l} \qquad \textbf{(4.36.c)}$$

4.5 – Comparação entre os Algoritmos

A tabela 1 [42] resume o desempenho computacional para os três algoritmos estudados. Pode-se perceber porque ainda hoje o FBP é o mais utilizado dos algoritmos.

Método	**Nº de multiplicações**	**Nº de multiplicações para imagem 64 x 64**
FBP	~ m N	193.000
SIRT	4 M N	49.413.000
ART	4 M n	983.000

Tabela 4.1 – Quadro comparativo entre os desempenhos computacionais dos algoritmos de reconstrução de cortes tomográficos.

Como detalhado na *Motivação* deste trabalho, o objetivo é comparar o desempenho dos algoritmos do ponto de vista das aplicações clínicas, buscando responder se as técnicas algébricas (ART e SIRT), embora às expensas de maiores recursos computacionais, apresentam melhores resultados que o FBP.

Capítulo 5 – Implementação dos Algoritmos de Reconstrução Tomográfica

5.1 – Características dos Programas e Funções Implementadas

A partir dos resultados e equações desenvolvidas no capítulo anterior, implementa-se os algoritmos de reconstrução tomográfica neste trabalho seguindo as seguintes equações:

- Interpolação (matrizes de pesos - $\boldsymbol{W}$) : equação 4.2;
- FBP : equações 4.20 e 4.28;
- ART : equações 4.29, 4.31, 4.32 e 4.34;
- SIRT : equações 4.29, 4.31, 4.32, 4.33 e 4.34.

Para a reconstrução dos cortes tomográficos foram implementados, além dos algoritmos de reconstrução (FBP, ART e SIRT), mais cinco programas auxiliares : ***framecut*** **,** ***matrizW***, ***bp***, ***bpw*** e ***fbpw*** , cujas funções serão detalhadas a seguir.

Todos os algoritmos (com exceção do ***matrizW***) foram implementados sob a forma de rotinas do aplicativo *Octave* (versão 2.1.31 rodando em sistema operacional *Linux – Madrake 7.2*). Este aplicativo de interface matemática trabalha com uma linguagem de alto nível, permitindo a programação e implementação de algoritmos bastante complexos em forma de funções, utilizando uma sintaxe muito semelhante a da linguagem C. Dentre os motivos para a escolha desta plataforma para a implementação dos algoritmos destacam-se: *software free*, manuseio e programação simples e, dentre os pacotes inclusos no *software* algoritmos de FFT (fft) e FFT inversa (ifft) (testes realizados sobre os algoritmos – *Anexo C*).

Já o programa para cálculo dos pesos para as interpolações (***matrizW***) foi implementado em linguagem C com a geração de um executável, pois se faz necessária a utilização de matrizes tri – dimensionais, recurso que o *Octave* não possui. Para a compilação deste programa utilizou-se o compilador *gcc* versão 2.95 do sistema operacional *Mandrake 7.2 – Linux*.

Dentre as limitações impostas pelas máquinas utilizadas (processadores de 750MHz com 256 Mb de Ram), destacam-se o tempo de processamento e o tamanho máximo das matrizes das imagens suportadas. O tempo de processamento para o algoritmo de reconstrução FBP é de 0,9 horas e para os algébricos (ART e SIRT) de até 1,5 horas por iteração. Já o tamanho das matrizes das imagens é limitado a [64], pois, matrizes maiores não são suportadas pelos sistemas. Esta limitação advém do tamanho das matrizes de peso $\boldsymbol{W}$ (uma única matriz possui 262.144 elementos para imagens de [64] e 2.097.152 elementos para matrizes [128] , requerendo precisão dupla, o que não é suportado pelo sistema no caso de matrizes [128]). Na prática, a segunda característica limita a resolução espacial das imagens.

5.2 – Funções e Inter-relação dos Algoritmos Implementados

Encontra–se a seguir a descrição da função de cada algoritmo implementado com seus respectivos programas auxiliares e o relacionamento entre estes para a reconstrução dos corte tomográficos (figura 5.1) :

- ***framecut.m*** (anexo D1) – as projeções adquiridas pelas câmeras cintiladoras são normalmente armazenadas em arquivos de 512x512 *pixels* ([512]), contendo 60 projeções de [64]. Este programa foi gerado para fazer o corte das 60 projeções em matrizes de [64], separando-os em matrizes distintas e passando seu código de cores para a escala de cinzas de 8 *bits*, construindo um sinograma[1] da linha do corte a ser reconstruído;

[1] Sinograma = matriz / imagem construída a partir de uma mesma linha de todos as projeções. Possuí o aspecto de uma função senoidal devido a transformação do sistema de coordenadas na câmera cintiladora (figura 4.2).

- ***matrizW.c*** (anexo D2) – programa responsável pelo cálculo das matrizes de pesos (**W**) que darão as coordenadas ponderadas na interpolação dos raios no plano de reconstrução. Dois programas auxiliares foram implementados para verificação deste primeiro: ***bp.m*** e ***bpw.m*** – algoritmos de reconstrução de cortes tomográficos por retro - projeção sem a utilização de filtros (equação 4.3). O primeiro apenas retro-projeta os raios sem levar em conta o quanto do seu comprimento passa pelo pixel (peso), enquanto que o segundo executa a retro-projeção multiplicando os valores das projeções de cada raio pelos respectivos pesos. As coordenadas e pesos de cada raio para cada ângulo de visada estão armazenados nas matrizes calculadas pelo programa ***matrizW.c*** ;
- ***fbp.m*** – programa de reconstrução de cortes tomográficos que utiliza a técnica analítica FBP, ocorrendo a retro-projeção das projeções filtradas com a interpolação dada pelos pesos das matrizes **W**. Também se implementou um programa auxiliar (***fbpw.m* -** anexo D3) que, quando da retro-projeção, há a multiplicação pelos pesos. Além do filtro rampa, pode-se efetuar com estes algoritmos a filtragem da imagem com uma janela do tipo *Butterworth*, com opções de três parâmetros de controle;
- ***art.m*** (anexo D4) - programa de reconstrução de cortes tomográficos que utiliza a técnica algébrica ART. Este programa permite reconstruir os cortes com as opções de correção aditiva ou multiplicativa e critérios de convergência por número de iterações, Entropia ou Desvio Padrão;
- ***sirt.m*** (anexo D5) - programa de reconstrução de cortes tomográficos que utiliza a técnica algébrica SIRT. O programa ***sirt.m*** possibilita a reconstrução de cortes com as opções de correção aditiva ou

multiplicativa e critérios de convergência por número de iterações, Entropia ou Desvio Padrão.

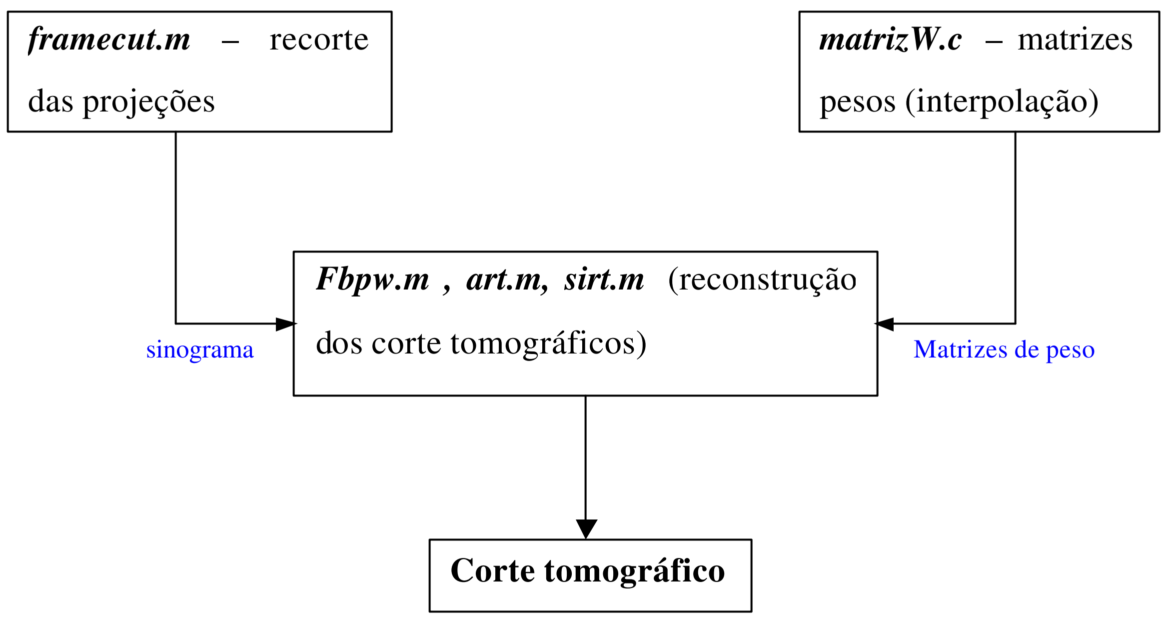


Figura 5.1 – Relação entre os programas elaborados para reconstruir os cortes tomográficos.

5.3 – Funções e Programas – Provas Qualitativas

Com o objetivo de se verificar o correto funcionamento dos algoritmos implementados, executou-se algumas funções destes e reconstruiu-se alguns cortes tomográficos de *phantoms* matemáticos para uma avaliação qualitativa dos mesmos. Os testes de performance serão abordados no capítulo seguinte.

5.3.1 – framecut.m

Tomando-se um conjunto de 60 projeções de um ECT de cervical (figura 5.2), em uma matriz [512], executa-se a função ***framecut.m*** sobre o mesmo. Como resultado, obtém-se o conjunto de 60 projeções em escala de cinza (figura 5.3), as 60 projeções separadas e salvas em matrizes distintas de [64] (exemplo da projeção de $\theta = 0^{o}$ - figura 5.4.a) e o sinograma da linha 32 das projeções em uma matriz 60x64 (figura 5.4.b), o que confirma o correto funcionamento do programa.

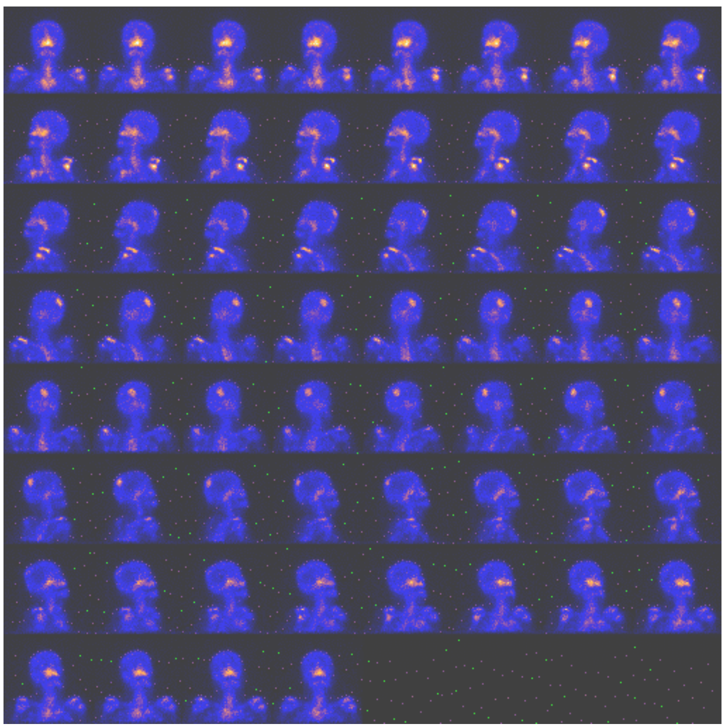


Figura 5.2 – Projeções de um ECT de cervical – matriz [512] com escala de cores originais.

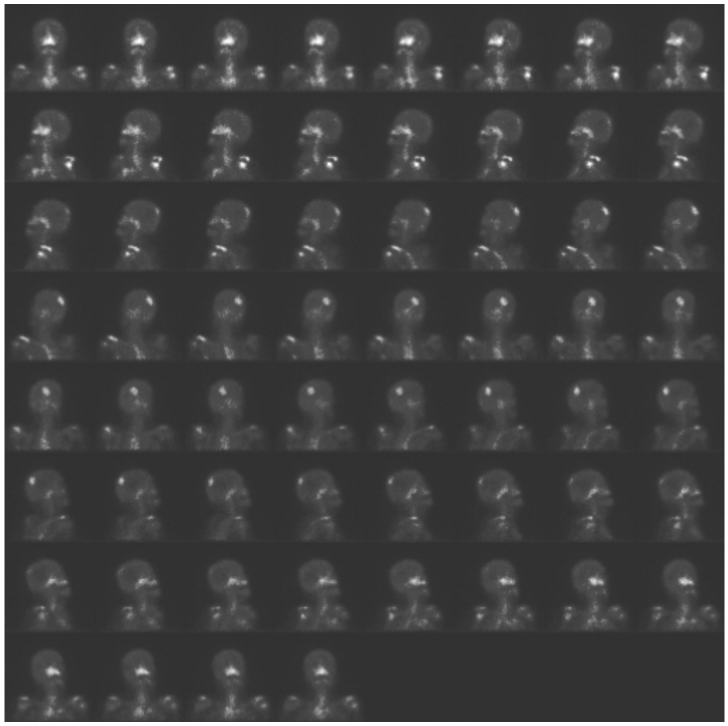
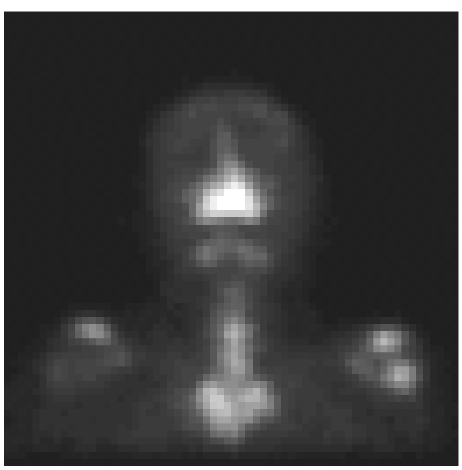
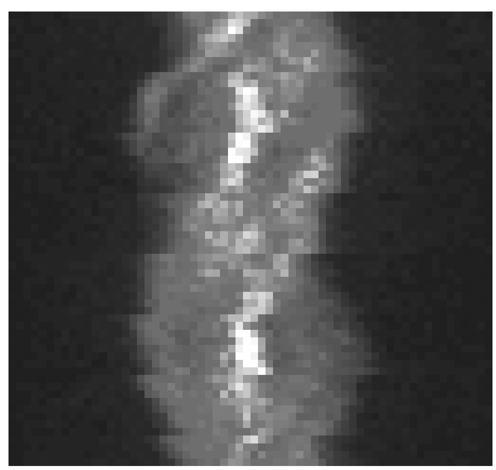


Figura 5.3 - Projeções de um ECT de cervical – matriz [512] com escala de cores originais transformada para *gray scale* – 8 *bits* por *framecut.m*.



(a)



(b)

Figura 5.4 – (a) Projeção em $\theta = 0^{o}$ - matriz [64] com *zoom* = x3. (b) Sinograma da linha 32 das 60 projeções – matriz [64] com *zoom* = x3. Deve–se notar o padrão de função senoidal na imagem, o que indica que o corte foi realizado corretamente.

5.3.2 – matrizW.c

O programa de confecção das matrizes de pesos e interpolação (***W***) foi confeccionado com base na equação 4.2. Para a implementação deste segue-se a notação das coordenadas dos pixels e raios anotada no esquema apresentado na figura 5.5 (exemplo de matriz [8]).

Calculando-se os pontos de intersecção das retas sobre as linhas de grade da matriz de retro-projeção dados pela equação e pela distância do raio *j* ao centro do pixel, ordena-se estes de forma decrescente por distância ao eixo y.

Com as coordenadas dos pontos (*n,m*) são calculadas as distâncias entre estes, ou seja, o quanto do comprimento de cada raio atravessa o *pixel*. Os valores de *n* e *m* são então interpolados para os índices das matrizes, e então armazenados em uma matriz tri-dimensional onde os índices *n,m,j* de cada elemento representam a linha (*n*), a coluna (*m*) do pixel e o raio (*j*) que o interseciona (quando o valor do elemento é igual zero, implica que o raio não passa por aquele pixel naquele ângulo de visada).

Utilizando os programas auxiliares ***bp.m*** e ***bpw.m*** , verifica-se a interpolação e os valores dos pesos das matrizes ***W*** confeccionadas. Os testes levaram em conta ângulos de visada simétricos e raios que passam pelo centro do sistema de coordenadas e eqüidistantes a este. Além das imagens[2] também foram obtidos perfis dos objetos retro-projetados para uma avaliação objetiva.

[2] Todas as imagens deste trabalho são apresentadas com *zoom* = x3, salvo quando anotado em contrário .

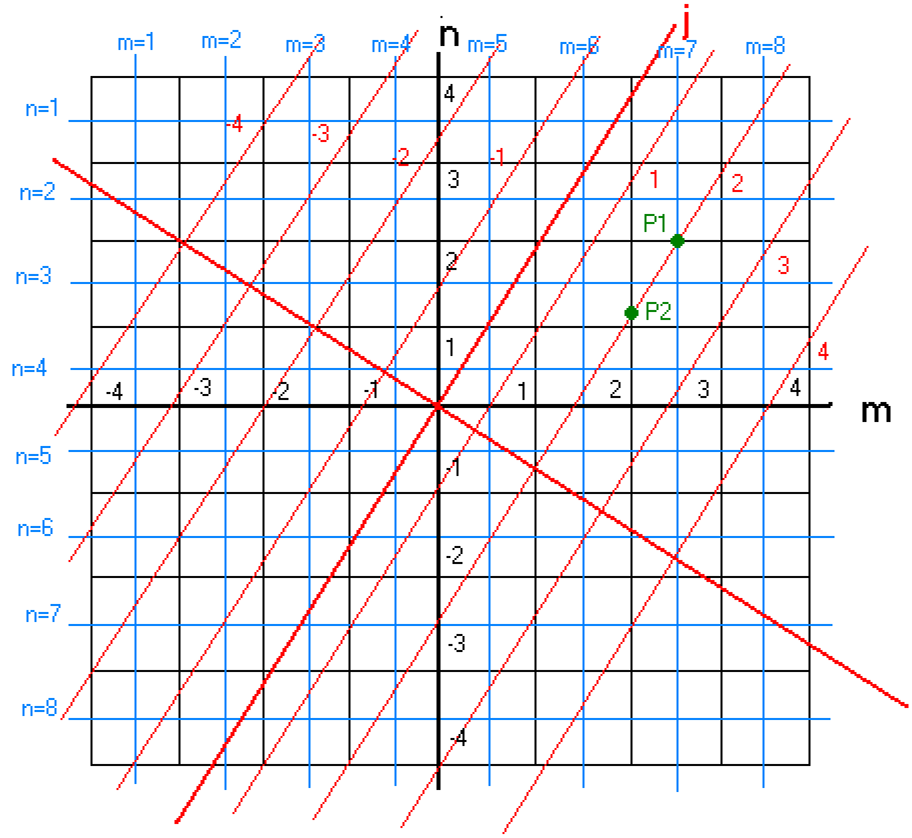
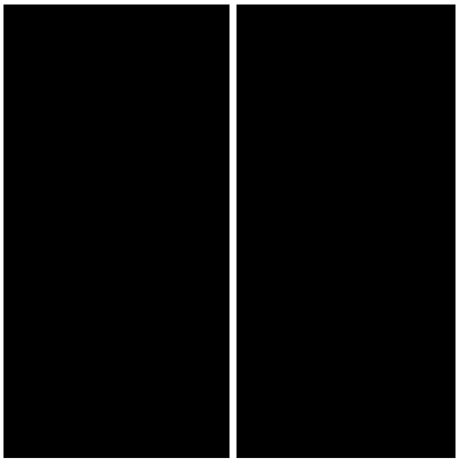
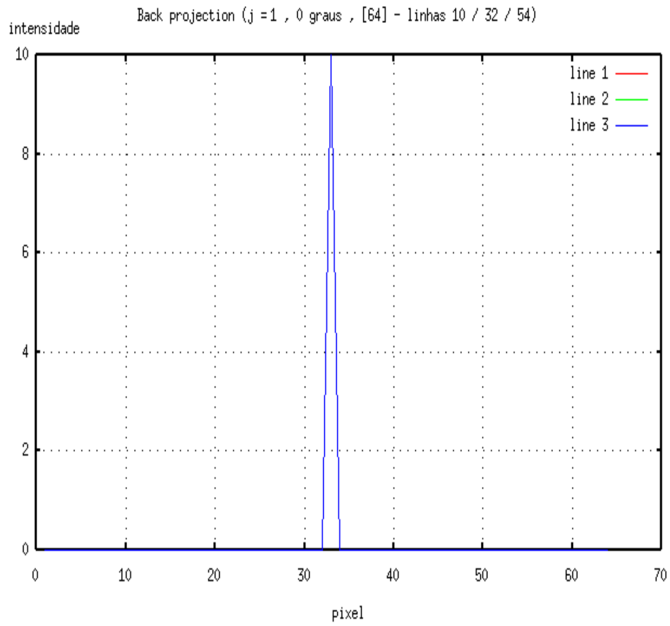


Figura 5.5 – Representação das coordenadas para o cálculo dos pesos e interpolação para a equação 4.2 (j = m Cos θ + n Sin θ) . As coordenadas (*n,m*) e *j* são as utilizadas na equação para encontrar as coordenadas dos pontos dos raios j que intersecionam os valores inteiros de *n* e *m* (como exemplo são mostrados os pontos *P1* e *P2*) . Com estes pontos em ordem decrescente em y, calcula–se a distância entre os mesmos (peso). O valor do peso é associado ao pixel (*n,m*) por meio de interpolação.

Os primeiros testes foram efetuados utilizando-se o programa ***bp.m.*** O objetivo é verificar a correta interpolação da retro- projeção de raios que passam pelo centro do plano de reconstrução (figuras 5.6 à 5.11) ou que são eqüidistantes a este (figuras 5.11 à 5.17), comparando–se os perfis das imagens de ângulos com simetrias de θ+π, π-θ e 2π –θ, atestando-se a correta construção das matrizes ***W***.

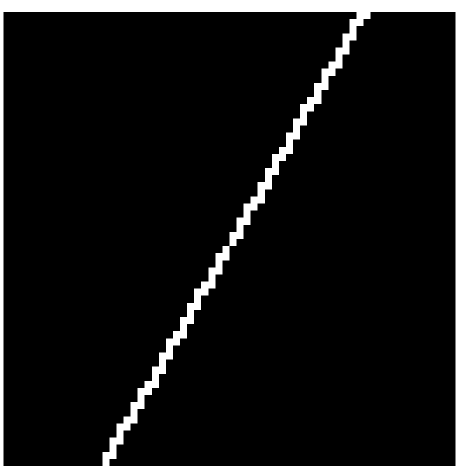


(a)

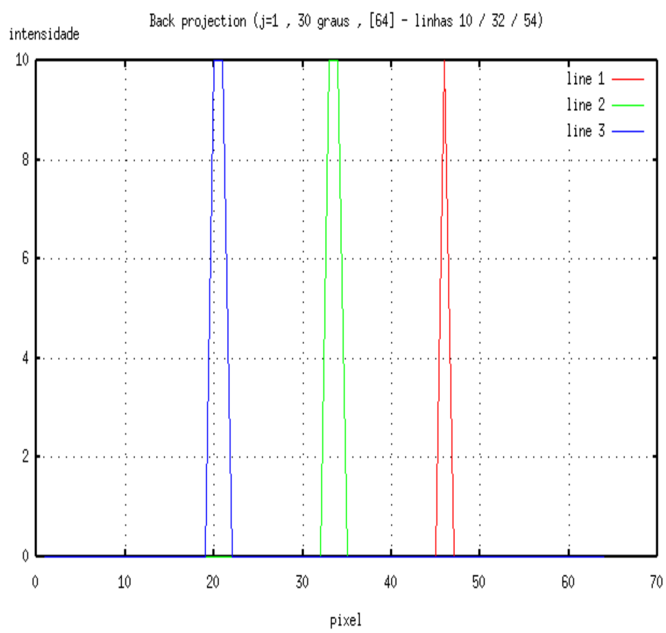


(b)

Figura 5.6 – (a) Retro – projeção (j=1, θ = 0º), [64] . (b) Perfis das linhas 10 / 32 / 54 de 5.6.a .

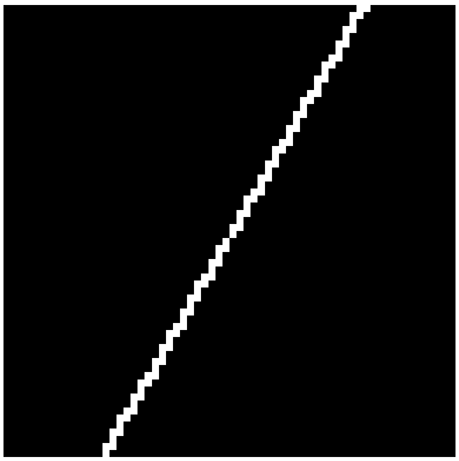


(a)

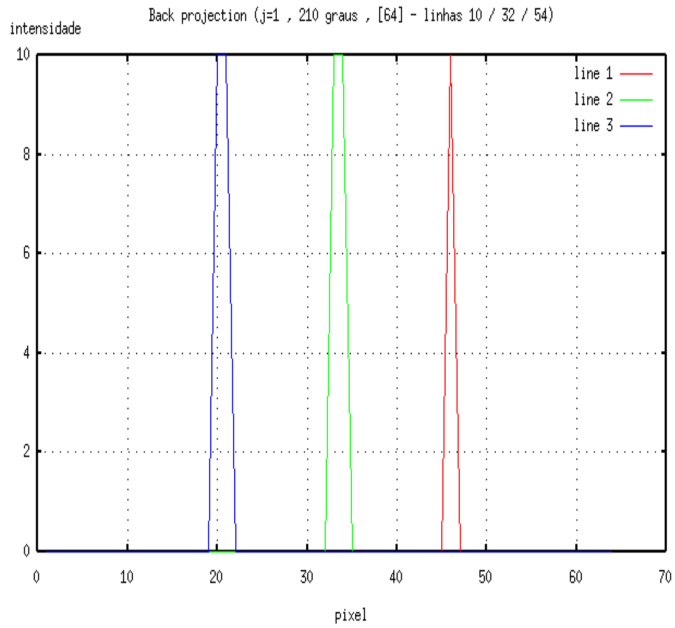


(b)

Figura 5.7 - (a) Retro – projeção (j=1, θ = 30º), [64] . (b) Perfis das linhas 10 / 32 / 54 de 5.7.a.

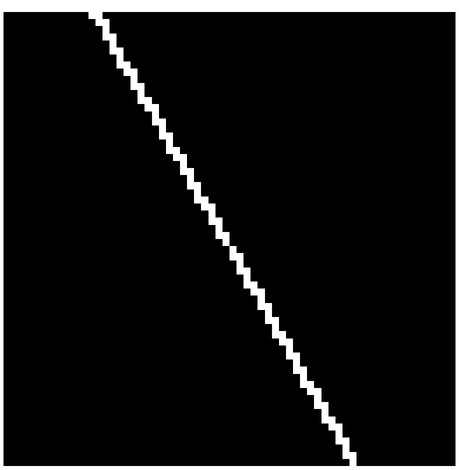


(a)

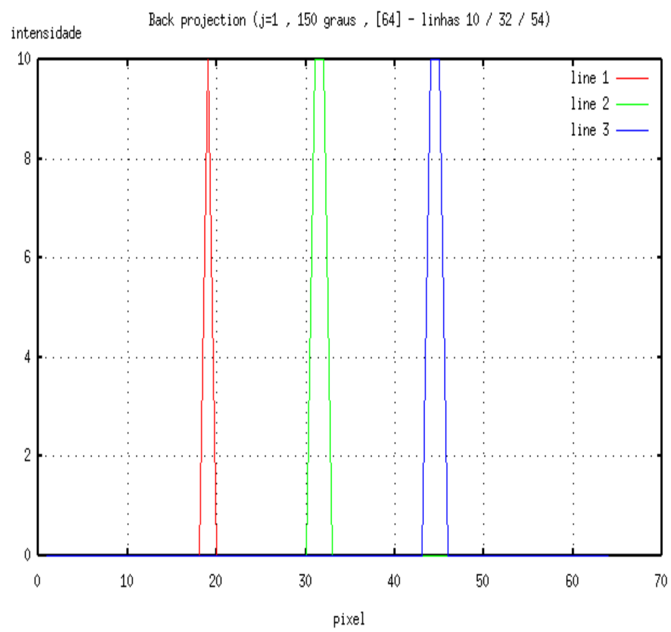


(b)

Figura 5.8 - (a) Retro – projeção (j=1, θ = 210º), [64] . (b) Perfis das linhas 10 / 32 / 54 de 5.8.a .

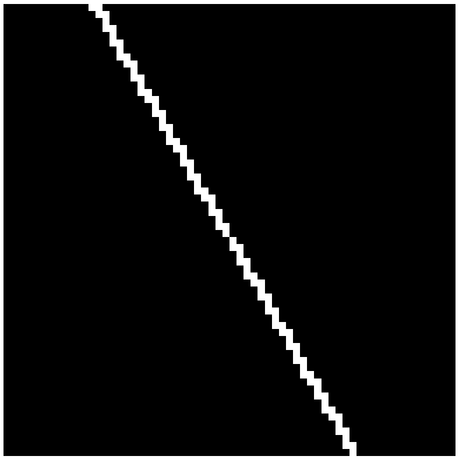


(a)

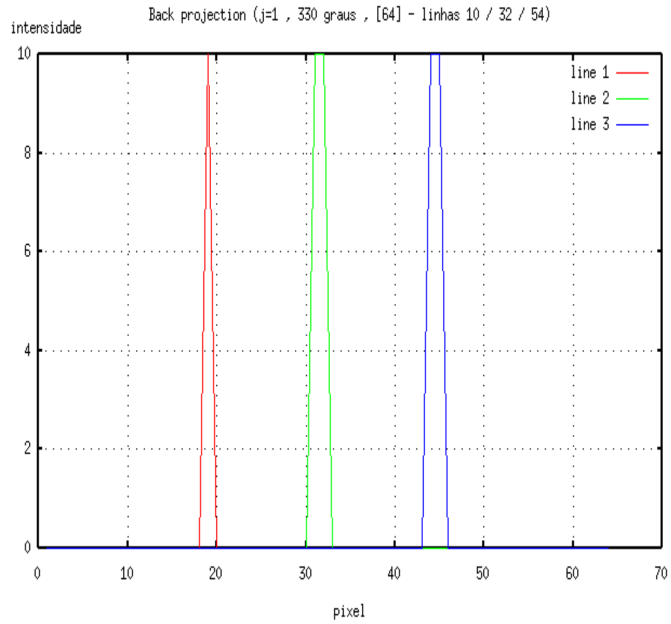


(b)

Figura 5.9 - (a) Retro – projeção (j=1, θ = 150º), [64] . (b) Perfis das linhas 10 / 32 / 54 de 5.9.a.

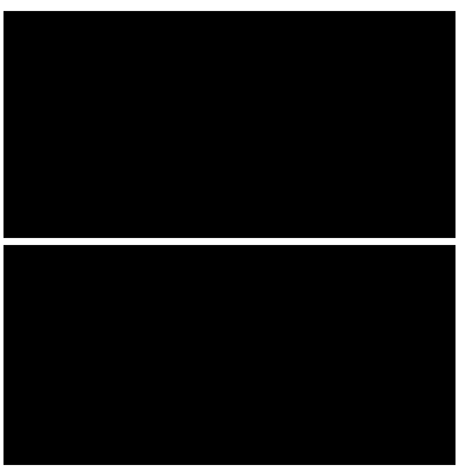


(a)

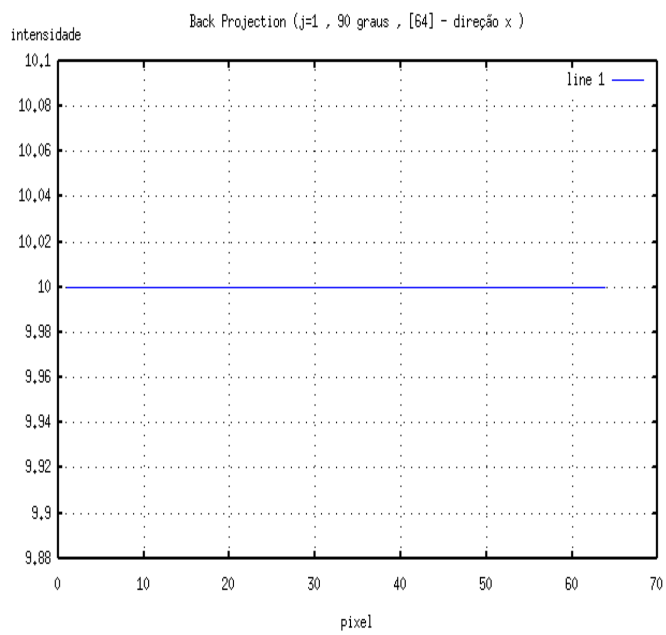


(b)

Figura 5.10 - (a) Retro – projeção (j=1, θ = 330º), [64] . (b) Perfis das linhas 10/ 32 / 54 de 5.10.a.

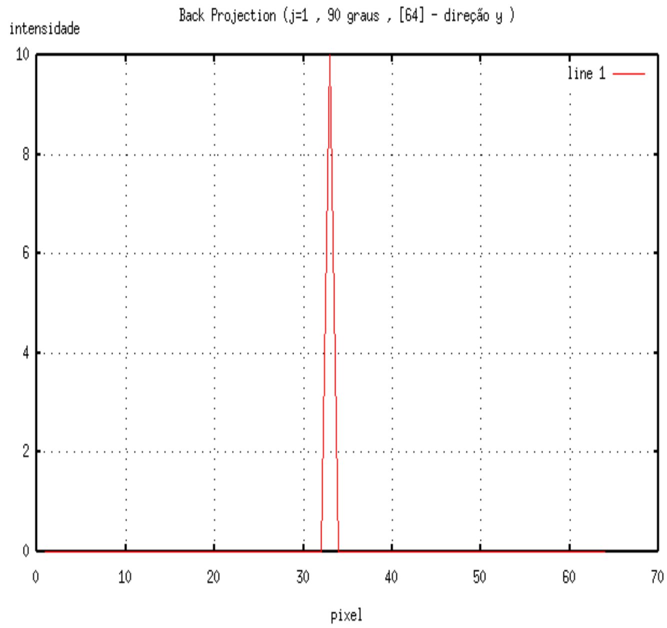


(a)



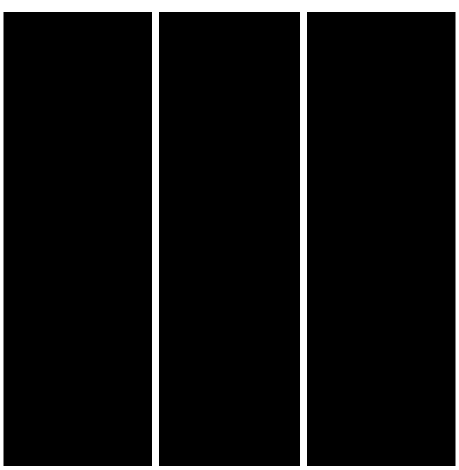
(b)

Legenda na pg. 82

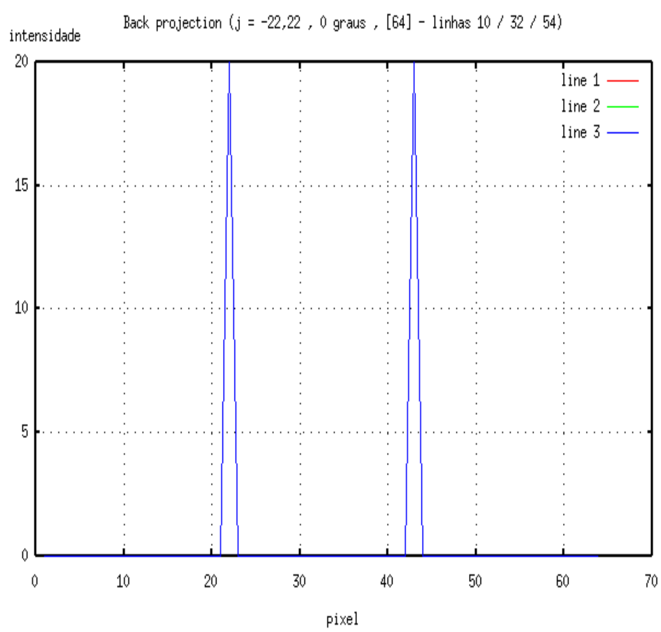


(c)

Figura 5.11 - (a) Retro – projeção (j=1, θ =90º), matriz [64] . (b) Perfil da linha 32 de 5.11.a. (c) Perfil da coluna 32 de 5.11.a.

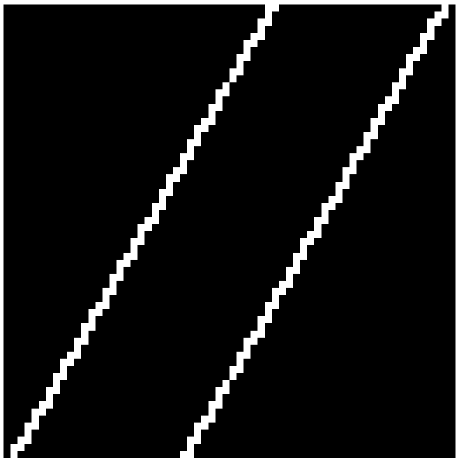


(a)

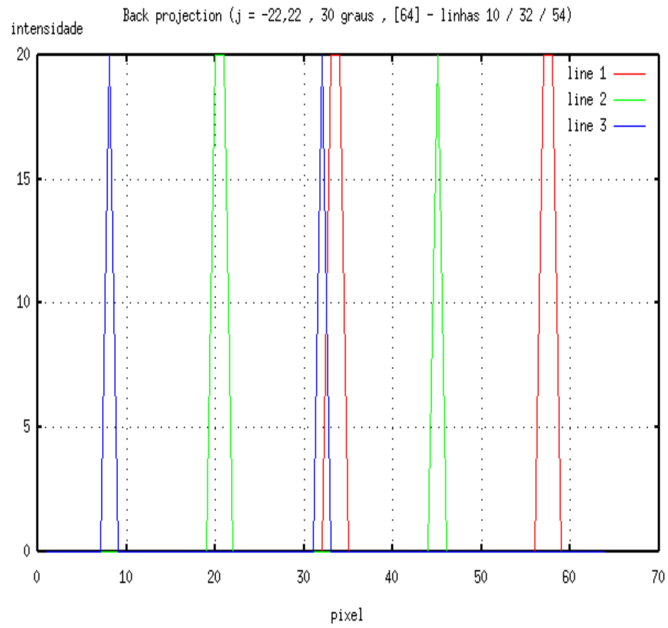


(b)

Figura 5.12 - (a) Retro – projeção (j = - 22 / 22, θ = 0º), [64] . (b) Perfis das linhas 10 / 32 / 54 de 5.12.a .

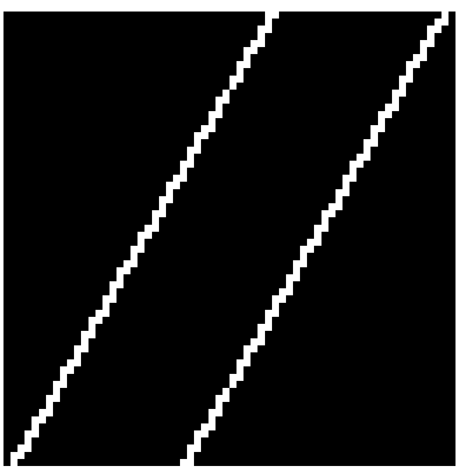


(a)

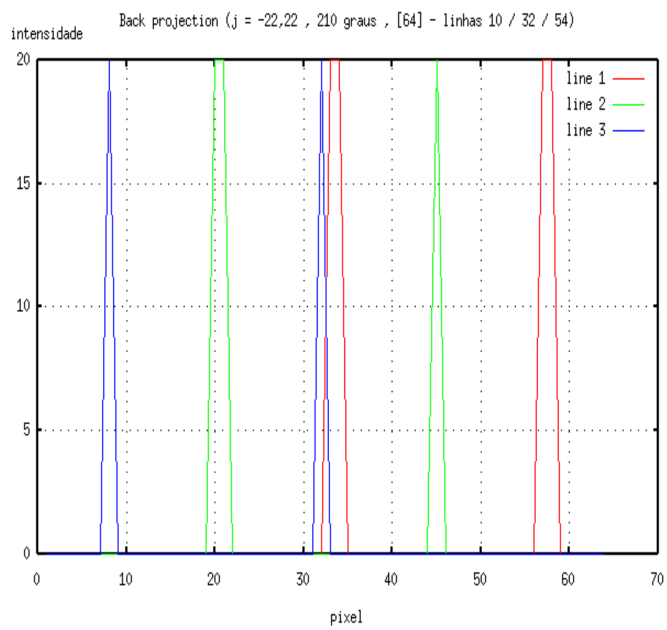


(b)

Figura 5.13 - (a) Retro – projeção (j = - 22 / 22, θ = 30º), [64] . (b) Perfis das linhas 10 / 32 / 54 de 5.13.a .

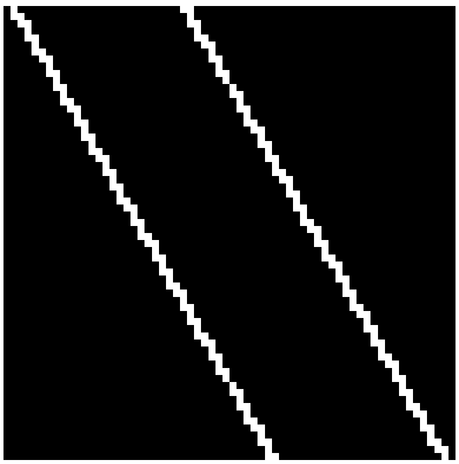


(a)

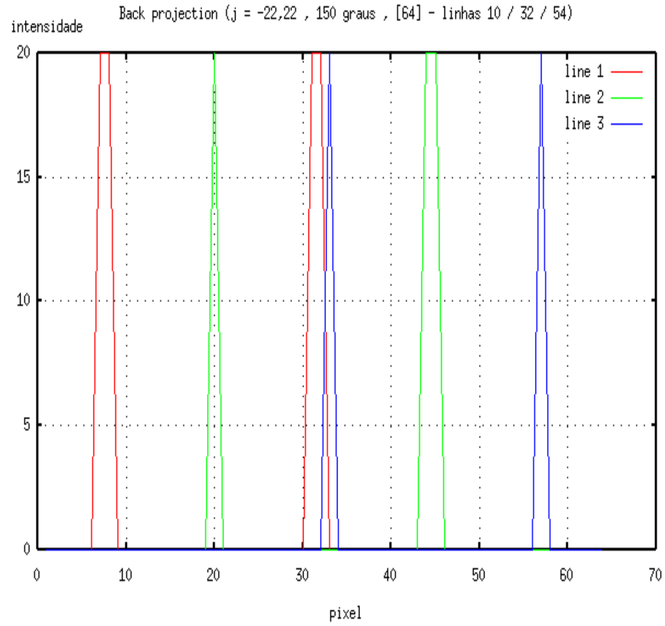


(b)

Figura 5.14 - (a) Retro – projeção (j = - 22 / 22, θ = 210º), [64] . (b) Perfis das linhas 10/ 32 / 54 de 5.14.a .

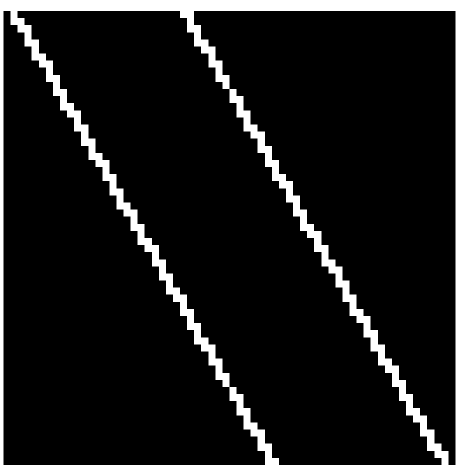


(a)

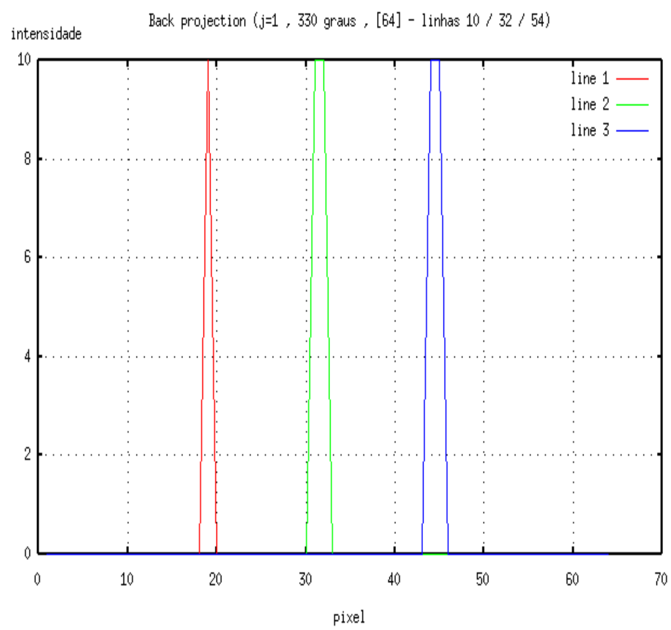


(b)

Figura 5.15 - (a) Retro – projeção (j = - 22 / 22, θ = 150º), [64] . (b) Perfis das linhas 10/ 32 / 54 de 5.15a .

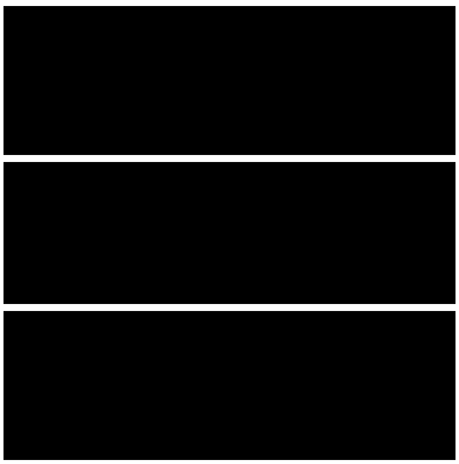


(a)

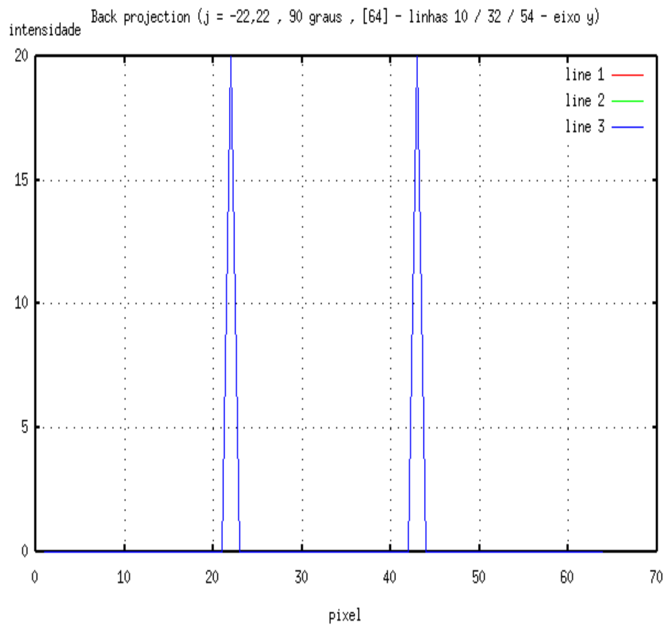


(b)

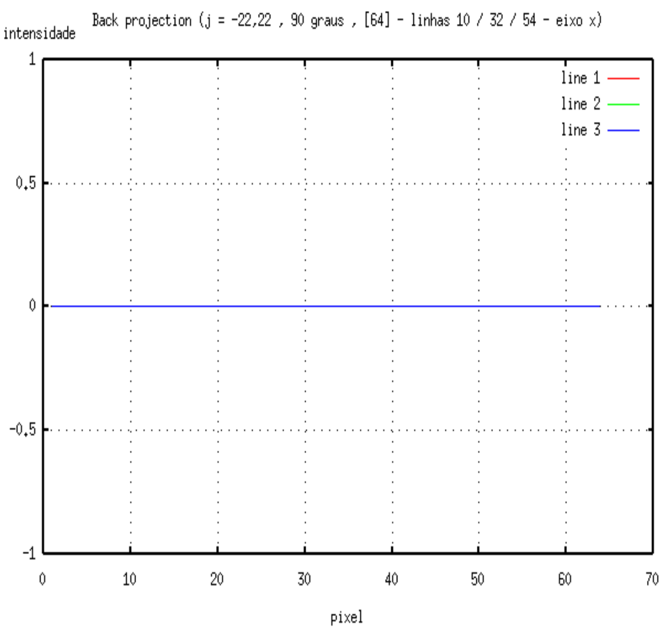
Figura 5.16 - (a) Retro – projeção (j = - 22 / 22, θ = 330º), [64] . (b) Perfis das linhas 10/ 32 / 54 de 5.16.a .



(a)



(b)

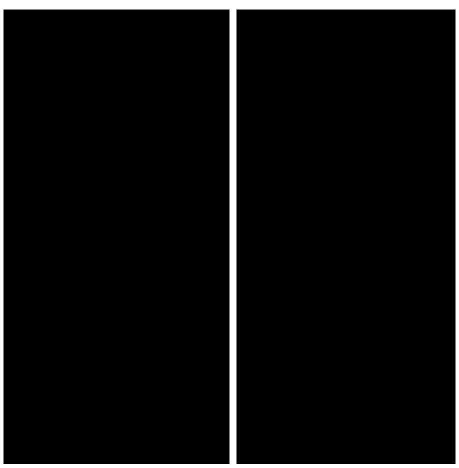


(c)

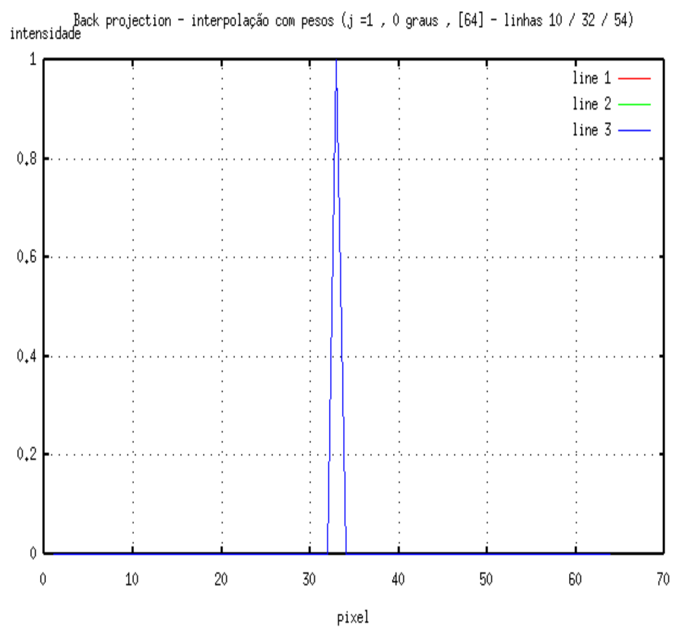
Figura 5.17 - (a) Retro – projeção (j = - 22 / 22 , θ =90º), [64] . (b) Perfil das linhas 10 / 32 / 54 de 5.17.a . (c) Perfil das colunas 10 / 32 / 54 de 5.17.a .

Nas figuras de 5.6 à 5.17, os perfis para os ângulos θ = 0º e 90º apresentam-se como esperado. Quando comparados os perfis de θ = 30º e 210º e θ = 150º e 330º (ângulos com simetria de θ +π), os perfis apresentam–se idênticos e, para os ângulos θ = 30º e 150º e θ = 210º e 330º (ângulos com simetria π - θ) os perfis apresentam-se invertidos em relação aos raios, ou seja, $j_{\theta 1} = 64 - j_{\theta 2} + 1$. Os resultados destes testes apresentam os padrões esperados da retro-projeção dos raios, comprovando a correta interpolação dada por ***W***.

Os testes executados com o programa ***bp.m*** são repetidos (mesmos ângulos e raios) com o programa ***bpw.m*** . O objetivo é verificar a correta construção dos valores dos pesos das matrizes ***W*** aplicados. Os resultados apresentados nas figuras 5.18 à 5.28 comprovam que os valores das ***W*** produzidas estão corretos.

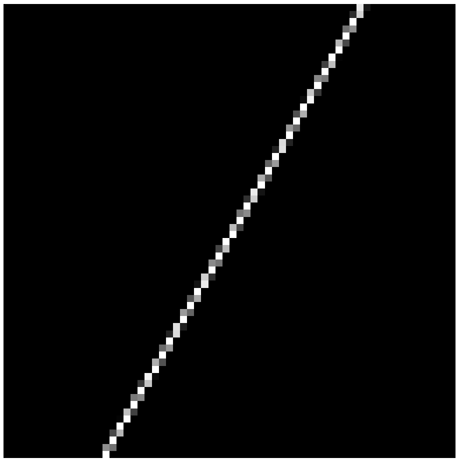


(a)

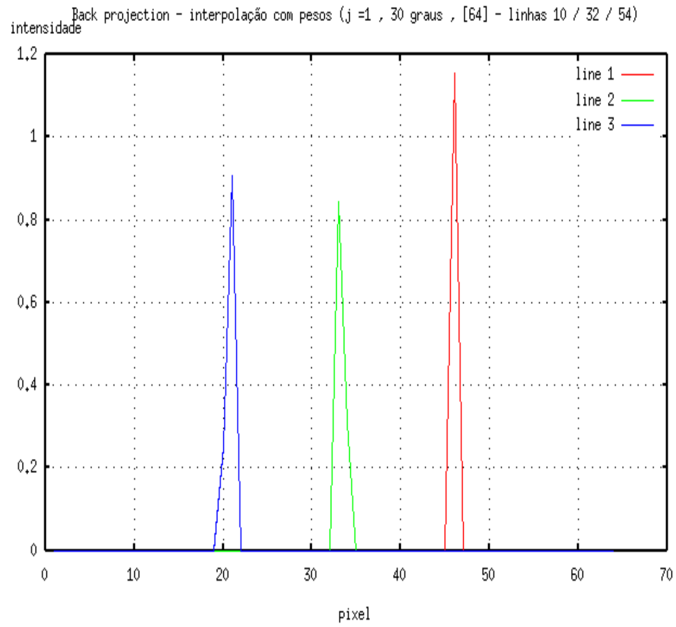


(b)

Figura 5.18 - (a) Retro – projeção - *W* (j = 1, θ = 0º), [64] . (b) Perfis das linhas 10/ 32 / 54 de 5.18.a .

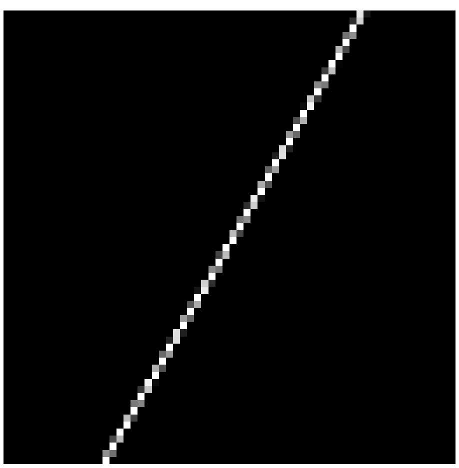


(a)

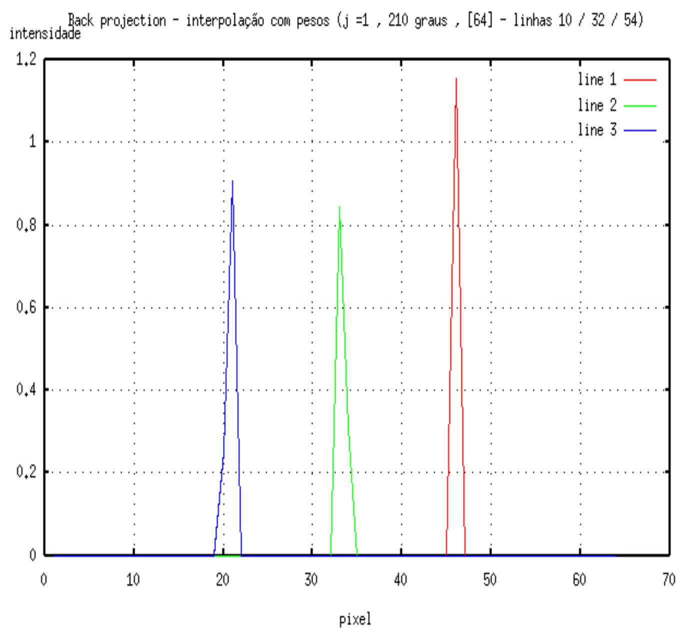


(b)

Figura 5.19 - (a) Retro – projeção - W (j = 1, θ =30º), [64] . (b) Perfis das linhas 10/ 32 / 54 de 5.19.a .

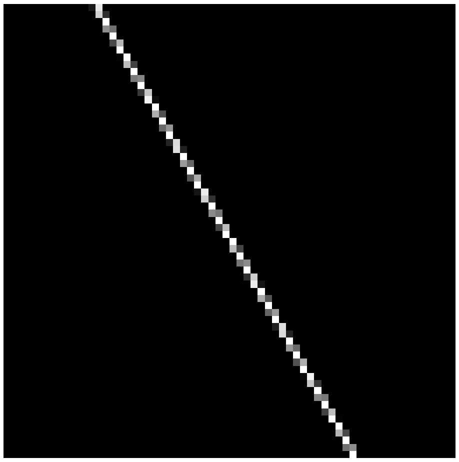


(a)

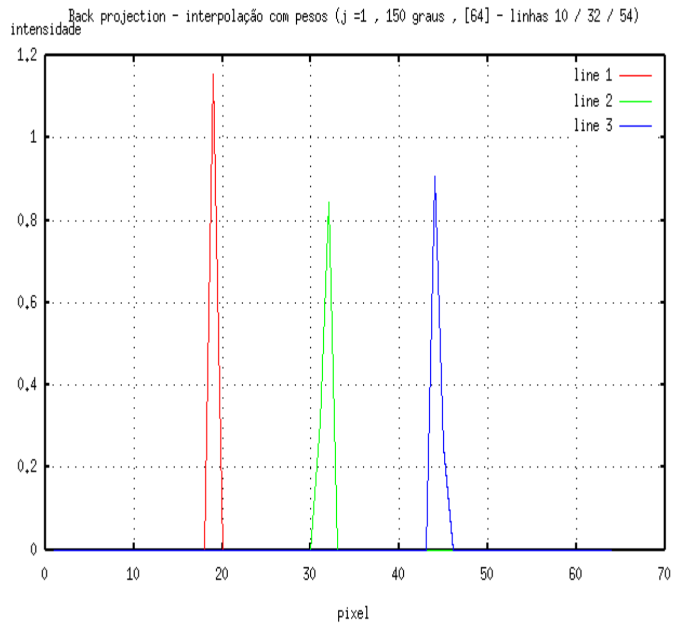


(b)

Figura 5.20 - (a) Retro – projeção - W (j = 1, θ = 210º), [64] . (b) Perfis das linhas 10/ 32 / 54 de 5.20.a .

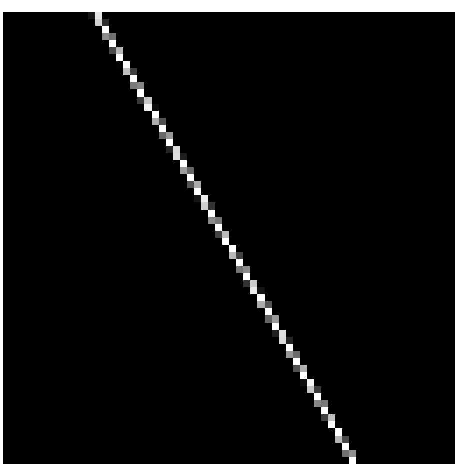


(a)

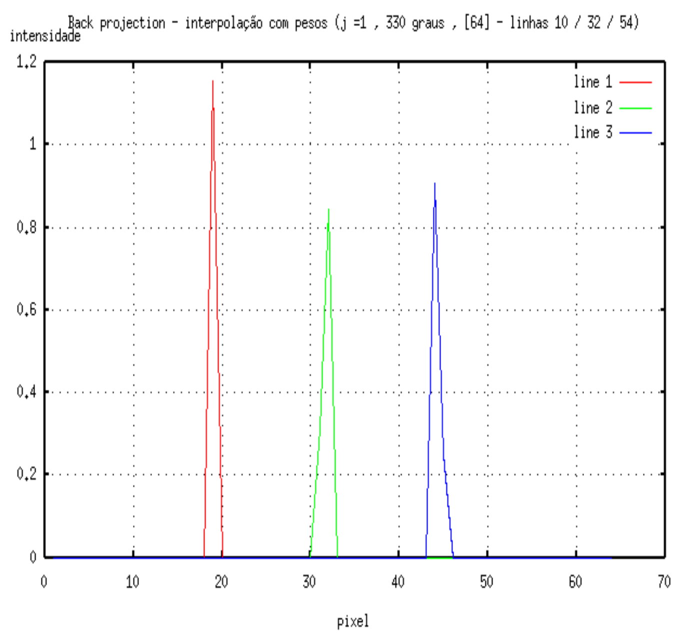


(b)

Figura 5.21 - (a) Retro – projeção - *W* (j = 1, θ = 150º), [64] . (b) Perfis das linhas 10/ 32 / 54 de 5.21.a .

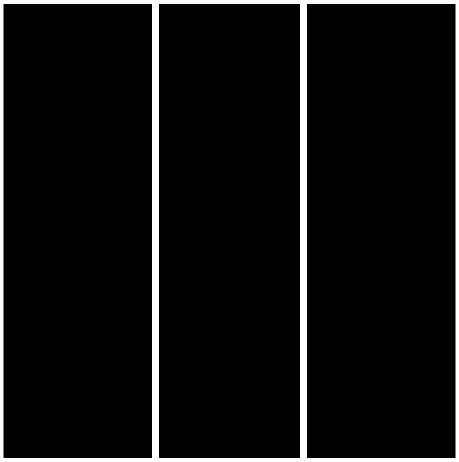


(a)

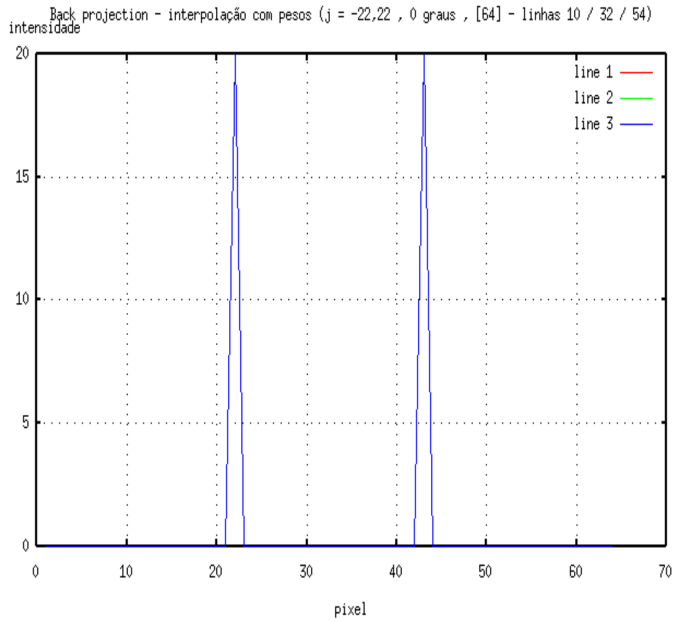


(b)

Figura 5.22 - (a) Retro – projeção - *W* (j = 1, θ =330º), [64] . (b) Perfis das linhas 10 / 32 / 54 de 5.22.a .

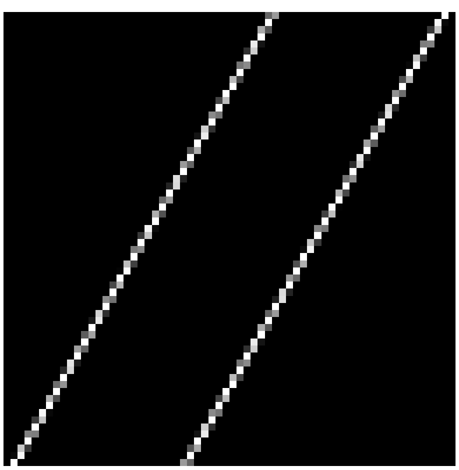


(a)

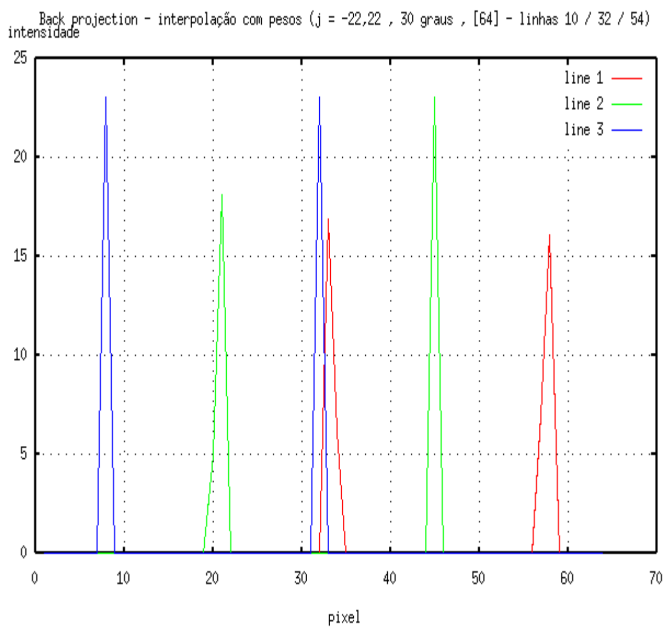


(b)

Figura 5.23 - (a) Retro – projeção - *W* (j = -22 / 22, θ = 0º), [64] . (b) Perfis das linhas 10/ 32 / 54 de 5.23.a .

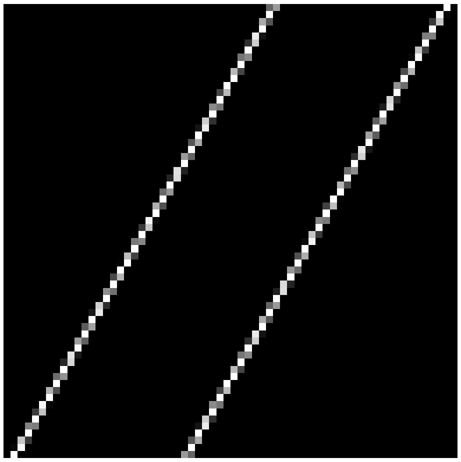


(a)

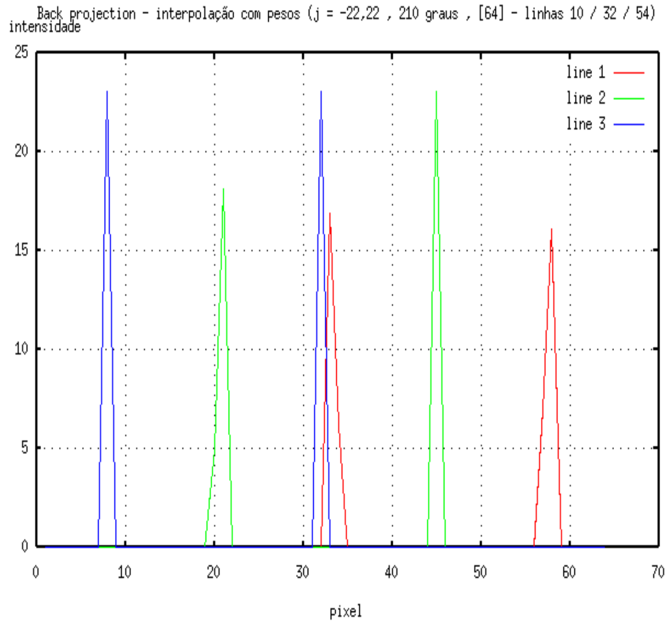


(b)

Figura 5.24 - (a) Retro – projeção - *W* (j =- 22 / 22, θ = 30º), [64] . (b) Perfis das linhas 10/ 32 / 54 de 5.24.a .

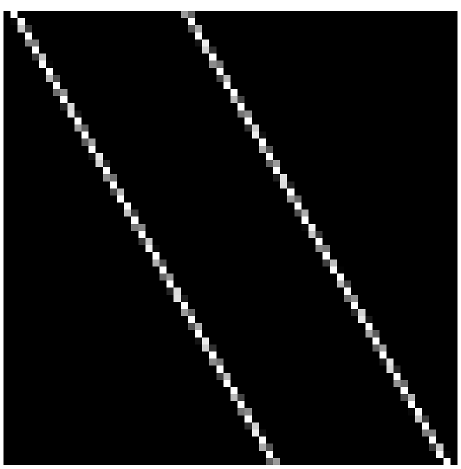


(a)

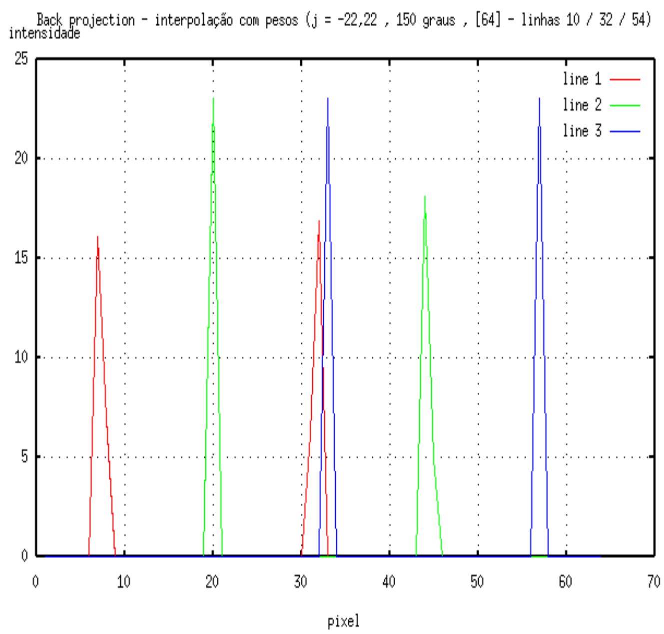


(b)

Figura 5.25 - (a) Retro – projeção - *W* (j = -22 / 22, θ = 210º), [64] . (b) Perfis das linhas 10/ 32 / 54 de 5.25.a.

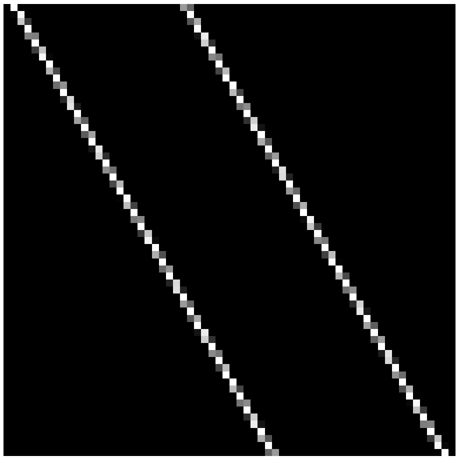


(a)

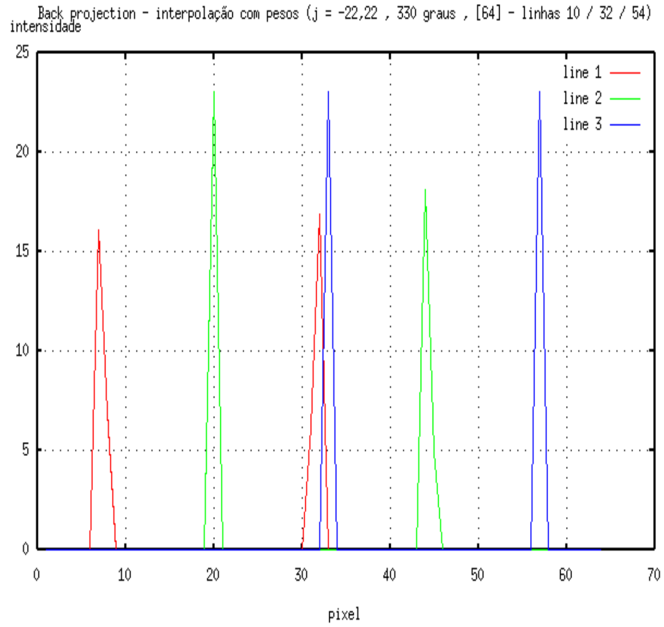


(b)

Figura 5.26 - (a) Retro – projeção - *W* (j = -22 / 22, θ =150º), [64] . (b) Perfis das linhas 10/ 32 / 54 de 5.26.a.

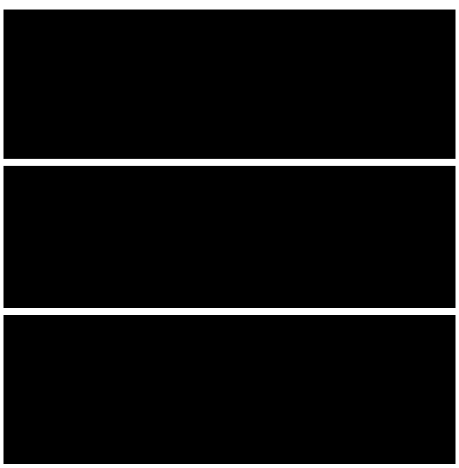


(a)

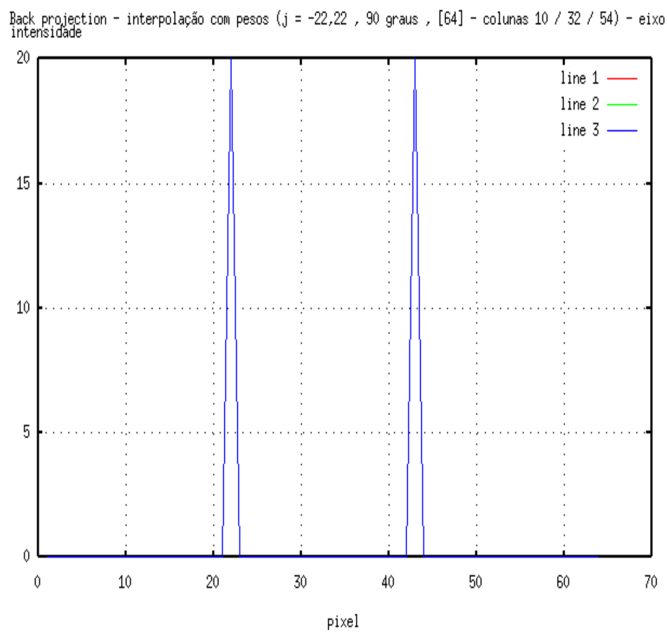


(b)

Figura 5.27 - (a) Retro – projeção - W (j = -22 / 22, θ = 330º), [64] . (b) Perfis das linhas 10/ 32 / 54 de 5.27.a.



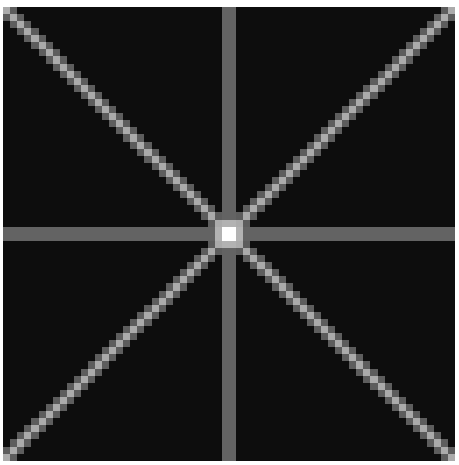
(a)



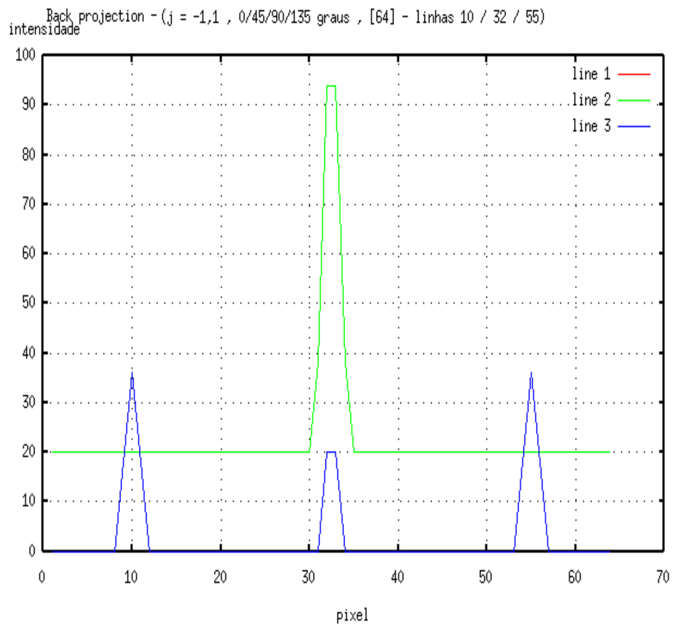
(b)

Figura 5.28 - (a) Retro – projeção - W (j =-22 / 22, θ = 90º), [64] . (b) Perfis das colunas 10/ 32 / 54 de 5.28.a.

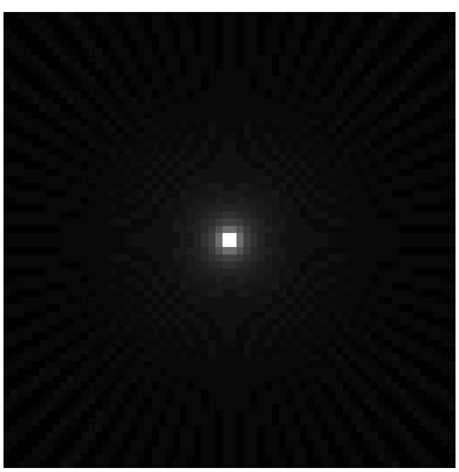
Utilizando-se de projeções nos ângulos de visada 0º / 45º / 90º / 135º de uma fonte puntiforme posicionada no centro da matriz de imagem, testa-se a formação da imagem do objeto com as retro-projeções multiplicadas pelos pesos das matrizes $\boldsymbol{W}$ correspondentes a estes ângulos. O resultado (figura 5.29) demonstra a formação correta da imagem e o efeito estrela (borramento) esperado.



(a)



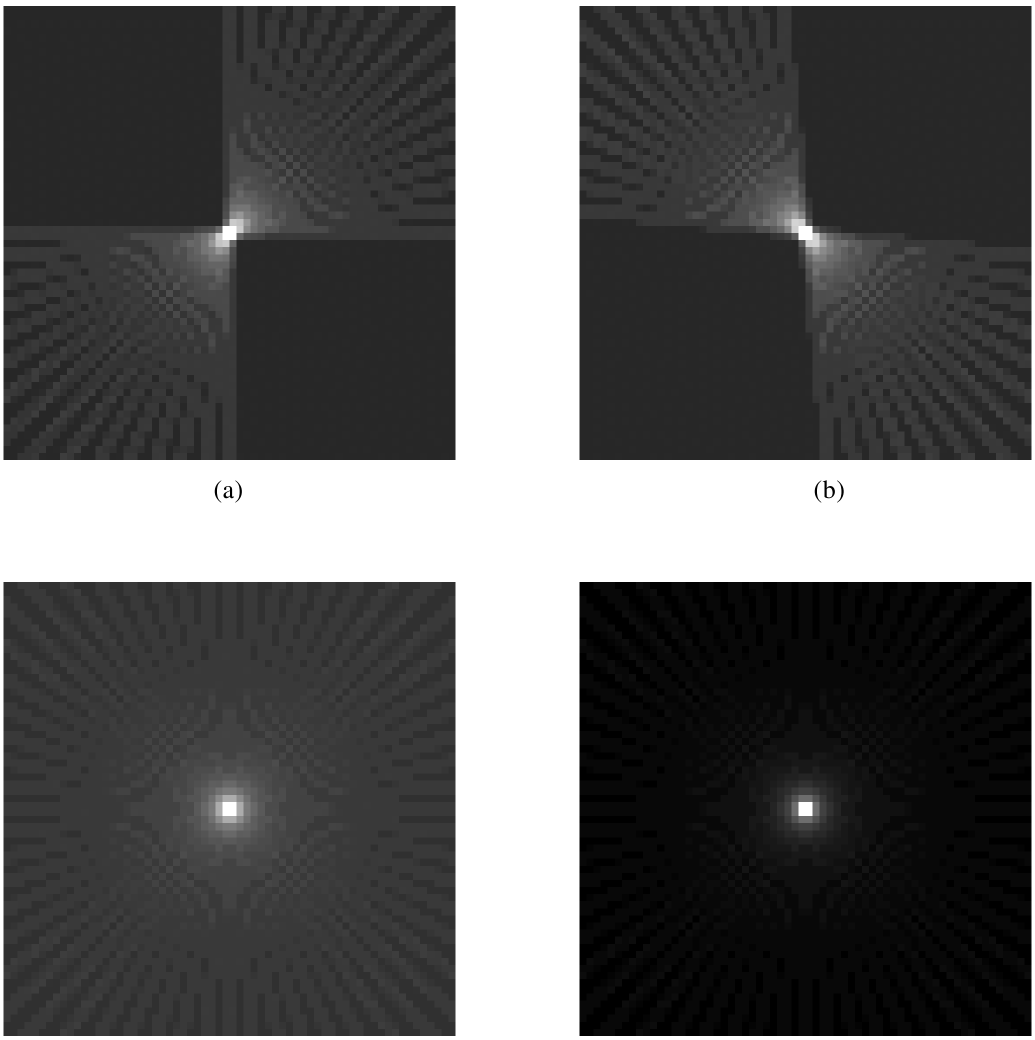
(b)



(c)

Figura 5.29 - (a) Retro – projeção - $\boldsymbol{W}$ (j = - 1 / 1) fonte puntiforme, [64] . (b) Perfis das linhas 10/ 32 / 55 de 5.29.a . (c) Objeto original – *phantom* matemático de uma fonte puntiforme.

Como último teste para o programa ***matrizW.c*** , foram retro-projetadas com o programa ***bpw.m*** 30 projeções de uma órbita de 180º (*step* de 6º) da mesma fonte puntiforme usada na imagem da figura 5.29, verificando a reconstrução da imagem paras os ângulos de 0º à 90º e de 96º à 174º, somando-se as imagens em seguida. Os resultados (figura 5.30) apresentam a reconstrução de uma fonte puntiforme no centro do plano com o efeito estrela presente , como esperado.



(a) (b)

(c) (d)

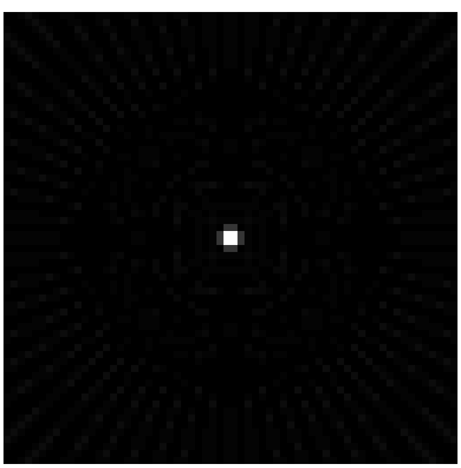
Figura 5.30 – (a) Retro – projeção com *W* das projeções de fonte puntiforme de 5.29.c - θ = 0º à 90º . (b) Idem (a) para θ = 96º à 174º . (c) Soma das imagens em (a) e (b) – As imagens tiveram sua intensidade e contraste alterados para a visualização do borramento – efeito estrela. (d) Imagem 5.30.c com contraste e intensidade originais.

Para efeito de comparação, foram reconstruídas algumas imagens com as matrizes $\boldsymbol{W}$ sem a simetria apropriada, evidenciando problemas nas imagens. Estas se encontram no *Anexo E*.

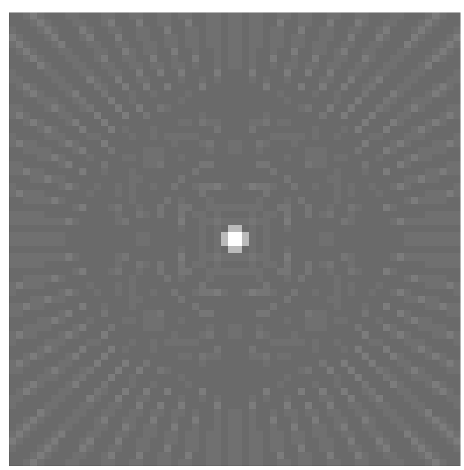
5.3.3 – *fbp.m* e *fbpw.m*

Com base nos algoritmos ***bp.m*** e ***bpw.m,*** foram implementados os programas ***fbp.m*** e ***fbpw.m*** para a reconstrução de cortes tomográficos por retro-projeções filtradas (rampa e janela), conforme descrito no ítem 5.2 deste capítulo.

Para testar qualitativamente os algoritmos, foram reconstruídas as imagens de três objetos: uma fonte puntiforme com distribuição gaussiana da radiação (centrada no plano – figura 5.29.c), uma fatia de um cilindro com distribuição uniforme de radiação (círculo) e um objeto elipsóide (*phantom* matemático). Para tanto, um programa para a construção de sinogramas a partir dos *phantoms* matemáticos foi desenvolvido, objetivando a reconstrução de matrizes [64] com 60 projeções de uma órbita de 360º com *steps* de 6º. Os resultados obtidos são apresentados nas figuras 5.31 à 5.36.

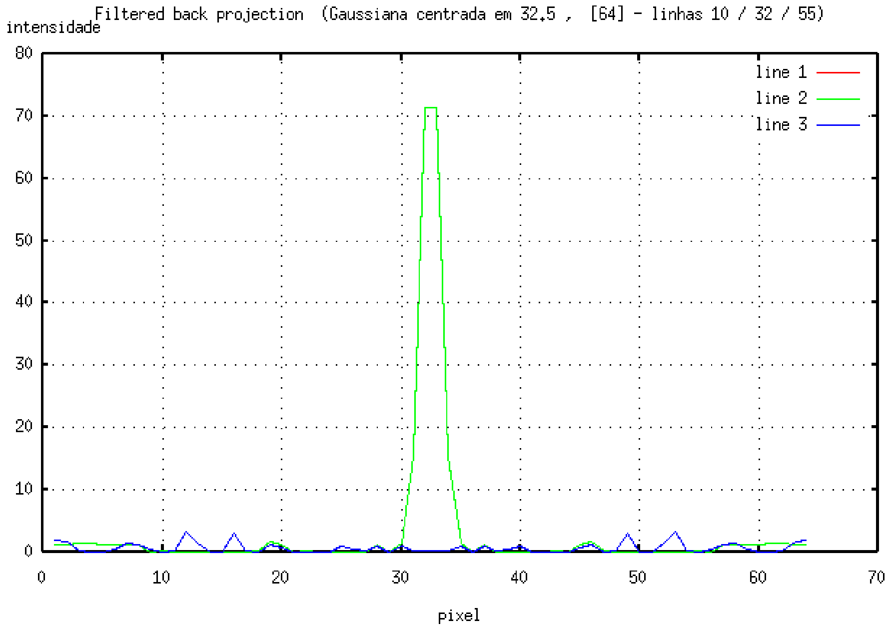


(a)



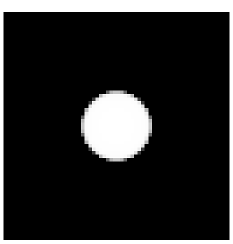
(b)

Legenda na pg. 95

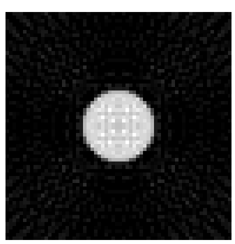


(c)

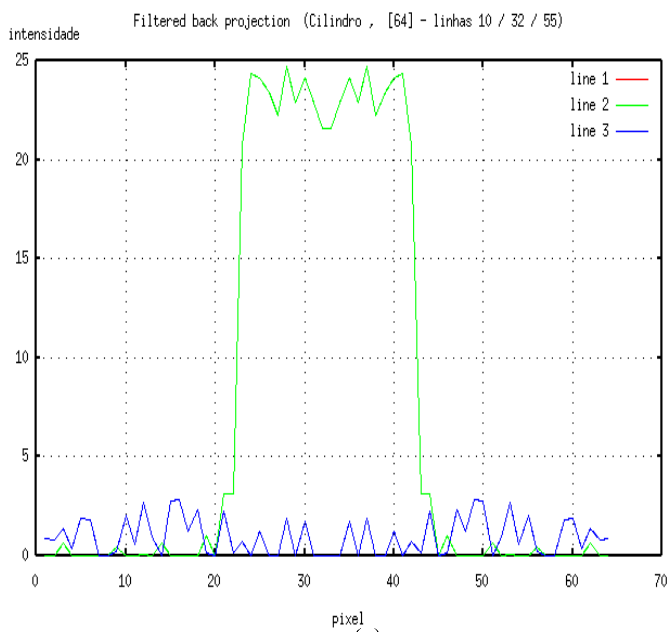
Figura 5.31 – (a) Reconstrução de uma fonte puntiforme centrada no *pixel* 32,5 com distribuição gaussiana com *fbp.m* (sem peso). (b) Imagem 5.31a com contraste e intensidade alterados para comparação com 5.30c, onde se observa o efeito do filtro rampa. (c) Perfis das linhas 10 / 32 / 55 de 5.31.a .



(a)

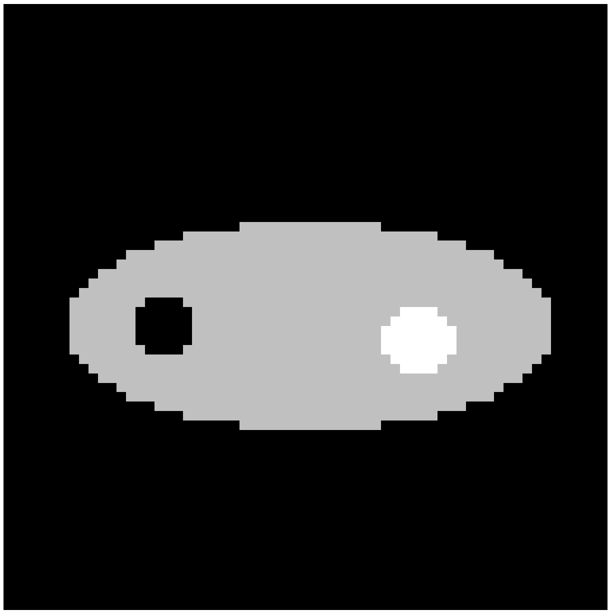


(b)

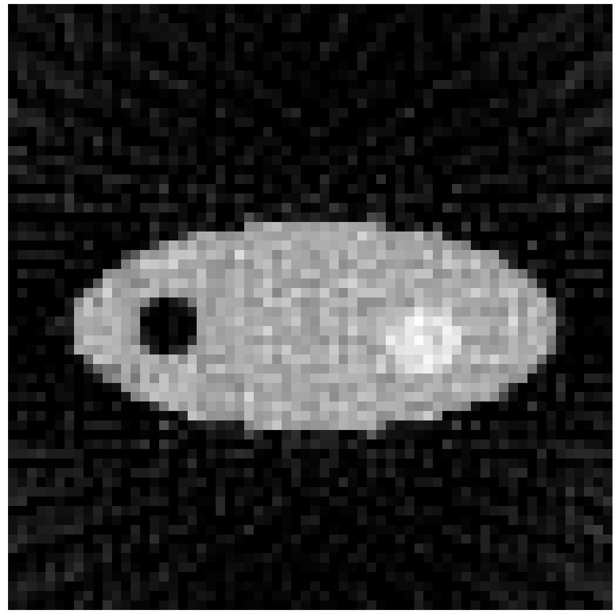


(c)

Figura 5.32 – (a) *Phantom* matemático – corte de um cilindro com 20 *pixels* de diâmetro (aproximadamente 13,2 cm) com distribuição uniforme de radiação – zoom = x2 . (b) Reconstrução de 5.32.a com *fbp.m* (sem peso). (c) Perfis das linhas 10 / 32 / 55 de 5.32.a .

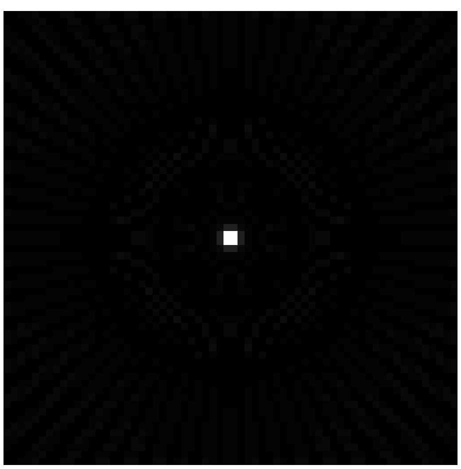


(a)

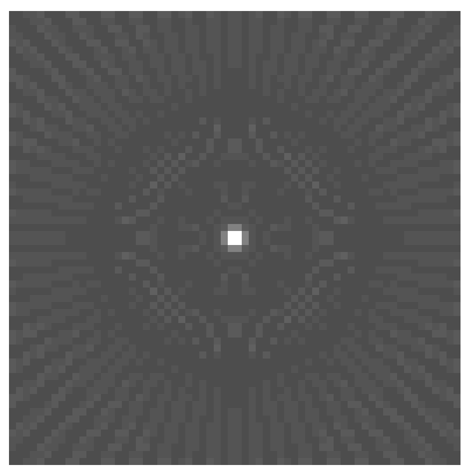


(b)

Figura 5.33 - (a) Objeto elipsóide - *phantom* matemático. (b) Reconstrução do objeto 5.33.a com *fbp.m* (sem peso).

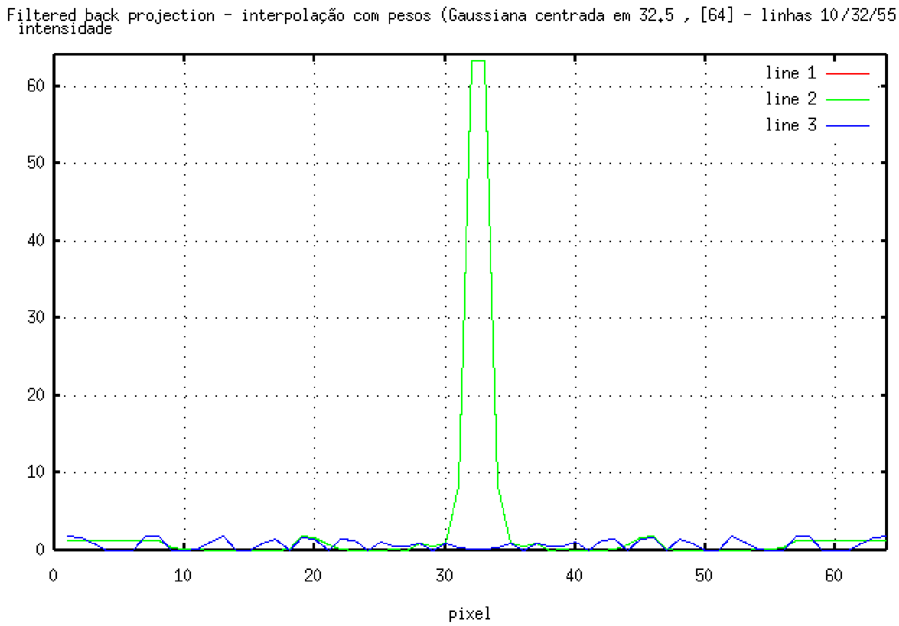


(a)



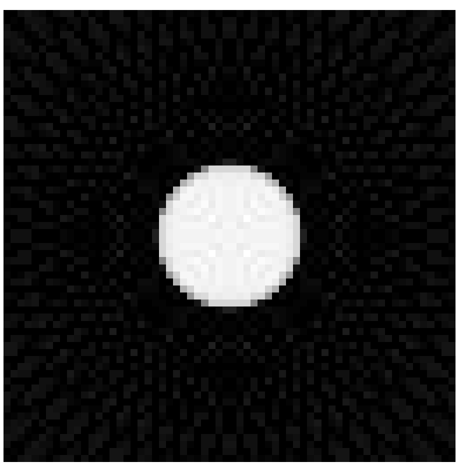
(b)

Legenda na pg. 97

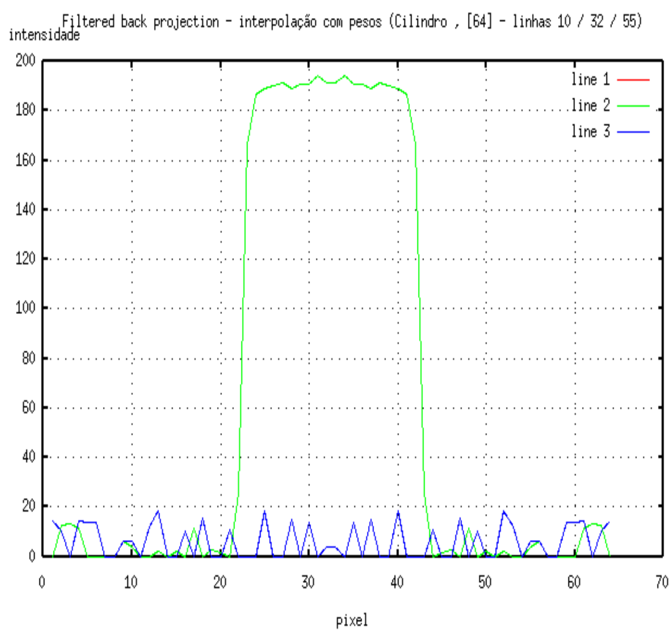


(c)

Figura 5.34 – (a) Reconstrução de uma fonte puntiforme centrada no *pixel* 32,5 com distribuição gaussiana com *fbpw.m* (com peso). (b) Imagem 5.34a com contraste e intensidade alterados para comparação com 5.30c, onde se observa o efeito do filtro rampa. (c) Perfis das linhas 10 / 32 / 55 de 5.34a .

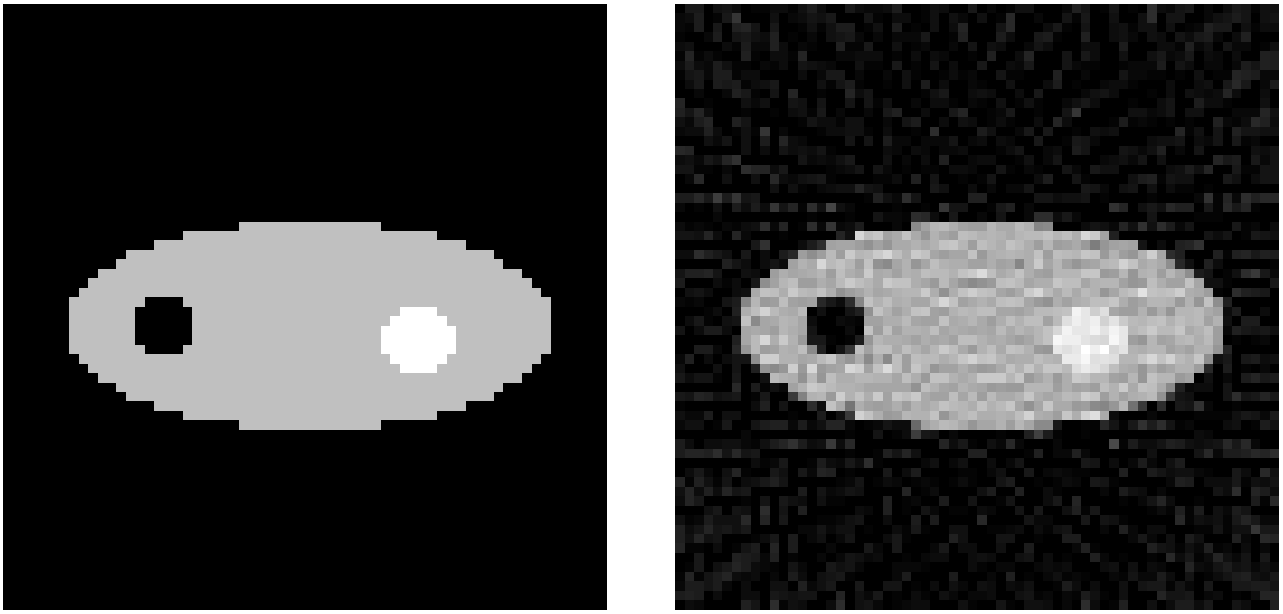


(a)



(b)

Figura 5.35 - (a) Reconstrução de um corte seccional de um cilindro com distribuição uniforme de radiação (figura 5.32.a) com *fbpw.m* (com peso). (b) Perfis das linhas 10 / 32 / 55 de 5.35a .



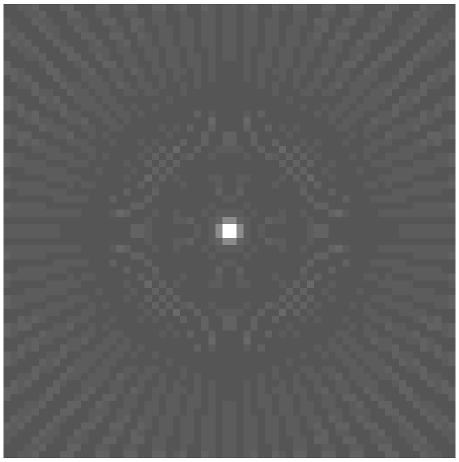
(a) (b)

Figura 5.36 - (a) Objeto elipsóide – *phantom* matemático . (b) Reconstrução do objeto 5.36a com *fbpw.m* (com peso).

A observação das figuras de 5.31 à 5.36 demonstra que os programas ***fbp.m*** e ***fbpw.m*** estão reconstruindo corretamente os objetos propostos nos testes, onde também se pode observar o efeito do filtro rampa e da multiplicação pelos pesos das matrizes ***W*** na diminuição do efeito estrela, principalmente entre as figuras 5.30c, 5.31b e 5.34b , reproduzidas na figura 5.37 para comparação.

(a) (b)

Legenda na pg. 99



(c)

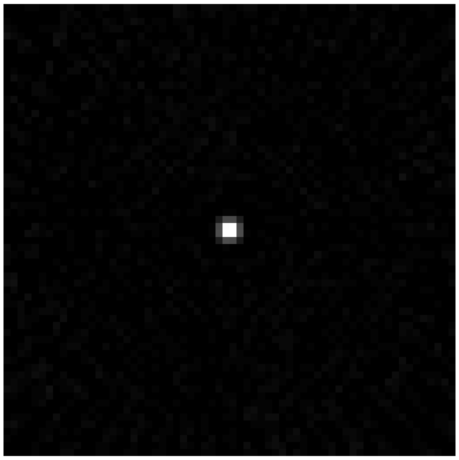
Figura 5.37 – (a) Figura 5.30c. (b) Figura 5.31b. (c) Figura 5.34b. – Intensidades e contrastes alterados.

5.3.4 – *art.m*

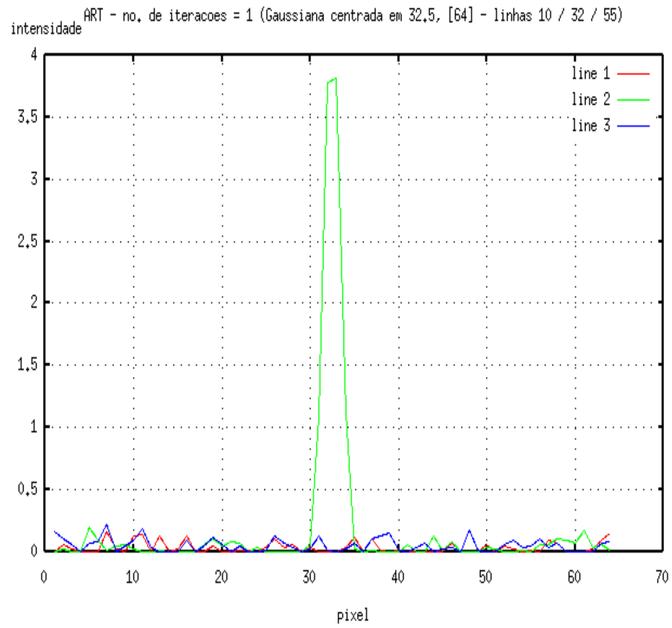
A implementação do programa com a técnica de reconstrução iterativa ART deu-se como descrito no ítem 5.1 deste capítulo.

Para análise qualitativa do programa, foram reconstruídos os mesmos objetos (exceto o cilindro, onde se alterou o diâmetro para 30 *pixels* – aproximadamente 19,8 cm) utilizados para os testes dos programas ***fbp.m*** e ***fbpw.m***, com um número de iterações igual a 6 (valor escolhido aleatoriamente), tanto para a técnica com correção do tipo aditiva como a multiplicativa. O objetivo destes testes é de verificar a correta reconstrução dos objetos.

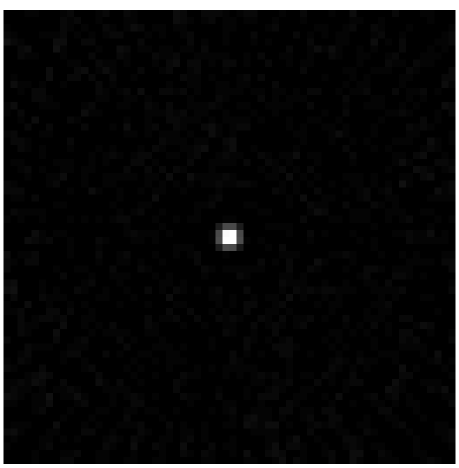
Os resultados apresentados nas figuras de 5.38 à 5.43, mostram as imagens obtidas na primeira e última iteração, sem se fazer menção aos critérios de convergência (Entropia / Desvio Padrão), sendo estes e as imagens intermediárias objetos de estudo posterior neste trabalho.



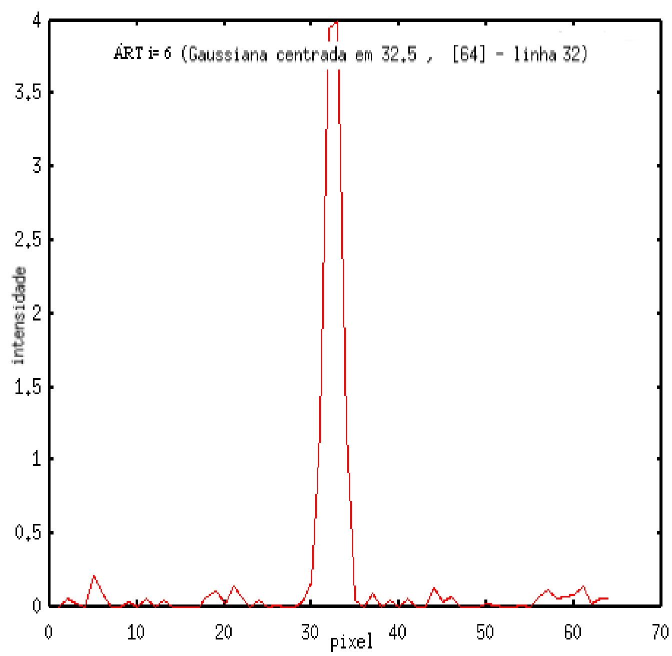
(a)



(b)

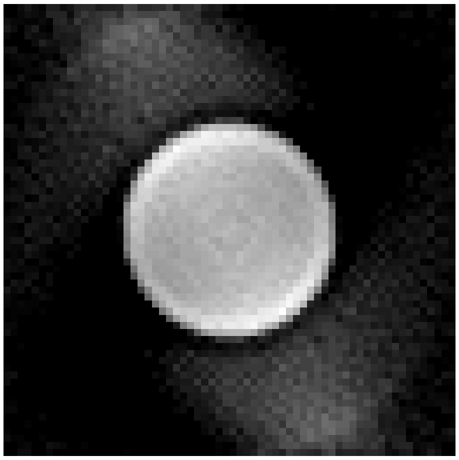


(c)

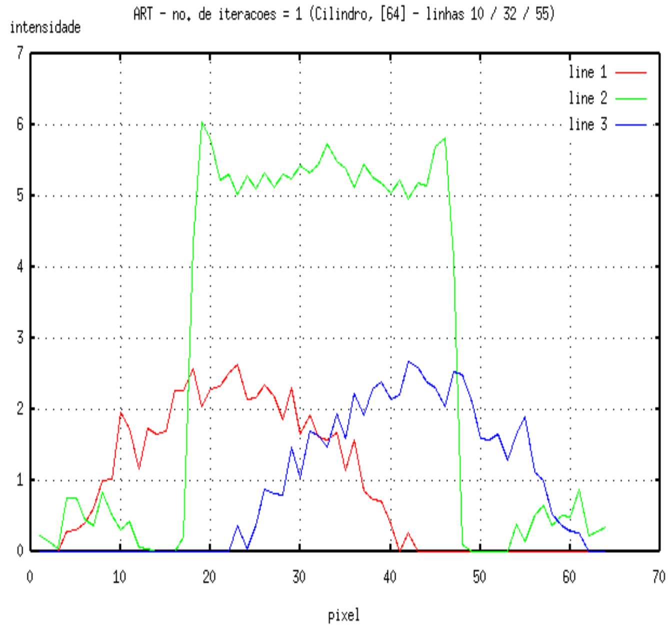


(d)

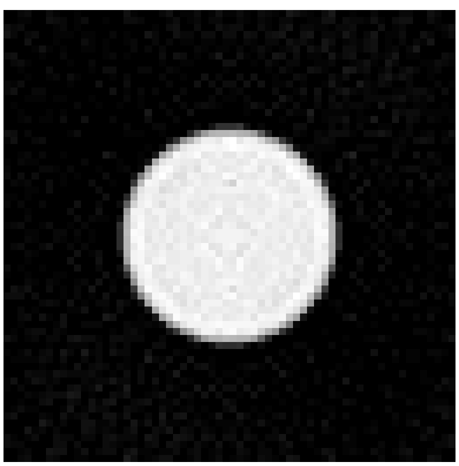
Figura 5.38 – (a) Reconstrução de uma fonte puntiforme centrada no *pixel* 32,5 com distribuição gaussiana com *art.m*. – iteração = 1 / correção aditiva. (b) Perfis das linhas 10 / 32 / 55 de 5.38a . (c) Idem 5.38a - iteração = 6. (d) Perfil da linha 32 de 5.38 c .



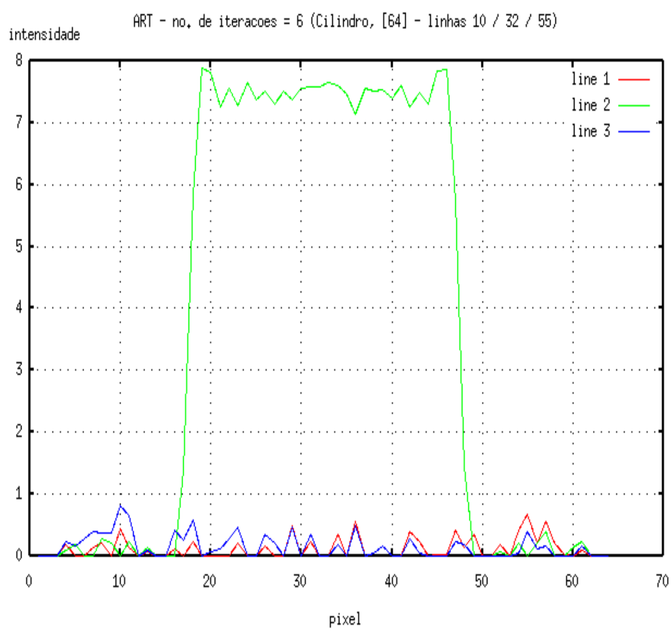
(a)



(b)

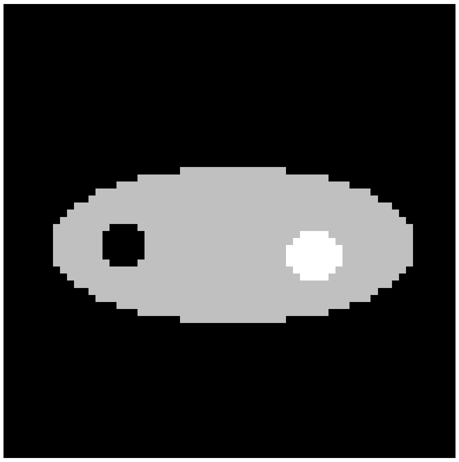


(c)

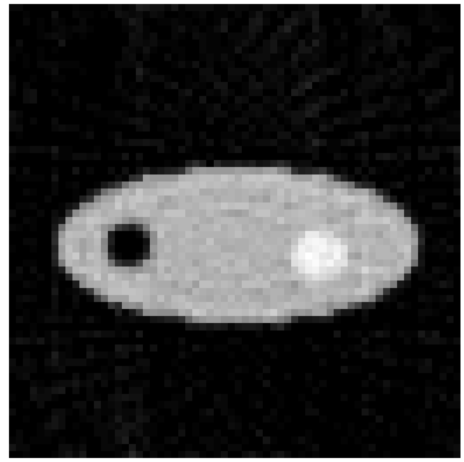


(d)

Figura 5.39 - (a) Reconstrução do corte seccional de um cilindro com distribuição de radiação uniforme com *art.m*. – iteração = 1 / correção aditiva. (b) Perfis das linhas 10 / 32 / 55 de 5.39a . (c) Idem 5.39a - iteração = 6. (d) Perfis das linhas 10 / 32 / 55 de 5.39 c .

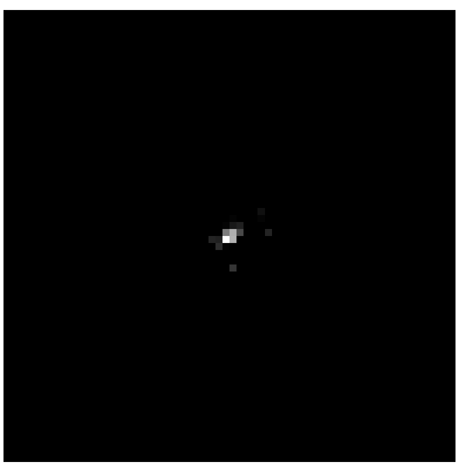


(a)

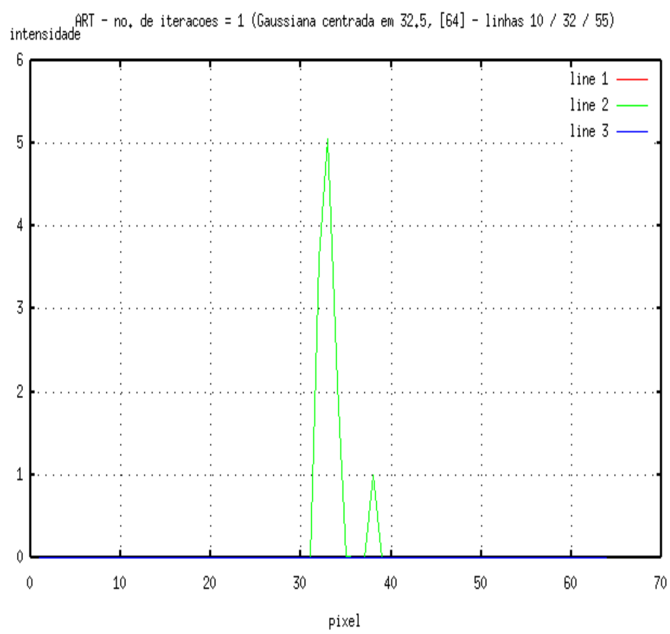


(b)

Figura 5.40 - (a) Objeto elipsóide – *phantom* matemático. (b) Reconstrução do objeto 5.40a com *art.m* - iteração = 6 / correção aditiva.

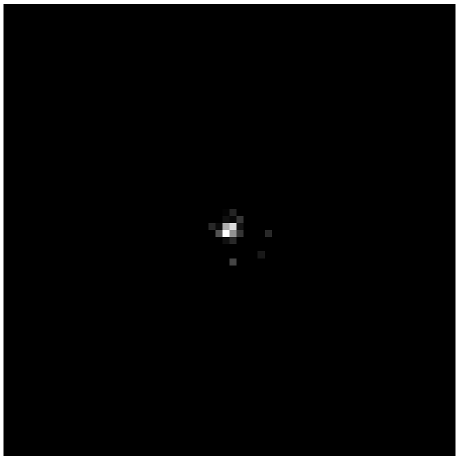


(a)

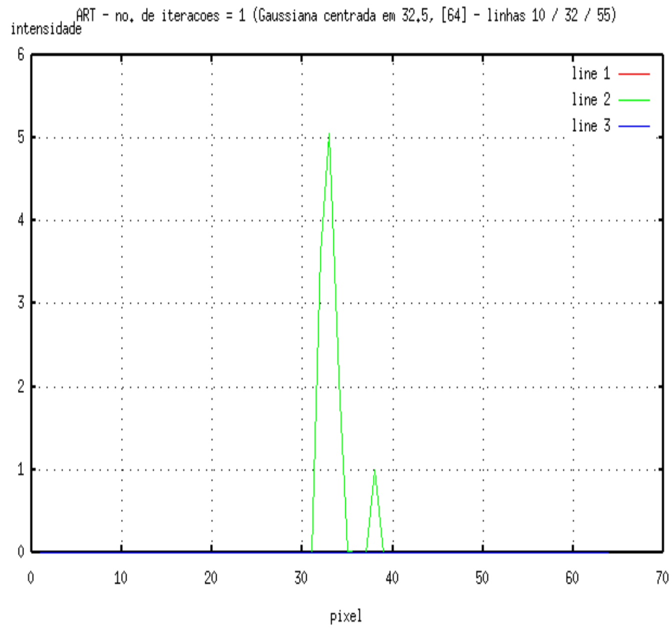


(b)

Legenda na pg. 103

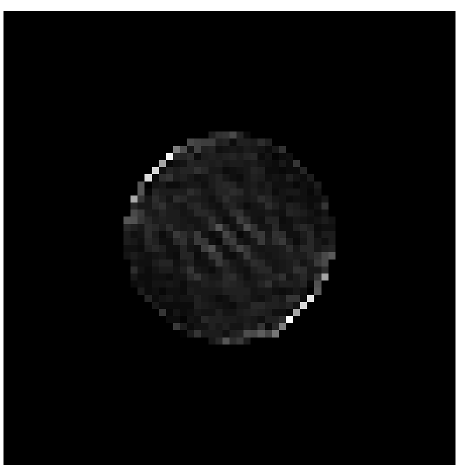


(c)

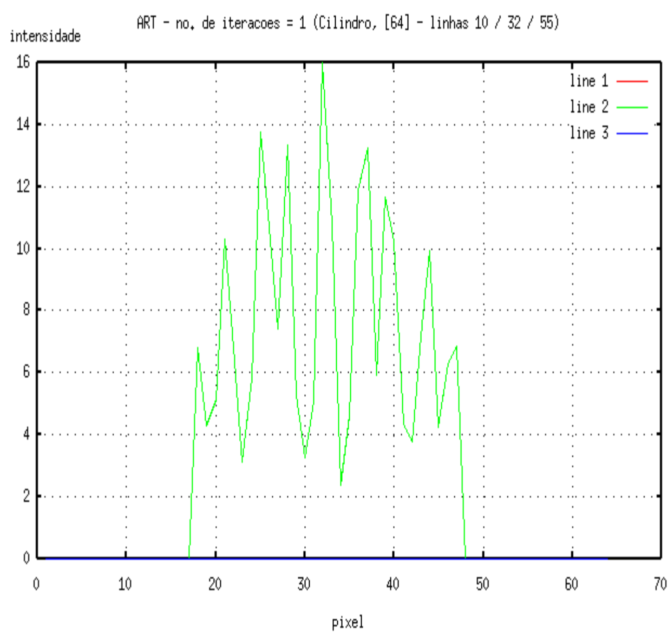


(d)

Figura 5.41 - (a) Reconstrução de uma fonte puntiforme centrada no *pixel* 32,5 com distribuição gaussiana com *art.m*. – iteração = 1 / correção multiplicativa. (b) Perfis das linhas 10 / 32 / 55 de 5.41a . (c) Idem 5.41a - iteração = 6. (d) Perfis das linhas 10 / 32 / 55 de 5.41 c .

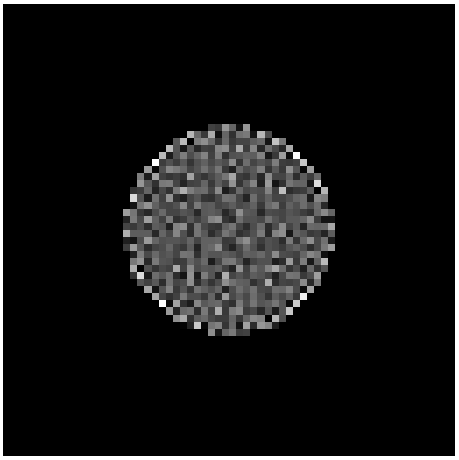


(a)

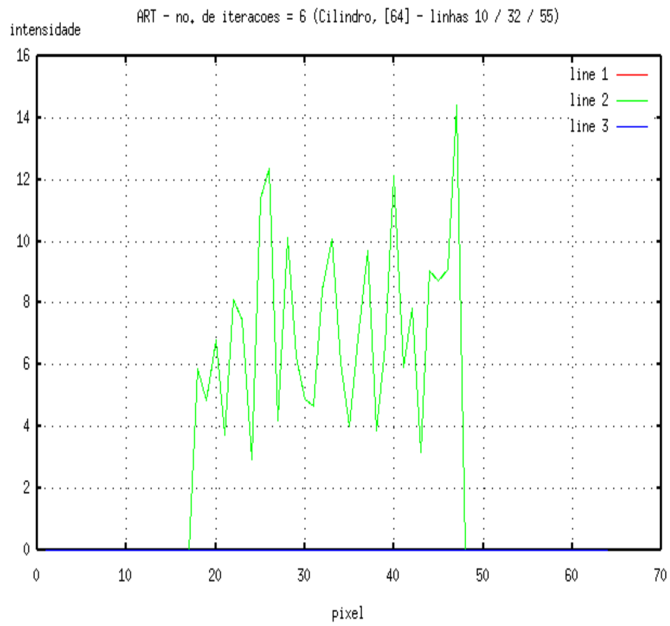


(b)

Legenda na pg. 104

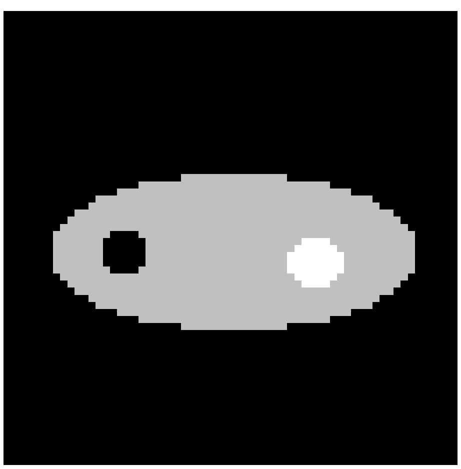


(c)

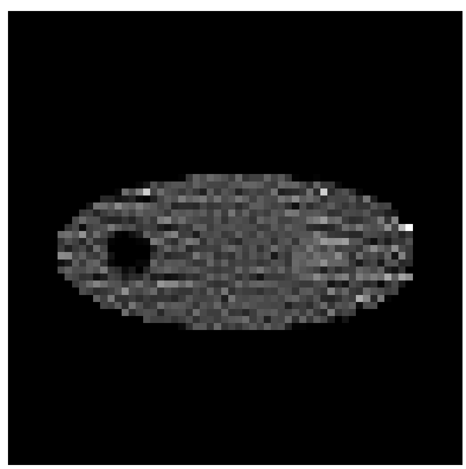


(d)

Figura 5.42 - (a) Reconstrução do corte seccional de um cilindro com distribuição de radiação uniforme com *art.m*. – iteração = 1 / correção multiplicativa. (b) Perfis das linhas 10 / 32 / 55 de 5.42a . (c) Idem 5.42a - iteração = 6. (d) Perfis das linhas 10 / 32 / 55 de 5.42 c .



(a)



(b)

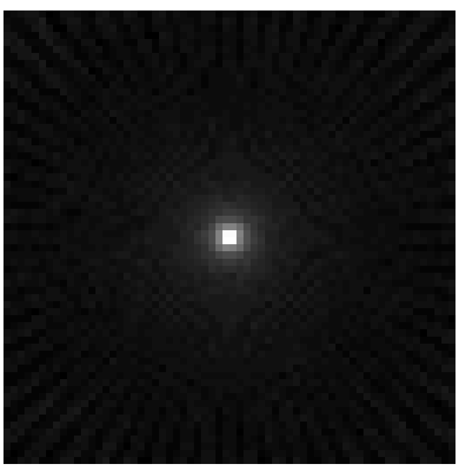
Figura 5.43 - (a) Objeto elipsóide – *phantom* matemático. (b) Reconstrução do objeto 5.43a com *art.m* - iteração = 6 / correção multiplicativa.

Observando as figuras de 5.38 à 5.43, verifica–se que o programa ***art.m*** reconstrói os objetos propostos corretamente, tanto para a opção com correção aditiva (equação 4.32.a) como multiplicativa (equação 4.32.b) .

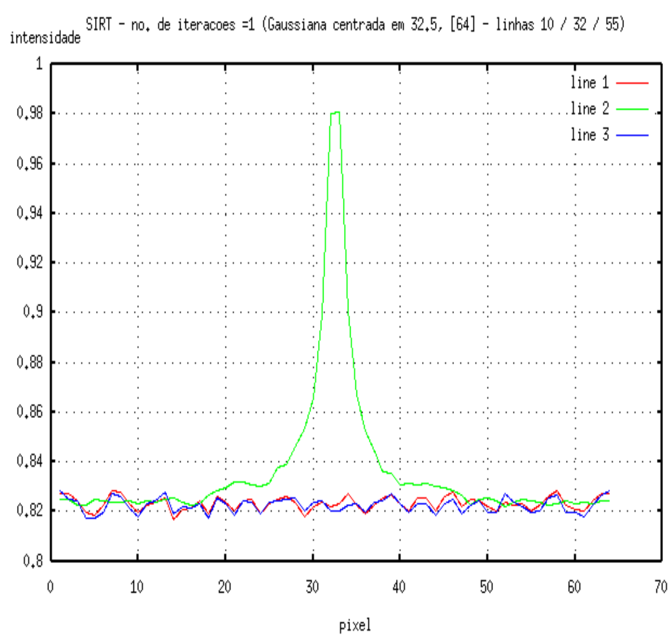
5.3.5 – *sirt.m*

O programa com técnica de reconstrução algébrica SIRT foi confeccionado, como descrito no ítem 5.1 deste capítulo, utilizando-se as equações 4.29, 4.31 à 4.34.

Para análise qualitativa do programa, foram reconstruídos os mesmos objetos (exceto o cilindro onde o diâmetro passou a ser de 40 *pixels* – aproximadamente 26,4 cm) utilizados para os testes dos programas ***fbp.m***, ***fbpw.m*** e ***art.m***, com um número de iterações igual 6 (novamente escolhido aleatoriamente), tanto para a técnica com a correção do tipo aditiva como a multiplicativa. Os resultados apresentados nas figuras de 5.44 à 5.49 mostram as imagens obtidas na primeira e última iteração, sem se fazer menção aos critérios de convergência (Entropia / Desvio Padrão), sendo estes e as imagens intermediárias objetos de estudo posterior neste trabalho.

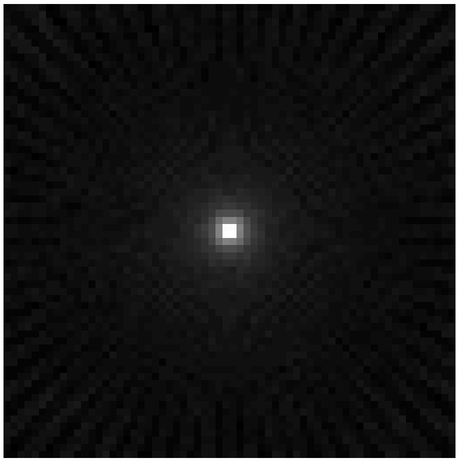


(a)

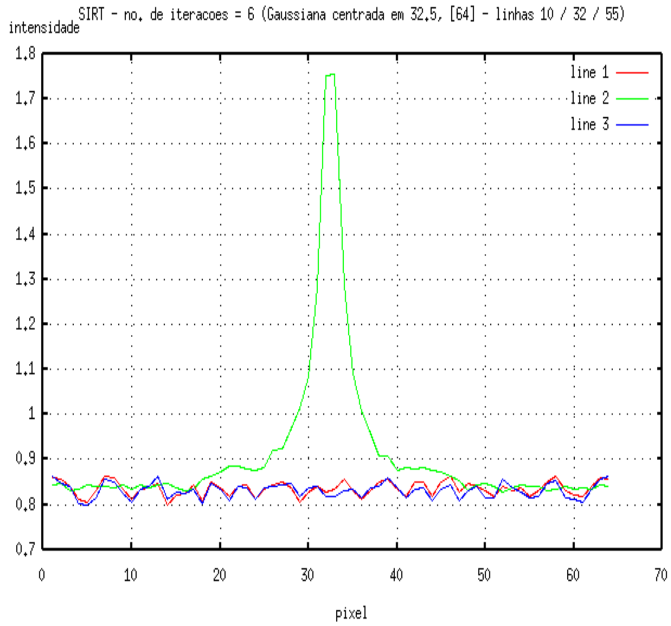


(b)

Legenda na pg. 106

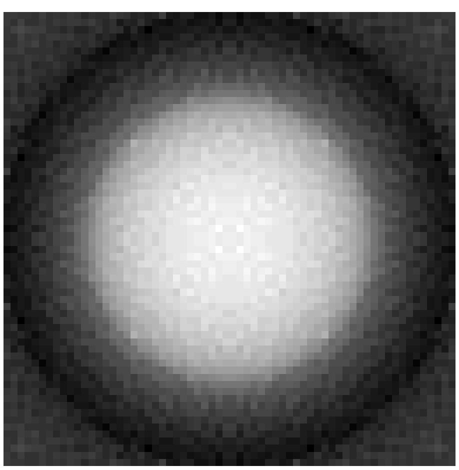


(a)

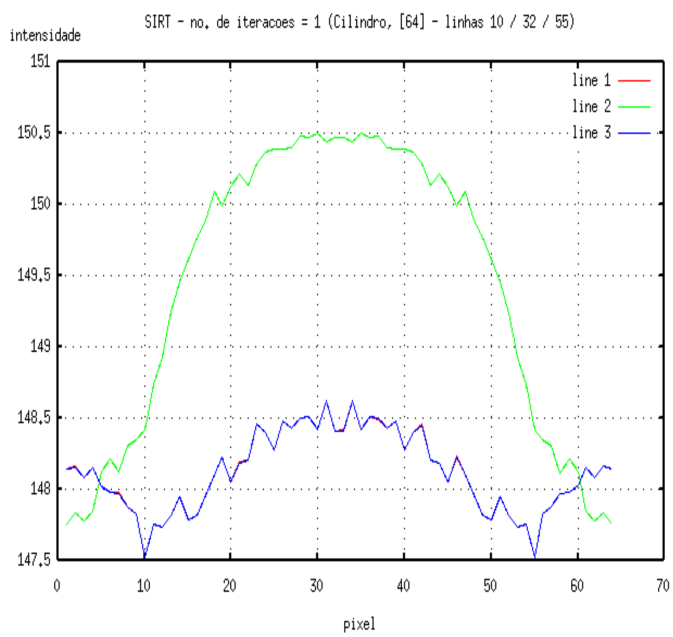


(b)

Figura 5.44 - (a) Reconstrução de uma fonte puntiforme centrada no *pixel* 32,5 com distribuição gaussiana com *sirt.m*. – iteração = 1 / correção aditiva. (b) Perfis das linhas 10 / 32 / 55 de 5.44a . (c) Idem 5.44a - iteração = 6. (d) Perfis das linhas 10 / 32 / 55 de 5.44 c .

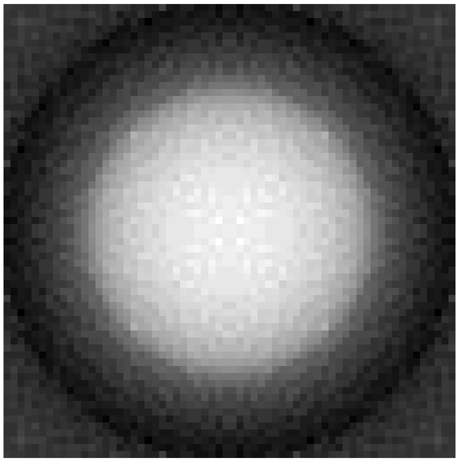


(a)

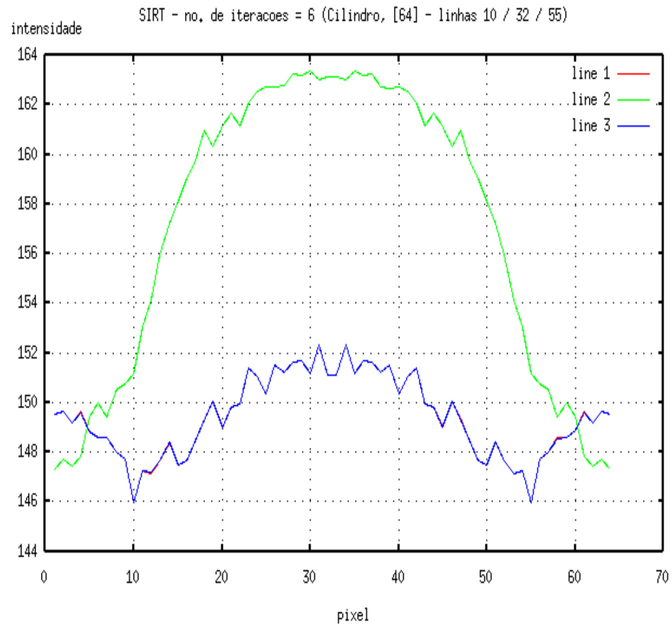


(b)

Legenda na pg 107

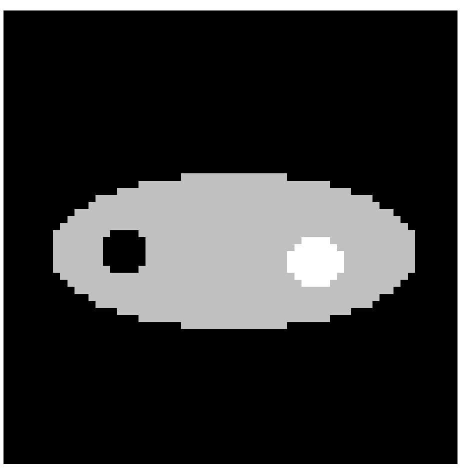


(c)

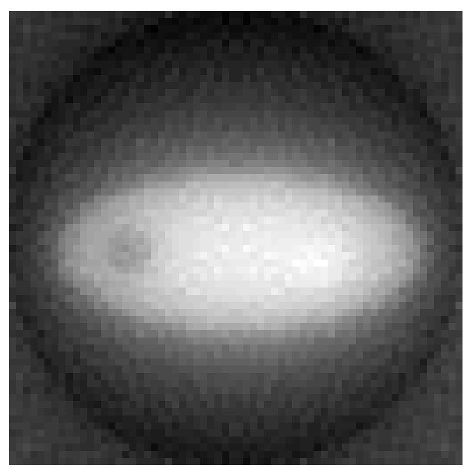


(b)

Figura 5.45 - (a) Reconstrução do corte seccional de um cilindro com distribuição de radiação uniforme com *sirt.m*. – iteração = 1 / correção aditiva. (b) Perfis das linha 10 / 32 / 55 de 5.45a . (c) Idem 5.45a - iteração = 6. (d) Perfis das linhas 10 / 32 / 55 de 5.45 c .

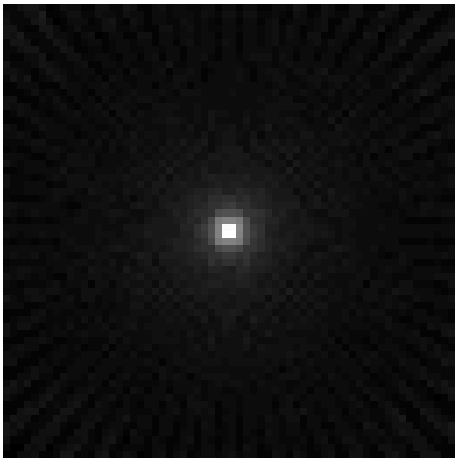


(a)

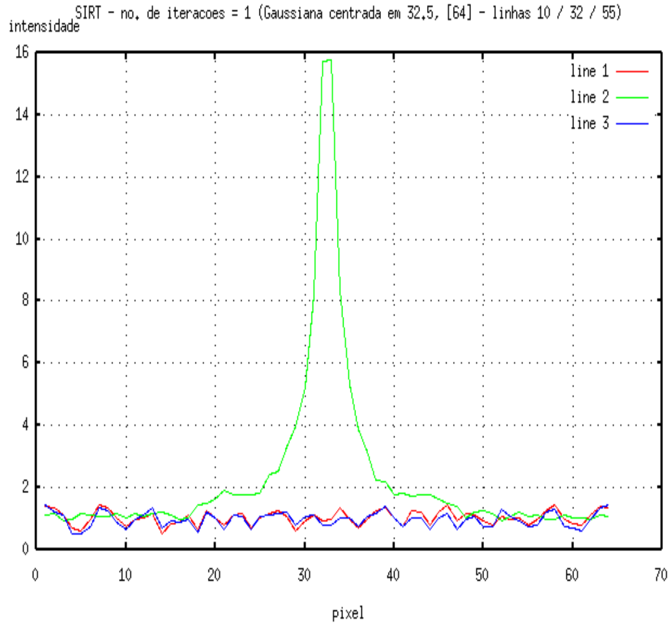


(b)

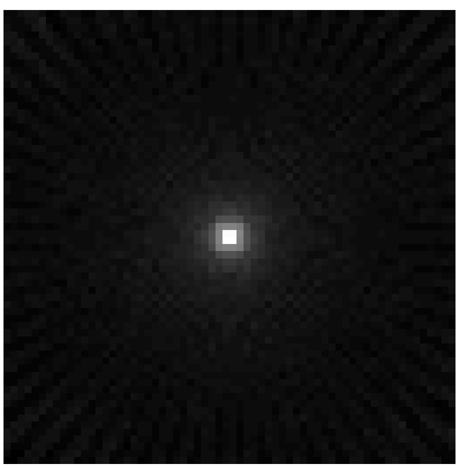
Figura 5.46 - (a) Objeto elipsóide – *phantom* matemático. (b) Reconstrução do objeto 5.46a com *sirt.m* - iteração = 6 / correção aditiva.



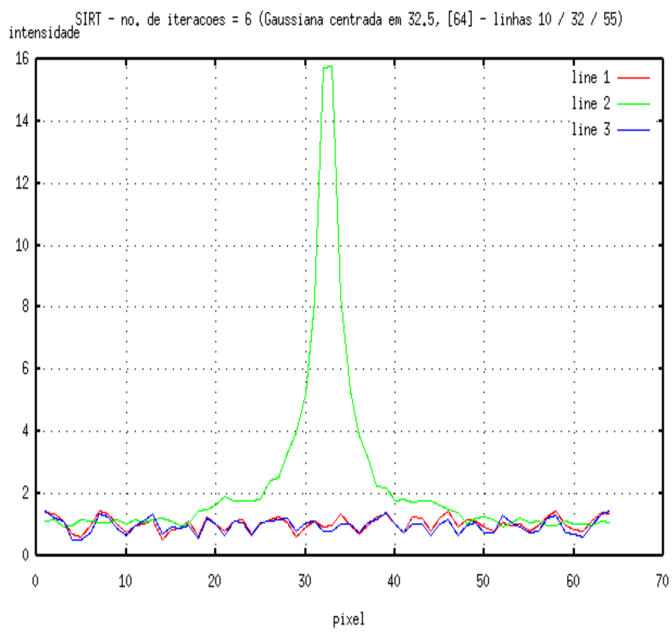
(a)



(b)

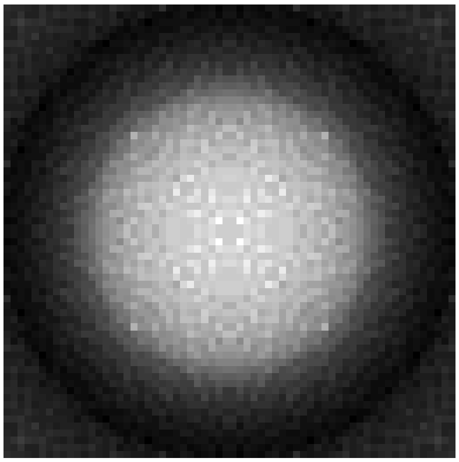


(c)

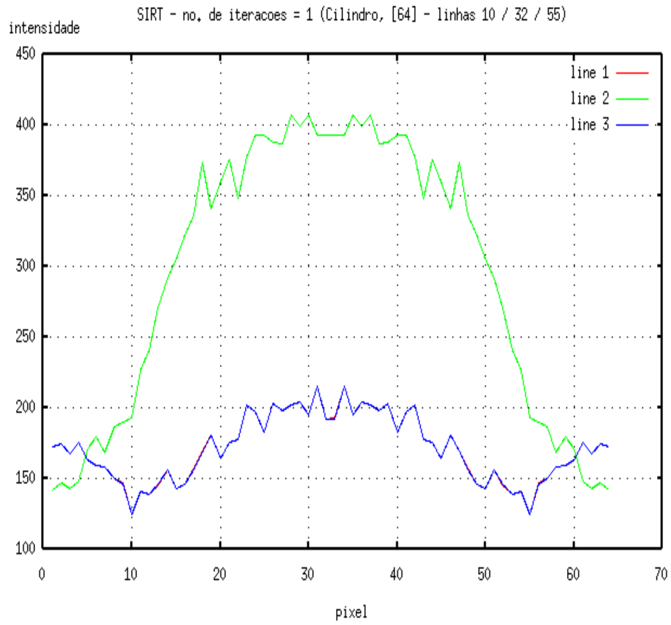


(d)

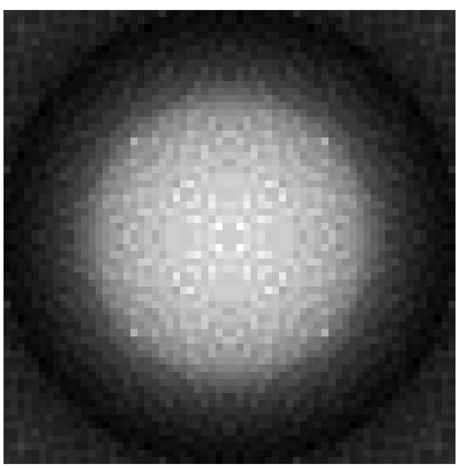
Figura 5.47 - (a) Reconstrução de uma fonte puntiforme centrada no *pixel* 32,5 com distribuição gaussiana com s*irt.m*. – iteração = 1 / correção multiplicativa. (b) Perfis das linha 10 / 32 / 55 de 5.47a . (c) Idem 5.47a - iteração = 6. (d) Perfis das linhas 10 / 32 / 55 de 5.47 c .



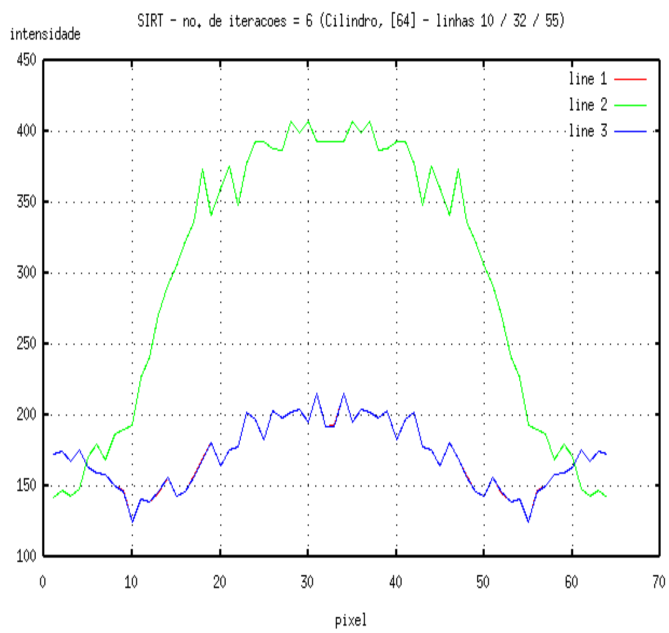
(a)



(b)

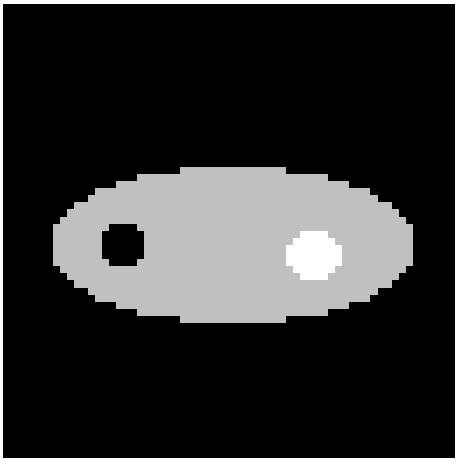


(c)

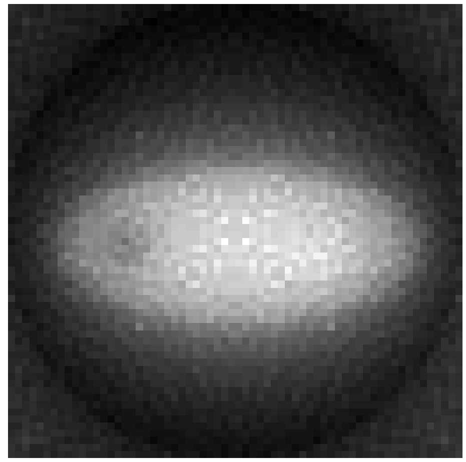


(d)

Figura 5.48 - (a) Reconstrução do corte seccional de um cilindro com distribuição de radiação uniforme com *sirt.m*. – iteração = 1 / correção multiplicativa. (b) Perfis das linhas 10 / 32 / 55 de 5.48a . (c) Idem 5.48a - iteração = 6. (d) Perfis das linhas 10 / 32 / 55 de 5.48 c .



(a)



(b)

Figura 5.49 - (a) Objeto elipsóide – *phantom* matemático. (b) Reconstrução do objeto 5.49a com *sirt.m* - iteração = 6 / correção multiplicativa.

Observando as figuras de 5.44 a 5.48, verifica–se que o programa ***sirt.m*** reconstrói os objetos propostos corretamente, tanto para a opção com correção aditiva (equação 4.33.a) como multiplicativa (equação 4.33.b). Observa-se também, como esperado [42], a convergência mais lenta na formação da imagem em relação ao algoritmo ART.

Com os resultados apresentados neste capítulo pode-se concluir que os programas implementados são capazes de reconstruir os cortes tomográficos necessários para a realização das comparações propostas na motivação deste trabalho. O capítulo seguinte trata da análise da performance dos algoritmos, tanto com *phantoms* matemáticos como com *phantoms* físicos.

Capítulo 6 – Funções e Programas – Provas Quantitativas (*Phantoms* Matemáticos *e Phantoms*[1])

A partir da reconstrução de cortes tomográficos de *phantoms* matemáticos (figuras geométricas conhecidas) e de *phantoms*, faz-se uma avaliação quantitativa dos programas implementados.

As provas procuram avaliar a uniformidade das imagens e a visibilização de objetos com variadas freqüências espaciais. Além disso, procura-se determinar para os algoritmos iterativos, qual o melhor critério de convergência quando da reconstrução das imagens clínicas.

Os estudos foram efetuados com *phantoms* matemáticos sem e com ruído aditivo, a fim de se obter simulações mais próximas a realidade no segundo caso, e com imagens de *phantoms* com vários *inserts*, adquiridas no SMN– HC / Unicamp.

6.1 – Testes sem a Adição de Ruído

6.1.1 – Uniformidade

O objetivo deste teste é avaliar a capacidade dos algoritmos em reconstruir uma imagem simples, medindo-se a uniformidade tomográfica do corte reconstruído. Também se verifica o comportamento dos algoritmos iterativos (iteração à iteração) para, com base também nos resultados obtidos nos testes de Uniformidade com ruído e de Contraste (com e sem ruído) , definir-se o critério de convergência . A avaliação da uniformidade se dá com a reconstrução da secção de um cilindro de 40 pixels (26,4

[1] *Phantom* – dispositivo utilizado para a verificação da performance de equipamentos, podendo ou não imitar um órgão do corpo humano. Normalmente são construídos em acrílico (ρ (g/cm^3) ˜ tecido humano).

cm) de diâmetro preenchido por uma distribuição uniforme de radiação (ρ igual a 10). A imagem do corte tomográfico esperada é um circulo, onde os valores dos pixels internos dão uma uniformidade igual a zero.

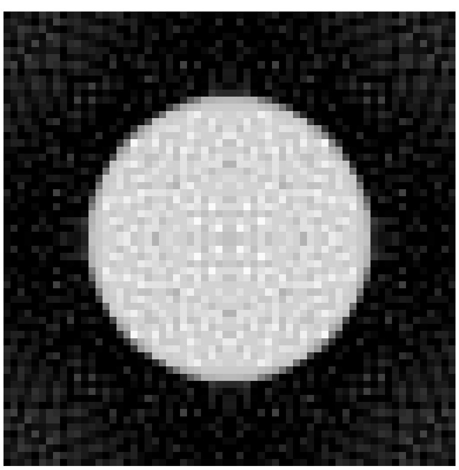
A IAEA sugere em seus estudos mais recentes[2] como método para o cálculo da uniformidade tomográfica o valor calculado pela equação 6.1:

$$U = \frac{(C_{\max} - C_{\min})*100}{C_{media}} \qquad \textbf{(6.1)}$$

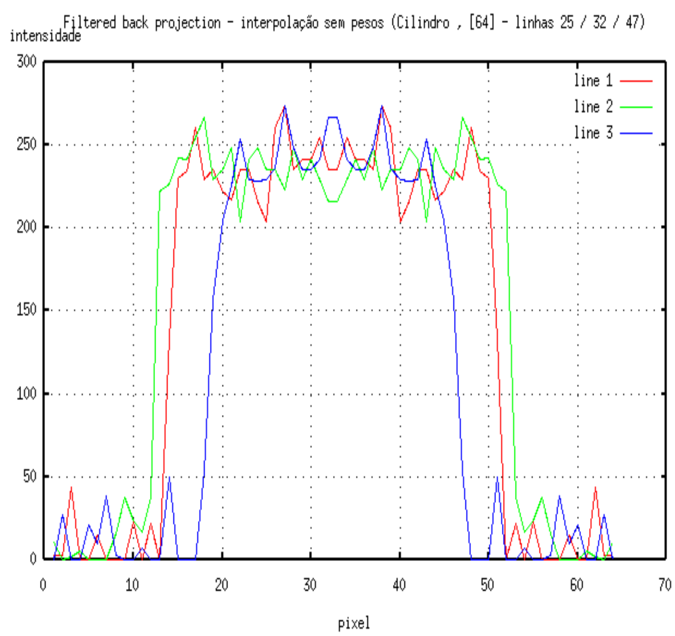
onde: U = uniformidade tomográfica, C_{max} = o número máximo, C_{min} o número mínimo e C_{media} a média das contagens de um perfil com largura de 3 a 5 *pixels* passando pelo centro de rotação na direção do eixo X e outro do Y.

Definiu–se inicialmente de forma aleatória para os programas iterativos 12 iterações. Os resultados – imagens, perfis e valores encontram–se nas figuras de 6.1 à 6.13 e Tabelas 6.1 à 6.7.

- ***FBP - Filtered Back Projection***



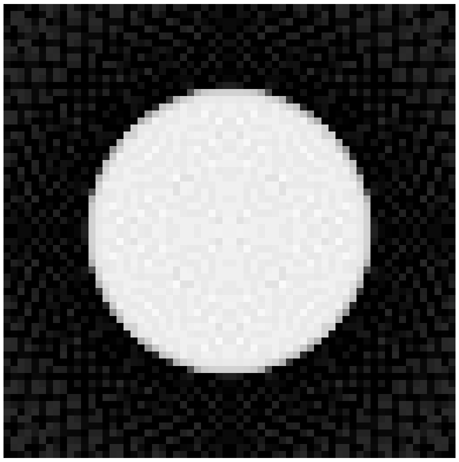
(a)



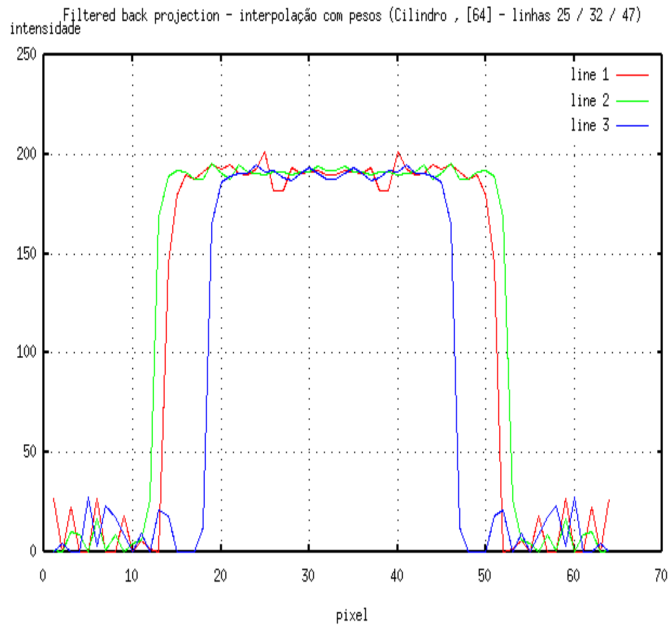
(b)

[2] Conforme contato pessoal com representante.

Legenda na pg. 113



(c)



(d)

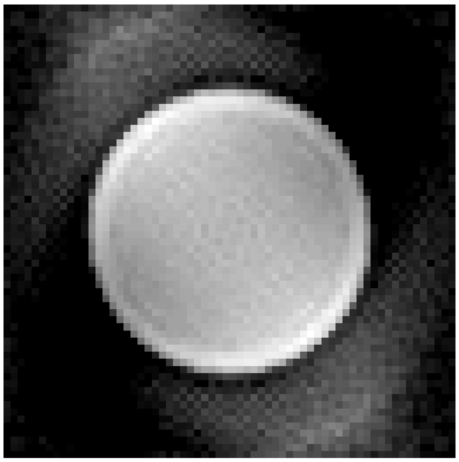
Figura 6.1 – (a) Corte tomográfico reconstruído com o programa *fbp.m*. (b) Perfis das linhas 25 / 32 / 47 da figura 6.1.a . (c) Corte tomográfico reconstruído com o programa *fbpw.m*. (d) Perfis das linhas 25 / 32 / 47 da figura 6.1.c .

	x				y				
Imagem	***Cmax***	***Cmin***	***Cmedia***	***Unif tomo***	***Cmax***	***Cmin***	***Cmedia***	***Unif tomo***	***Unif. Media***
6.1.a	248,00	178,00	209,80	33,37	248,00	178,00	209,05	33,48	33,42
6.1.c	250,00	210,00	235,80	16,96	250,00	210,00	235,80	16,96	16,96

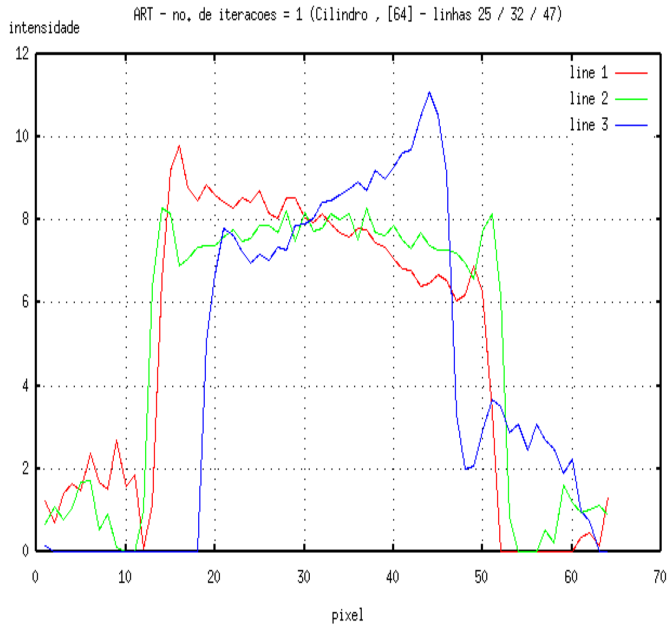
Tabela 6.1 – Valores da Uniformidade tomográfica calculados com a equação 6.1 para as reconstruções com os algoritmos *fbp.m* (figura 6.1.a) e *fbpw.m* (figura 6.1.c).

Verifica–se pelos valores anotados na Tabela 6.1 que a introdução da multiplicação pelos pesos de W na retro-projeção dos valores das projeções filtradas, implica numa melhora significativa da uniformidade do objeto reconstruído. Além disso, como as projeções não possuem ruído, a eficácia do filtro rampa para eliminação do efeito estrela é muito grande (o artefato restante possuí no máximo 2,5% do valor da intensidade dos *pixels* internos ao círculo, no caso de ***fbpw.m***). O valor *Unif. Média* na Tabela 6.1 traz a média entre as uniformidades tomográficas calculadas para os eixos X e Y.

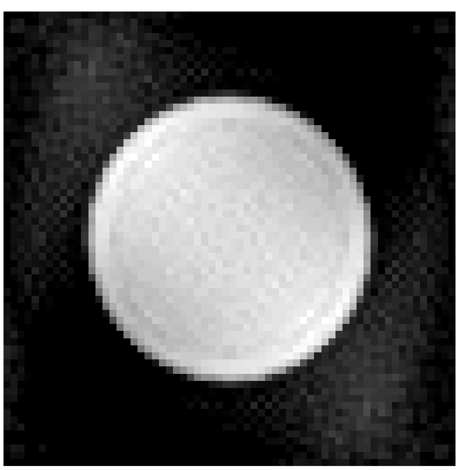
- ***Art – aditivo***



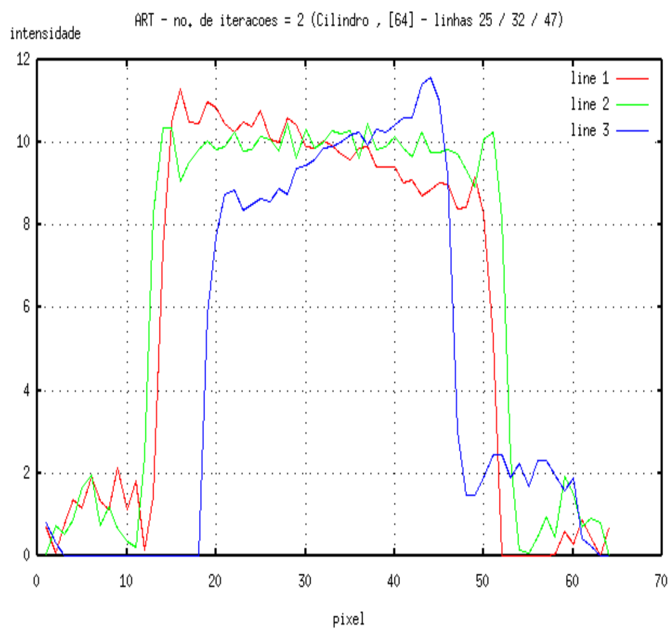
(a)



(b)

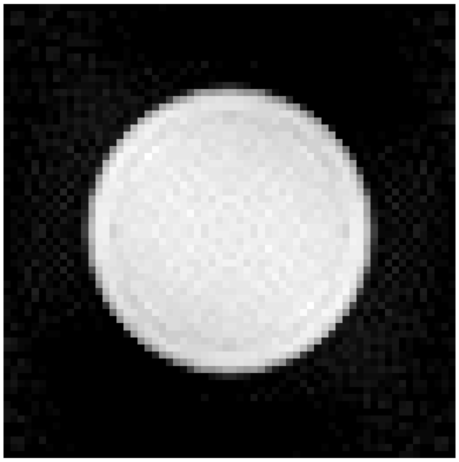


(c)

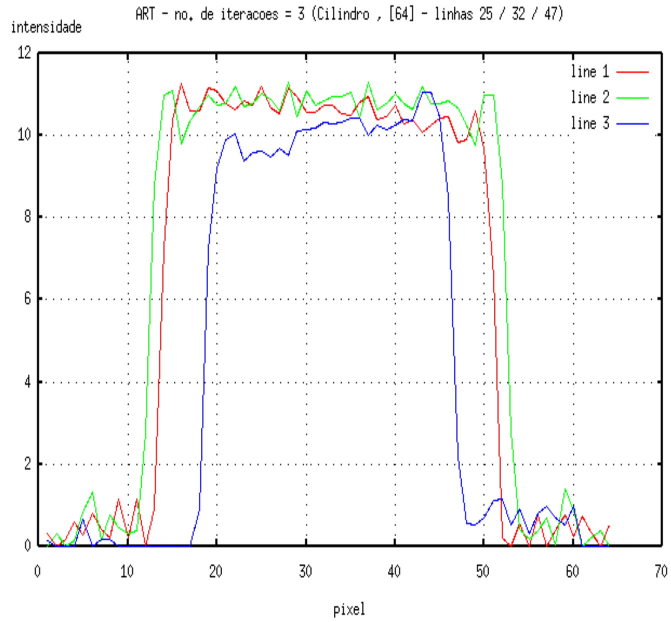


(d)

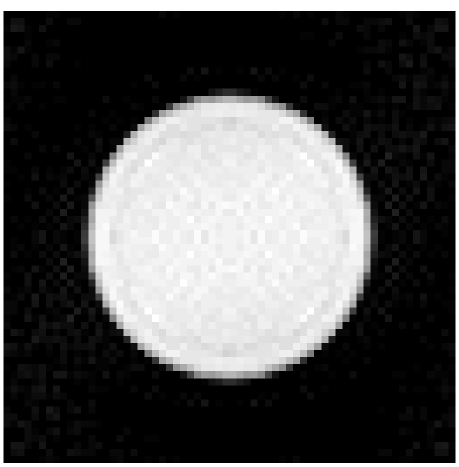
Figura 6.2 – (a) Corte tomográfico reconstruído com o programa *art.m* – nº de iterações = 1 . (b) Perfis das linhas 25 / 32 / 47 da figura 6.2.a . (c) Corte tomográfico reconstruído com o programa *art.m* – nº de iterações = 2 . (d) Perfis das linhas 25 / 32 / 47 da figura 6.2.c .



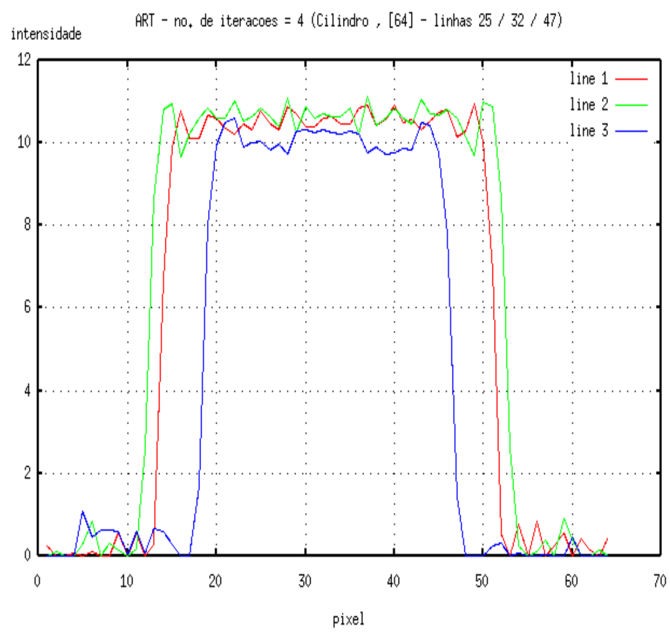
(e)



(f)

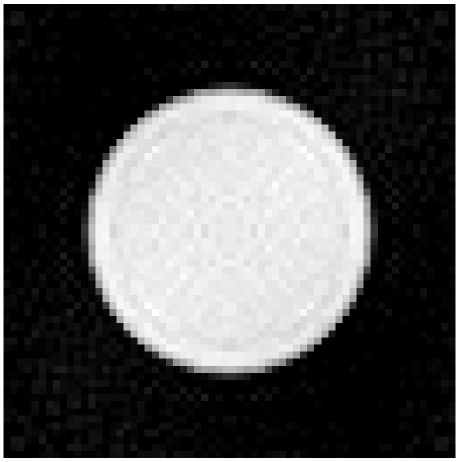


(g)

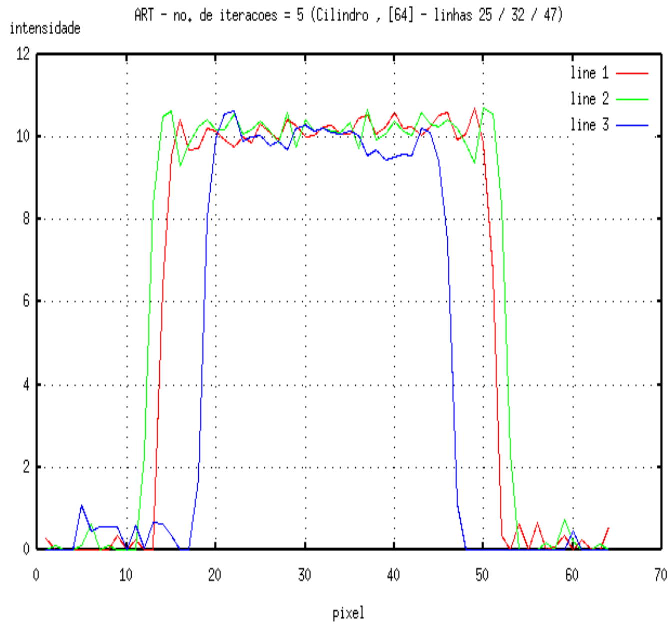


(h)

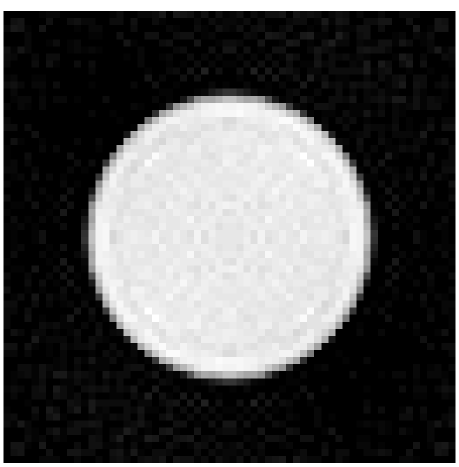
Figura 6.2 - (e) Corte tomográfico reconstruído com o programa *art.m* – nº de iterações = 3 . (f) Perfis das linhas 25 / 32 / 47 da figura 6.2.e . (g) Corte tomográfico reconstruído com o programa *art.m* – nº de iterações = 4 . (h) Perfis das linhas 25 / 32 / 47 da figura 6.2.g .



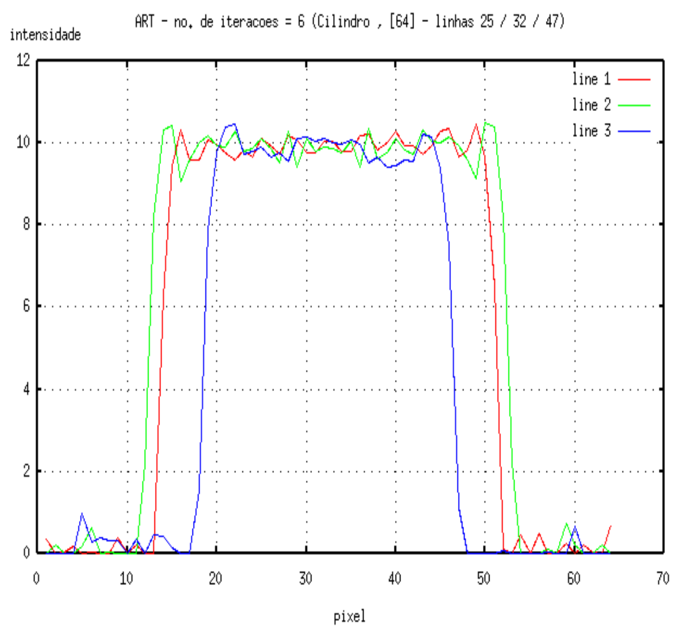
(i)



(j)

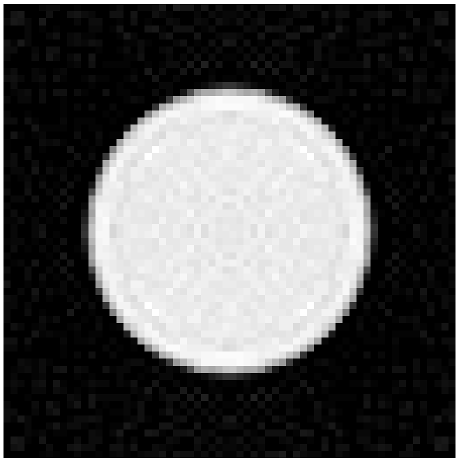


(k)

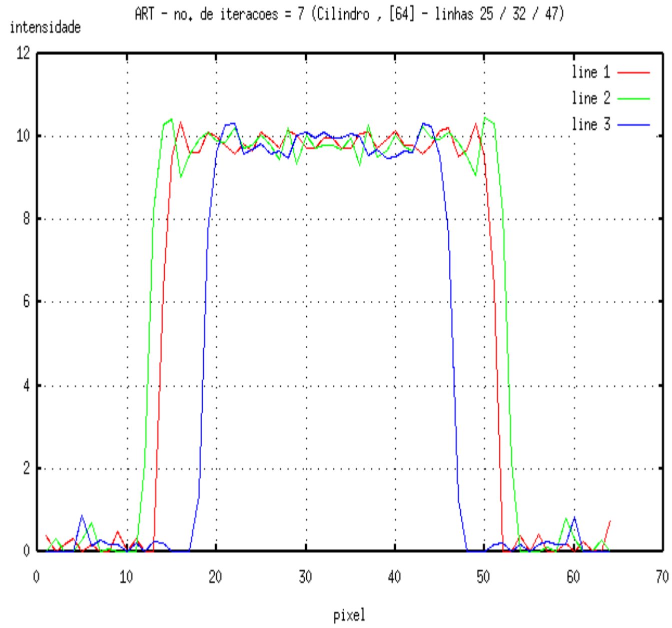


(l)

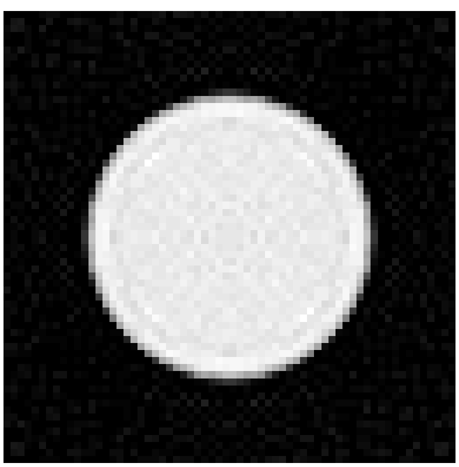
Figura 6.2 – (i) Corte tomográfico reconstruído com o programa *art.m* – nº de iterações = 5 . (j) Perfis das linhas 25 / 32 / 47 da figura 6.2.i . (k) Corte tomográfico reconstruído com o programa *art.m* – nº de iterações = 6 . (l) Perfis das linhas 25 / 32 / 47 da figura 6.2.k .



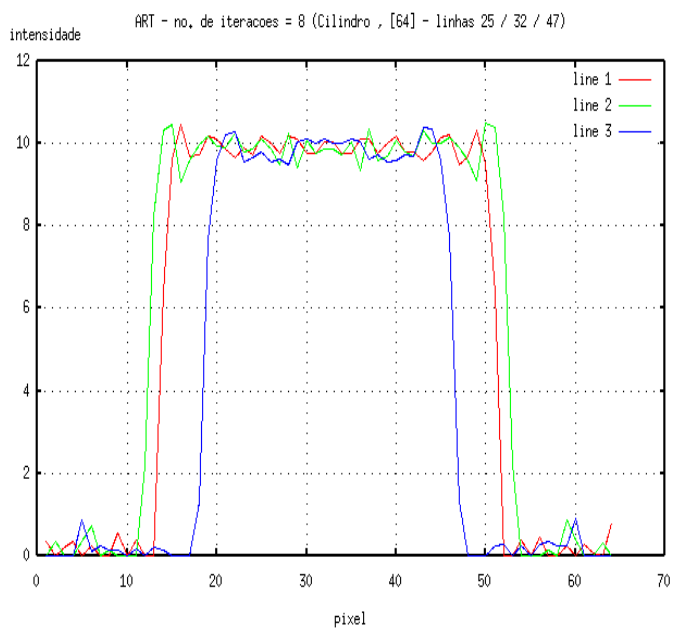
(m)



(n)

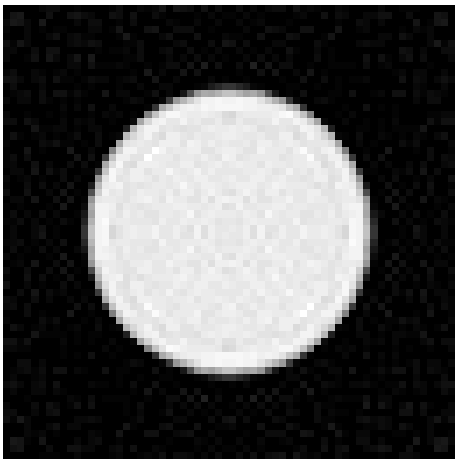


(o)

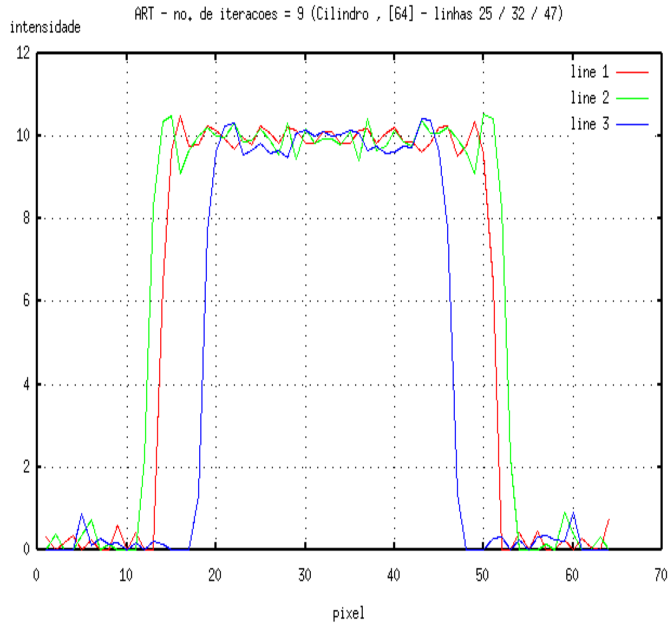


(p)

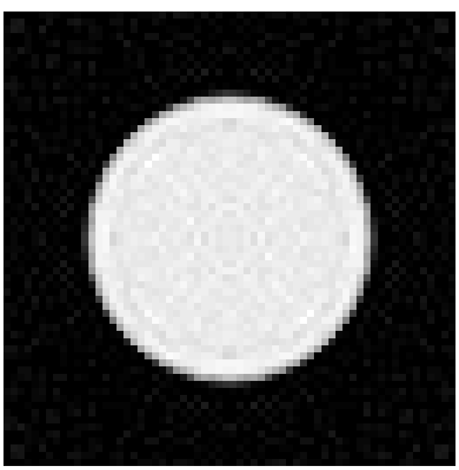
Figura 6.2 - (m) Corte tomográfico reconstruído com o programa *art.m* – nº de iterações = 7 . (n) Perfis das linhas 25 / 32 / 47 da figura 6.2.m . (o) Corte tomográfico reconstruído com o programa *art.m* – nº de iterações = 8 . (p) Perfis das linhas 25 / 32 / 47 da figura 6.2.o .



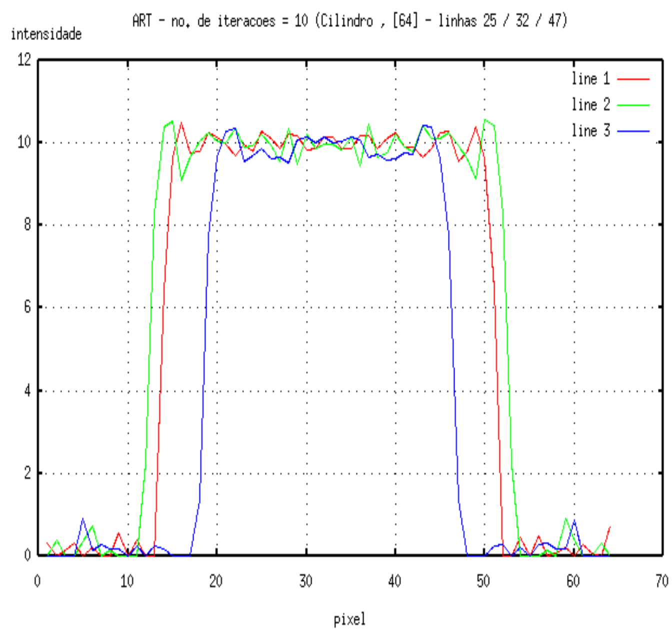
(q)



(r)

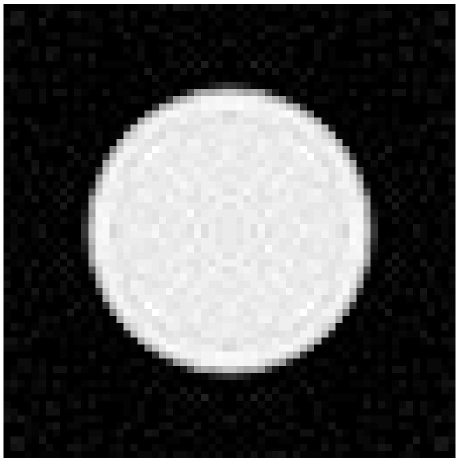


(s)

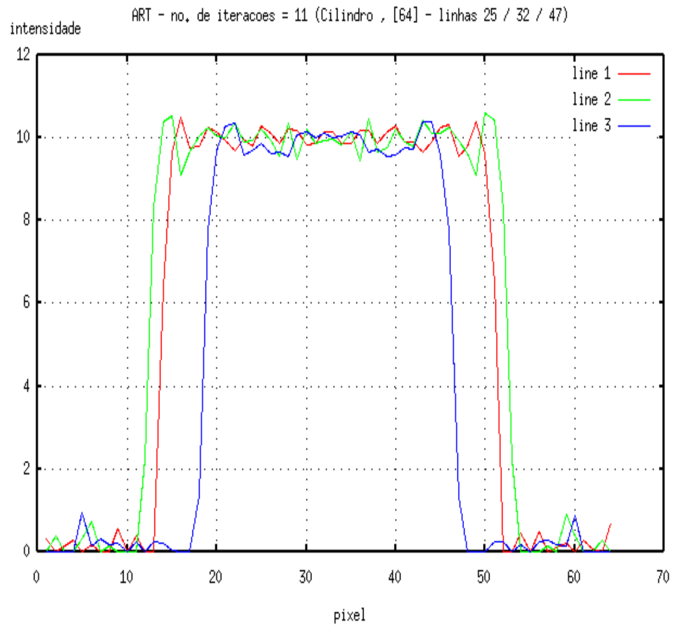


(t)

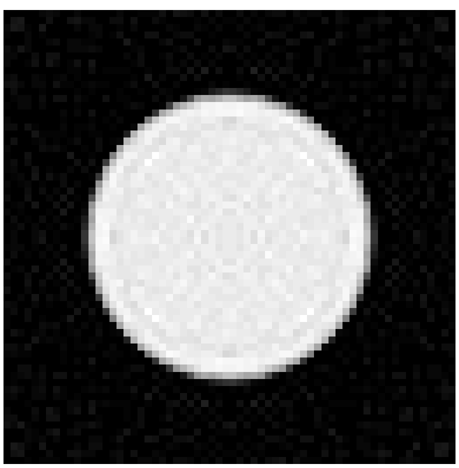
Figura 6.2 – (q) Corte tomográfico reconstruído com o programa *art.m* – nº de iterações = 9 . (r) Perfis das linhas 25 / 32 / 47 da figura 6.2.q . (s) Corte tomográfico reconstruído com o programa *art.m* – nº de iterações = 10 . (t) Perfis das linhas 25 / 32 / 47 da figura 6.2.s .



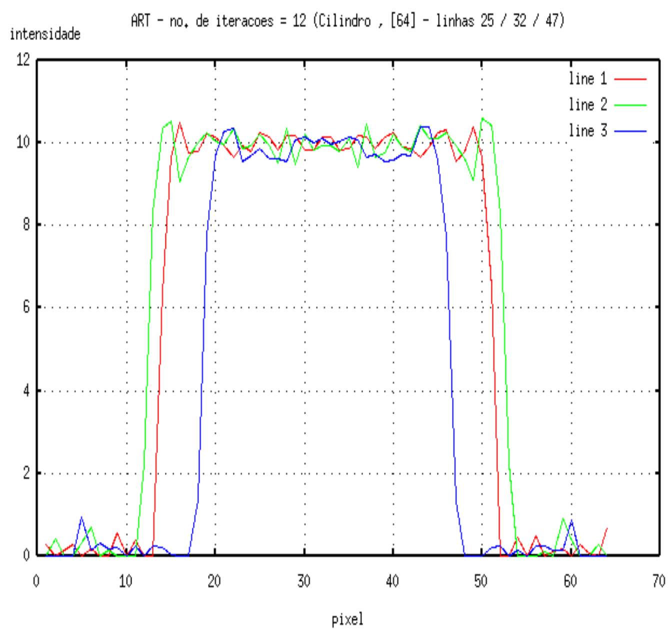
(u)



(v)



(x)



(z)

Figura 6.2 – (u) Corte tomográfico reconstruído com o programa *art.m* – nº de iterações = 11 . (v) Perfis das linhas 25 / 32 / 47 da figura 6.2.u . (x) Corte tomográfico reconstruído com o programa *art.m* – nº de iterações = 12 . (z) Perfis das linhas 25 / 32 / 47 da figura 6.2.x .

		x				y				
Imagem	**Iterações**	**Cmax**	**Cmin**	**Cmedia**	**Unif tomo**	**Cmax**	**Cmin**	**Cmedia**	**Unif tomo**	**Unif. Media**
6.2.a	1,00	194,00	125,00	173,54	39,76	235,00	170,00	189,70	34,26	37,01
6.2.c	2,00	231,00	162,00	214,00	32,24	227,00	162,00	214,01	30,37	31,31
6.2.e	3,00	243,00	178,00	231,83	28,04	243,00	158,00	225,09	37,76	32,90
6.2.g	4,00	251,00	182,00	238,35	28,95	251,00	166,00	233,19	36,45	32,70
6.2.i	5,00	247,00	178,00	232,20	29,72	243,00	174,00	230,60	29,92	29,82
6.2.k	6,00	247,00	178,00	231,45	29,81	251,00	178,00	231,90	31,48	30,65
6.2.m	7,00	251,00	178,00	231,95	31,47	251,00	178,00	233,10	31,32	31,39
6.2.o	8,00	251,00	178,00	231,50	31,53	251,00	178,00	232,10	31,45	31,49
6.2.q	9,00	251,00	178,00	231,80	31,49	247,00	178,00	232,05	29,73	30,61
6.2.s	10,00	251,00	178,00	232,25	31,43	247,00	178,00	232,50	29,68	30,55
6.2.u	11,00	251,00	178,00	232,75	31,36	251,00	178,00	232,85	31,35	31,36
6.2.x	12,00	251,00	178,00	233,00	31,33	251,00	178,00	232,90	31,34	31,34

Tabela 6.2 - Valores de Uniformidade tomográfica calculados com a equação 6.1 para as reconstruções com o algoritmo *art.m* (aditivo).

Os resultados para a Uniformidade tomográfica contidos na Tabela 6.2, mostram uma convergência destes valores a partir da iteração de número 6. Esta convergência pode ser observada nos perfis das imagens de 6.2.k à 6.2.x .

Isto implica que há um fator limitante para a qualidade da imagem pois, calculando-se o parâmetro de controle α para a Entropia e Desvio Padrão (figura 6.4) das imagens com as equações 4.35.b e 4.36.c respectivamente, observa-se que a uniformidade tomográfica deveria ainda continuar melhorando. Porém, verificando-se os perfis da linha 32 para as imagens com número de iterações igual a 1 / 3 / 6 / 9 / 12 (figura 6.3) constata-se que este fato não ocorre pois, após a iteração de número 6 , os valores associados aos *pixels* praticamente não mais se alteram. O fator limitante para a convergência será então a resolução espacial, dada pelo tamanho das matrizes utilizadas ([64]) pois, mesmo ainda ocorrendo uma convergência tanto por parte da Entropia da imagem como do Desvio Padrão das projeções calculadas, o valor da Uniformidade mantém-se com variações insignificantes.

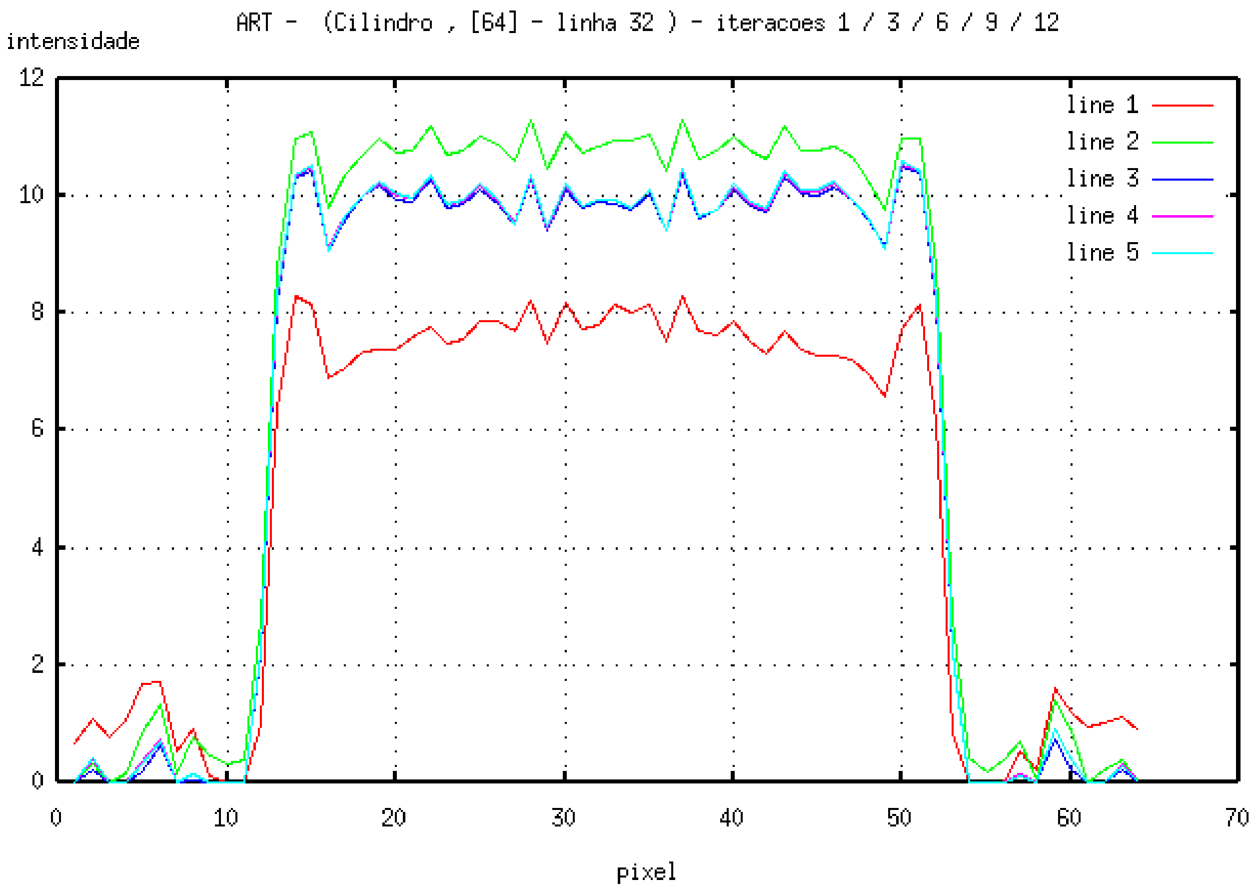
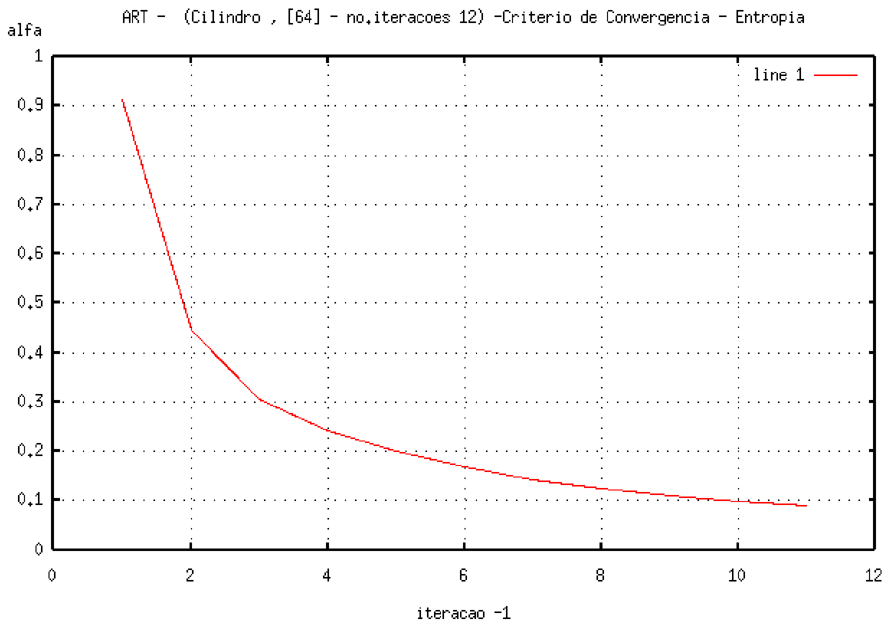
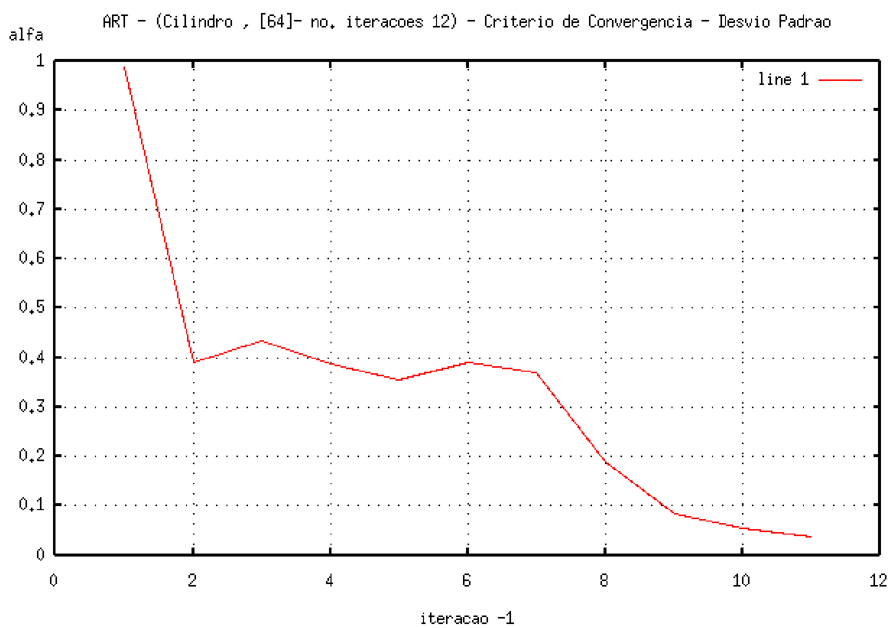


Figura 6.3 - Perfis das linhas 32 para as imagens de 1 / 3 / 6 / 9 /12 iterações .



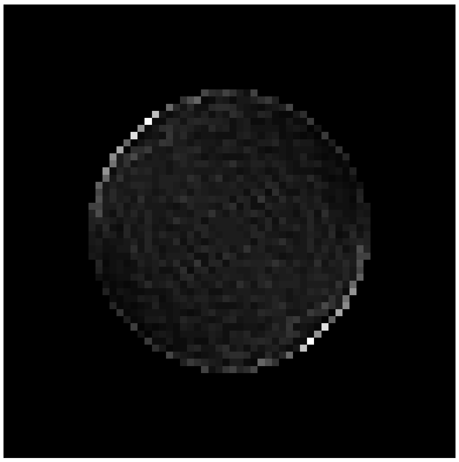
(a)



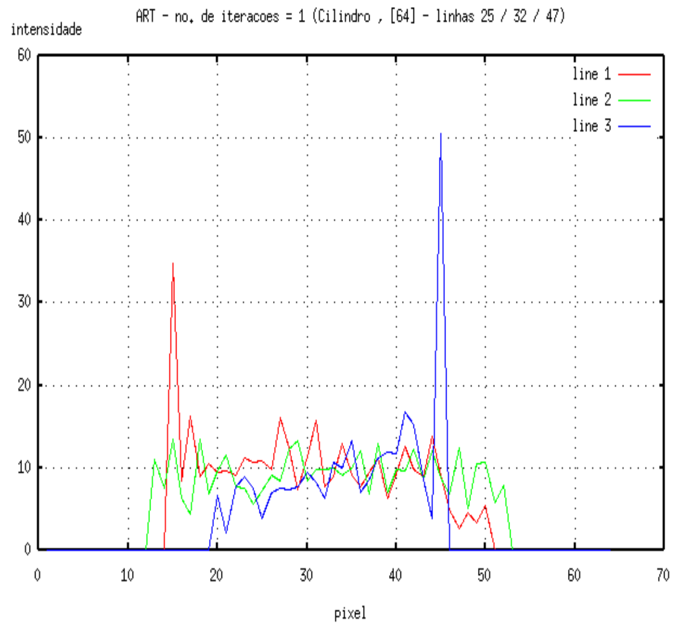
(b)

Figura 6.4 – (a) Parâmetro de controle α para a Entropia das imagens da Figura 6.2 . (b) Parâmetro de controle α para Desvio Padrão das projeções calculadas paras as imagens da Figura 6.2.

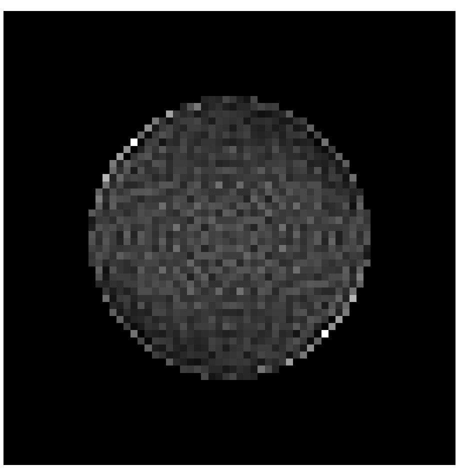
- ***Art (multiplicativo)***



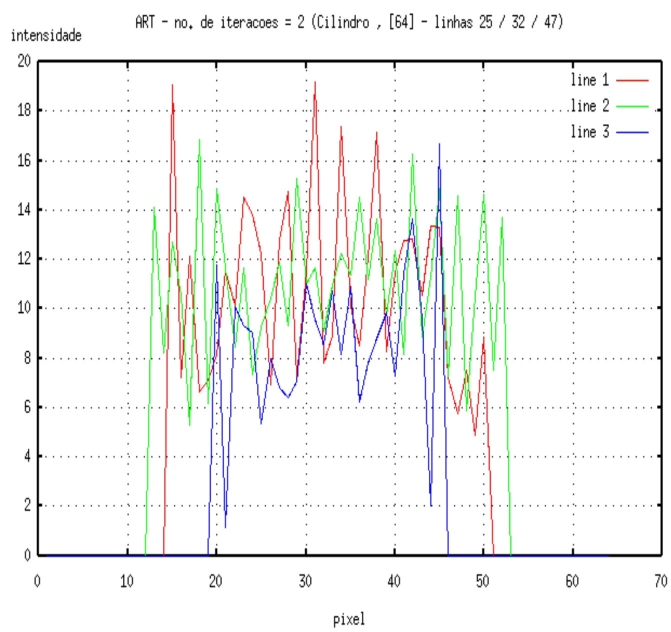
(a)



(b)

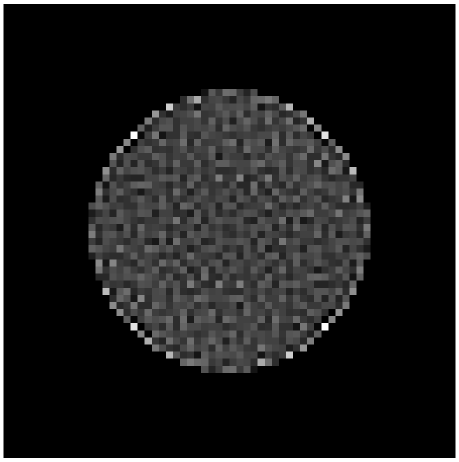


(c)

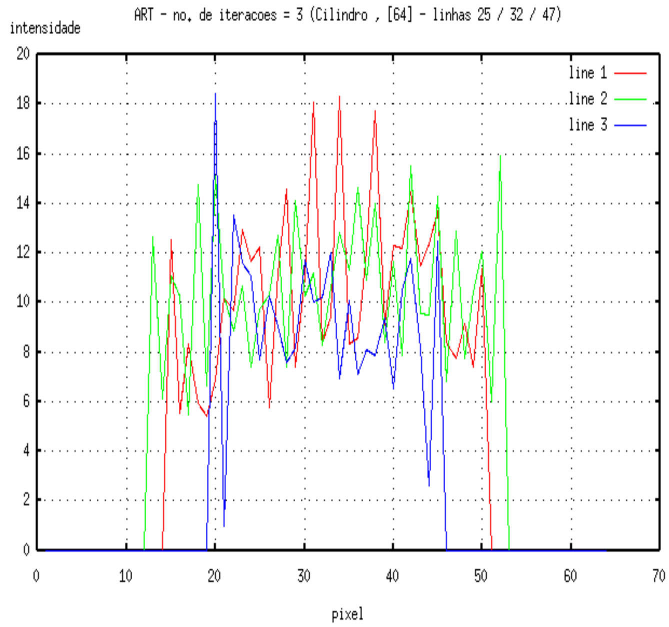


(d)

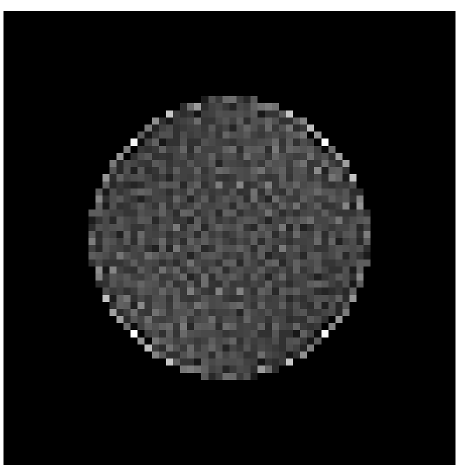
Figura 6.5 - (a) Corte tomográfico reconstruído com o programa *art.m* – nº de iterações = 1 . (b) Perfis das linhas 25 / 32 / 47 da figura 6.5.a . (c) Corte tomográfico reconstruído com o programa *art.m* – nº de iterações = 2 . (d) Perfis das linhas 25 / 32 / 47 da figura 6.5.c .



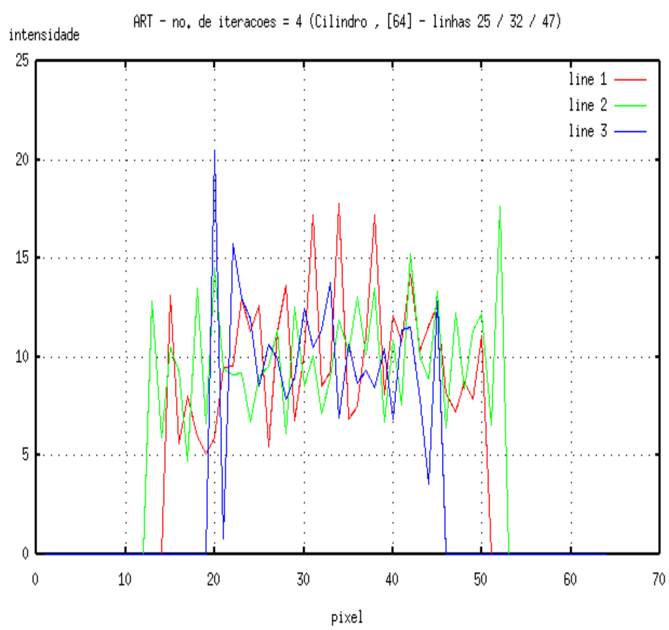
(e)



(f)

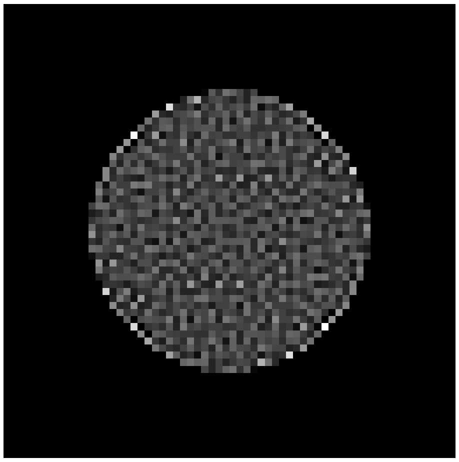


(g)

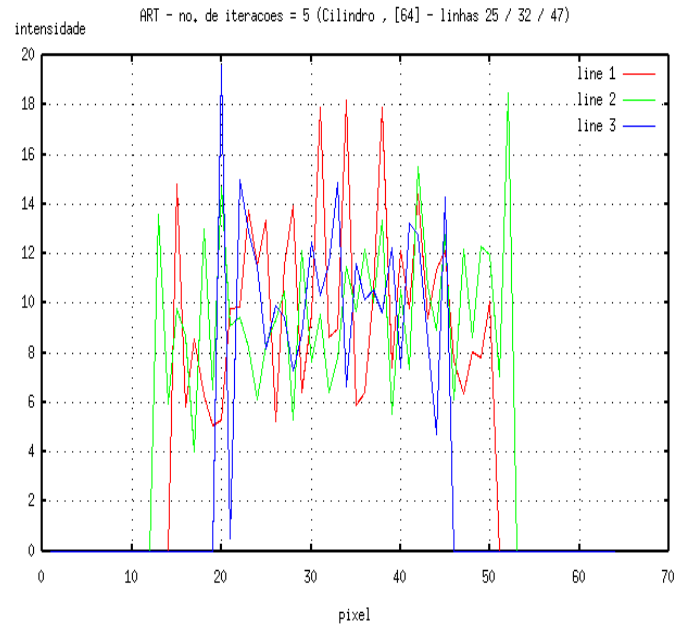


(h)

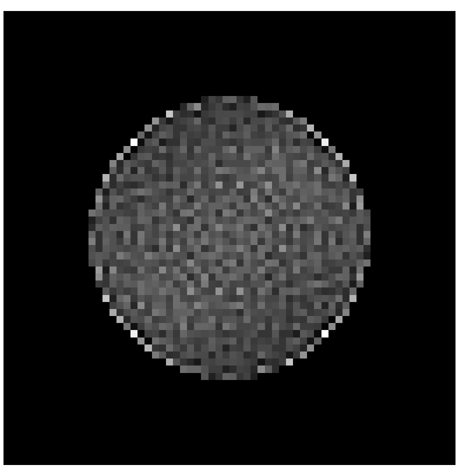
Figura 6.5 - (e) Corte tomográfico reconstruído com o programa *art.m* – nº de iterações = 3 . (f) Perfis das linhas 25 / 32 / 47 da figura 6.5.e . (g) Corte tomográfico reconstruído com o programa *art.m* – nº de iterações = 4 . (h) Perfis das linhas 25 / 32 / 47 da figura 6.5.g .



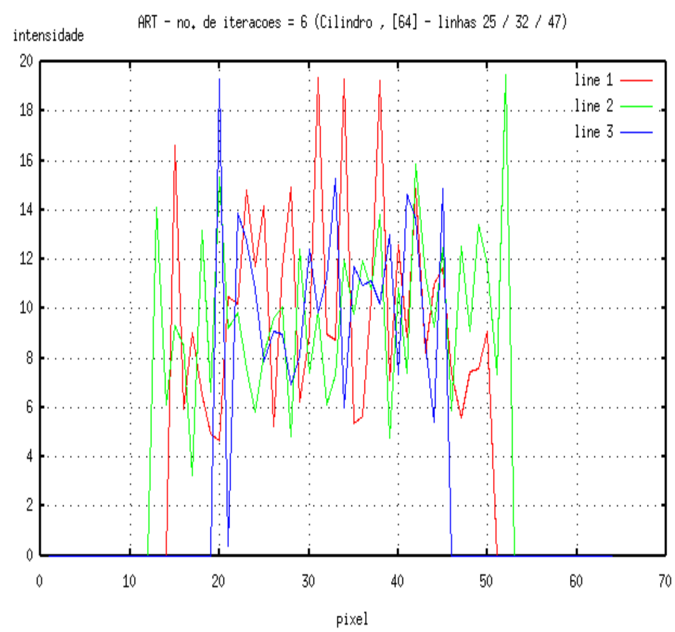
(i)



(j)

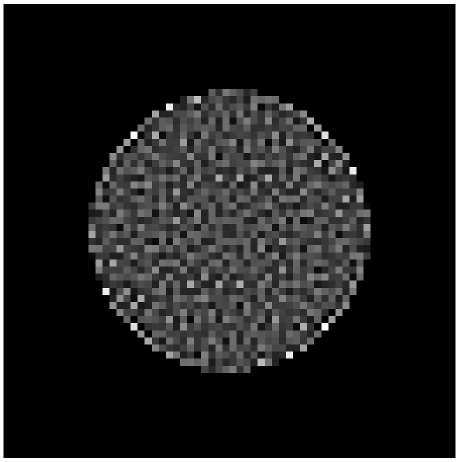


(k)

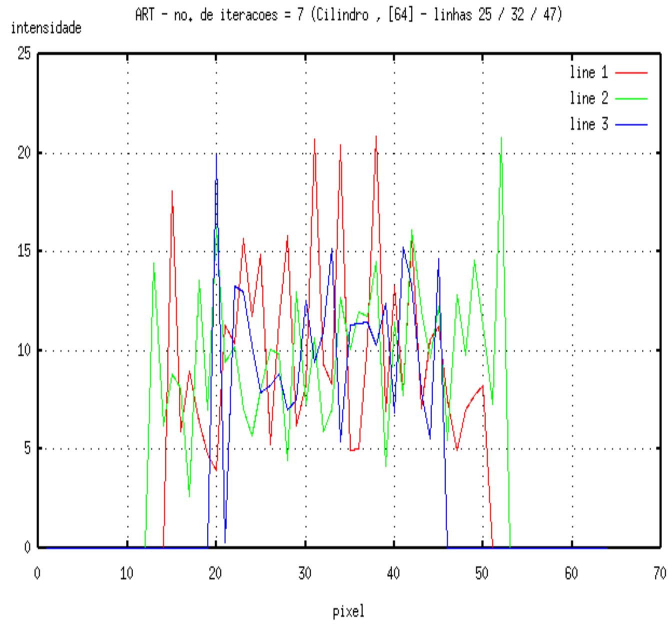


(l)

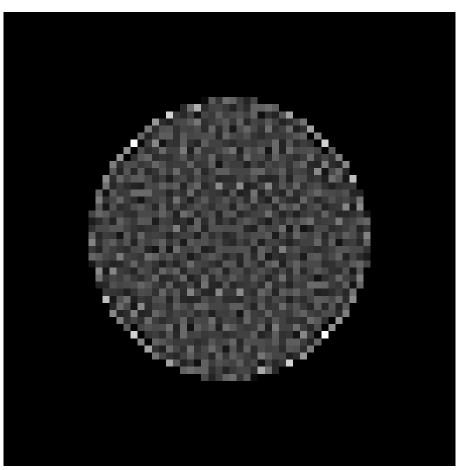
Figura 6.5 - (i) Corte tomográfico reconstruído com o programa *art.m* – nº de iterações = 5 . (j) Perfis das linhas 25 / 32 / 47 da figura 6.5.i . (k) Corte tomográfico reconstruído com o programa *art.m* – nº de iterações = 6 . (l) Perfis das linhas 25 / 32 / 47 da figura 6.5.k .



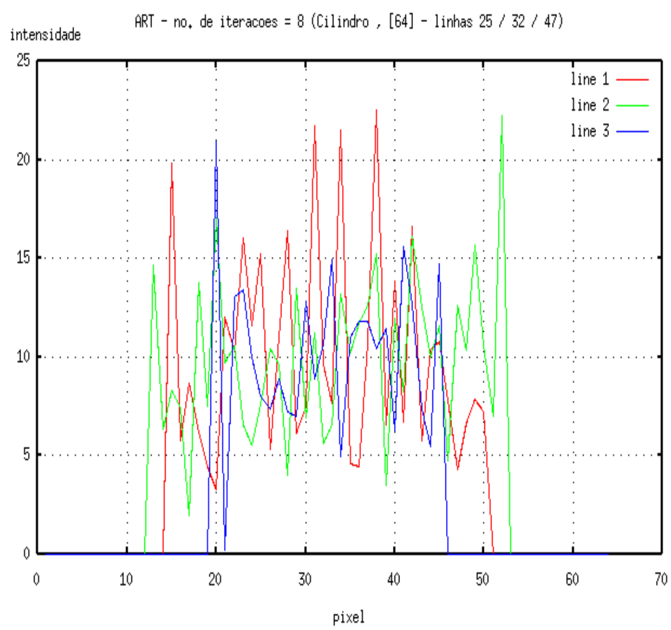
(m)



(n)

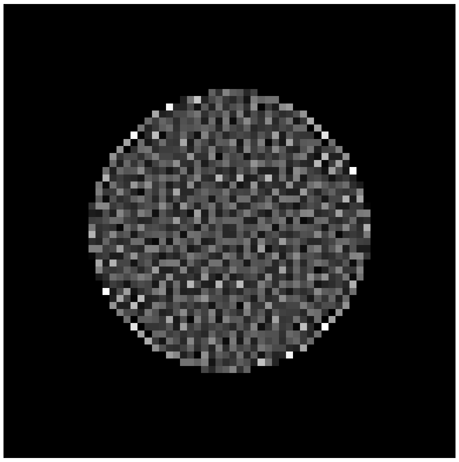


(o)

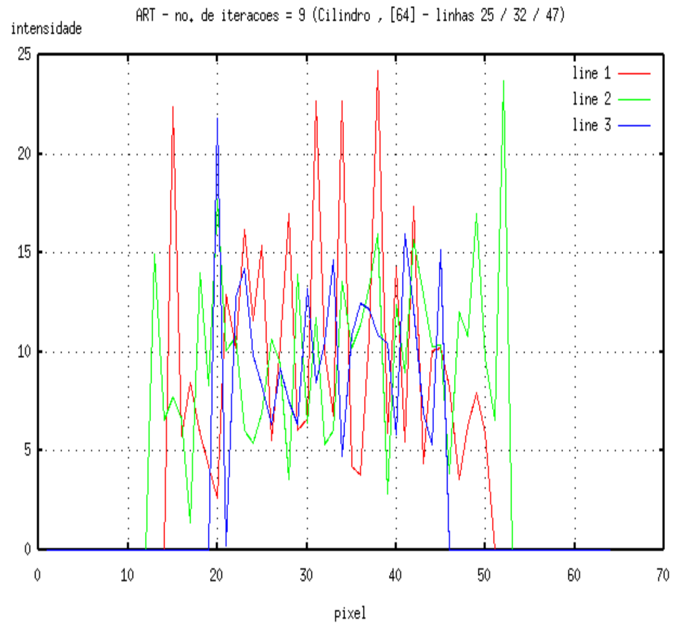


(p)

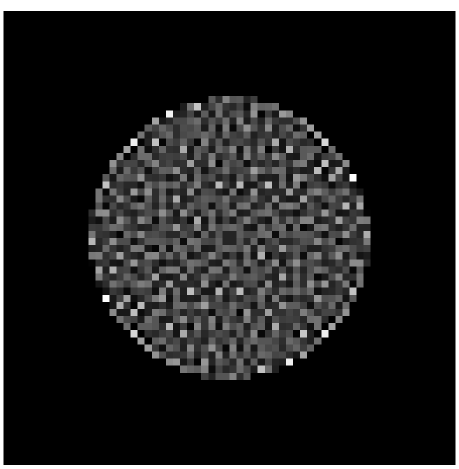
Figura 6.5 - (m) Corte tomográfico reconstruído com o programa *art.m* – nº de iterações = 7 . (n) Perfis das linhas 25 / 32 / 47 da figura 6.5.m . (o) Corte tomográfico reconstruído com o programa *art.m* – nº de iterações = 8 . (p) Perfis das linhas 25 / 32 / 47 da figura 6.5.o .



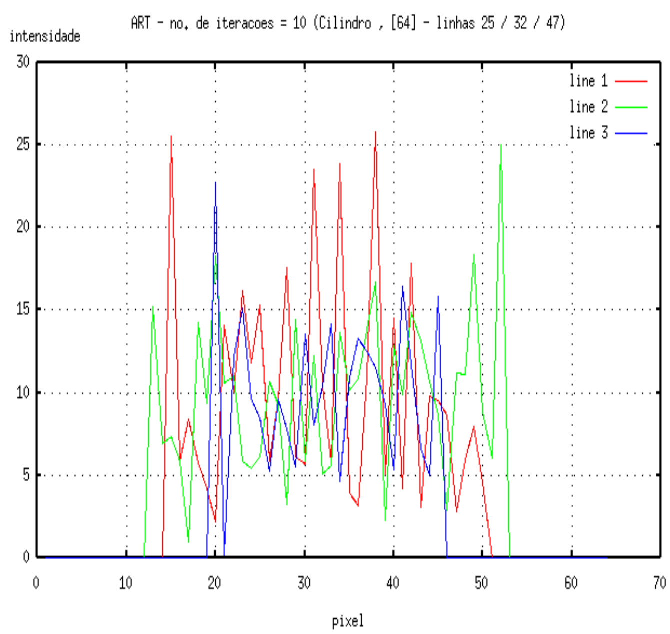
(q)



(r)

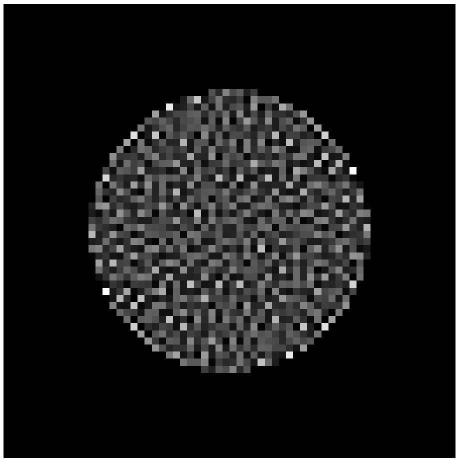


(s)

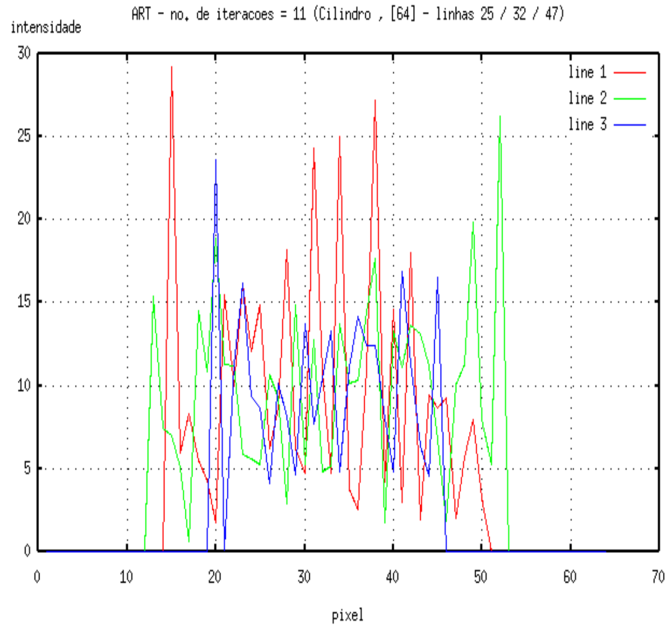


(t)

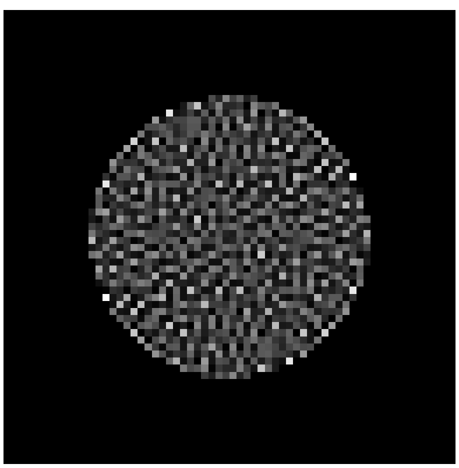
Figura 6.5 - (q) Corte tomográfico reconstruído com o programa *art.m* – nº de iterações = 9 . (r) Perfis das linhas 25 / 32 / 47 da figura 6.5.q . (s) Corte tomográfico reconstruído com o programa *art.m* – nº de iterações = 10 . (t) Perfis das linhas 25 / 32 / 47 da figura 6.5.s .



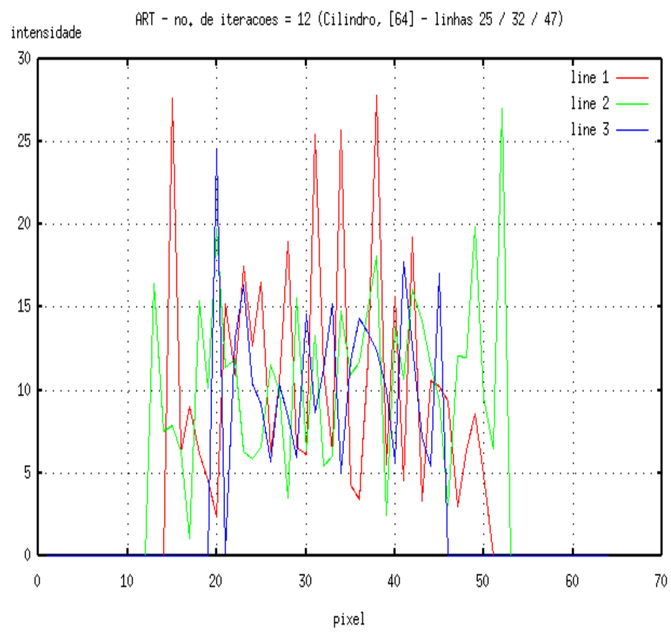
(u)



(v)



(x)



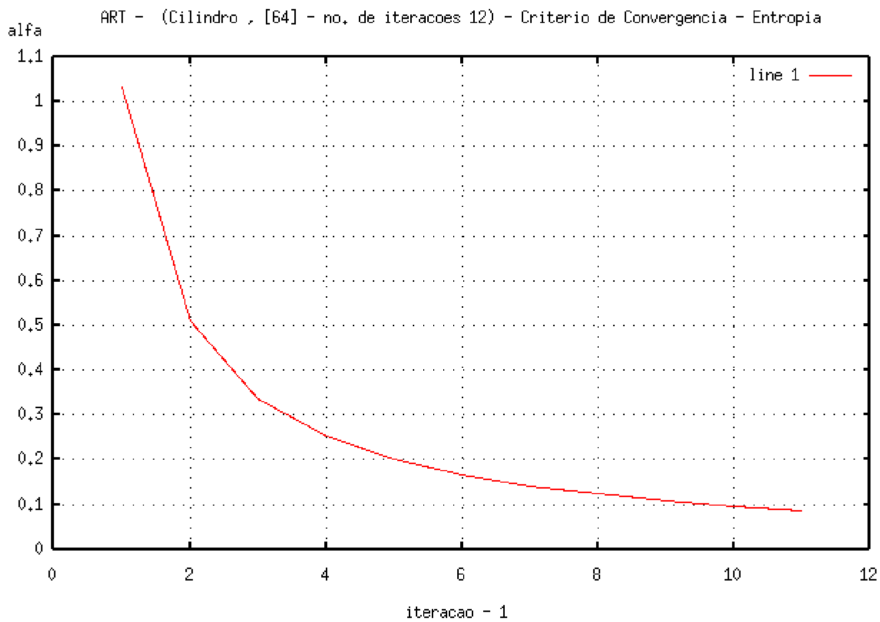
(z)

Figura 6.5 - (u) Corte tomográfico reconstruído com o programa *art.m* – nº de iterações = 11 . (v) Perfis das linhas 25 / 32 / 47 da figura 6.5.u . (x) Corte tomográfico reconstruído com o programa *art.m* – nº de iterações = 12 . (z) Perfis das linhas 25 / 32 / 47 da figura 6.5.x .

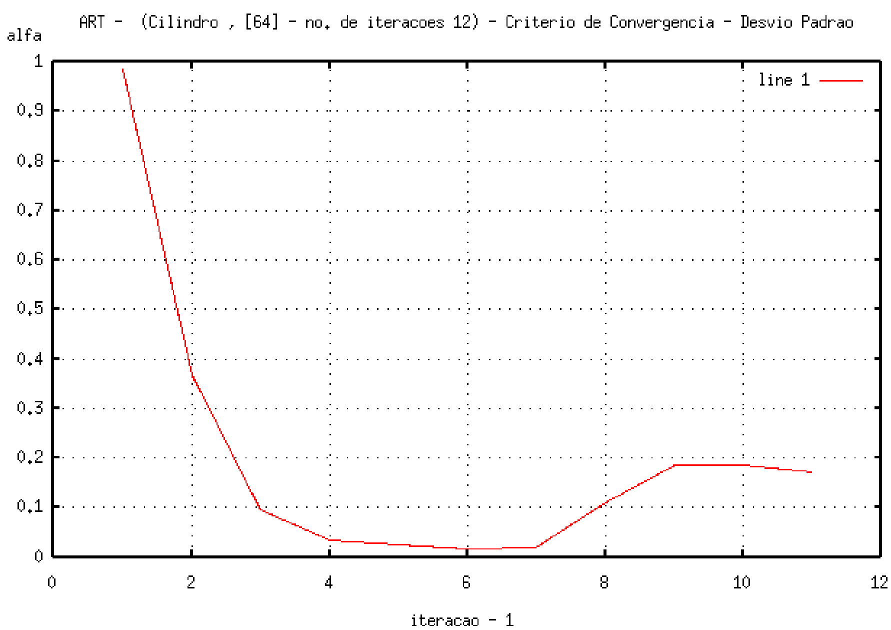
		x				y				
Imagem	**Iterações**	**Cmax**	**Cmin**	**Cmedia**	**Unif tomo**	**Cmax**	**Cmin**	**Cmedia**	**Unif tomo**	**Unif. Media**
6.5.a	1,00	45,00	12,00	26,64	123,87	65,00	12,00	27,99	189,35	156,61
6.5.c	2,00	113,00	32,00	66,01	122,71	113,00	24,00	58,48	152,19	137,45
6.5.e	3,00	142,00	45,00	84,41	114,92	150,00	32,00	79,98	147,54	131,23
6.5.g	4,00	125,00	32,00	71,70	129,71	134,00	36,00	73,14	133,99	131,85
6.5.i	5,00	125,00	24,00	65,72	153,68	130,00	24,00	68,01	155,86	154,77
6.5.k	6,00	130,00	20,00	65,59	167,71	138,00	20,00	66,78	176,70	172,20
6.5.m	7,00	146,00	16,00	70,19	185,21	158,00	16,00	69,74	203,61	194,41
6.5.o	8,00	162,00	12,00	73,36	204,47	174,00	12,00	72,68	222,89	213,68
6.5.q	9,00	178,00	8,00	74,80	227,27	186,00	8,00	74,18	239,96	233,61
6.5.s	10,00	182,00	4,00	71,66	248,40	186,00	4,00	71,11	255,94	252,17
6.5.u	11,00	182,00	4,00	69,01	257,93	186,00	4,00	68,25	266,67	262,30
6.5.x	12,00	186,00	0,00	66,94	277,86	186,00	0,00	66,16	281,14	279,50

Tabela 6.3 - Valores de Uniformidade tomográfica calculados com a equação 6.1 para as reconstruções com o algoritmo *art.m* (multiplicativo).

As imagens reconstruídas com o algoritmo ***art.m*** com fator de correção multiplicativo e os respectivos valores de Uniformidade tomográfica (Tabela 6.3) apresentam uma diferença significativa de performance em relação aos obtidos com correção aditiva. Apesar da Entropia da imagem apresentar convergência (vide parâmetro α – figura 6.6.a) o mesmo não ocorre com o Desvio padrão das projeções calculadas (figura 6.6.b), após a oitava iteração. Tal fato ocorre porque, quando a correção multiplicativa atribui zero a um *pixel*, este não mais se altera, o que impede o aumento significativo do valor de $C_{média}$ a cada iteração. Para efeito de comparação, o histograma da imagem 6.5.a (uma iteração) apresenta 24 pixels com o valor zero, a imagem 6.5.k (8 iterações) 32 e a 6.5.x (12 iterações) 55.



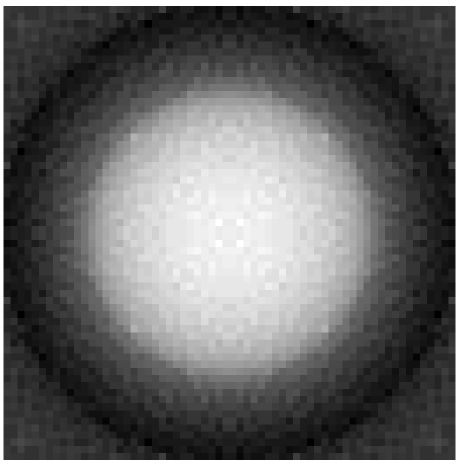
(a)



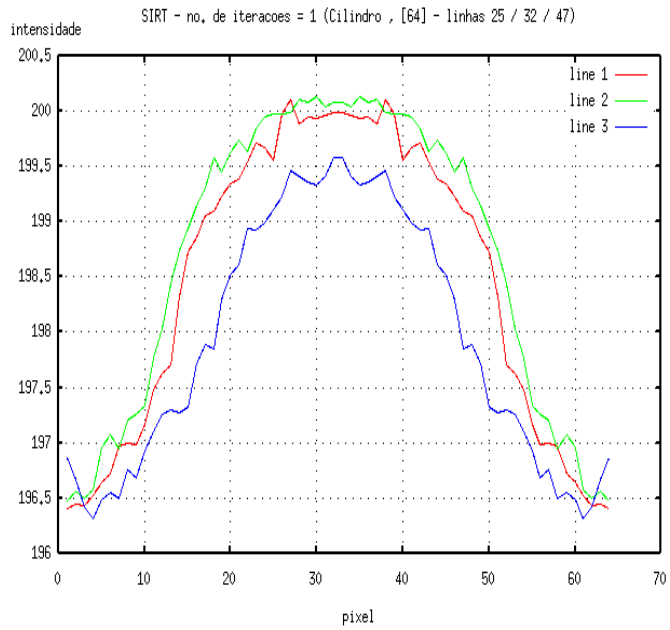
(b)

Figura 6.6 – (a) Parâmetro de controle α para a Entropia das imagens da Figura 6.5 . (b) Parâmetro de controle α para Desvio Padrão das projeções calculadas paras as imagens da Figura 6.5.

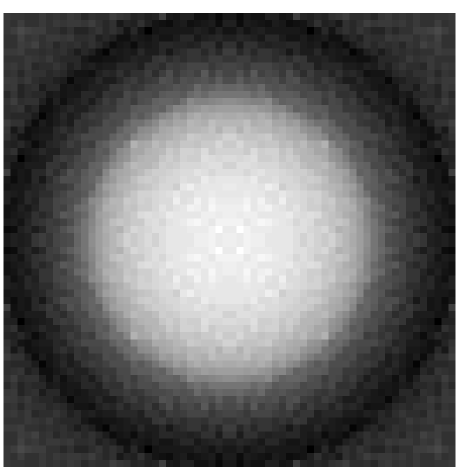
- ***Sirt (Aditivo)***



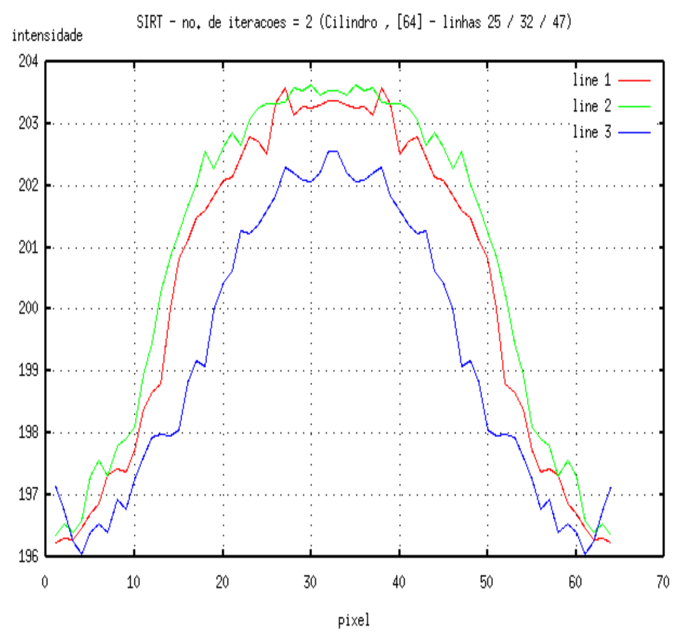
(a)



(b)

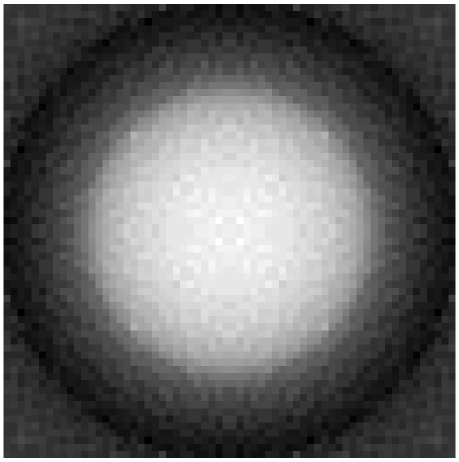


(c)

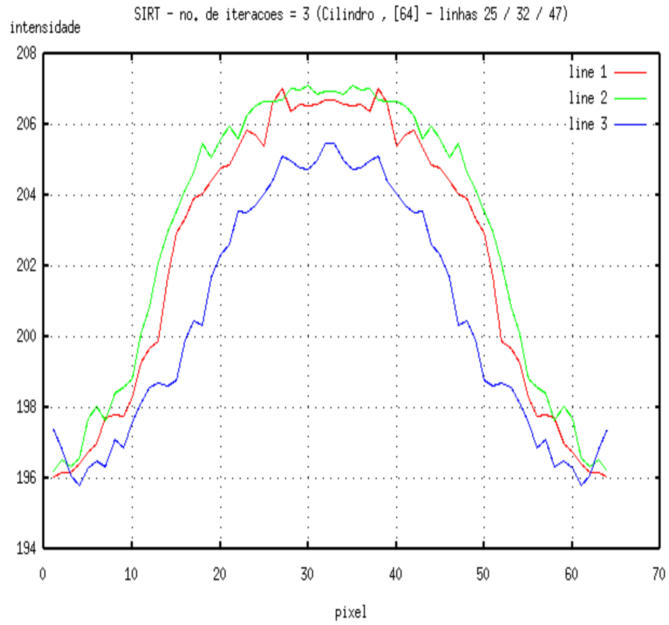


(d)

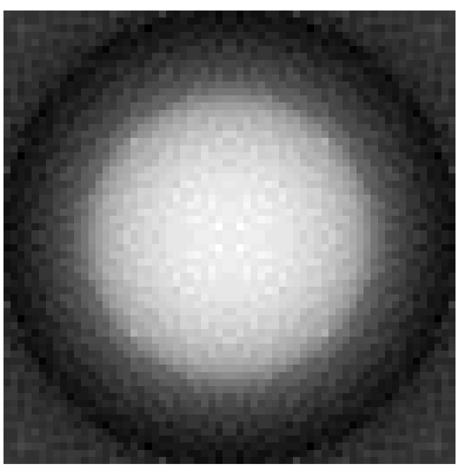
Figura 6.7 - (a) Corte tomográfico reconstruído com o programa *sirt.m* – nº de iterações = 1 . (b) Perfis das linhas 25 / 32 / 47 da figura 6.7.a . (c) Corte tomográfico reconstruído com o programa *sirt.m* – nº de iterações = 2 . (d) Perfis das linhas 25 / 32 / 47 da figura 6.7.c .



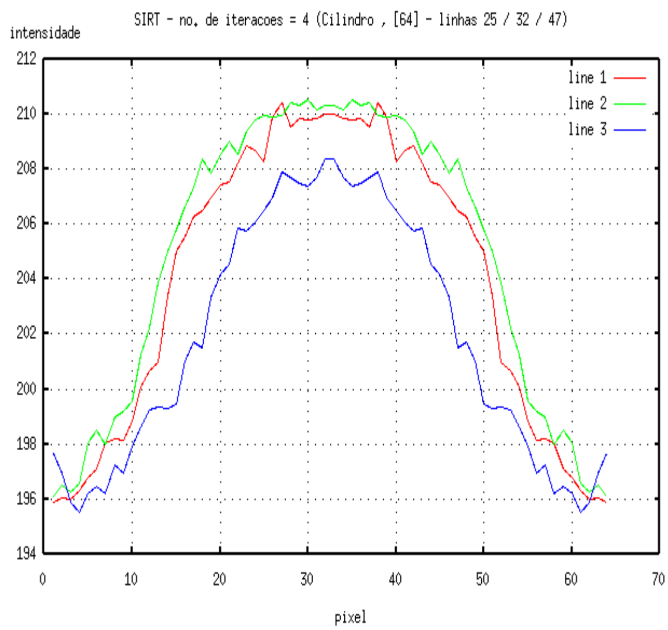
(e)



(f)

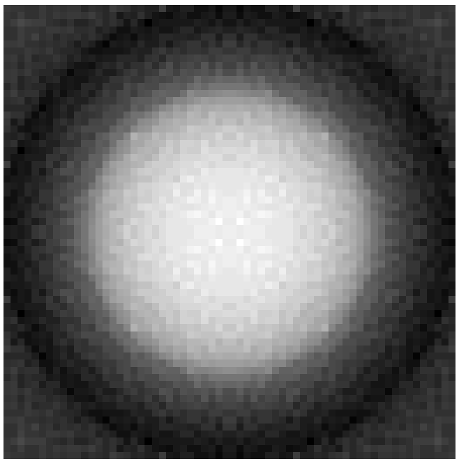


(g)

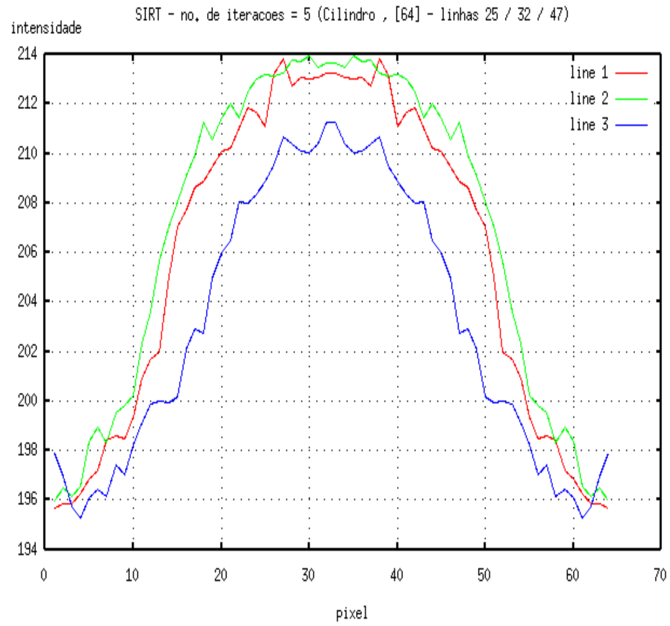


(h)

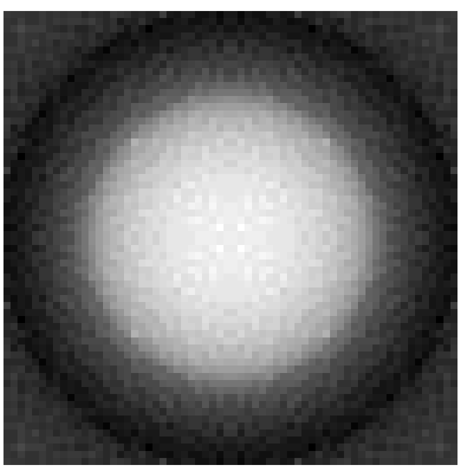
Figura 6.7 - (e) Corte tomográfico reconstruído com o programa *sirt.m* – nº de iterações = 3 . (f) Perfis das linhas 25 / 32 / 47 da figura 6.7.e . (g) Corte tomográfico reconstruído com o programa *sirt.m* – nº de iterações = 4 . (h) Perfis das linhas 25 / 32 / 47 da figura 6.7.g .



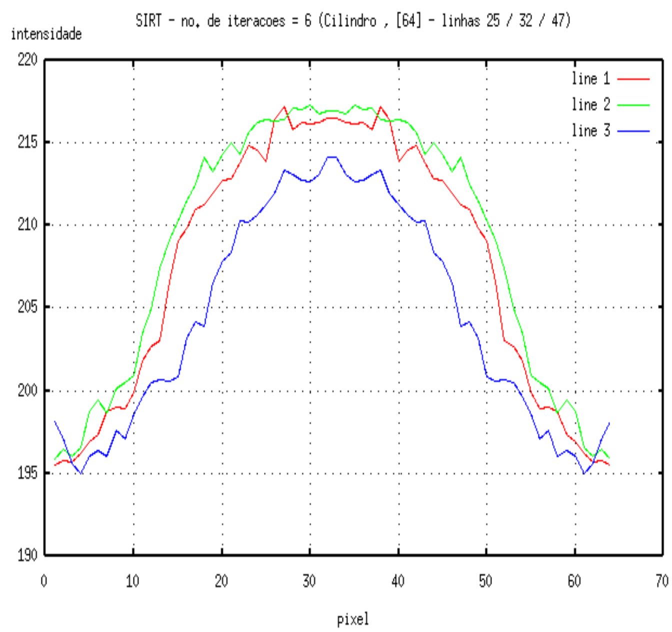
(i)



(j)

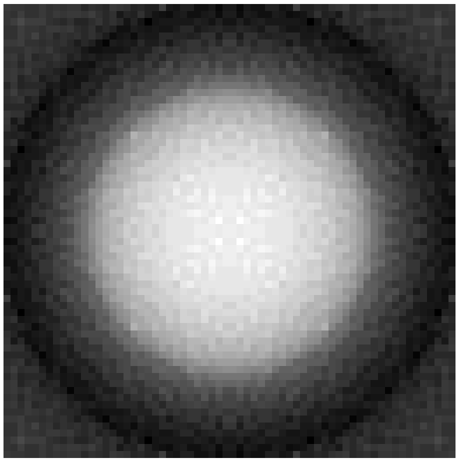


(k)

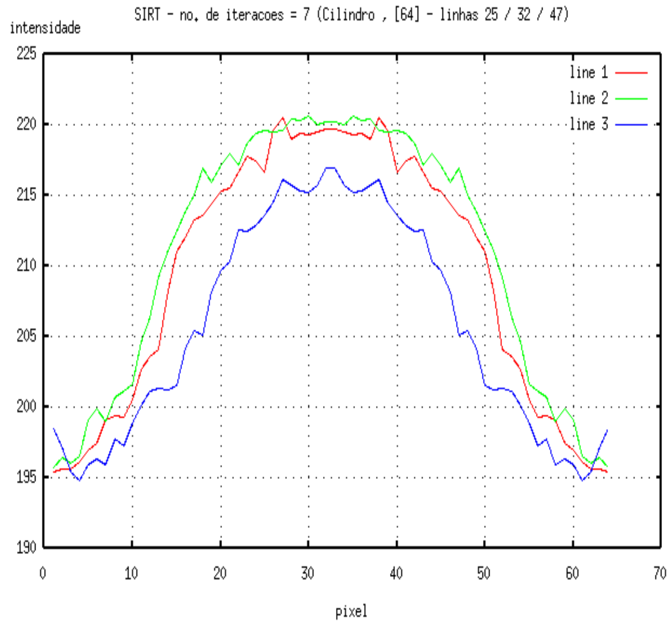


(l)

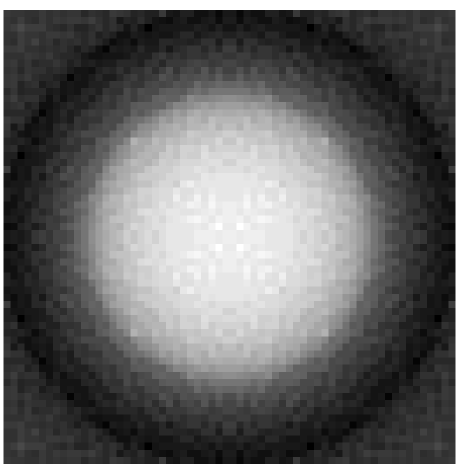
Figura 6.7 - (i) Corte tomográfico reconstruído com o programa *sirt.m* – nº de iterações = 5 . (j) Perfis das linhas 25 / 32 / 47 da figura 6.7.i . (k) Corte tomográfico reconstruído com o programa *sirt.m* – nº de iterações = 6 . (l) Perfis das linhas 25 / 32 / 47 da figura 6.7.k .



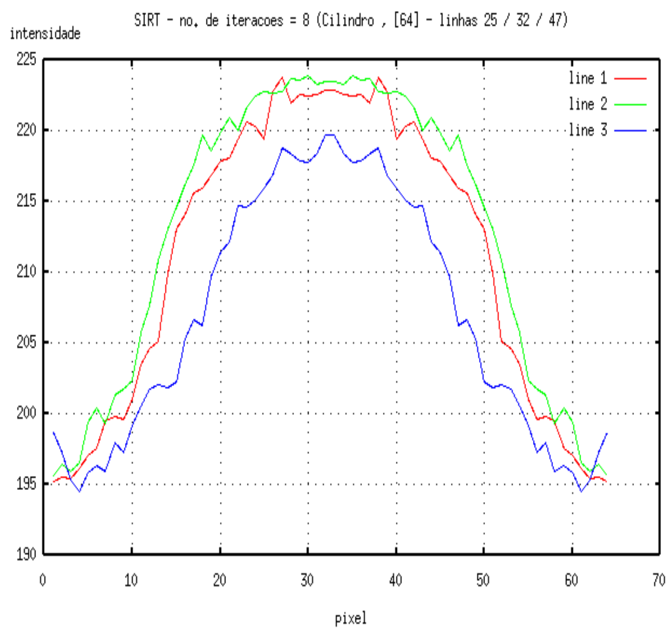
(m)



(n)

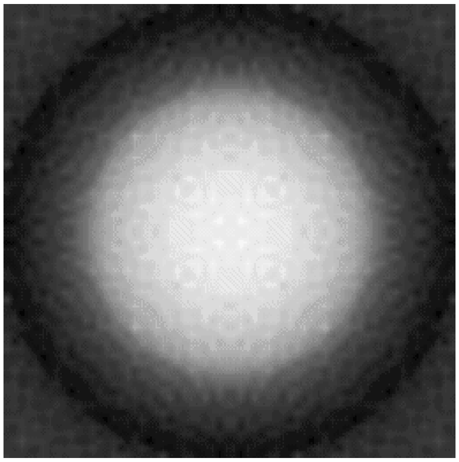


(o)

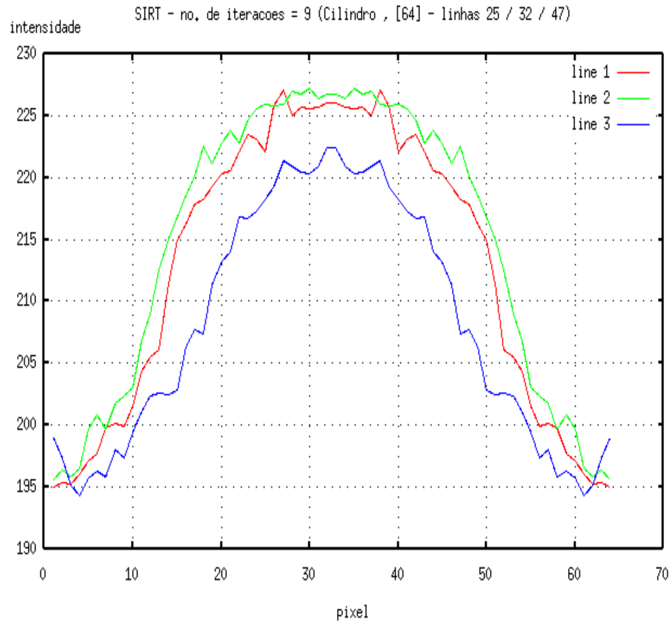


(p)

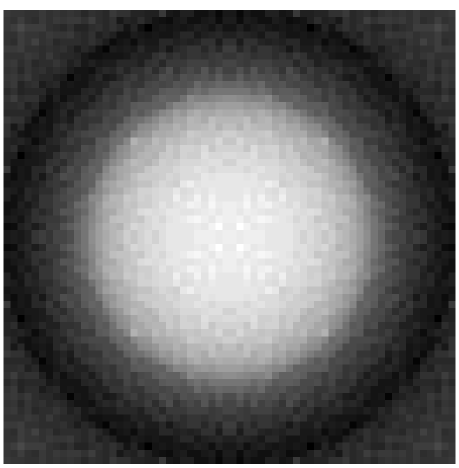
Figura 6.7 - (m) Corte tomográfico reconstruído com o programa *sirt.m* – nº de iterações = 7 . (n) Perfis das linhas 25 / 32 / 47 da figura 6.7.m . (o) Corte tomográfico reconstruído com o programa *sirt.m* – nº de iterações = 8 . (p) Perfis das linhas 25 / 32 / 47 da figura 6.7.o .



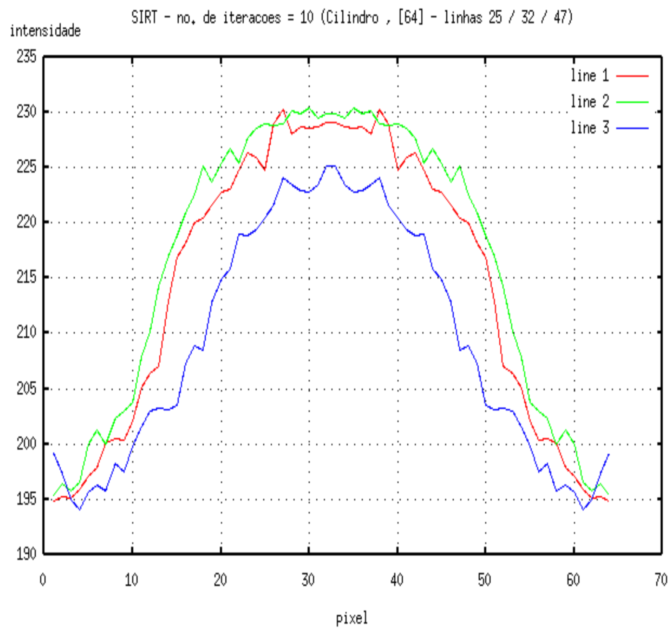
(q)



(r)

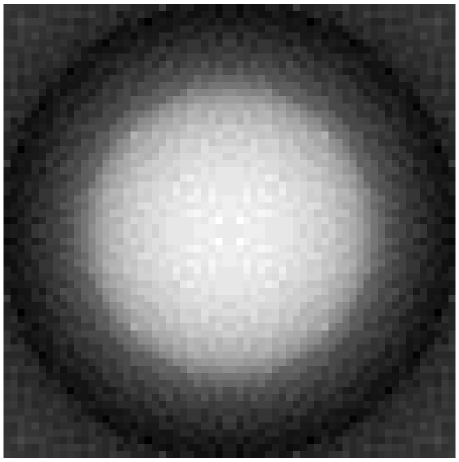


(s)

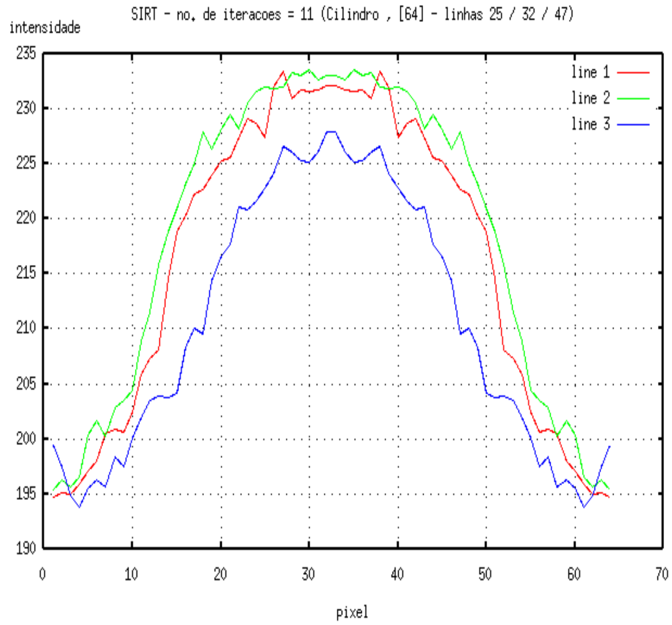


(t)

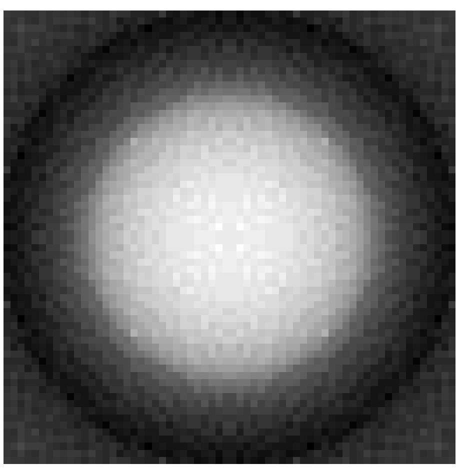
Figura 6.7 - (q) Corte tomográfico reconstruído com o programa *sirt.m* – nº de iterações = 9 . (r) Perfis das linhas 25 / 32 / 47 da figura 6.7.q . (s) Corte tomográfico reconstruído com o programa *sirt.m* – nº de iterações = 10 . (t) Perfis das linhas 25 / 32 / 47 da figura 6.7.s .



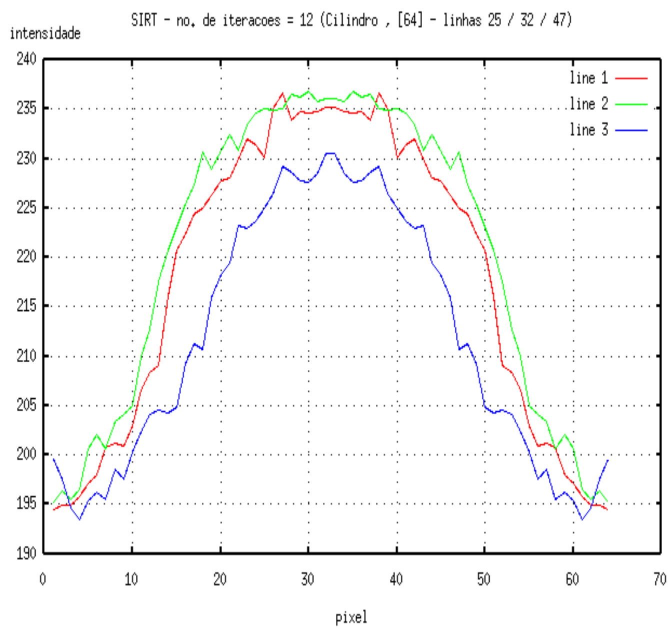
(u)



(v)



(x)



(z)

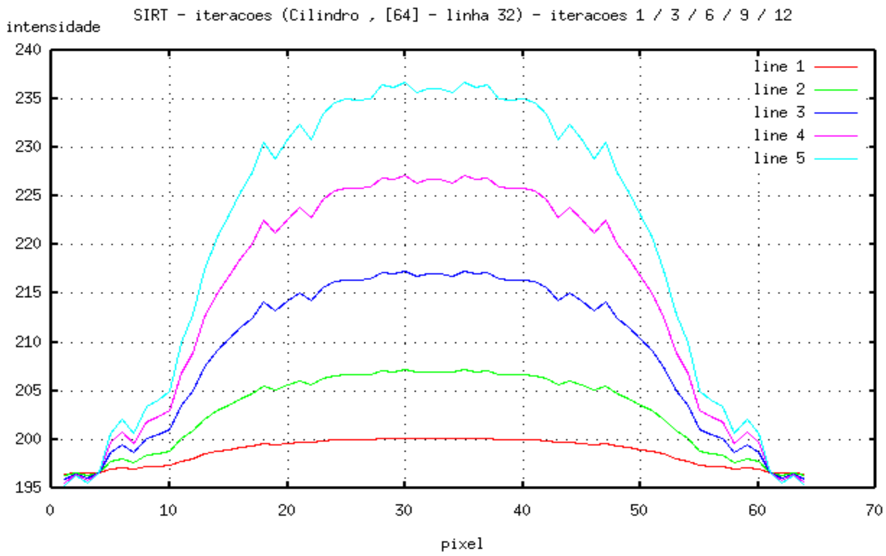
Figura 6.7 - (u) Corte tomográfico reconstruído com o programa *sirt.m* – nº de iterações = 11 . (v) Perfis das linhas 25 / 32 / 47 da figura 6.7.u . (x) Corte tomográfico reconstruído com o programa *sirt.m* – nº de iterações = 12 . (z) Perfis das linhas 25 / 32 / 47 da figura 6.7.x .

		x				y				
Imagem	***Iterações***	***Cmax***	***Cmin***	***Cmedia***	***Unif tomo***	***Cmax***	***Cmin***	***Cmedia***	***Unif tomo***	***Unif. Media***
6.7.a	1,00	255,00	138,00	212,30	55,11	255,00	138,00	212,30	55,11	55,11
6.7.c	2,00	255,00	138,00	212,10	55,16	255,00	138,00	212,25	55,12	55,14
6.7.e	3,00	255,00	138,00	212,10	55,16	255,00	138,00	212,15	55,15	55,16
6.7.g	4,00	255,00	138,00	211,90	55,21	255,00	138,00	211,85	55,23	55,22
6.7.i	5,00	255,00	138,00	211,85	55,23	255,00	138,00	211,70	55,27	55,25
6.7.k	6,00	255,00	138,00	211,65	55,28	255,00	138,00	211,70	55,27	55,27
6.7.m	7,00	255,00	138,00	211,50	55,32	255,00	138,00	211,50	55,32	55,32
6.7.o	8,00	255,00	138,00	211,50	55,32	255,00	138,00	211,50	55,32	55,32
6.7.q	9,00	255,00	138,00	211,50	55,32	255,00	138,00	211,50	55,32	55,32
6.7.s	10,00	255,00	138,00	211,50	55,32	255,00	138,00	211,50	55,32	55,32
6.7.u	11,00	255,00	134,00	211,20	57,29	255,00	134,00	211,05	57,33	57,31
6.7.x	12,00	255,00	134,00	211,00	57,35	255,00	134,00	211,00	57,35	57,35

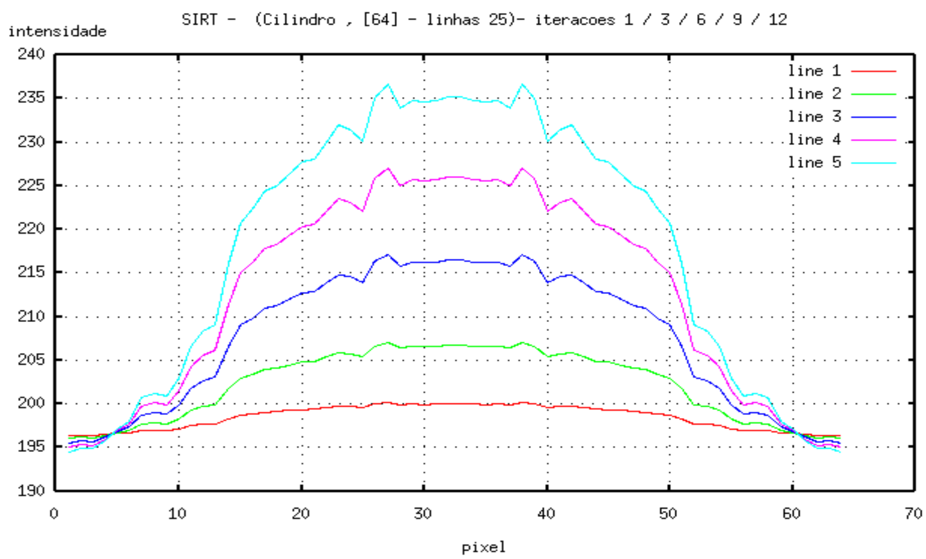
Tabela 6.4 - Valores de Uniformidade tomográfica calculados com a equação 6.1 para as reconstruções com o algoritmo *sirt.m* (aditivo) .

Os perfis e os valores de Uniformidade tomográfica para o programa ***sirt.m*** com correção aditiva demonstram a lentidão na convergência deste algoritmo. Para efeito de comparação, observa-se nos perfis das linhas 25 e 32 das iterações 1 / 3 / 6 / 9 / 12 (figura 6.8) que a proporção entre os *pixels* destes não se alteram. Este fato indica que o parâmetro de controle da Entropia converge , porém o do Desvio Padrão das projeções calculadas mantém-se inalterado[3] (figura 6.9), levando à lentidão da convergência.

[3] Diferentemente do algoritmo ART, onde o Desvio Padrão (SD) da projeção calculada (p_c) é obtido para o maior valor dentre os p_c do ângulo de visada , para o algoritmo SIRT, o valor de SD é calculado para cada pixel , tomando –se o maior valor de todos a cada iteração (4096 *pixels*).

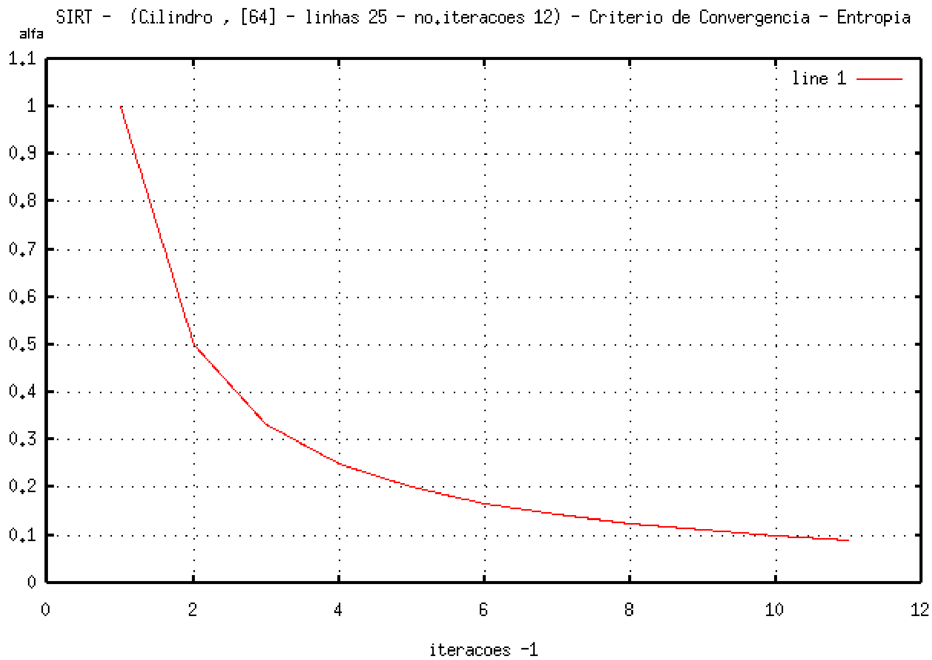


(a)

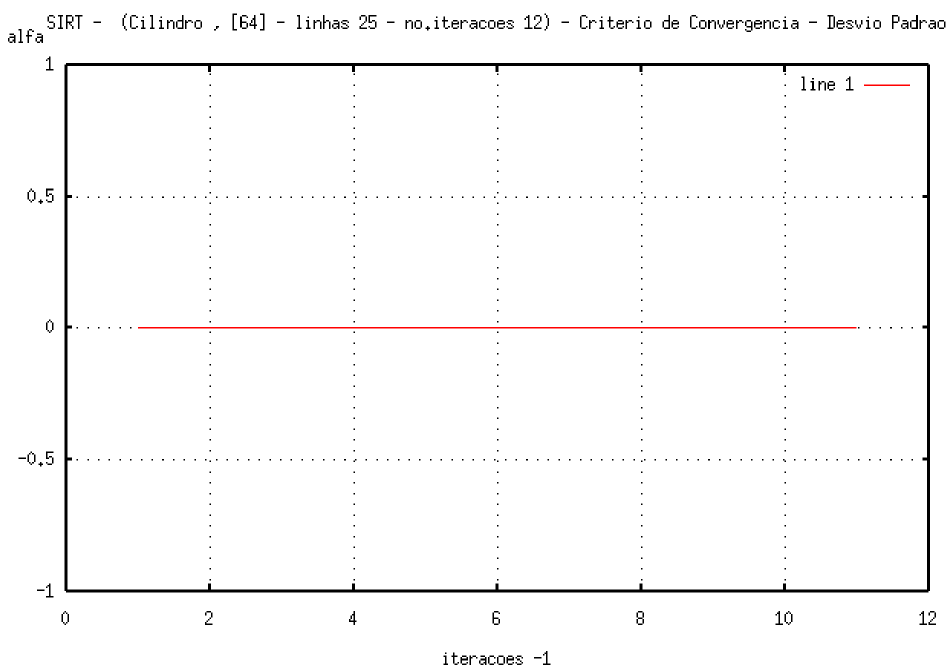


(b)

Figura 6.8 – (a) Perfis das linhas 32 para as iterações 1 / 3 / 6 / 9 / 12 para o programa *sirt.m* (aditivo) . (b) Perfis das linhas 25 para as iterações 1 / 3 / 6 / 9 / 12 para o programa *sirt.m* (aditivo).



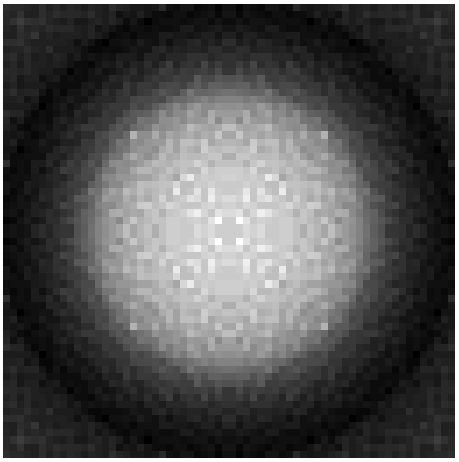
(a)



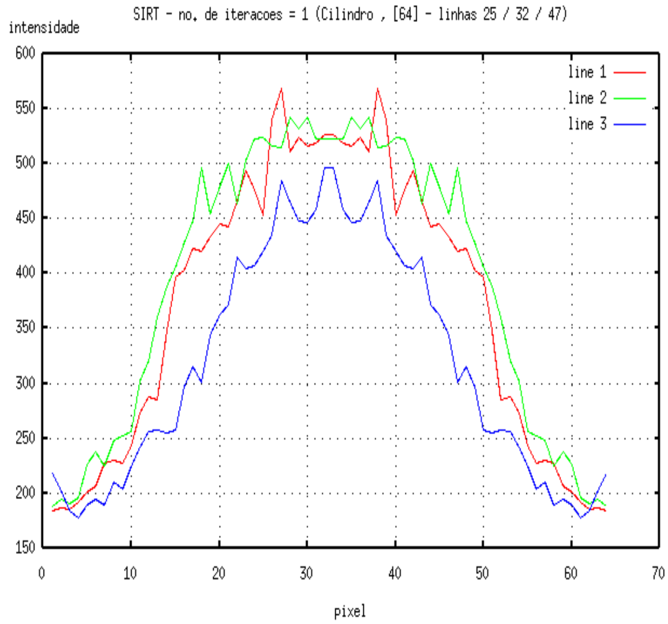
(b)

Figura 6.9 - (a) Parâmetro de controle α para a Entropia das imagens da Figura 6.7 . (b) Parâmetro de controle α para Desvio Padrão das projeções calculadas paras as imagens da Figura 6.7.

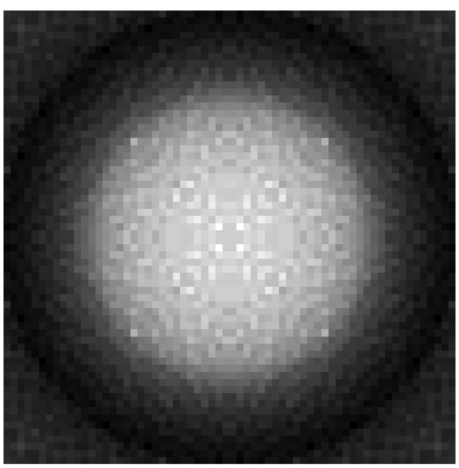
- ***Sirt (multiplicativo)***



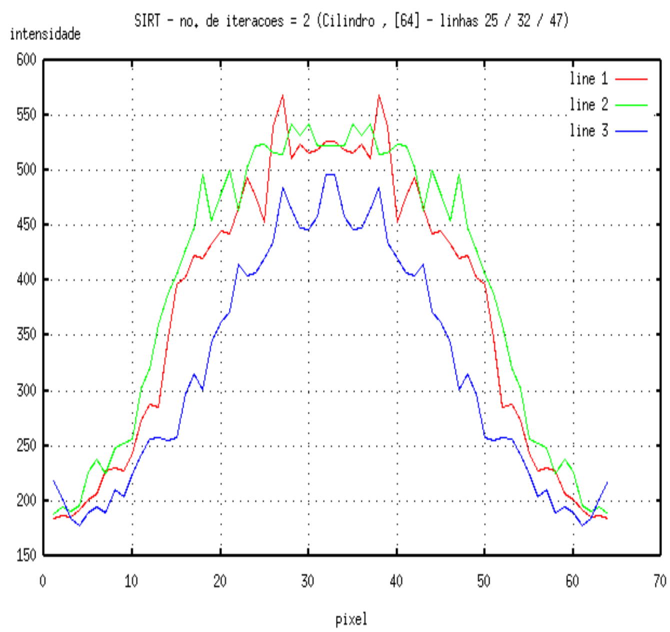
(a)



(b)

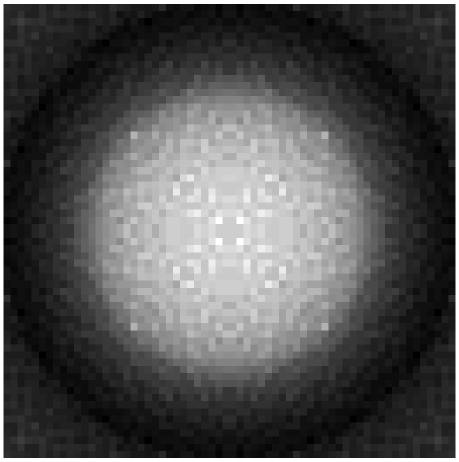


(c)

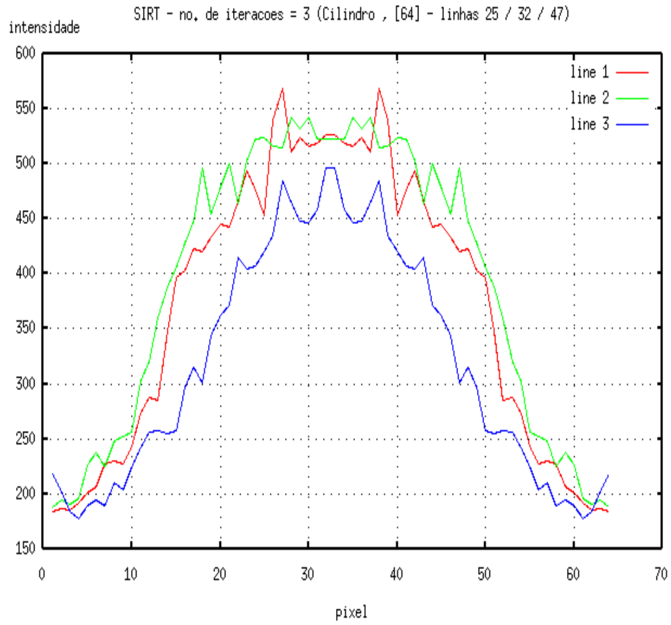


(d)

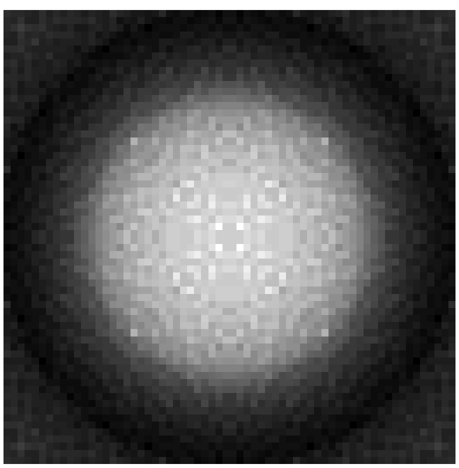
Figura 6.10 - (a) Corte tomográfico reconstruído com o programa *sirt.m* – nº de iterações = 1 . (b) Perfis das linhas 25 / 32 / 47 da figura 6.10.a . (c) Corte tomográfico reconstruído com o programa *sirt.m* – nº de iterações = 2 . (d) Perfis das linhas 25 / 32 / 47 da figura 6.10.c .



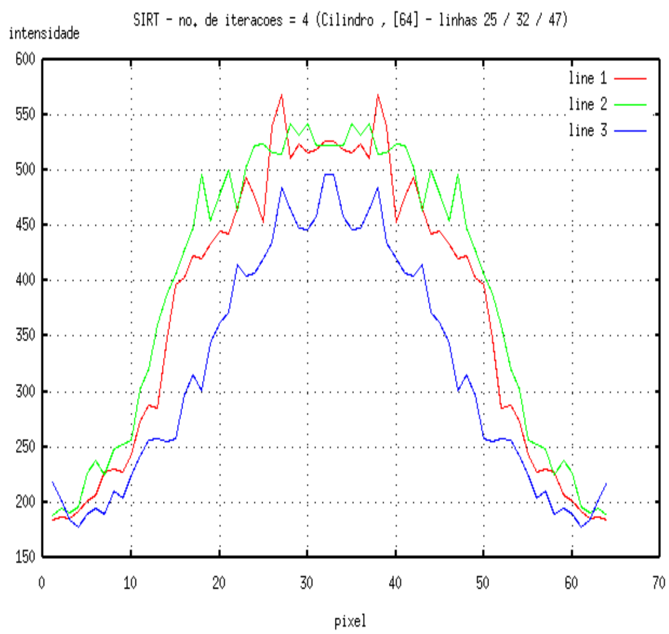
(e)



(f)

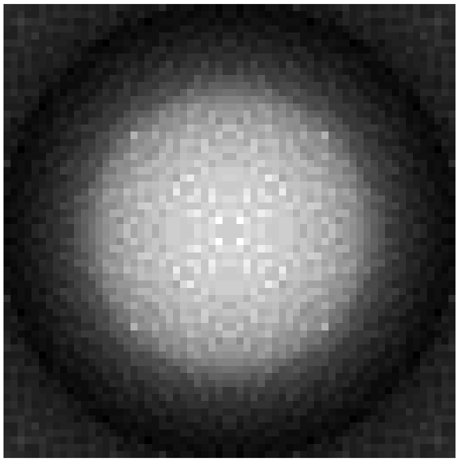


(g)

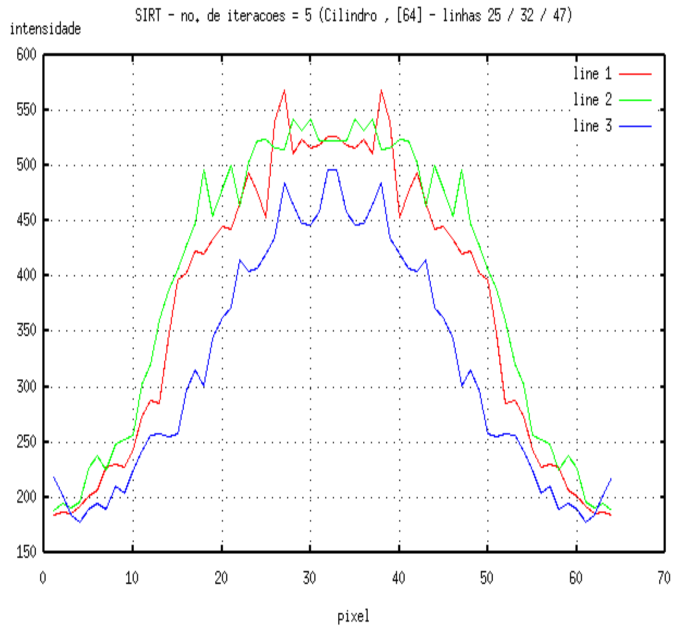


(h)

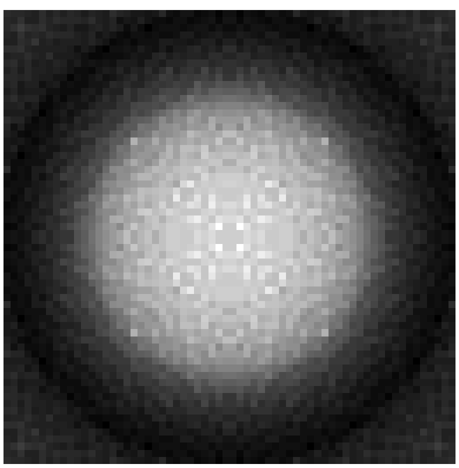
Figura 6.10 - (e) Corte tomográfico reconstruído com o programa *sirt.m* – nº de iterações = 3 . (f) Perfis das linhas 25 / 32 / 47 da figura 6.10.e . (g) Corte tomográfico reconstruído com o programa *sirt.m* – nº de iterações = 4 . (h) Perfis das linhas 25 / 32 / 47 da figura 6.10.g .



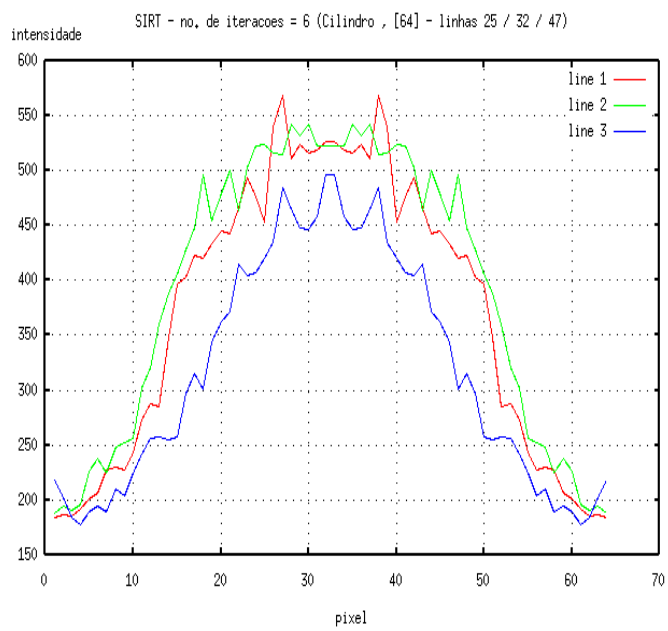
(i)



(j)

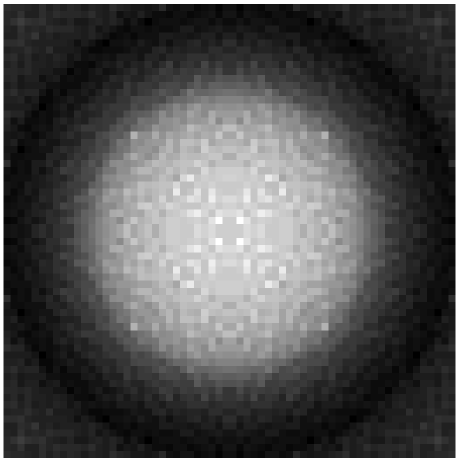


(k)

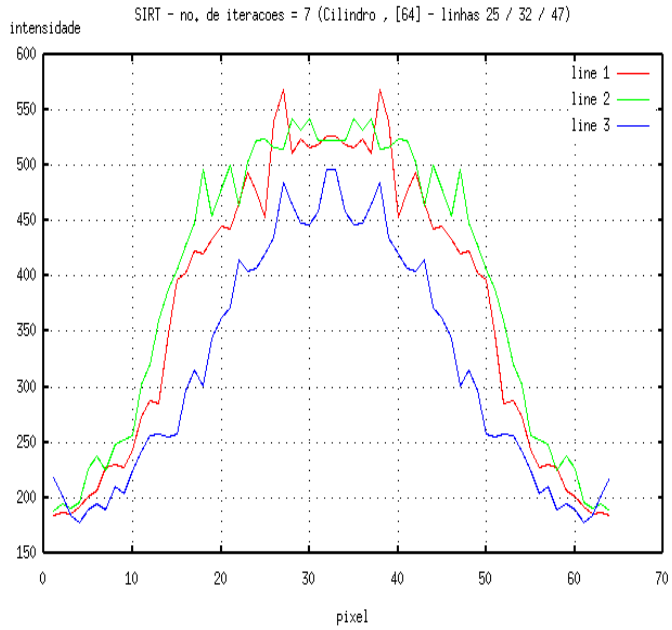


(l)

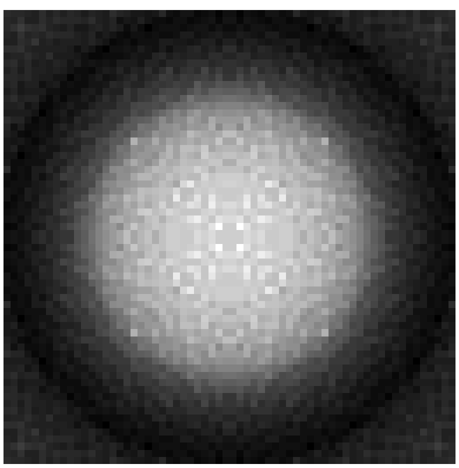
Figura 6.10 - (i) Corte tomográfico reconstruído com o programa *sirt.m* – nº de iterações = 5 . (j) Perfis das linhas 25 / 32 / 47 da figura 6.10.i . (k) Corte tomográfico reconstruído com o programa *sirt.m* – nº de iterações = 6 . (l) Perfis das linhas 25 / 32 / 47 da figura 6.10.k .



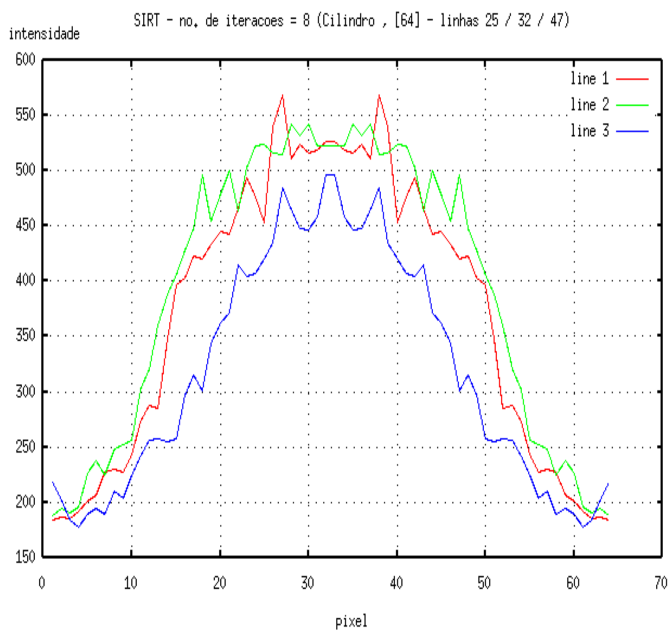
(m)



(n)

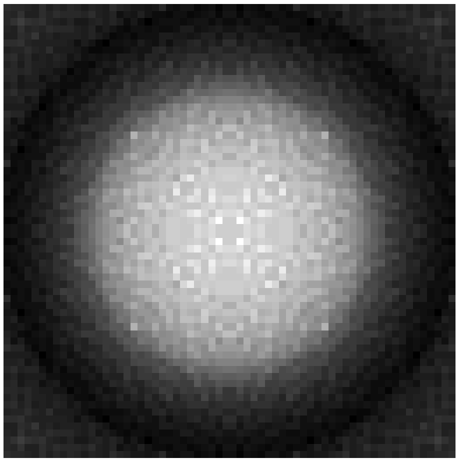


(o)

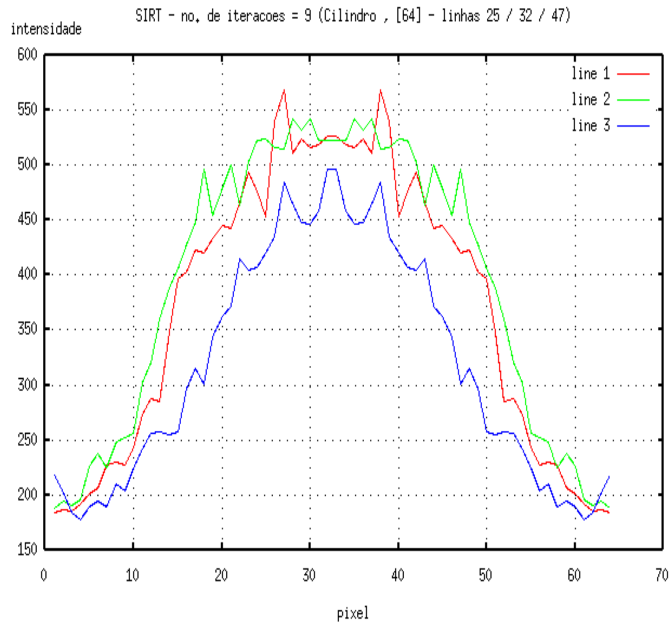


(p)

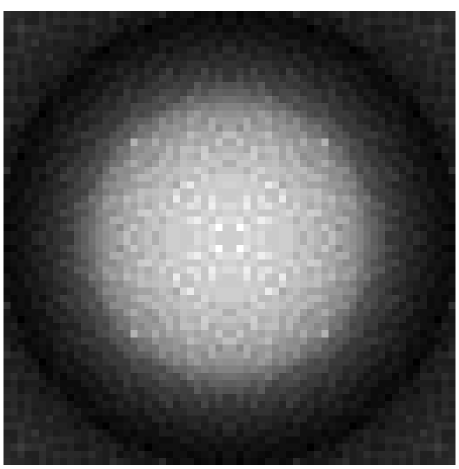
Figura 6.10 - (m) Corte tomográfico reconstruído com o programa *sirt.m* – nº de iterações = 7 . (n) Perfis das linhas 25 / 32 / 47 da figura 6.10.m . (o) Corte tomográfico reconstruído com o programa *sirt.m* – nº de iterações = 8 . (p) Perfis das linhas 25 / 32 / 47 da figura 6.10.o .



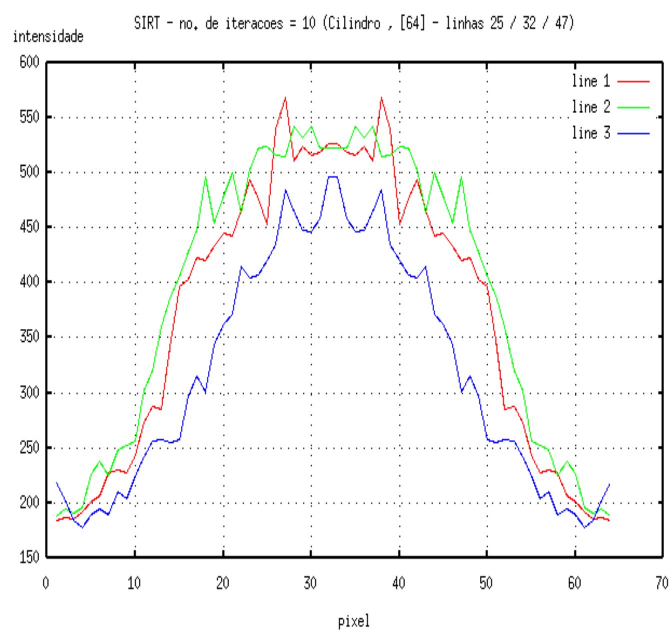
(q)



(r)

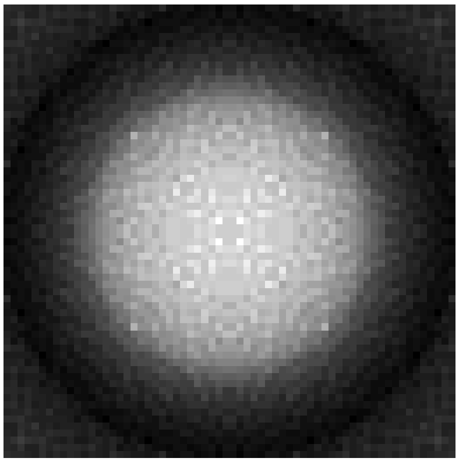


(s)

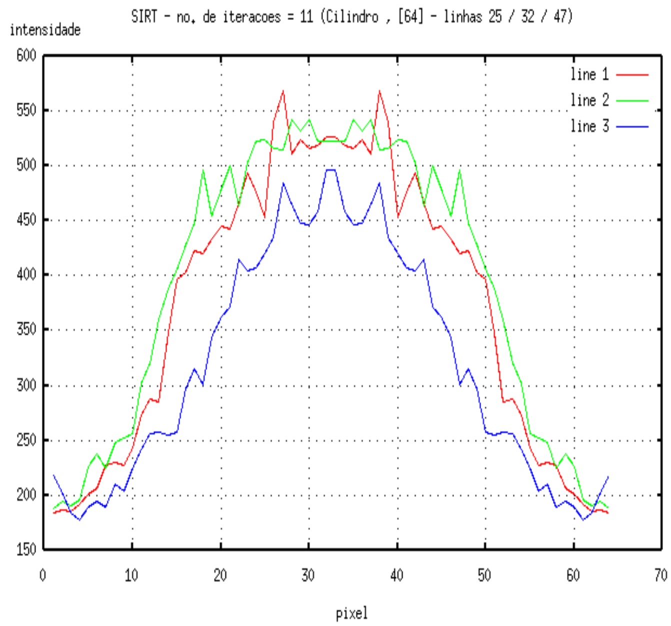


(t)

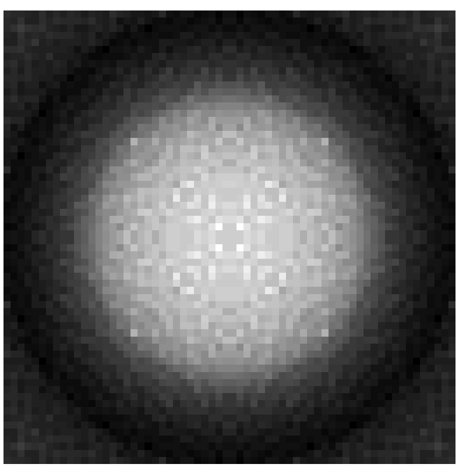
Figura 6.10 - (q) Corte tomográfico reconstruído com o programa *sirt.m* – nº de iterações = 9 . (r) Perfis das linhas 25 / 32 / 47 da figura 6.10.q . (s) Corte tomográfico reconstruído com o programa *sirt.m* – nº de iterações = 10 . (t) Perfis das linhas 25 / 32 / 47 da figura 6.10.s .



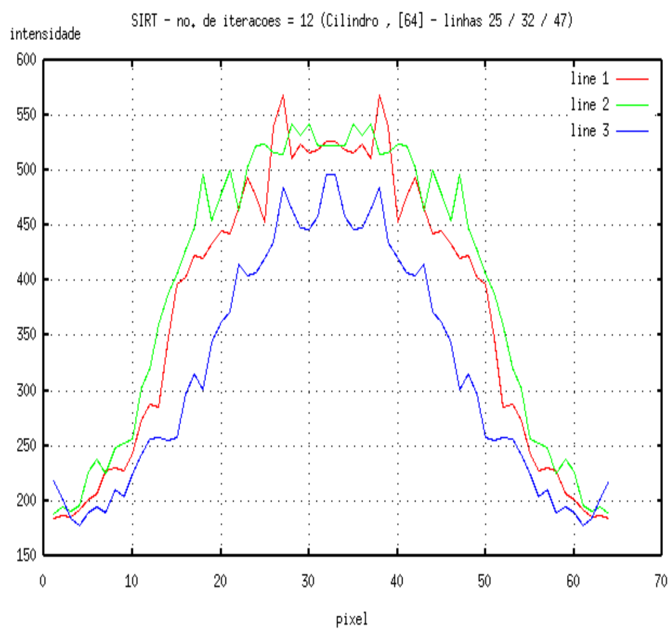
(u)



(v)



(x)



(z)

Figura 6.10 - (u) Corte tomográfico reconstruído com o programa *sirt.m* – nº de iterações = 11 . (v) Perfis das linhas 25 / 32 / 47 da figura 6.10.u . (x) Corte tomográfico reconstruído com o programa *sirt.m* – nº de iterações = 12 . (z) Perfis das linhas 25 / 32 / 47 da figura 6.10.x .

		x				y				
Imagem	**Iterações**	**Cmax**	**Cmin**	**Cmedia**	**Unif tomo**	**Cmax**	**Cmin**	**Cmedia**	**Unif tomo**	**Unif. Media**
6.10.a	1,00	255,00	109,00	185,05	78,90	255,00	109,00	185,00	78,92	78,91
6.10.c	2,00	255,00	109,00	185,05	78,90	255,00	109,00	185,00	78,92	78,91
6.10.e	3,00	255,00	109,00	185,05	78,90	255,00	109,00	185,00	78,92	78,91
6.10.g	4,00	255,00	109,00	185,05	78,90	255,00	109,00	185,00	78,92	78,91
6.10.i	5,00	255,00	109,00	185,05	78,90	255,00	109,00	185,00	78,92	78,91
6.10.k	6,00	255,00	109,00	185,05	78,90	255,00	109,00	185,00	78,92	78,91
6.10.m	7,00	255,00	109,00	185,05	78,90	255,00	109,00	185,00	78,92	78,91
6.10.o	8,00	255,00	109,00	185,05	78,90	255,00	109,00	185,00	78,92	78,91
6.10.q	9,00	255,00	109,00	185,05	78,90	255,00	109,00	185,00	78,92	78,91
6.10.s	10,00	255,00	109,00	185,05	78,90	255,00	109,00	185,00	78,92	78,91
6.10.u	11,00	255,00	109,00	185,05	78,90	255,00	109,00	185,00	78,92	78,91
6.10.x	12,00	255,00	109,00	185,05	78,90	255,00	109,00	185,00	78,92	78,91

Tabela 6.5 - Valores de Uniformidade tomográfica calculados com a equação 6.1 para as reconstruções com o algoritmo *sirt.m* (multiplicativo) .

Os valores da Tabela 6.5 e as imagens / perfis da figura 6.10 demonstram que os resultados para a Uniformidade tomográfica obtidos para o programa ***sirt.m*** com correção multiplicativa estão muito aquém dos obtidos com a correção aditiva. Este fato se deve a rapidez com que o algoritmo converge nas projeções calculadas (parâmetro α - figura 6.11) do corte em estudo, não possibilitando alterações de iteração para iteração.

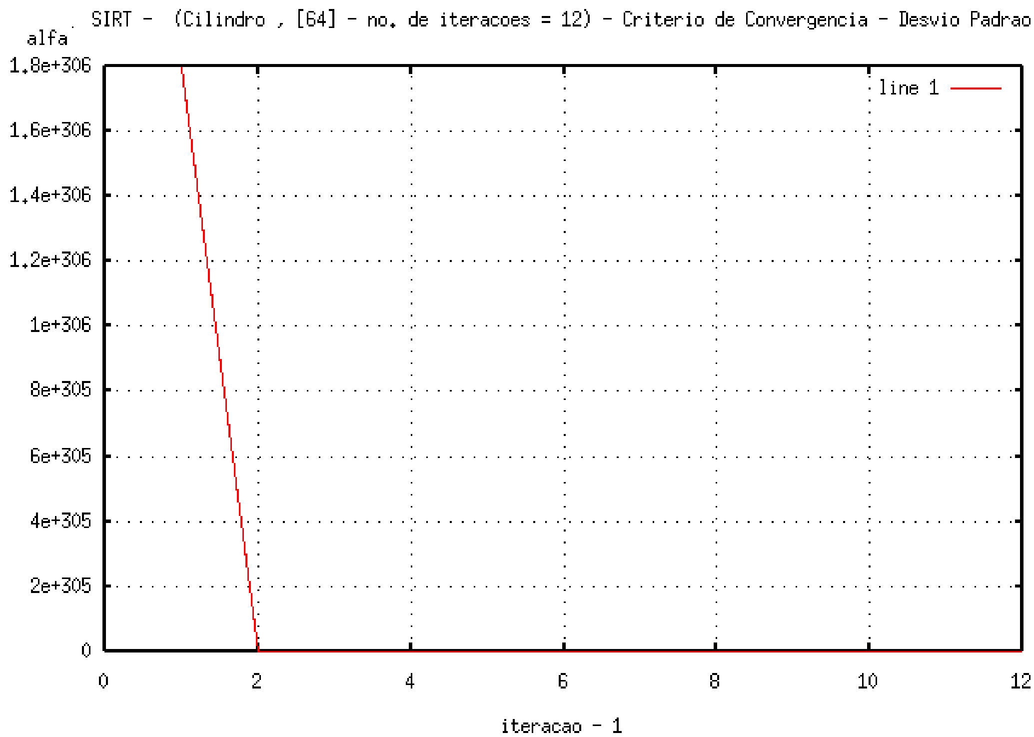
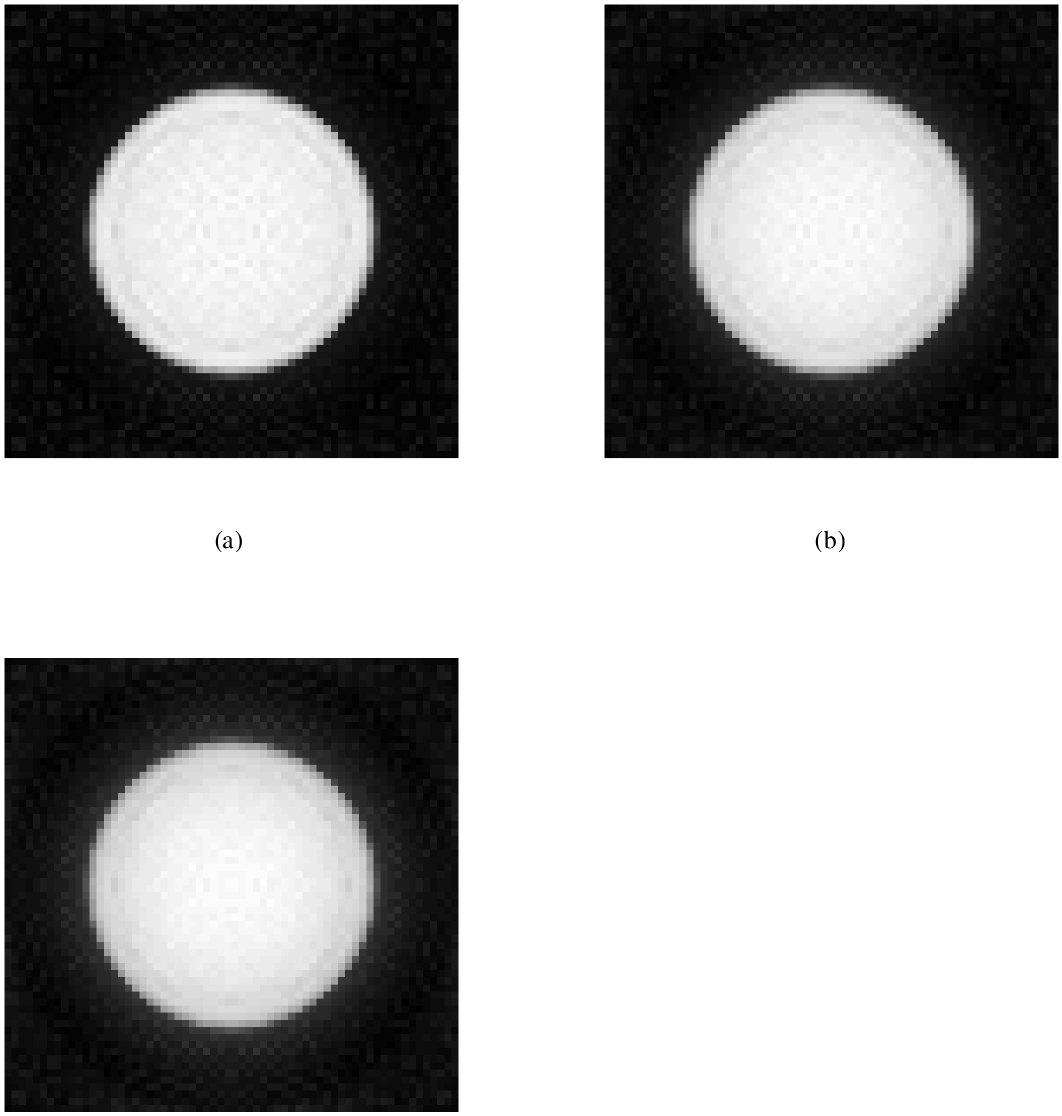


Figura 6.11 - Parâmetro de controle α para Desvio Padrão das projeções calculadas paras as imagens da Figura 6.10.

- ***Art (aditivo) + Sirt (aditivo - 3 iterações)***

Alguns textos sugerem a técnica híbrida [42]. Realizam-se algumas iterações com o algoritmo ART, com convergência mais rápida, porém mais susceptível a ruídos e em seguida algumas iterações com o algoritmo SIRT, de convergência mais lenta, porém menos susceptível a ruído. Apesar destes testes ainda utilizarem projeções sem ruído, foram executadas 3 iterações do algoritmo SIRT com correção aditiva a partir da imagem 6.1.x. Os resultados obtidos encontram-se na figura 6.12 e Tabela 6.6.



(a) (b)

(c)

Figura 6.12 – (a) Corte tomográfico reconstruído com o programa *sirt.m* – nº de iterações = 1 a partir da imagem da figura 6.1.m. (b) Corte tomográfico reconstruído com o programa *sirt.m* – nº de iterações = 2 a partir da imagem da figura 6.1.m . (c) Corte tomográfico reconstruído com o programa *sirt.m* – nº de iterações= 3 a partir da imagem da figura 6.1.m.

		x				y				
Imagem	**Iterações**	**Cmax**	**Cmin**	**Cmedia**	**Unif tomo**	**Cmax**	**Cmin**	**Cmedia**	**Unif tomo**	**Unif. Media**
6.12.a	1,00	251,00	174,00	235,90	32,64	251,00	174,00	236,90	32,50	32,57
6.12.b	2,00	255,00	170,00	234,53	36,24	251,00	170,00	235,00	34,47	35,36
6.12.c	3,00	255,00	162,00	233,10	39,90	255,00	166,00	233,53	38,11	39,00

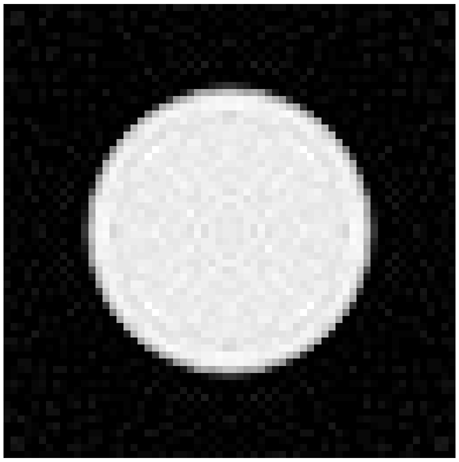
Tabela 6.6 - Valores de Uniformidade tomográfica calculados com a equação 6.1 para as reconstruções com o algoritmo *sirt.m* (aditivo) a partir da imagem 6.2.x .

A tabela 6.6 mostra valores de Uniformidade tomográfica inferiores aquele alcançado pela imagem 6.1.x porém, este teste foi executado sem a presença de ruído nas projeções, sendo apenas indicativo.

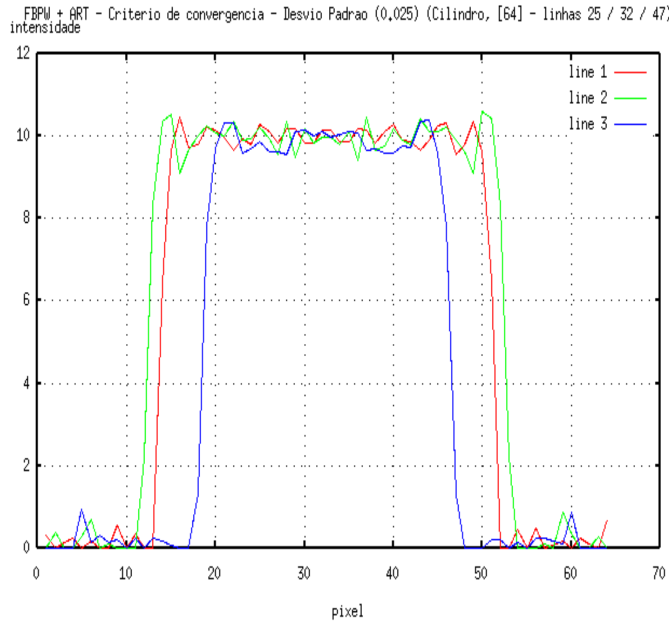
- ***FBPW + Art (aditivo)***

Os equipamentos recentemente fabricados para estudos em Medicina Nuclear utilizam este algoritmo híbrido, onde a imagem inicial, ao invés de dada pela equação 4.34 é obtida por uma reconstrução com o algoritmo FBP conjuntamente com uma função janela.

Para a obtenção da imagem neste teste, utilizou-se o critério de convergência de Desvio Padrão (equação 4.36) com α igual 0.025, resultando num total de 18 iterações. A imagem inicial utilizada é a da figura 6.1.c , ou seja, reconstruída sem a utilização da função janela, apenas com o filtro rampa. Isto porque ainda tratam-se de imagens sem ruído. Os resultados estão apresentados na figura 6.13 e Tabela 6.7.



(a)



(b)

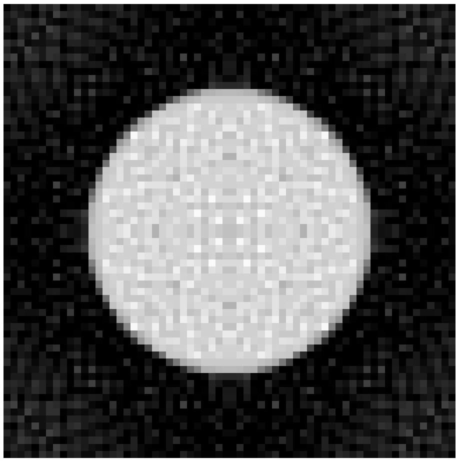
Figura 6.13 - (a) Corte tomográfico reconstruído com o programa *fbpw.m* + *art.m* . (b) Perfis das linhas 25 / 32 / 47 da figura 6.13.a .

		x				y				
Imagem	**Iterações**	**Cmax**	**Cmin**	**Cmedia**	**Unif tomo**	**Cmax**	**Cmin**	**Cmedia**	**Unif tomo**	**Unif. Media**
6.13.a	18,00	251,00	178,00	232,85	31,35	251,00	178,00	232,95	31,34	31,34

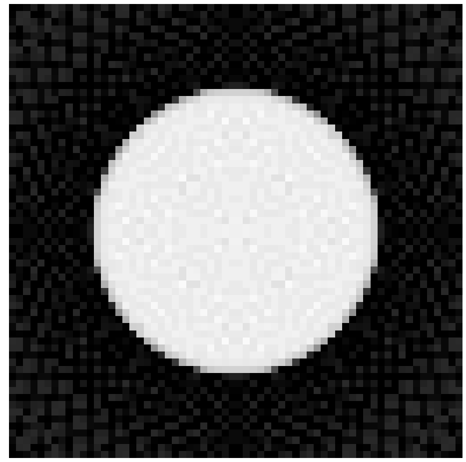
Tabela 6.7 -Valores de Uniformidade tomográfica calculados com a equação 6.1 para as reconstruções com o algoritmo *fbpw.m* + *art.m* (aditivo) a partir da imagem 6.1.c .

- ***Resumo e Análise dos resultados***

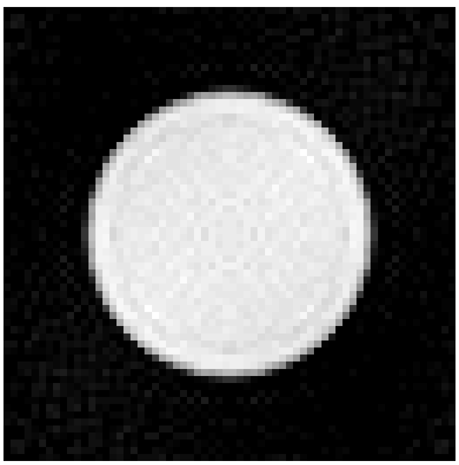
Para avaliação reúnem-se as imagens de todos os algoritmos testados e, no caso dos que possuem mais de uma, apresenta-se a de melhor resultado (figura 6.14). A Tabela 6.8 trás os valores de Uniformidade tomográfica obtidos pelas imagens da figura 6.14 e o gráfico da figura 6.15 representa estes valores.



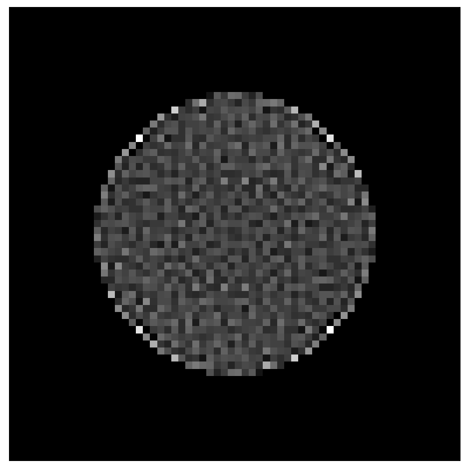
(a)



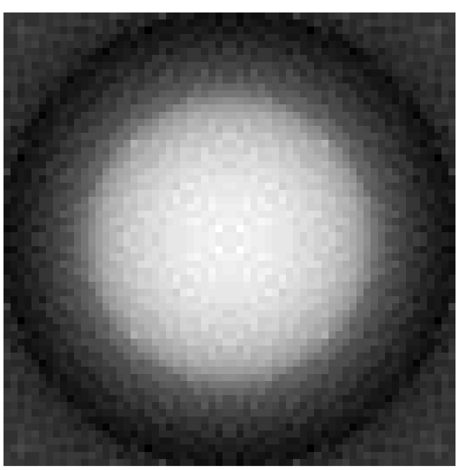
(b)



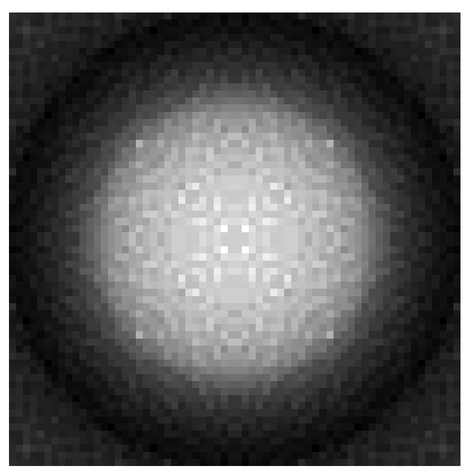
(c)



(d)

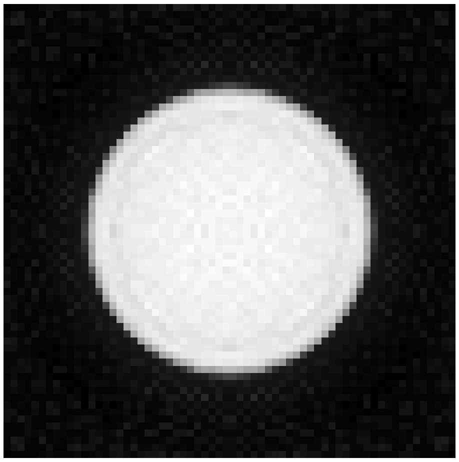


(e)

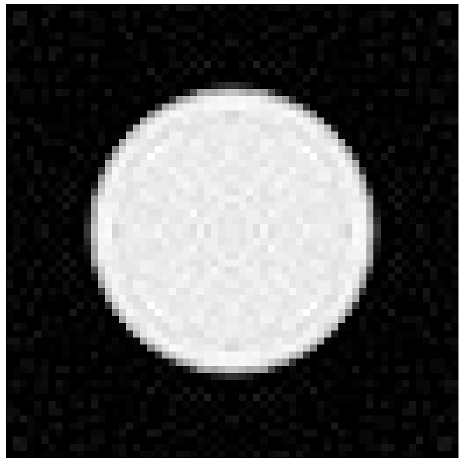


(f)

Legenda na pg. 152



(g)



(h)

Figura 6.14 – (a) Corte reconstruído com o algoritmo *fbp.m* (figura 6.1.a) . (b) Corte reconstruído com o algoritmo *fbpw.m* (figura 6.1.c) . (c) Corte reconstruído com o algoritmo *art.m* - aditivo (figura 6.2.i) . (d) Corte reconstruído com o algoritmo *art.m* - multiplicativo (figura 6.5.e) . (e) Corte reconstruído com o algoritmo *sirt.m* - aditivo (figura 6.7.a) . (f) Corte reconstruído com o algoritmo *sirt.m* - multiplicativo (figura 6.10.a) . (g) Corte reconstruído com o algoritmo *art.m* + *sirt.m* (figura 6.12.a) . (h) Corte reconstruído com o algoritmo *fbpw.m* + *art.m* - aditivo (figura 6.13.a) .

	x				y				
Imagem	*Cmax*	*Cmin*	*Cmedia*	*Unif tomo*	*Cmax*	*Cmin*	*Cmedia*	*Unif tomo*	*Unif. Media*
6.1.a	248,00	178,00	209,80	33,37	248,00	178,00	209,05	33,48	33,42
6.1.c	250,00	210,00	235,80	16,96	250,00	210,00	235,80	16,96	16,96
6.2.s	251,00	178,00	232,25	31,43	247,00	178,00	232,50	29,68	30,55
6.5.e	142,00	45,00	84,41	114,92	150,00	32,00	79,98	147,54	131,23
6.7.a	255,00	138,00	212,30	55,11	255,00	138,00	212,30	55,11	55,11
6.10.a	255,00	109,00	185,05	78,90	255,00	109,00	185,00	78,92	78,91
6.12.a	251,00	174,00	235,90	32,64	251,00	174,00	236,90	32,50	32,57
6.13.a	251,00	178,00	232,85	31,35	251,00	178,00	232,95	31,34	31,34

Tabela 6.8 – Resumo dos valores de Uniformidade tomográfica encontrados para as reconstruções efetuadas com *phantom* matemático sem ruído.

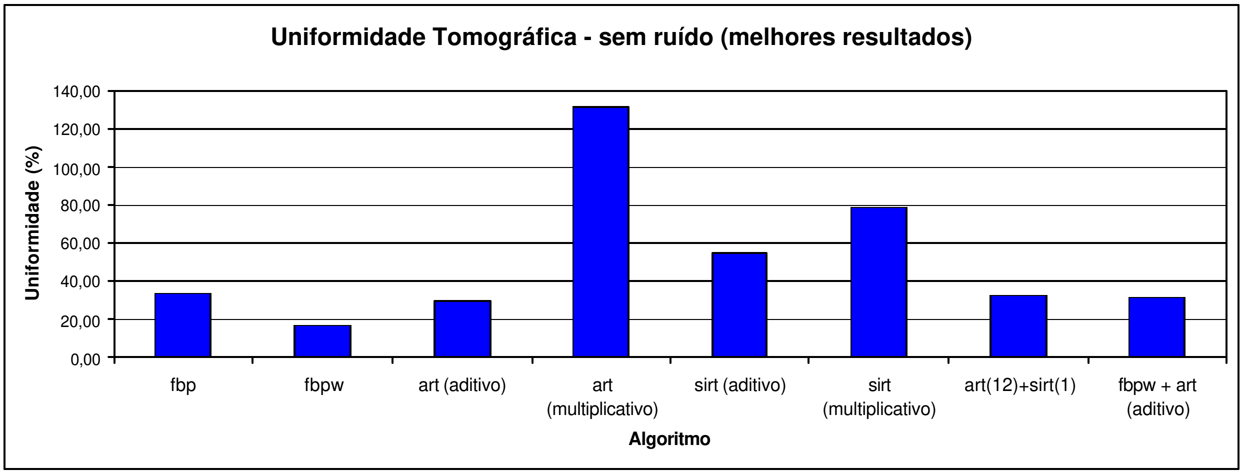


Figura 6.15 – Representação gráfica da Tabela 6.8.

Analisando os resultados obtidos, pode-se afirmar que os algoritmos ART – multiplicativo e SIRT (aditivo / multiplicativo) não apresentam boas performances, não merecendo maior atenção. Os resultados apresentados por estes algoritmos talvez possam ter uma melhora quando da aplicação de filtros pós processamento e da utilização de valores nas matrizes peso com precisão *long double*, possibilitando valores mais refinados para o truncamento das reconstruções.

Além disso, observando-se as imagens e resultados obtidos com as técnicas ART (aditivo) e FBPW + ART, ainda torna-se difícil definir um critério de convergência. Desse modo, mantém-se os mesmos critérios de convergência para o teste seguinte para os algoritmos ART e FBPW + ART.

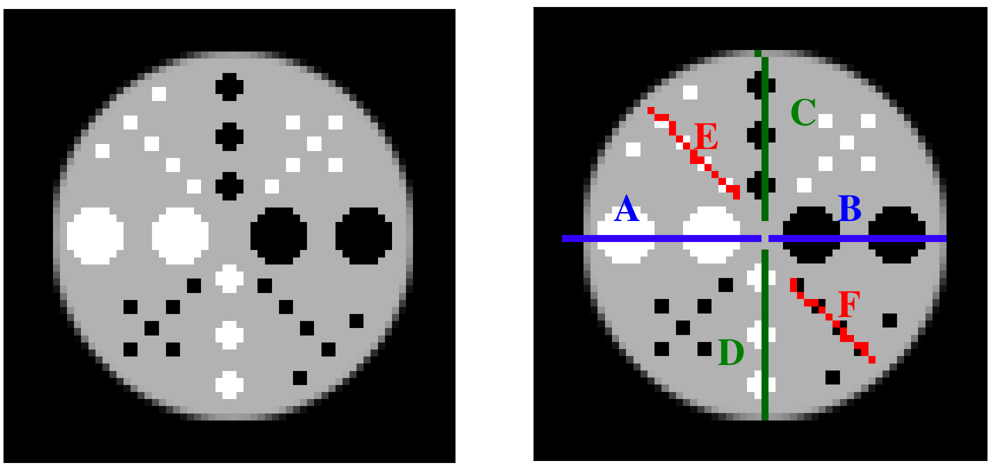
Portanto, para os estudos de contraste a seguir, utilizam-se apenas os algoritmos FBP/FBPW , ART aditivo (critério de convergência por número de iterações), FBPW + ART aditivo com critério de convergência por Desvio padrão e ART + SIRT aditivo com número de iterações igual a 3 (SIRT(3)).

6.1.2 – Contraste

O segundo teste realizado objetiva avaliar a capacidade de reconstrução de objetos quentes e frios com freqüências espaciais variadas, analisando o contraste entre estes e o *back ground* [4] (BG – cinza) de um *phantom* matemático.

O *phantom* elaborado para o estudo é apresentado na figura 6.16.a e os objetos que têm seu contraste avaliado na figura 6.16.b.

O contraste de objetos periódicos é calculado tomando-se a média das contagens dos objetos (quentes ou frios) e a média das contagens dos objetos formados pelo BG entre os mesmos objetos. O cálculo do contraste se dá pela equação 6.2, onde Cont é o contraste observado, C_{obj} = o número médio de contagens por *pixel* nos objetos e C_{BG} = o número médio de contagens por pixel da região de BG.



(b)

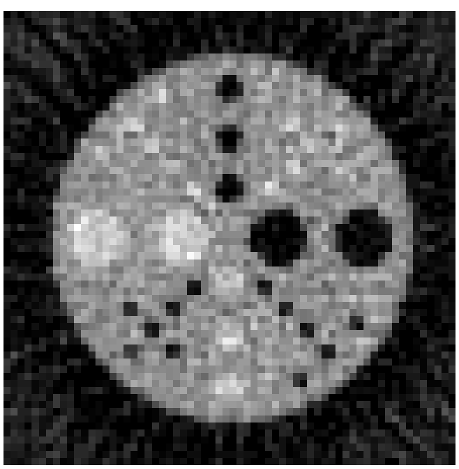
(a)

Figura 6.16 – (a) *Phantom* matemático. (b) Objetos periódicos : A = círculos quentes de 8 *pixels* de diâmetro, B = círculos frios de 8 *pixels* de diâmetro, C = círculos frios de 4 *pixels* de diâmetro , D = círculos quentes de 4 *pixels* de diâmetro, E = círculos quentes de 2 *pixels* de diâmetros, F = círculos frios de 2 *pixels* de diâmetro.

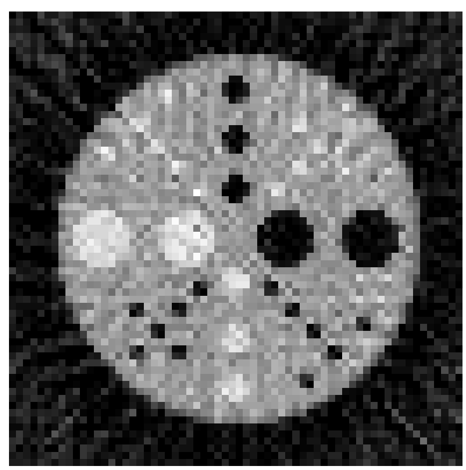
[4] Para o *phantom* matemático, considera–se como BG todo *pixel* com tom de cinza, em oposição ao *pixel* de um objeto quente (branco) ou frio (preto).

$$Cont = \frac{abs(C_{obj} - C_{BG})}{(C_{obj} + C_{BG})} \qquad \textbf{(6.2)}$$

Para a reconstrução do *phantom* matemático proposto na figura 6.16 , primeiro gera-se as projeções e o sinograma do objeto com um programa auxiliar. As imagens reconstruídas com os algoritmos FBP/FBPW apresentam-se apenas com a aplicação de filtro rampa. Para a imagem reconstruída com o algoritmo ART (aditivo) utilizou-se ainda como critério de convergência o número de iterações (8, pois, como visto no teste de Uniformidade sem ruído, um número maior de iterações não apresenta melhoras significativas na imagem). O mesmo foi feito com o ART + SIRT (1) (uma iteração). Por último, para o algoritmo híbrido FBPW+ART utilizou-se como critério de parada o parâmetro de controle α de Desvio Padrão igual 0.01.

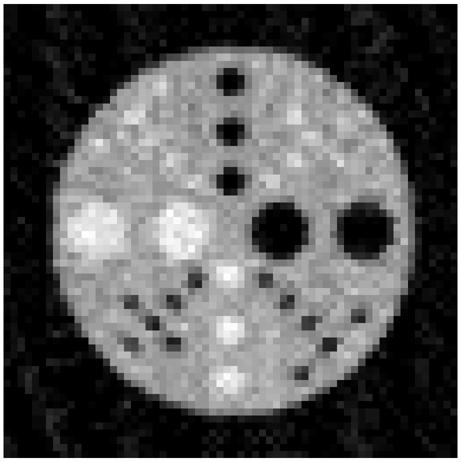


(a)

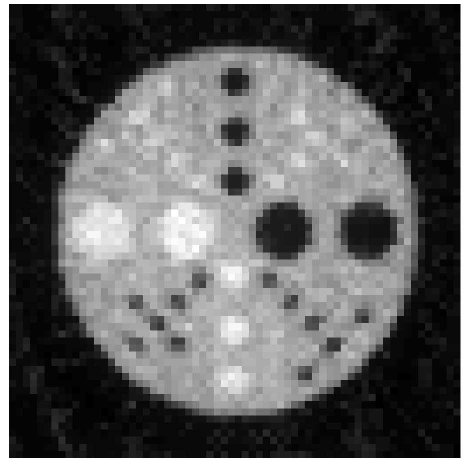


(b)

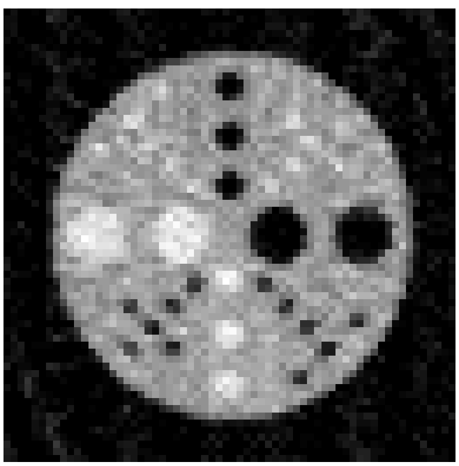
Legenda na pg. 156



(c)



(d)



(e)

Figura 6.17 – (a) *Phantom* matemático da figura 6.16.a reconstruído com o algoritmo *fbp.m* (apenas com o filtro rampa). (b) *Phantom* matemático da figura 6.16.a reconstruído com o algoritmo *fbpw.m* (apenas com o filtro rampa). (c) *Phantom* matemático da figura 6.16.a reconstruído com o algoritmo *art.m* (critério de parada – nº de iterações = 8). (d) *Phantom* matemático da figura 6.16.a reconstruído com o algoritmo *art.m* + *sirt.m* (critério de convergência ART – nº de iterações = 8, critério de convergência SIRT – nº de iterações = 1). (e) *Phantom* matemático da figura 6.16.a reconstruído com o algoritmo *fbpw.m* + *art.m* (critério de convergência ART - α Desvio Padrão = 0.01).

Seguindo a equação 6.2, calcula-se o contraste dos objetos propostos na figura 6.16.b. Os resultados são apresentados nas Tabelas de 6.9 à 6.13. Dentre os valores apresentados nas Tabelas, o chamado ***% / teórico*** indica o quanto o valor encontrado na reconstrução se aproximou do observado na figura 6.16.ª

6.17.a	Objeto	branco	cinza	Contraste	Contraste 6.16.a	% / teórico
	A	198,72	146,76	***0,150***	***0,178***	***84,49***
		cinza	preto			
	B	135,84	16,38	***0,785***	***1,000***	***78,48***
		preto	cinza			
	C	25,14	138,06	***0,692***	***1,000***	***69,19***
		cinza	branco			
	D	145,57	193,28	***0,141***	***0,178***	***79,10***
		cinza	branco			
	E	146,67	193,83	***0,139***	***0,178***	***77,81***
		preto	cinza			
	F	33,58	132,60	***0,596***	***1,000***	***59,59***

Tabela 6.9 – Valores de contraste para o *phantom* matemático da figura 6.16.a, reconstruído com o algoritmo *fbp.m* – imagem 6.17.a .

6.17.b	Objeto	branco	cinza	Contraste	Contraste 6.16.a	% / teórico
	A	214,50	156,94	***0,155***	***0,178***	***87,06***
		cinza	preto			
	B	148,41	14,30	***0,824***	***1,000***	***82,42***
		preto	cinza			
	C	16,86	149,86	***0,798***	***1,000***	***79,77***
		cinza	branco			
	D	153,35	212,28	***0,161***	***0,178***	***90,55***
		cinza	branco			
	E	157,60	207,42	***0,136***	***0,178***	***76,68***
		preto	cinza			
	F	26,58	148,53	***0,696***	***1,000***	***69,64***

Tabela 6.10 – Valores de contraste para o *phantom* matemático da figura 6.16.a, reconstruído com o algoritmo *fbpw.m* – imagem 6.17.b .

6.17.c	Objeto	branco	cinza	Contraste	Contraste 6.16.a	% / teórico
	A	221,83	166,31	***0,143***	***0,178***	***80,36***
		cinza	preto			
	B	150,70	16,73	***0,800***	***1,000***	***80,02***
		preto	cinza			
	C	20,69	148,69	***0,756***	***1,000***	***75,57***
		cinza	branco			
	D	166,90	225,11	***0,148***	***0,178***	***83,42***
		cinza	branco			
	E	168,40	199,17	***0,084***	***0,178***	***47,03***
		preto	cinza			
	F	58,92	144,53	***0,421***	***1,000***	***42,08***

Tabela 6.11 – Valores de contraste para o *phantom* matemático da figura 6.16.a, reconstruído com o algoritmo *art.m* – imagem 6.17.c .

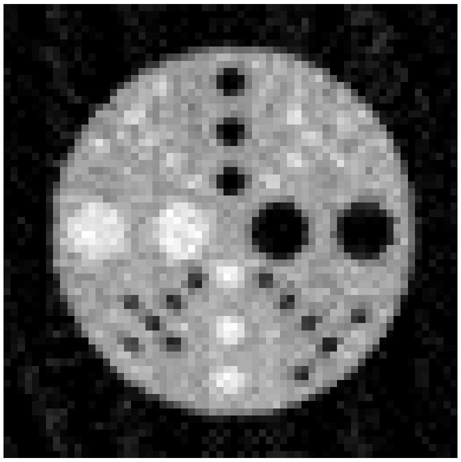
6.17.d	Objeto	branco	cinza	Contraste	Contraste 6.16.a	% / teórico
	A	224,70	177,88	***0,116***	***0,178***	***65,34***
		cinza	preto			
	B	158,16	38,00	***0,613***	***1,000***	***61,26***
		preto	cinza			
	C	47,53	157,56	***0,536***	***1,000***	***53,65***
		cinza	branco			
	D	175,70	223,64	***0,120***	***0,178***	***67,44***
		cinza	branco			
	E	176,40	203,92	***0,072***	***0,178***	***40,65***
		preto	cinza			
	F	80,67	152,13	***0,307***	***1,000***	***30,70***

Tabela 6.12 - Valores de contraste para o *phantom* matemático da figura 6.16.a, reconstruído com o algoritmo *art.m (8) + sirt.m(1)* – imagem 6.17.d .

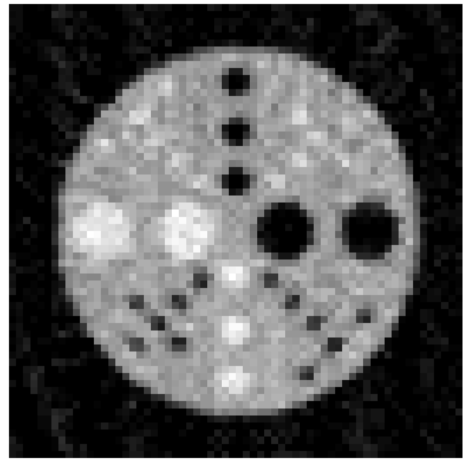
6.17.e	Objeto	branco	cinza	Contraste	Contraste 6.16.a	% / teórico
	A	219,92	164,19	***0,145***	***0,178***	***81,51***
		cinza	preto			
	B	149,20	15,90	***0,807***	***1,000***	***80,74***
		preto	cinza			
	C	17,92	147,47	***0,783***	***1,000***	***78,33***
		cinza	branco			
	D	164,40	211,86	***0,126***	***0,178***	***70,86***
		cinza	branco			
	E	164,40	199,25	***0,096***	***0,178***	***53,84***
		preto	cinza			
	F	52,92	143,40	***0,461***	***1,000***	***46,09***

Tabela 6.13 - Valores de contraste para o *phantom* matemático da figura 6.16.a, reconstruído com o algoritmo *fbpw.m + art.m* – imagem 6.17.e .

Para a imagem reconstruída com o algoritmo ART (figura 6.17.c), observa-se que, caso o critério de convergência utilizado fosse o parâmetro de controle α para o Desvio Padrão igual 0.01, os resultados obtidos para esta reconstrução seriam praticamente os mesmos (Figura 6.18 e Tabela 6.14). A figura 6.19 apresenta o parâmetro de controle α para o Desvio Padrão das projeções calculadas, onde se observa que na iteração de número 6 alfa atinge o valor de 0.01. Para a verificação da pouca significância da aplicação de mais duas iterações neste caso, observa-se os valores dos *pixels* sobre a linha e coluna 32 (figura 6.20) por iteração, verificando-se que a partir da iteração de número 6 estes valores praticamente não se alteram.



(a)



(b)

Figura 6.18 – (a) *Phantom* matemático da figura 6.16.a reconstruído com o algoritmo *art.m* (critério de convergência – nº de iterações = 6). (b) *Phantom* matemático da figura 6.16.a reconstruído com o algoritmo *art.m* (critério de convergência – nº de iterações = 8).

	Objeto	branco	cinza	Contraste	Contraste 6.16.a	% / teórico
6.18.a	A	223,13	167,13	*0,143*	*0,178*	*80,61*
		cinza	preto			
	B	151,80	16,64	*0,802*	*1,000*	*80,24*
		preto	cinza			
	C	21,03	149,58	*0,753*	*1,000*	*75,35*
		cinza	branco			
	D	167,80	226,83	*0,150*	*0,178*	*84,04*
		cinza	branco			
	E	167,33	199,17	*0,087*	*0,178*	*48,81*
		preto	cinza			
	F	57,92	142,93	*0,423*	*1,000*	*42,33*

Tabela 6.14 - Valores de contraste para o *phantom* matemático da figura 6.16.a, reconstruído com o algoritmo *art.m* – imagem 6.18.a .

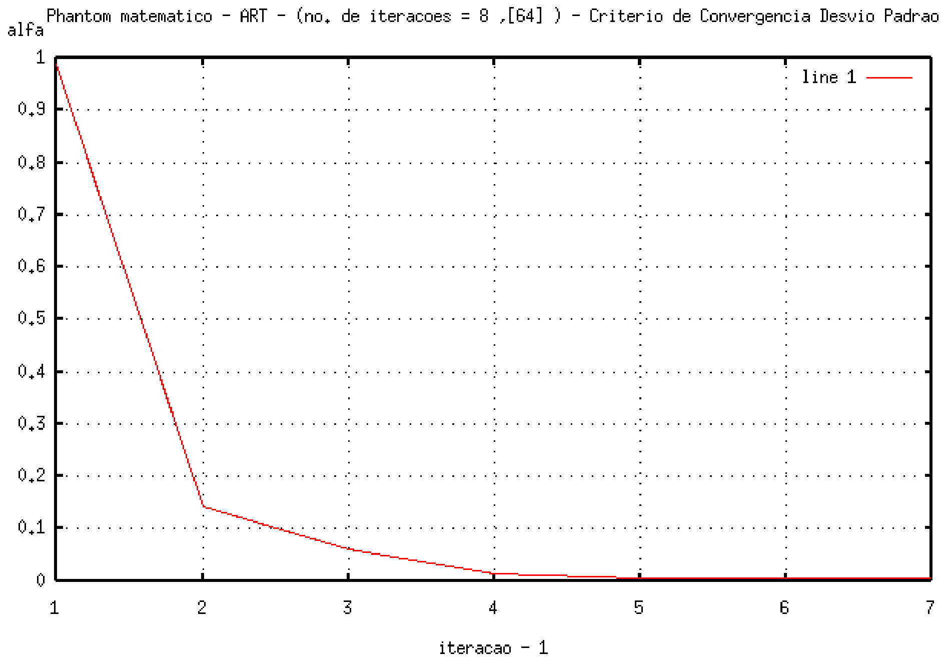
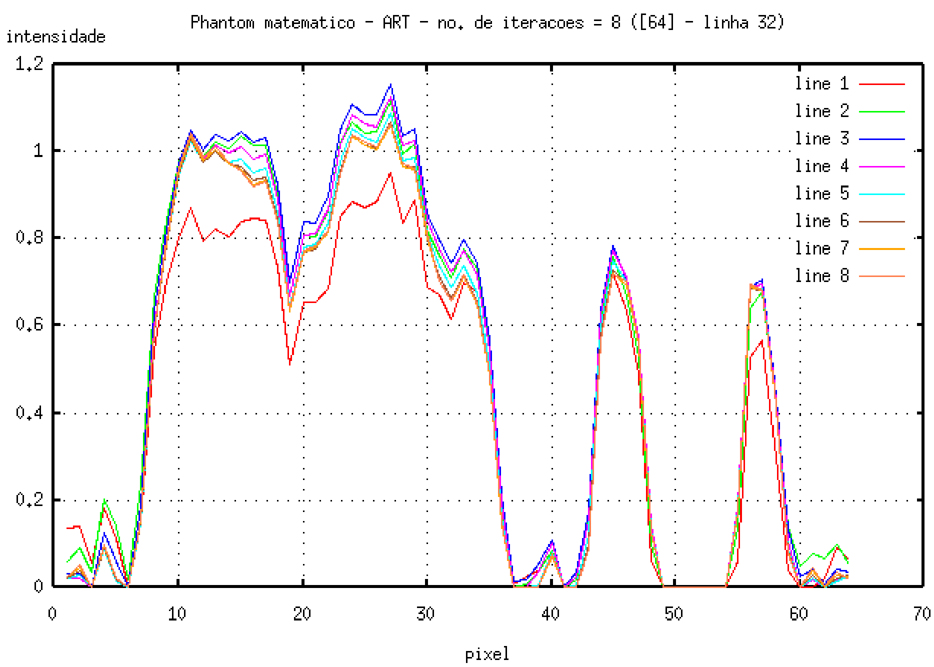
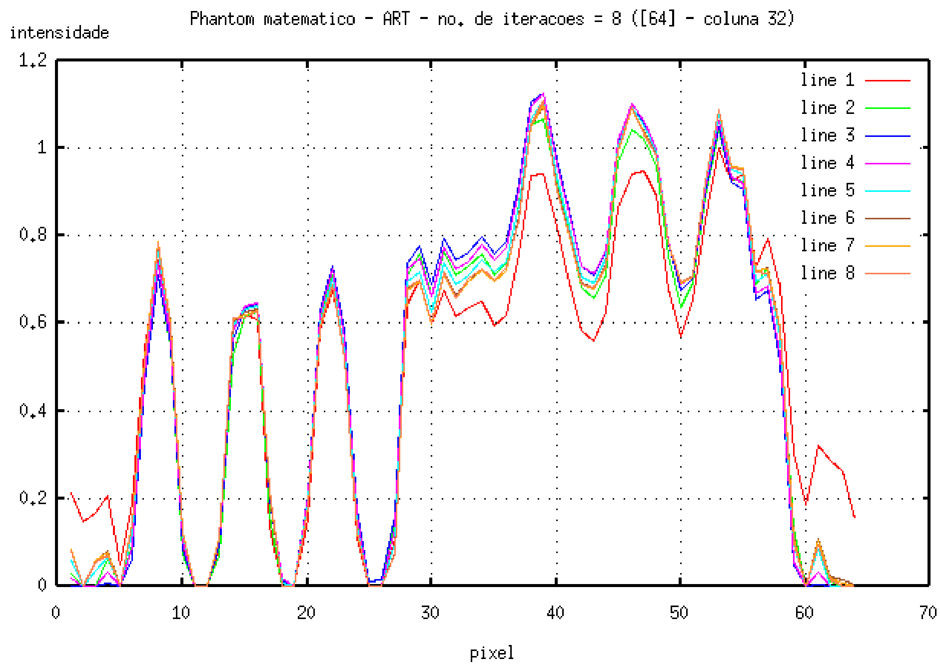


Figura 6.19 – Parâmetro de controle α para Desvio Padrão – reconstrução da figura 6.17.c .



(a)

Legenda na pg. 162



(b)

Figura 6.20 – (a) Perfis das linhas 32 das 8 iterações realizadas na reconstrução de 6.17.c. (b) Perfis das colunas 32 das 8 iterações realizadas na reconstrução de 6.17.c.

Com os resultados destas observações decide-se experimentar para os próximos testes como critério de convergência para o algoritmo art.m o parâmetro de controle α do Desvio Padrão das projeções calculadas igual 0.01.

Para resumir os resultados encontrados, a figura 6.21 trás os valores das Tabelas 6.9 à 6.13, onde os resultados da Tabela 6.10 são substituídos pelos da 6.14, pelos motivos supra citados. Neste gráfico, como na figura 6.15, verifica-se que o algoritmo ***fbp.m*** tem performance igual ou inferior ao ***fbpw.m,*** indicando a redundância de se continuar testando o algoritmo ***fbp.m***. O mesmo ocorre com o algoritmo híbrido ART+ SIRT que tendo uma performance inferior muito significativa em relação aos demais, principalmente em relação ao do algoritmo ART aditivo, torna-se desinteressante para a continuidade dos testes .

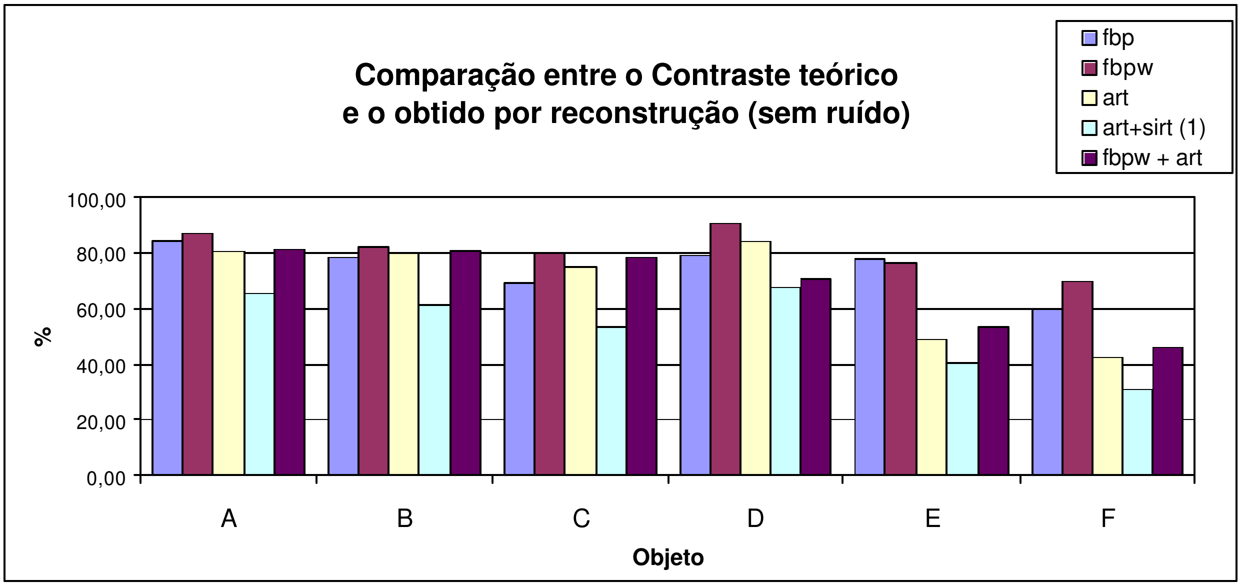


Figura 6.21 – Resumo dos contrastes encontrados para os testes realizados sem ruído.

6.2 – Testes com a Adição de Ruído

Nesta secção, testa-se os algoritmos com a adição de ruído nas projeções dos objetos a serem reconstruídos. Para manter a característica randômica do processo, o ruído adicionado tem distribuição Gaussiana centrada em 0 (zero) e a região positiva da dispersão apresentada na figura 6.22. A amplitude máxima do ruído adicionado varia em 10,4 e 2% do máximo valor dentro da projeção. Isto implica em uma uniformidade plana equivalente a estes números, o que equivaleria a um equipamento com a calibração de alta tensão dos PMT´s e tabelas de correção muito bem calibrados e ajustados (2%), razoavelmente calibrados e ajustados (4%) e mal calibrados (10%) .

A uniformidade plana é uma característica importante na reconstrução tomográfica (ver ítem 2.4). Fica evidente este aspecto quando se observa a curva experimental da figura 6.23 [20], a qual relaciona a uniformidade plana com a tomográfica para imagens reconstruídas com algoritmos de FBP (apenas com filtro rampa) em função da distância da fonte ao centro de rotação do equipamento. No caso dos testes executados, a relação diâmetro do cilindro / matriz, denota uma distância de aproximadamente 4 cm .

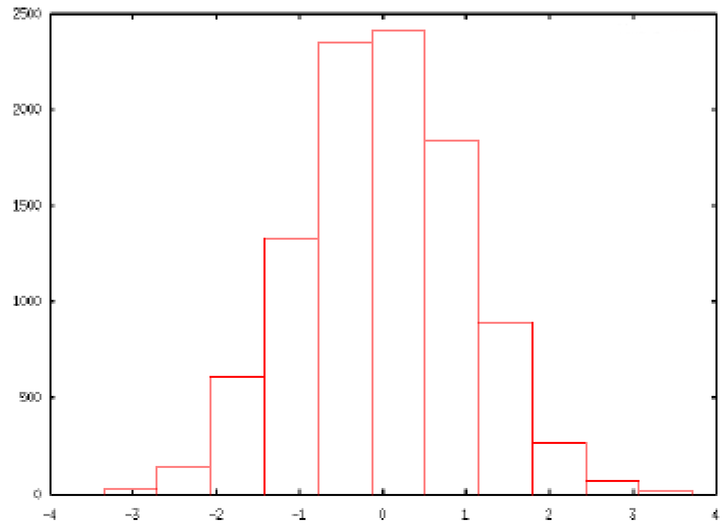


Figura 6.22 – Distribuição Gaussiana do ruído adicionado.

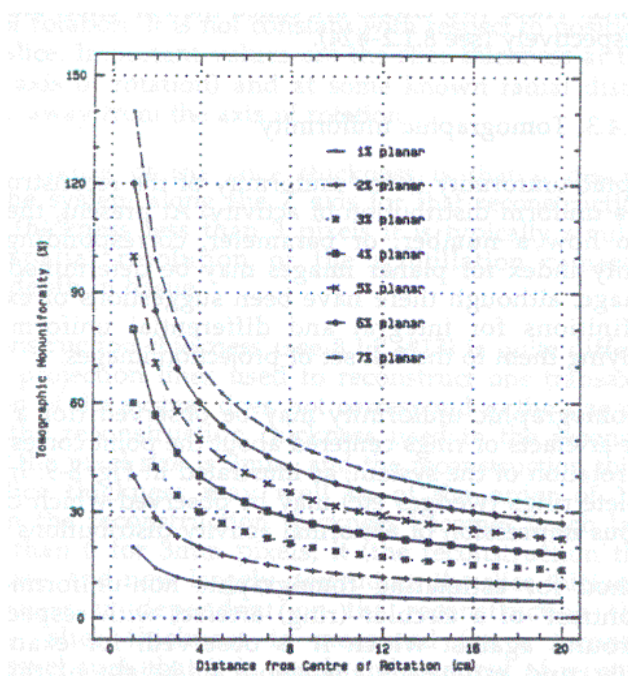


Figura 6.23 – Relação entre a uniformidade plana e a tomográfica em função da distanciada da fonte ao centro de rotação[44].

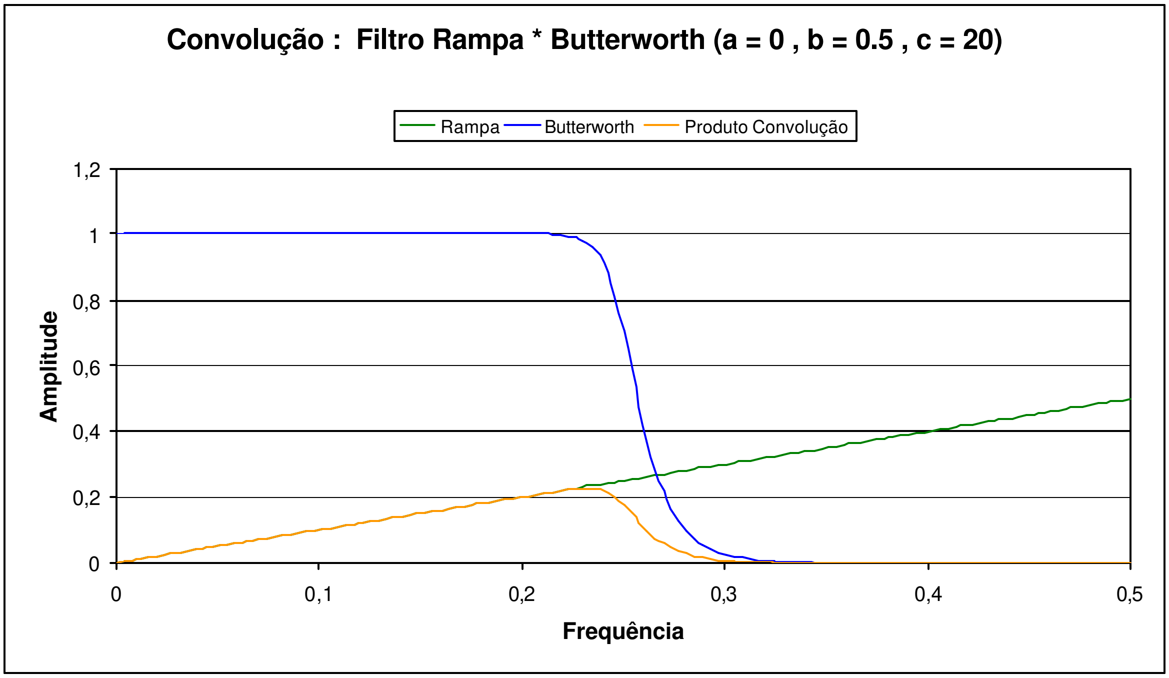
6.2.1 – Uniformidade

Devido a adição de ruído, passa a ser interessante também avaliar a performance de imagens reconstruídas com o algoritmo ***fbpw.m*** utilizando funções janela (equação 4.28.b). A janela escolhida é a Butterworth (equação 6.3) onde a,b e c são parâmetros de controle. Dois exemplos de conjuntos de parâmetros A (a = 0 , b = 0.5 e c = 20) , denominado buttA e, B (a =0 , b= 0.25 e c = 18) denominado buttB são apresentados na figura 6.24, as funções janela e o resultado da convolução com o filtro rampa (equação 4.16) são apresentados nesta figura.

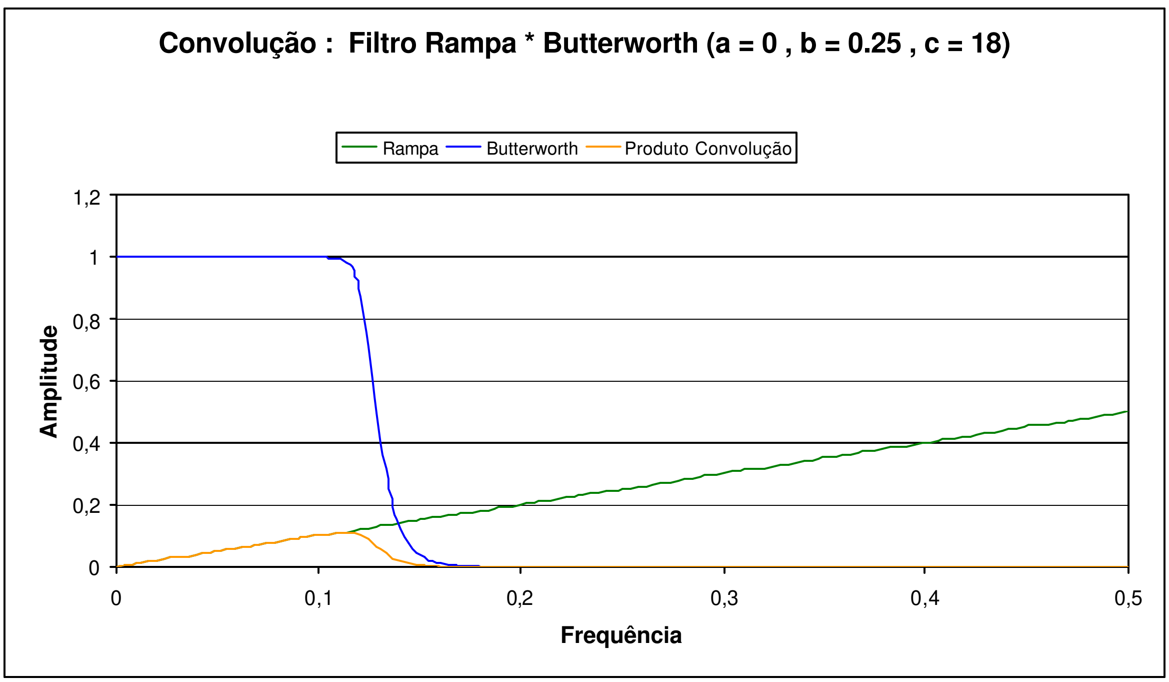
Os testes de uniformidade com a adição de ruído realizados foram os mesmos feitos no ítem 6.1.1, apenas porém para os algoritmos ***fbpw.m*** (com e sem janela), ***art.m*** (aditivo), ***art.m*** (aditivo) + ***sirt.m*** (aditivo) e ***fbpw.m*** com buttA + ***art.m*** (aditivo), pelos motivos já expostos anteriormente, utilizando-se como critério de convergência o parâmetro de controle α do Desvio Padrão das projeções calculadas.

$$w(f) = \frac{1 + a.f}{\sqrt{1 + \left(\dfrac{f}{b.f_c}\right)^{2.c}}} \qquad \textbf{(6.3)}$$

Onde : f = fração da freqüência de corte , no caso Nyquist = 0.5; a, b e c = parâmetros de controle.



(a)



(b)

Figura 6.24 – (a) Janela Butterworth com o conjunto A de parâmetros. (b) Janela Butterworth com o conjunto B de parâmetros.

- ***Uniformidade plana 10%***

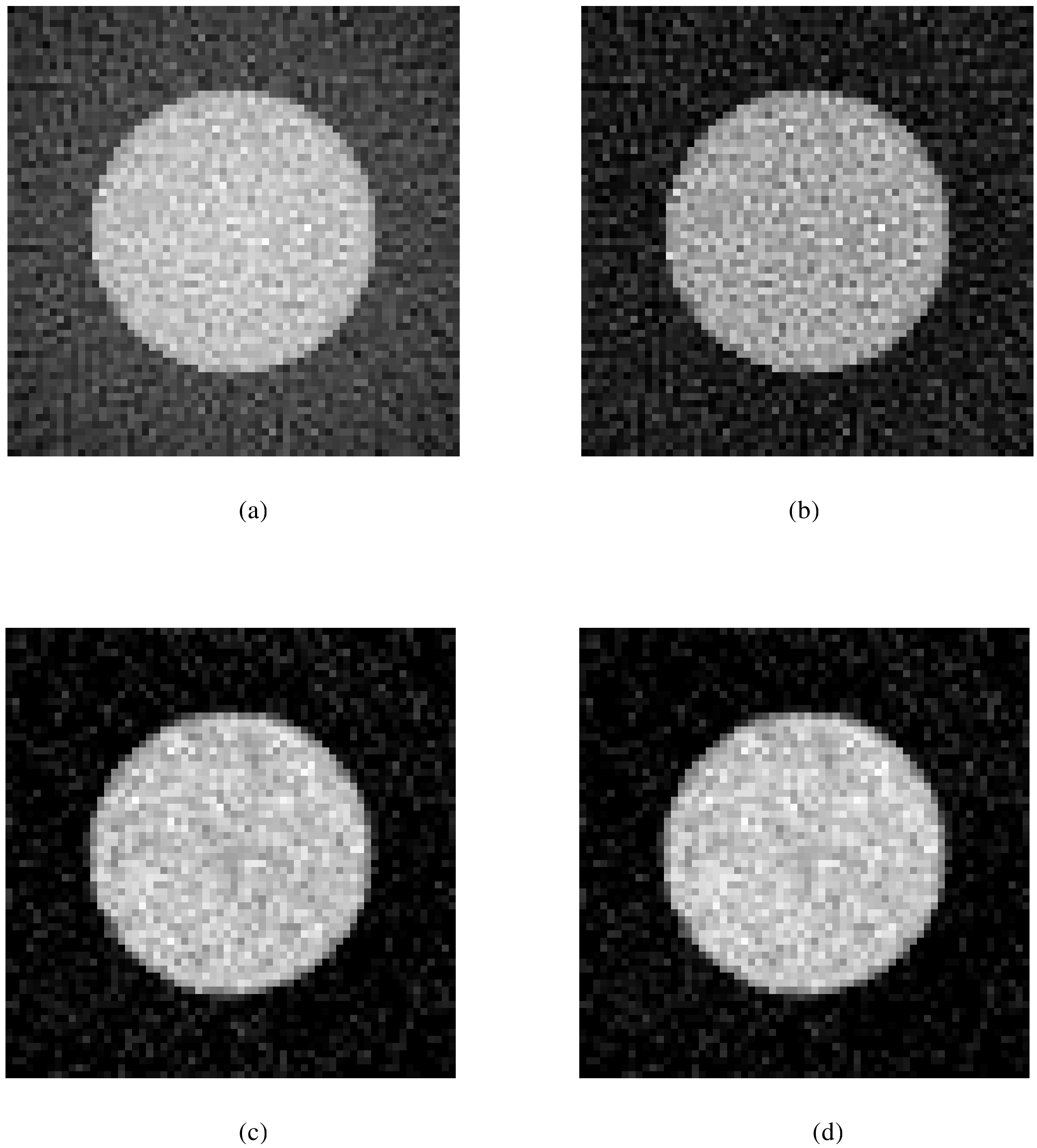


Figura 6.25 – (a) Corte reconstruído com o algoritmo *fbpw.m.* (b) Corte reconstruído com o algoritmo *fbpw.m* com janela buttA. (c) Corte reconstruído com o algoritmo *art.m* com critério de convergência α – Desvio Padrão = 0.01. (d) Corte reconstruído com o algoritmo *fbpw.m* com buttA + *art.m* com critério de convergência α – Desvio Padrão = 0.01.

	x				y				
Imagem	***Cmax***	***Cmin***	***Cmedia***	***Unif tomo***	***Cmax***	***Cmin***	***Cmedia***	***Unif tomo***	***Unif. Media***
6.25.a	250,00	98,00	166,65	91,21	226,00	106,00	165,69	72,42	81,82
6.25.b	255,00	101,00	166,11	92,71	223,00	105,00	164,77	71,61	82,16
6.25.c	243,00	125,00	192,05	61,44	251,00	117,00	193,29	69,33	65,38
6.25.d	243,00	125,00	191,70	61,55	251,00	117,00	193,09	69,40	65,48

Tabela 6.15 – Valores de Uniformidade tomográfica para as imagens da figura 6.25.

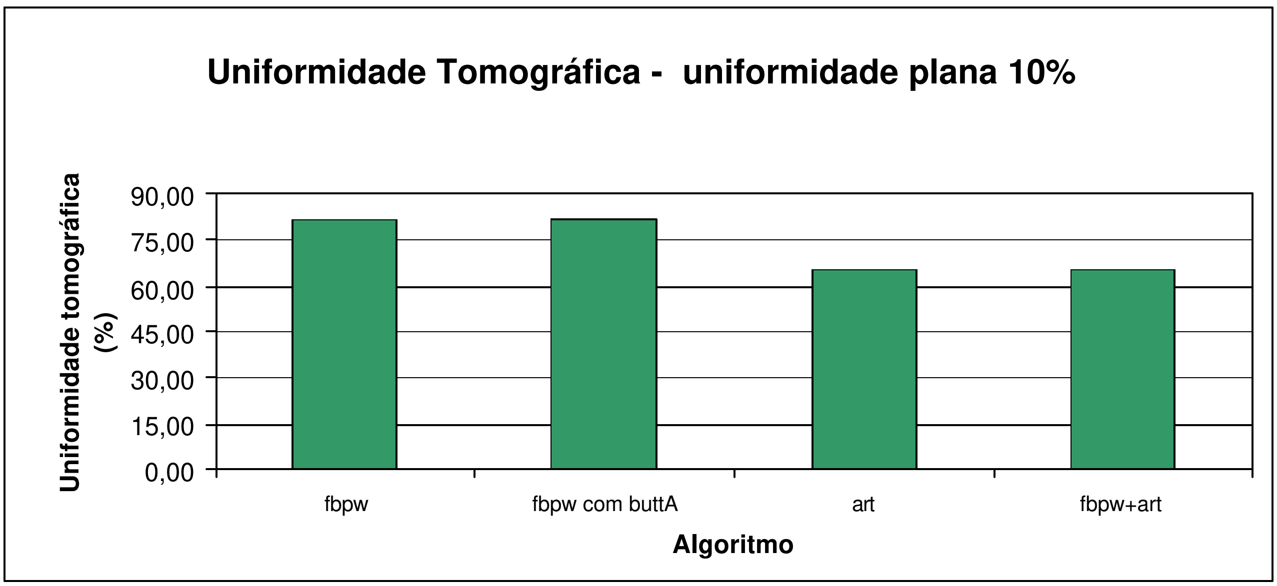
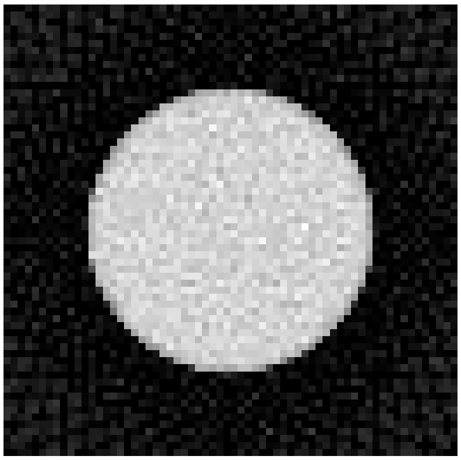
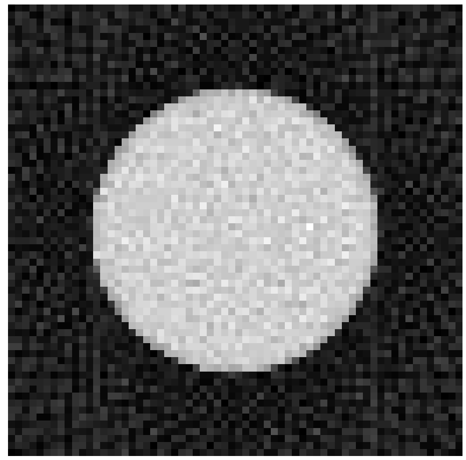


Figura 6.26 – Representação gráfica dos valores de Uniformidade tomográfica obtidos pelas imagens da figura 6.25 – uniformidade plana =10% .

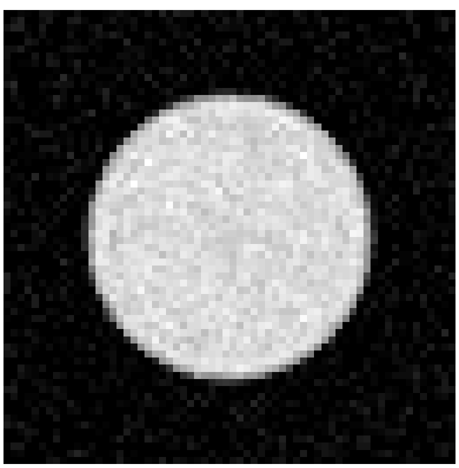
- ***Uniformidade plana 4%***



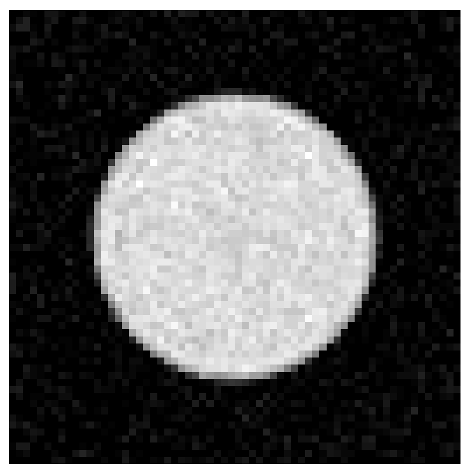
(a)



(b)



(c)



(d)

Figura 6.27 – (a) Corte reconstruído com o algoritmo *fbpw.m.* (b) Corte reconstruído com o algoritmo *fbpw.m* com janela buttA. (c) Corte reconstruído com o algoritmo *art.m* com critério de convergência α – Desvio Padrão = 0.01. (d) Corte reconstruído com o algoritmo *fbpw.m* com buttA+ *art.m* com critério de convergência α – Desvio Padrão = 0.01.

	x				y				
Imagem	***Cmax***	***Cmin***	***Cmedia***	***Unif tomo***	***Cmax***	***Cmin***	***Cmedia***	***Unif tomo***	***Unif. Media***
6.26.a	*255,00*	*162,00*	*209,77*	*44,33*	*243,00*	*166,00*	*208,98*	*36,85*	*40,59*
6.26.b	255,00	166,00	207,28	42,94	239,00	166,00	206,44	35,36	39,15
6.26.c	247,00	154,00	213,88	43,48	247,00	154,00	215,40	43,18	43,33
6.26.d	247,00	154,00	213,59	43,54	247,00	154,00	215,01	43,25	43,40

Tabela 6.16 - Valores de Uniformidade tomográfica para as imagens da figura 6.27.

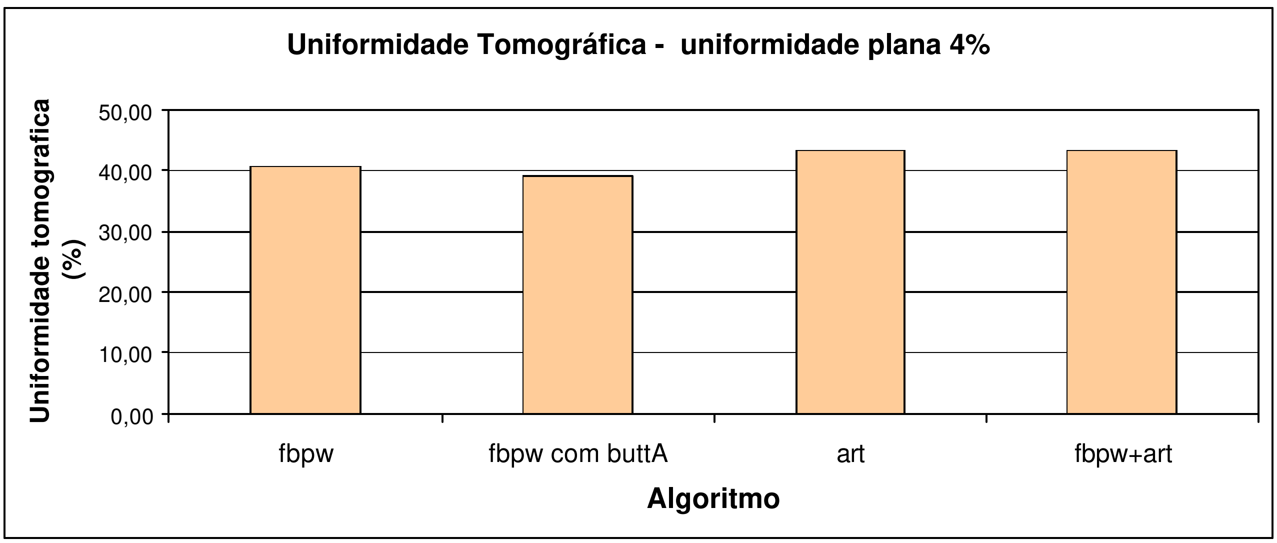
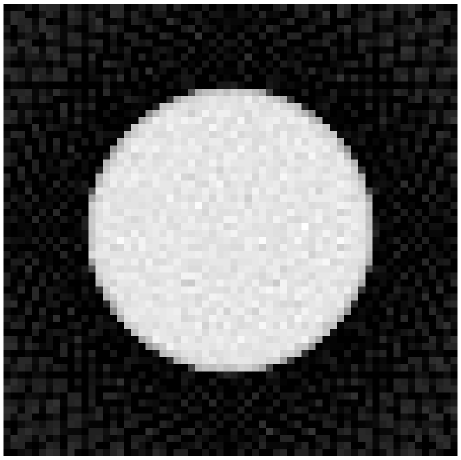
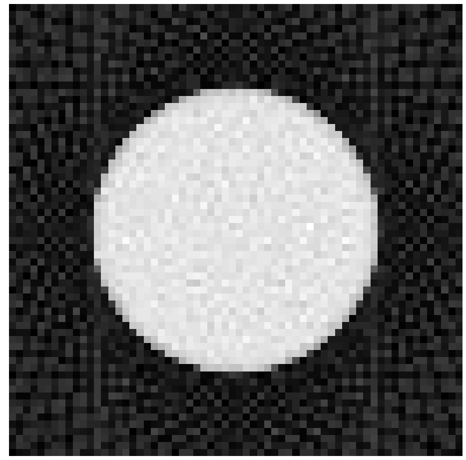


Figura 6.28 - Representação gráfica dos valores de Uniformidade tomográfica obtidos pelas imagens da figura 6.27 – uniformidade plana = 4% .

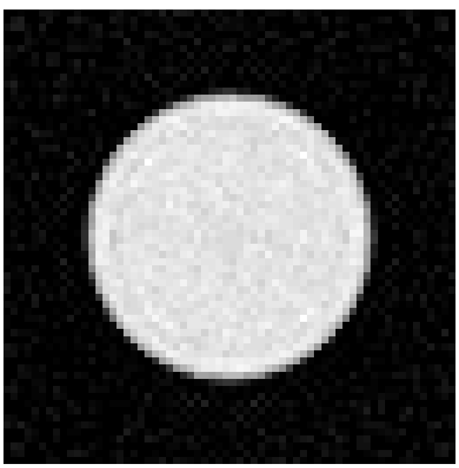
- ***Uniformidade plana 2%***



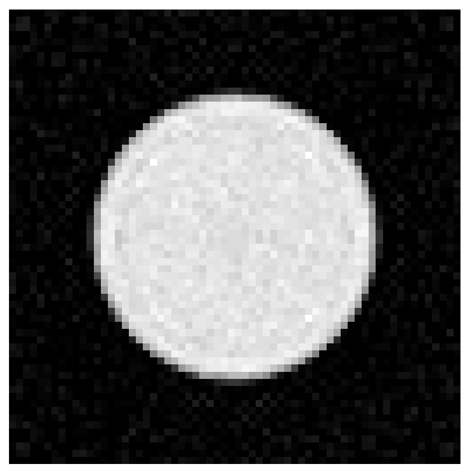
(a)



(b)



(c)



(d)

Figura 6.29 - (a) Corte reconstruído com o algoritmo *fbpw.m.* (b) Corte reconstruído com o algoritmo *fbpw.m* com janela buttA. (c) Corte reconstruído com o algoritmo *art.m* com critério de convergência α – Desvio Padrão = 0.01. (d) Corte reconstruído com o algoritmo *fbpw.m* com buttA + *art.m* com critério de convergência α – Desvio Padrão = 0.01.

	x				y				
Imagem	*Cmax*	*Cmin*	*Cmedia*	*Unif tomo*	*Cmax*	*Cmin*	*Cmedia*	*Unif tomo*	*Unif. Media*
6.28.a	*255,00*	*190,00*	*227,98*	*28,51*	*247,00*	*190,00*	*227,31*	*25,08*	*26,79*
6.28.b	255,00	194,00	226,87	26,89	247,00	194,00	226,37	23,41	25,15
6.28.c	247,00	166,00	223,24	36,28	247,00	166,00	224,73	36,04	36,16
6.28.d	247,00	166,00	223,11	36,30	247,00	166,00	224,34	36,11	36,21

Tabela 6.17 - Valores de Uniformidade tomográfica para as imagens da figura 6.29.

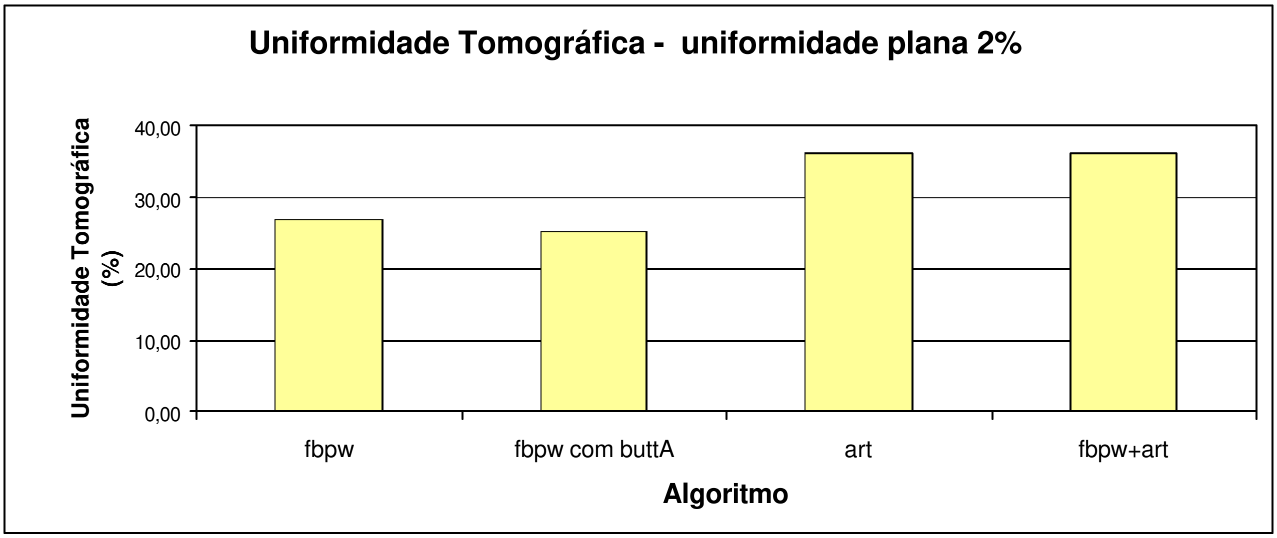


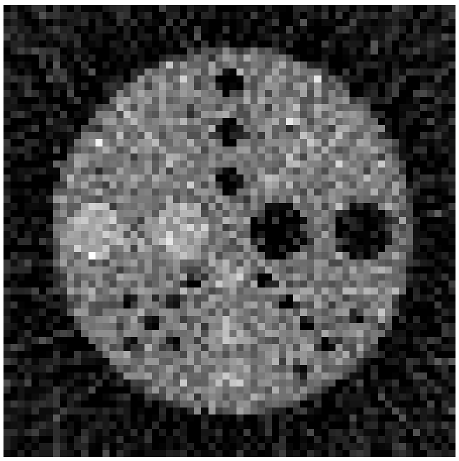
Figura 6.30 - Representação gráfica dos valores de Uniformidade tomográfica obtidos pelas imagens da figura 6.29 – uniformidade plana = 2% .

Observando os valores obtidos para a Uniformidade tomográfica dos cortes reconstruídos com o algoritmo ***fbpw.m*** , e comparando-se estes com os valores da curva da figura 6.23, para as Uniformidades planas iguais a 4 e 2 %, para a distância de 4 cm, verifica-se uma total concordância entres os valores.

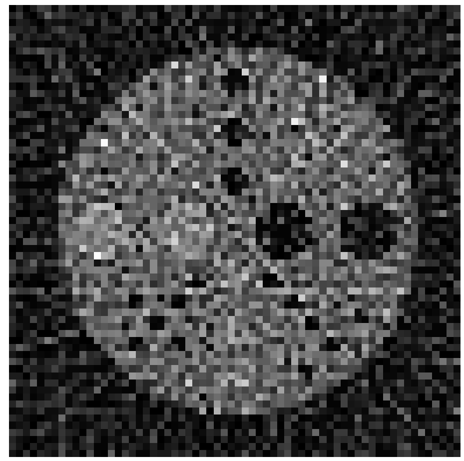
6.2.2 – Contraste

Repetem-se os testes do ítem 6.1.2 com a adição de ruído, para valores de uniformidade plana iguais a 10, 4 e 2%. Os algoritmos testados são o ***fbpw.m***, ***fbpw.m*** com buttA, ***art.m*** e ***fbpw.m*** com buttA + ***art.m***, onde os iterativos têm como critério de convergência o parâmetro de controle α do Desvio Padrão igual a 0.01.

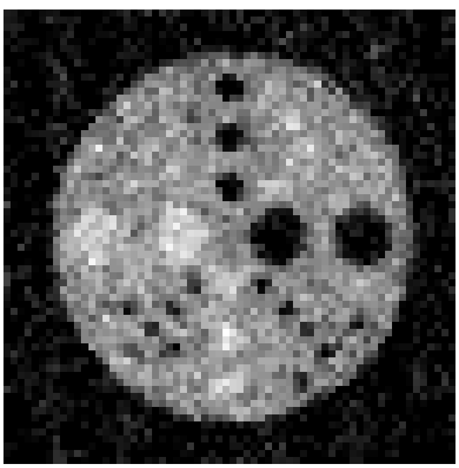
- ***Uniformidade plana 10%***



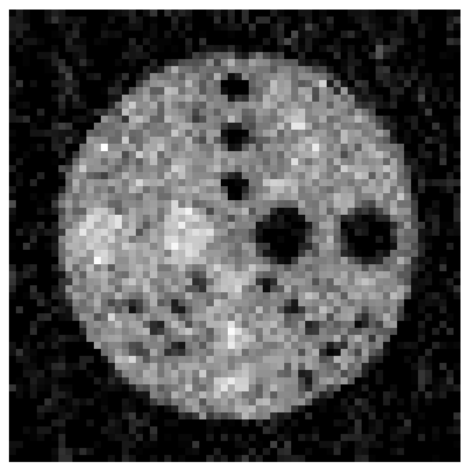
(a)



(b)



(c)



(d)

Figura 6.31 - (a) Corte reconstruído com o algoritmo *fbpw.m.* (b) Corte reconstruído com o algoritmo *fbpw.m* com janela buttA. (c) Corte reconstruído com o algoritmo *art.m* com critério de convergência α – Desvio Padrão = 0.01. (d) Corte reconstruído com o algoritmo *fbpw.m* com buttA + *art.m* com critério de convergência α – Desvio Padrão = 0.01.

6.31.a	Objeto	branco	cinza	Contraste	Contraste 6.16.a	% / teórico
	A	168,64	122,16	***0,160***	***0,178***	***89,79***
		cinza	preto			
	B	113,61	16,47	***0,747***	***1,000***	***74,68***
		preto	cinza			
	C	14,08	118,28	***0,787***	***1,000***	***78,72***
		cinza	branco			
	D	118,70	159,75	***0,147***	***0,178***	***82,82***
		cinza	branco			
	E	120,80	156,33	***0,128***	***0,178***	***72,03***
		preto	cinza			
	F	15,17	116,27	***0,769***	***1,000***	***76,92***

Tabela 6.18 - Valores de contraste para o *phantom* matemático da figura 6.16.a, reconstruído com o algoritmo *fbpw.m* – imagem 6.31.a .

6.31.b	Objeto	branco	cinza	Contraste	Contraste 6.16.a	% / teórico
	A	136,88	99,30	***0,159***	***0,178***	***89,39***
		cinza	preto	**cont**		
	B	94,89	24,81	***0,585***	***1,000***	***58,55***
		preto	cinza	**cont**		
	C	19,25	99,69	***0,676***	***1,000***	***67,63***
		cinza	branco	**cont**		
	D	95,23	129,53	***0,153***	***0,178***	***85,73***
		cinza	braco	**cont**		
	E	92,00	139,25	***0,204***	***0,178***	***114,79***
		preto	cinza	**cont**		
	F	10,75	117,07	***0,832***	***1,000***	***83,18***

Tabela 6.19 - Valores de contraste para o *phantom* matemático da figura 6.16.a, reconstruído com o algoritmo *fbpw.m* com buttA – imagem 6.31.b .

6.31.c	Objeto	branco	cinza	Contraste	Contraste 6.16.a	% / teórico
	A	194,50	145,52	***0,144***	***0,178***	***80,93***
		cinza	preto			
	B	130,27	20,66	***0,726***	***1,000***	***72,62***
		preto	cinza			
	C	16,50	140,42	***0,790***	***1,000***	***78,97***
		cinza	branco			
	D	147,57	197,81	***0,145***	***0,178***	***81,72***
		cinza	branco			
	E	140,33	174,67	***0,109***	***0,178***	***61,24***
		preto	cinza			
	F	38,00	123,60	***0,530***	***1,000***	***52,97***

Tabela 6.20 - Valores de contraste para o *phantom* matemático da figura 6.16.a, reconstruído com o algoritmo *art.m* – imagem 6.31.c .

6.31.d	Objeto	branco	cinza	Contraste	Contraste 6.16.a	% / teórico
	A	190,72	141,72	***0,147***	***0,178***	***82,81***
		cinza	preto			
	B	126,14	19,66	***0,730***	***1,000***	***73,03***
		preto	cinza			
	C	15,50	136,25	***0,796***	***1,000***	***79,57***
		cinza	branco			
	D	144,45	193,86	***0,146***	***0,178***	***82,05***
		cinza	branco			
	E	136,87	173,67	***0,119***	***0,178***	***66,57***
		preto	cinza			
	F	42,17	122,33	***0,487***	***1,000***	***48,73***

Tabela 6.21 - Valores de contraste para o *phantom* matemático da figura 6.16.a, reconstruído com o algoritmo *fbpw.m* com buttA + *art.m* – imagem 6.31.d .

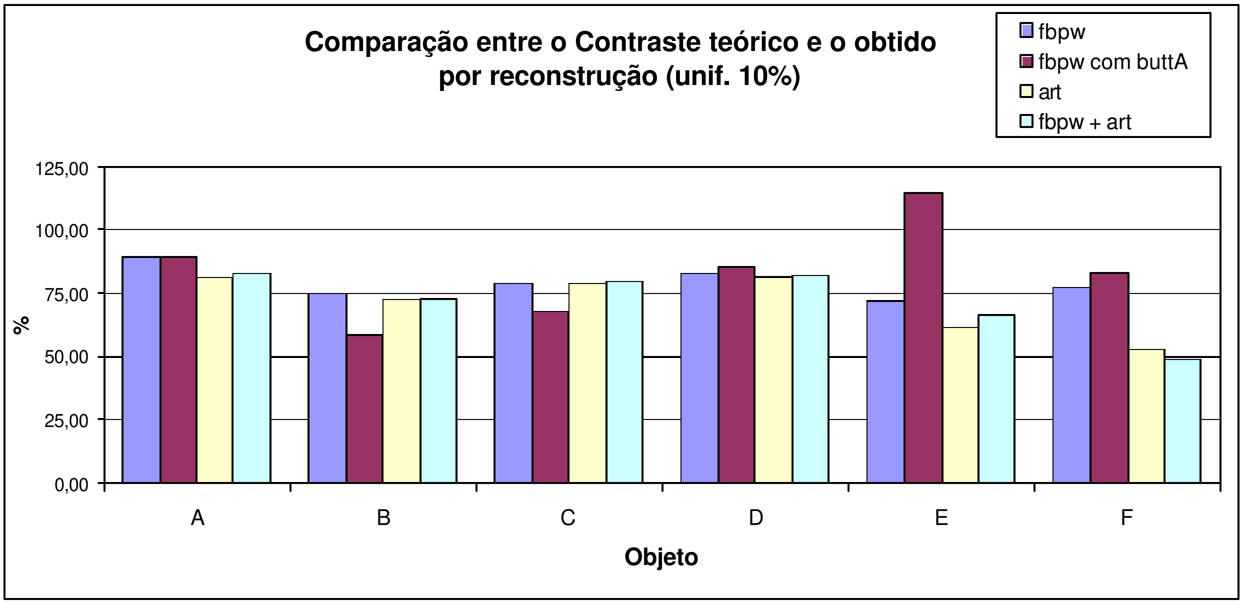
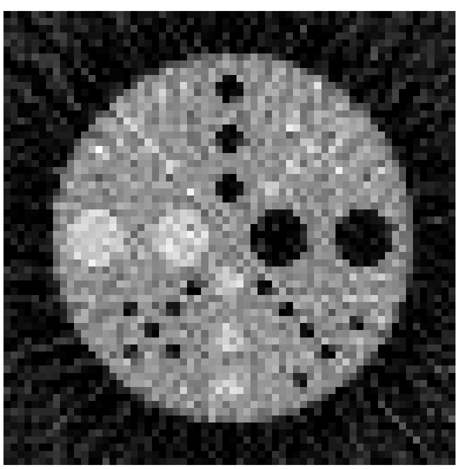
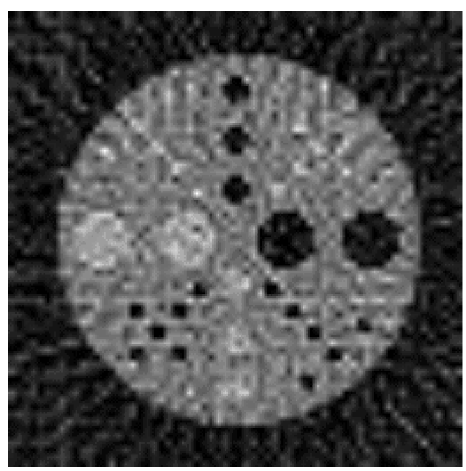


Figura 6.32 - Resumo dos contrastes encontrados para os testes realizados com ruído – uniformidade plana 10%.

- ***Uniformidade plana 4%***

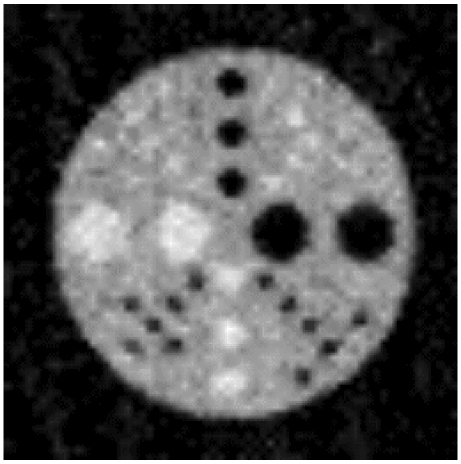


(a)

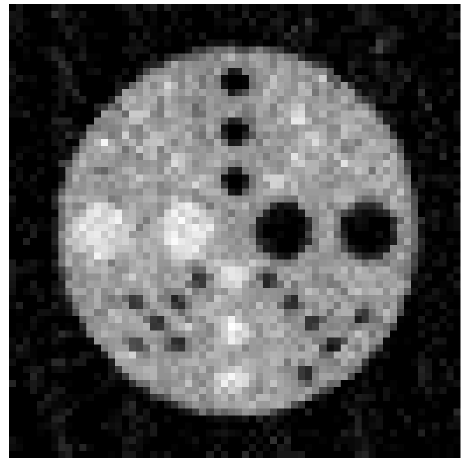


(b)

Legenda na pg. 177



(c)



(d)

Figura 6.33 - (a) Corte reconstruído com o algoritmo *fbpw.m.* (b) Corte reconstruído com o algoritmo *fbpw.m* com janela buttA. (c) Corte reconstruído com o algoritmo *art.m* com critério de convergência α – Desvio Padrão = 0.01. (d) Corte reconstruído com o algoritmo *fbpw.m* com buttA + *art.m* com critério de convergência α – Desvio Padrão = 0.01.

	Objeto	branco	cinza	Contraste	Contraste 6.16.a	% / teórico
6.33.a	A	199,90	146,23	0,155	0,178	87,11
		cinza	preto			
	B	136,97	14,69	0,806	1,000	80,63
		preto	cinza			
	C	14,53	140,06	0,812	1,000	81,20
		cinza	branco			
	D	141,90	194,56	0,157	0,178	87,93
		cinza	branco			
	E	146,07	190,33	0,132	0,178	73,92
		preto	cinza			
	F	18,25	139,27	0,768	1,000	76,83

Tabela 6.22 - Valores de contraste para o *phantom* matemático da figura 6.16.a, reconstruído com o algoritmo *fbpw.m* – imagem 6.33.a .

6.33.b	Objeto	branco	cinza	Contraste	Contraste 6.16.a	% / teórico
	A	157,55	119,20	***0,139***	***0,178***	***77,85***
		cinza	preto			
	B	112,36	27,56	***0,606***	***1,000***	***60,61***
		preto	cinza			
	C	32,82	115,00	***0,556***	***1,000***	***55,59***
		cinza	branco			
	D	116,16	153,03	***0,137***	***0,178***	***76,95***
		cinza	branco			
	E	117,73	155,98	***0,140***	***0,178***	***78,51***
		preto	cinza			
	F	37,77	120,93	***0,524***	***1,000***	***52,40***

Tabela 6.23 - Valores de contraste para o *phantom* matemático da figura 6.16.a, reconstruído com o algoritmo *fbpw.m* com buttA – imagem 6.33.b .

6.33.c	Objeto	branco	cinza	Contraste	Contraste 6.16.a	% / teórico
	A	205,81	155,61	***0,139***	***0,178***	***78,03***
		cinza	preto			
	B	138,30	21,22	***0,734***	***1,000***	***73,40***
		preto	cinza			
	C	30,65	138,46	***0,638***	***1,000***	***63,75***
		cinza	branco			
	D	158,57	207,45	***0,134***	***0,178***	***75,03***
		cinza	branco			
	E	157,37	179,76	***0,066***	***0,178***	***37,31***
		preto	cinza			
	F	65,39	124,84	***0,313***	***1,000***	***31,25***

Tabela 6.24 - Valores de contraste para o *phantom* matemático da figura 6.16.a, reconstruído com o algoritmo *art.m* – imagem 6.33.c .

6.33.d	Objeto	branco	cinza	Contraste	Contraste 6.16.a	% / teórico
	A	214,32	160,14	*0,145*	*0,178*	*81,29*
		cinza	preto			
	B	144,45	17,98	*0,779*	*1,000*	*77,86*
		preto	cinza			
	C	17,50	147,56	*0,788*	*1,000*	*78,80*
		cinza	branco			
	D	161,20	216,89	*0,147*	*0,178*	*82,75*
		cinza	branco			
	E	157,60	194,83	*0,106*	*0,178*	*59,35*
		preto	cinza			
	F	52,75	136,67	*0,443*	*1,000*	*44,30*

Tabela 6.25 - Valores de contraste para o *phantom* matemático da figura 6.16.a, reconstruído com o algoritmo *fbpw.m* com buttA + *art.m* – imagem 6.33.d .

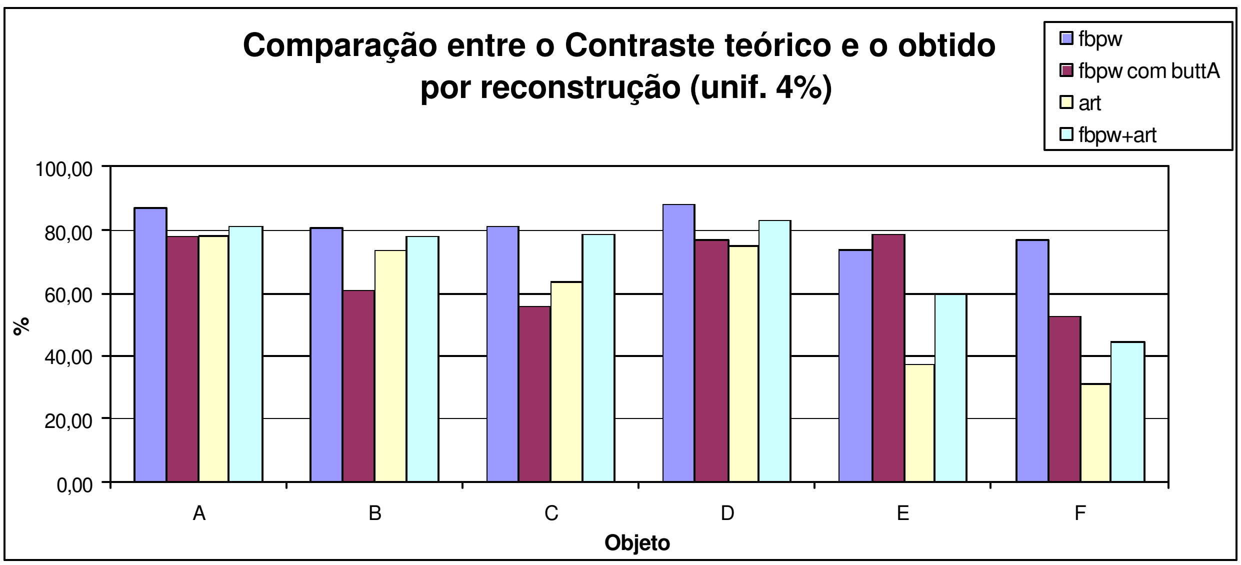
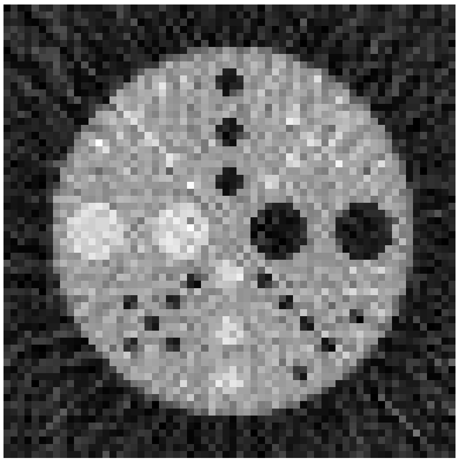
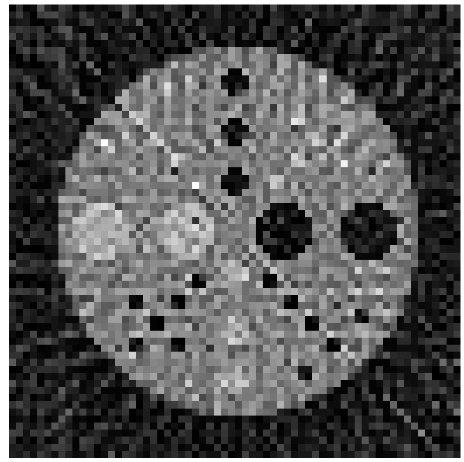


Figura 6.34 - Resumo dos contrastes encontrados para os testes realizados com ruído – uniformidade plana 4 % .

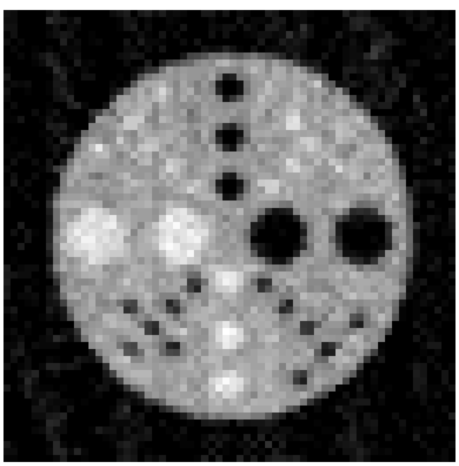
- ***Uniformidade plana 2 %***



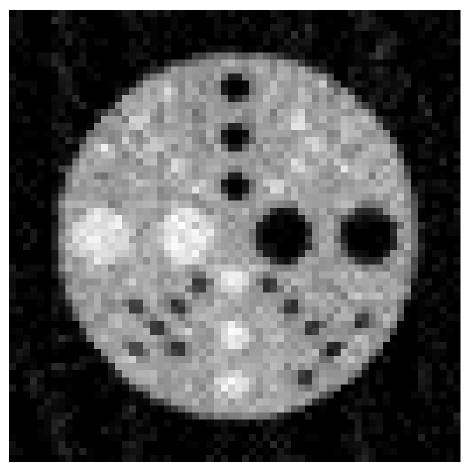
(a)



(b)



(c)



(d)

Figura 6.35 - (a) Corte reconstruído com o algoritmo *fbpw.m.* (b) Corte reconstruído com o algoritmo *fbpw.m* com janela buttA. (c) Corte reconstruído com o algoritmo *art.m* com critério de convergência α – Desvio Padrão = 0.01. (d) Corte reconstruído com o algoritmo *fbpw.m* com buttA + *art.m* com critério de convergência α – Desvio Padrão = 0.01.

6.35.a	Objeto	branco	cinza	Contraste	Contraste 6.16.a	% / teórico
	A	209,80	158,83	***0,138***	***0,178***	***77,68***
		cinza	preto			
	B	150,97	31,37	***0,656***	***1,000***	***65,59***
		preto	cinza			
	C	33,00	152,89	***0,645***	***1,000***	***64,50***
		cinza	branco			
	D	154,93	206,78	***0,143***	***0,178***	***80,53***
		cinza	branco			
	E	158,47	203,08	***0,123***	***0,178***	***69,32***
		preto	cinza			
	F	39,33	152,67	***0,590***	***1,000***	***59,03***

Tabela 6.26 - Valores de contraste para o *phantom* matemático da figura 6.16.a, reconstruído com o algoritmo *fbpw.m* – imagem 6.35.a .

6.35.b	Objeto	branco	cinza	Contraste	Contraste 6.16.a	% / teórico
	A	173,48	126,84	***0,155***	***0,178***	***87,25***
		cinza	preto			
	B	123,33	20,75	***0,712***	***1,000***	***71,20***
		preto	cinza			
	C	16,97	127,11	***0,764***	***1,000***	***76,44***
		cinza	branco			
	D	123,08	172,47	***0,167***	***0,178***	***93,88***
		cinza	branco			
	E	119,47	183,83	***0,212***	***0,178***	***119,21***
		preto	cinza			
	F	6,67	146,73	***0,913***	***1,000***	***91,30***

Tabela 6.27 - Valores de contraste para o *phantom* matemático da figura 6.16.a, reconstruído com o algoritmo *fbpw.m* com buttA – imagem 6.35.b .

6.35.c	Objeto	branco	cinza	Contraste	Contraste 6.16.a	% / teórico
	A	217,96	162,67	***0,145***	***0,178***	***81,61***
		cinza	preto			
	B	147,28	16,70	***0,796***	***1,000***	***79,63***
		preto	cinza			
	C	19,58	148,03	***0,766***	***1,000***	***76,64***
		cinza	branco			
	D	163,90	221,89	***0,150***	***0,178***	***84,45***
		cinza	branco			
	E	162,27	194,75	***0,091***	***0,178***	***51,11***
		preto	cinza			
	F	53,08	139,87	***0,450***	***1,000***	***44,98***

Tabela 6.28 - Valores de contraste para o *phantom* matemático da figura 6.16.a, reconstruído com o algoritmo *art.m* – imagem 6.35.c .

6.35.d	Objeto	branco	cinza	Contraste	Contraste 6.16.a	% / teórico
	A	220,41	164,94	***0,144***	***0,178***	***80,87***
		cinza	preto			
	B	149,45	17,23	***0,793***	***1,000***	***79,33***
		preto	cinza			
	C	19,06	149,83	***0,774***	***1,000***	***77,43***
		cinza	branco			
	D	165,10	222,78	***0,149***	***0,178***	***83,54***
		cinza	branco			
	E	163,33	200,25	***0,102***	***0,178***	***57,05***
		preto	cinza			
	F	55,00	140,47	***0,437***	***1,000***	***43,73***

Tabela 6.29 - Valores de contraste para o *phantom* matemático da figura 6.16.a, reconstruído com o algoritmo *fbpw.m* com buttA + *art.m* – imagem 6.35.d .

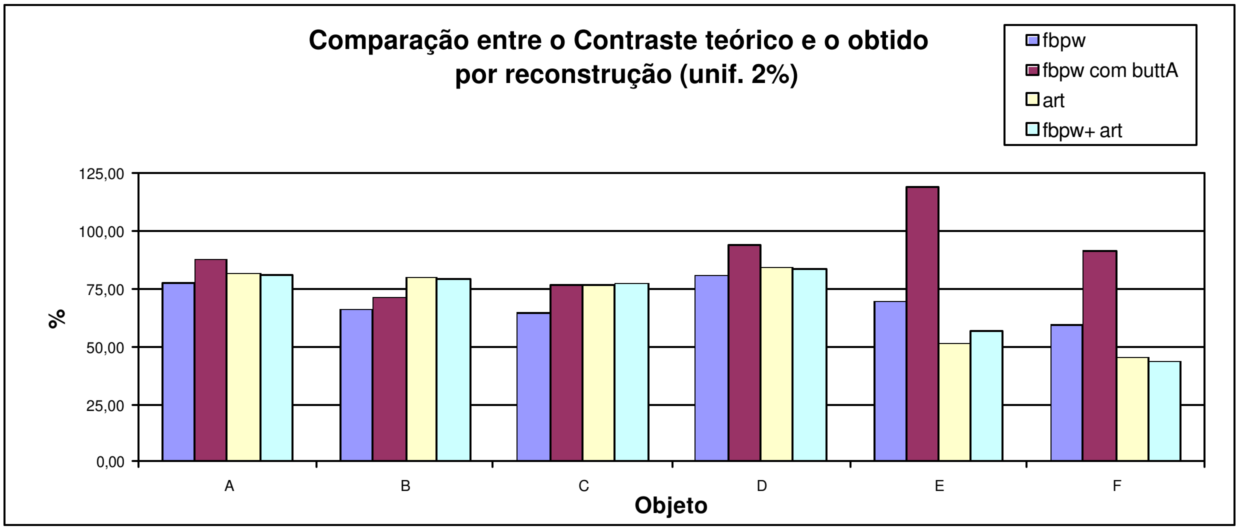


Figura 6.36 - Resumo dos contrastes encontrados para os testes realizados com ruído – uniformidade plana 2 % .

6.3 – Análise Parcial dos Resultados

Os testes até aqui executados (reconstruções de *phantoms* matemáticos), buscaram avaliar o comportamento quantitativo dos algoritmos implementados através de condições bem estabelecidas. Os resultados demonstraram que:

- os testes de uniformidade sem ruído para os algoritmos ART (multiplicativo) e SIRT (aditivo / multiplicativo) indicam que estes não merecem maior atenção no momento, pois apresentaram uma performance muito ruim e no caso do algoritmo SIRT aditivo uma convergência lenta demais, possivelmente pela precisão utilizada nas matrizes de peso ;
- com os testes de Uniformidade e Contraste sem ruído, verifica-se que o algoritmo FBP apresentou resultados iguais ou piores que o FBPW, optando-se assim pela continuidade dos testes apenas com o segundo;
- provavelmente devido a precisão usada nas matrizes pesos, os testes de Uniformidade e Contraste sem ruído com o algoritmo híbrido ART + SIRT

apresentaram resultados piores que os apresentados pelo algoritmo ART – aditivo, não apresentando por este aspecto, maior interesse;

- os testes aplicados ao algoritmo ART – aditivo sem ruído e ao híbrido FBPW + ART(aditivo) sugerem, confirmado pelos testes com ruído, a utilização do parâmetro de controle α do Desvio Padrão das projeções calculadas <= 0.01 como critério de convergência e
- apesar das imagens reconstruídas com o algoritmo ART aditivo apresentarem bons resultados, estas demonstram grande susceptibilidade a ruído.

Em termos específicos e por momento, o que se pode também verificar nos resultados alcançados é um comportamento muito parecido, entre os algoritmos ***fbpw.m*** (com ou sem janela) e ***art.m*** ou o híbrido (***fbpw.m*** + ***art.m***), tanto em uniformidade como em contraste, a menos para objetos pequenos (0.5 *pixel* $^{-1}$), onde o algoritmo FBP se sobressai.

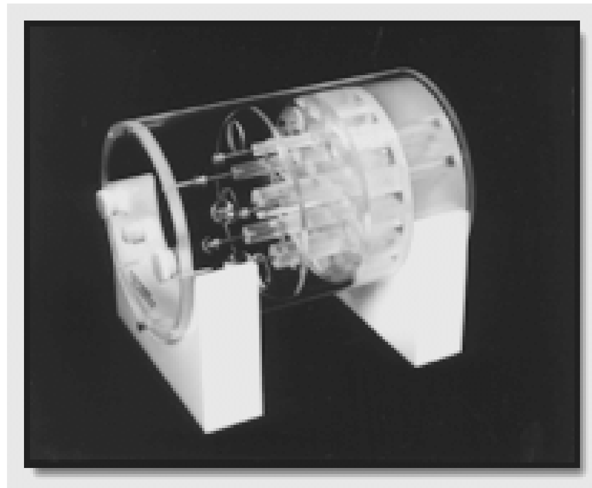
A partir destas informações, reconstrói-se alguns *phantoms*, cujas projeções foram adquiridos no SMN / HC – Unicamp. A reconstrução destas imagens tem por objetivo verificar qualitativa e quantitativamente o comportamento dos algoritmos com imagens reais buscando, se necessário, melhorias para a reconstrução das aquisições de baixa estatística.

6.4 – Reconstrução de Cortes Tomográficos de *Phantoms*

6.4.1 – Estudo Qualitativo

Foram utilizados *phantoms* Jaszczak e Carlson (figura 6.37) , cujos *inserts* são apresentados em conjunto com os cortes reconstruídos. Os *phantoms* foram preenchidos com uma solução de água e ^{99m}Tc (atividade de 20 mCi / 740 MBq). As projeções adquiridas são apresentadas na figura 6.38 e as linhas para os cortes reconstruídos representados pelas linhas vermelhas nas projeções de $\theta = 0º$.

(a)



(b)

Figura 6.37 – (a) *Phantom* Jaszczak . (b) *Phantom* Carlson em suporte.

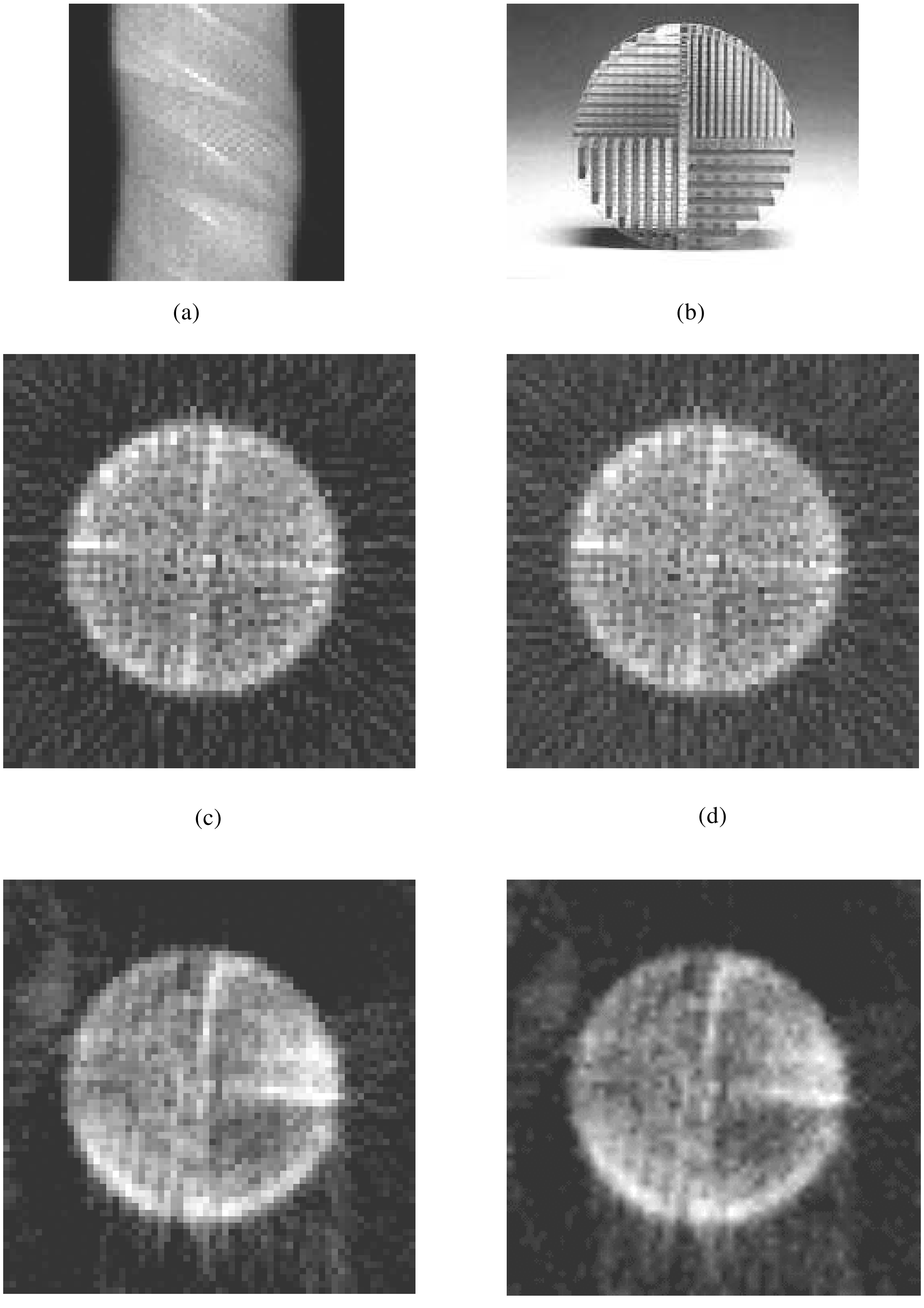
(a)

Legenda na pg. 186

(b)

Figura 6.38 – (a) Aquisição – *phantom* Jaszczak. As linhas vermelhas no frame θ = 0 º indicam os cortes a serem reconstruídos – linhas 25 e 40 . (b) Aquisição *phantom* Carlson. As linhas vermelhas no frame θ = 0 º indicam os cortes a serem reconstruídos – linha 10 , 22 e 40 (sem *insert*) .

Utilizando o programa ***framecut.m***, constrói-se os sinogramas das linhas desejadas. As reconstruções com os algoritmos ***fbpw.m***, ***fbpw.m*** com buttA , ***art.m*** e ***híbrido*** são apresentadas nas figuras 6.39 à 6.43, onde os iterativos utilizam o critério de convergência por parâmetro de controle α do Desvio Padrão das projeções calculadas igual a 0.01.



(a) (b)

(c) (d)

(e) (f)

Figura 6.39 - (a) Sinograma da linha 25 da aquisição da figura 6.38.a. (b) *Insert* utilizado para cálculo da resolução espacial tomográfica (MTF – *Modulation Transfer Function*). (c) Corte da linha 25 / figura 6.38.a – *fbpw.m* . (d) Corte da linha 25 / figura 6.38.a – *fbpw.m* com buttA . (e) Corte da linha 25 / figura 6.38.a – *art.m* . (f) Corte da linha 25 / figura 6.38.a – *hibrido* .

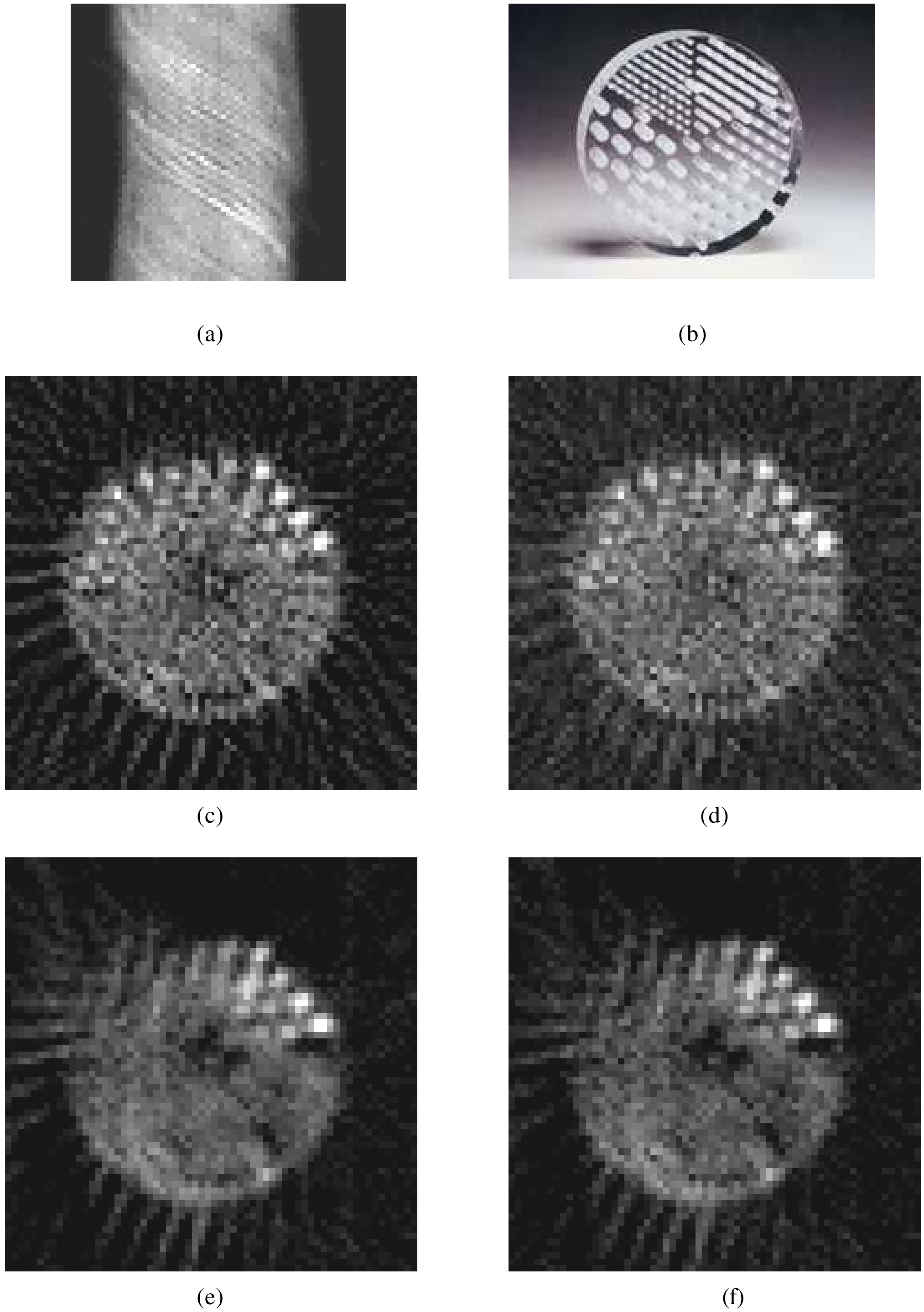


Figura 6.40 - (a) Sinograma da linha 40 da aquisição da figura 6.38.a. (b) *Insert* de furos quentes. (c) Corte da linha 40 / figura 6.38.a – *fbpw.m* . (d) Corte da linha 40 / figura 6.38.a – *fbpw.m* com buttA . (e) Corte da linha 40 / figura 6.38.a – *art.m* . (f) Corte da linha 40 / figura 6.38.a – *hibrido*. .

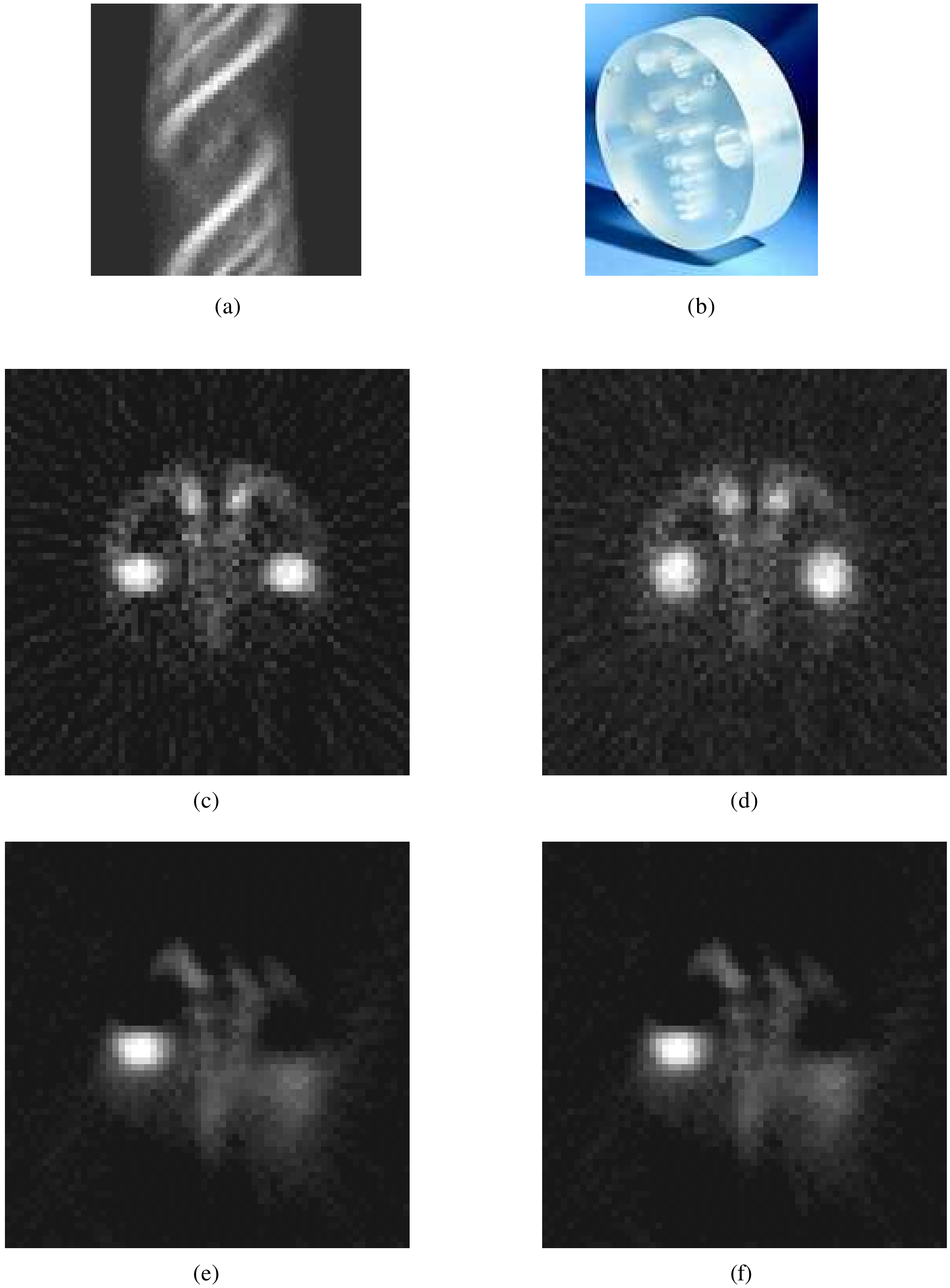


Figura 6.41 - (a) Sinograma da linha 10 da aquisição da figura 6.38.b. (b) *Insert* de furos quentes. (c) Corte da linha 10 / figura 6.38.b – *fbpw.m* . (d) Corte da linha 10 / figura 6.38.b – *fbpw.m* com buttA . (e) Corte da linha 10 / figura 6.38.b – *art.m*. (f) Corte da linha 10 / figura 6.38.b – *hibrido* .

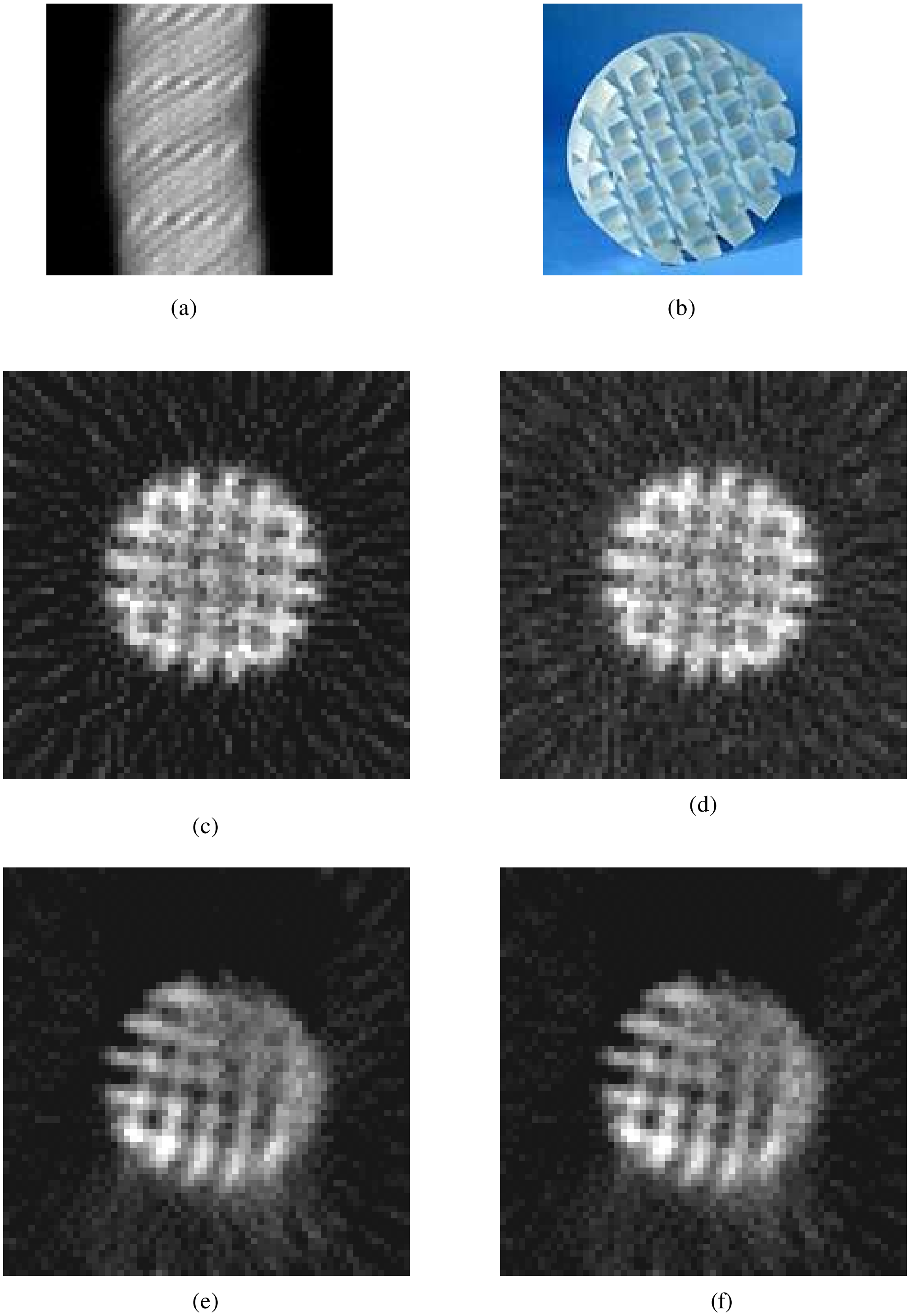
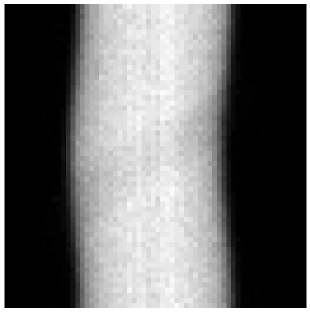
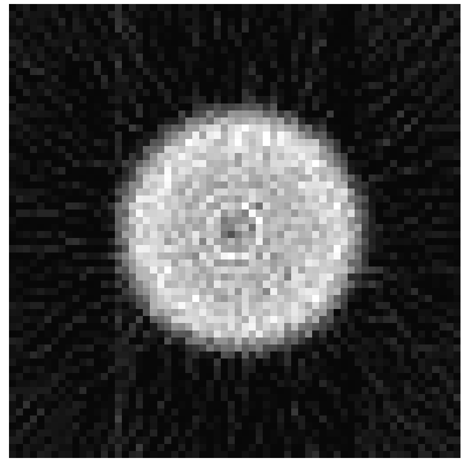


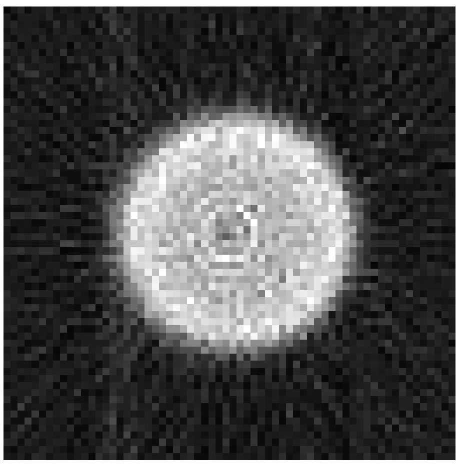
Figura 6.42 - (a) Sinograma da linha 22 da aquisição da figura 6.38.b. (b) *Insert* de linearidade. (c) Corte da linha 22 / figura 6.38.b – *fbpw.m* . (d) Corte da linha 22 / figura 6.38.b – *fbpw.m* com buttA . (e) Corte da linha 22 / figura 6.38.b – *art.m*. (f) Corte da linha 22 / figura 6.38.b – *hibrido*. .



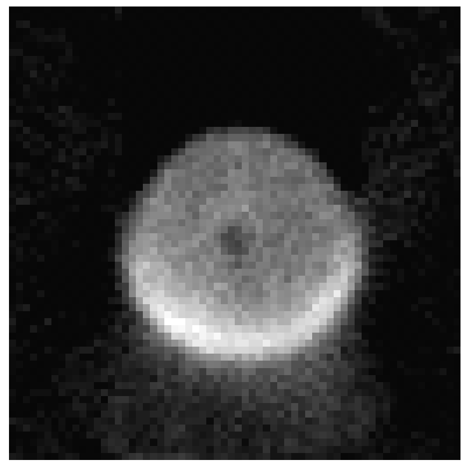
(a)



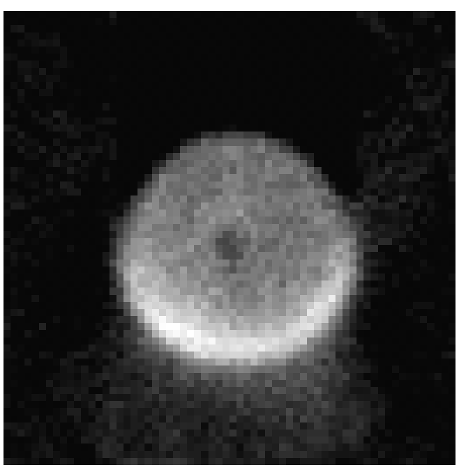
(b)



(c)



(d)



(e)

Figura 6.43 - (a) Sinograma da linha 40 da aquisição da figura 6.38.b. (b) Corte da linha 40 / figura 6.38.b – *fbpw.m* . (c) Corte da linha 40 / figura 6.38.b – *fbpw.m* com buttA . (d) Corte da linha 40 / figura 6.38.b – *art.m*. (e) Corte da linha 40 / figura 6.38.b – *hibrido* .

Observando os cortes reconstruídos, verifica-se que as imagens obtidas com os algoritmos ART e híbrido apresentam artefatos e deformações . As prováveis causas são :

1) a presença de ruído nas projeções o algoritmo ART é bastante susceptível a ruído [42] e

2) a seqüência dos ângulos de visada utilizados para as iterações com *step* de 6º privilegia os dois últimos quadrantes da imagem – figura 6.44.

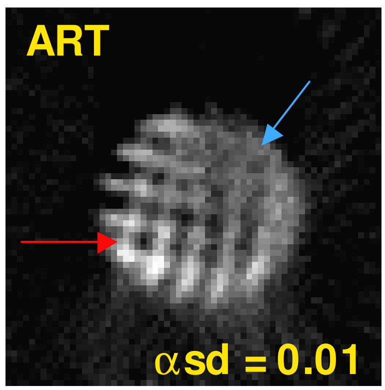


Figura 6.44 – Reconstrução do *insert* de linearidade com o algoritmo *art.m* . As setas indicam os artefatos / distorções na imagem.

Para se corrigir os artefatos e as distorções, implementam-se duas modificações no programa ***art.m*** :

1) introdução de um coeficiente de relaxação (*r*) no fator de correção da equação 4.31, conforme equação 6.4 para atenuar os efeitos da presença de ruído [45] e

2) alteração na seqüência dos ângulos de visada : de 0º,6º,12º ... (*step* de 6º) para 0º, 90º , 180º ... (*step* de 90º) , não privilegiando nenhum quadrante.

Implementa-se a mudança na seqüência e *step* dos ângulos de visada apenas alterando-se a seqüência de contadores e *loops* dentro do programa (*Anexo F* – ***artx.m***).

Quanto ao coeficiente de relaxação r , sugere-se em literatura a utilização de valores fixos entre 0.25 <= r <= 0.5 [45]. Porém, a título de experiência implementa-se também $r = r(\sigma)$, onde $r(\sigma)$ deve ser uma função que tende a zero quando Δp vai a zero e tende a 1 quando Δp muito grande. Heuristicamente implementa-se a função $r(\sigma)$ apresentada na equação 6.5, representada para alguns valores de σ na figura 6.45.

$$f_i^{\,l} = f_i^{\,l-1} + r.\sum_{j=1}^{M} \Delta f_{ij}^{\,l} \qquad \textbf{(6.4)}$$

$$r(\sigma) = 1 - \exp(-|\sigma.\Delta p|) \qquad \textbf{(6.5)}$$

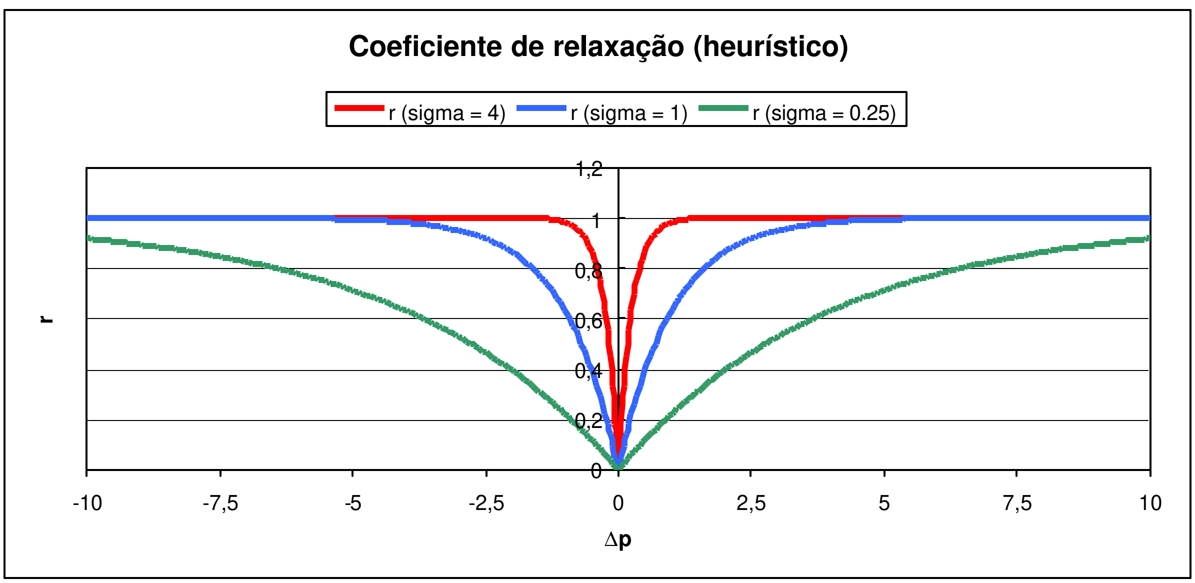
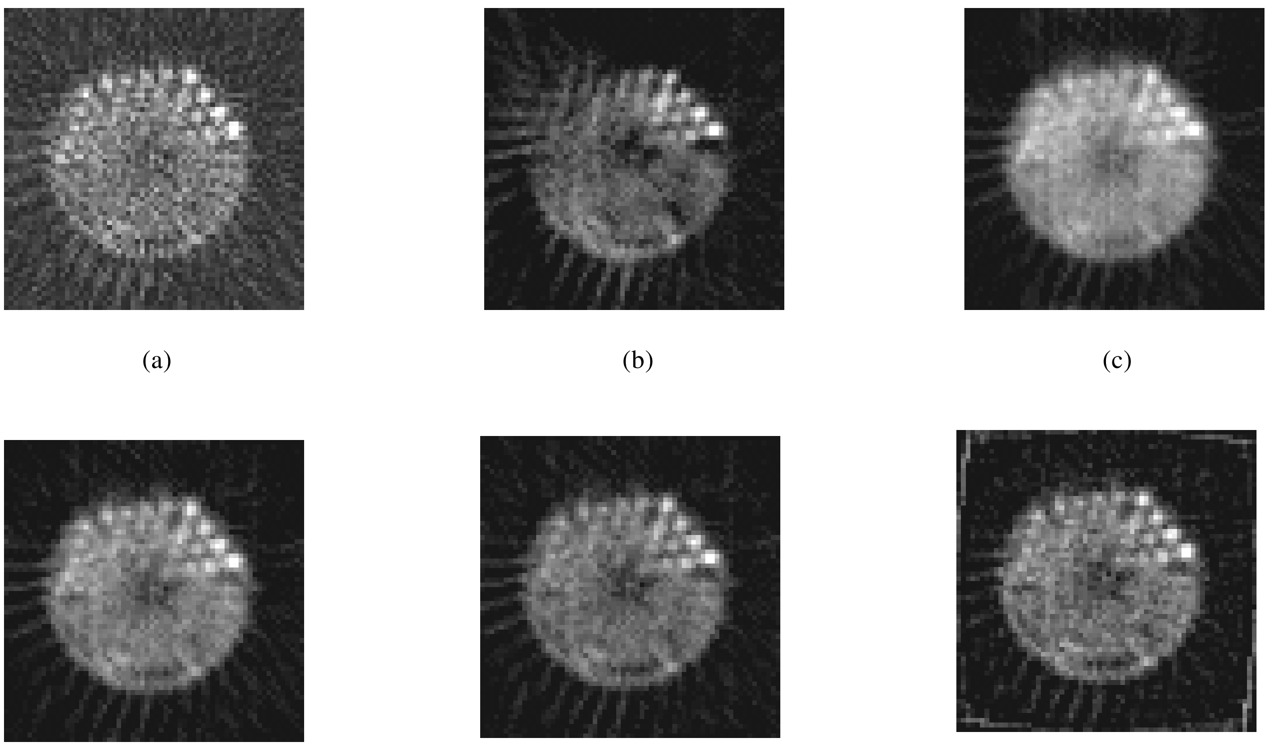


Figura 6.45 – Função r(σ) para os valores de σ = 4, 1, 0.25 .

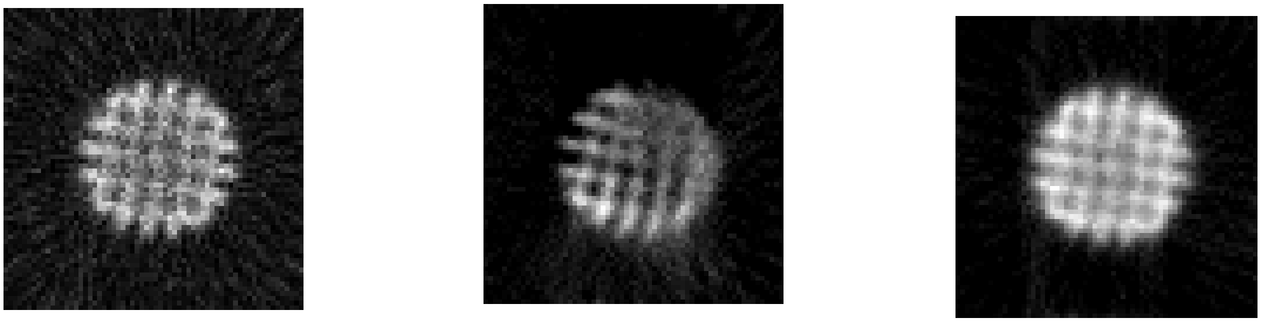
Com as modificações implementadas no programa ***art.m***, repete-se a reconstrução dos cortes das figuras 6.40 e 6.42 para testes com *r* fixo igual a 0.25 e *r*(σ) com σ = 4, 1 e 0.25. As imagens são apresentadas com zoom x2.



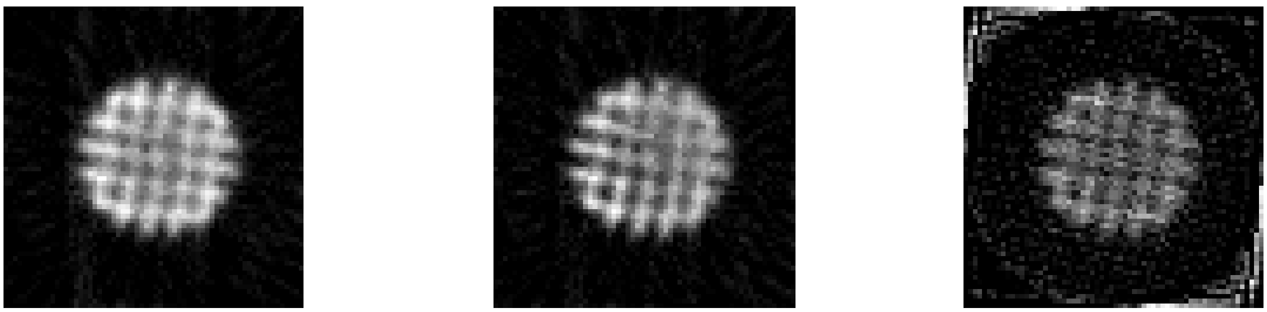
(a) (b) (c)

(d) (e) (f)

Figura 6.46 – (a) Corte reconstruído com programa *fbpw.m* e janela buttA (figura 6.40.d) . (b) Corte reconstruído com programa *art.m* (figura 6.40.e). (c) Corte reconstruído com programa *artx.m* e *r* (σ) , σ = 0.25 . (d) Corte reconstruído com programa *artx.m* e *r* (σ) , σ = 1 . (e) Corte reconstruído com programa *artx.m* e *r* (σ) , σ = 4 . (f) Corte reconstruído com programa *artx.m* e *r* =0.25.

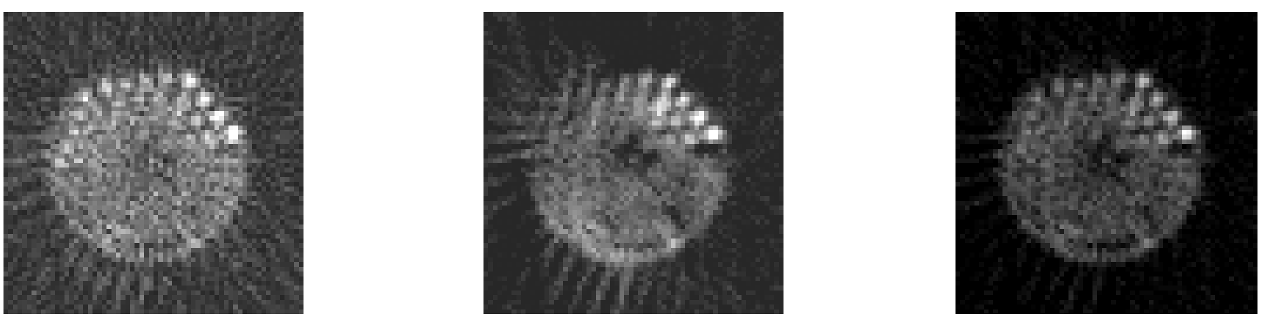


(a) (b) (c)



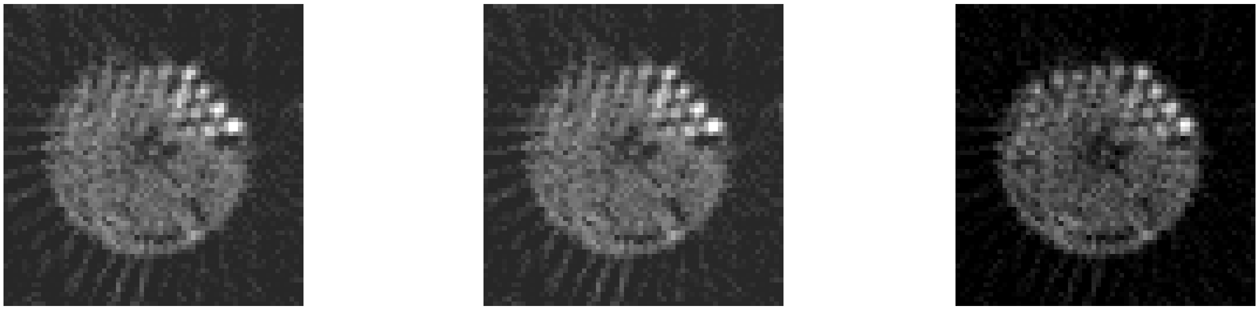
(d) (e) (f)

Figura 6.47 – (a) Corte reconstruído com programa *fbpw.m* e janela buttA (figura 6.42.d) . (b) Corte reconstruído com programa *art.m* (figura 6.42.e). (c) Corte reconstruído com programa *artx.m* e *r (σ)* , σ = 0.25 . (d) Corte reconstruído com programa *artx.m* e *r (σ)* , σ = 1 . (e) Corte reconstruído com programa *artx.m* e *r (σ)* , σ = 4 . (f) Corte reconstruído com programa *artx.m* e *r* =0.25.



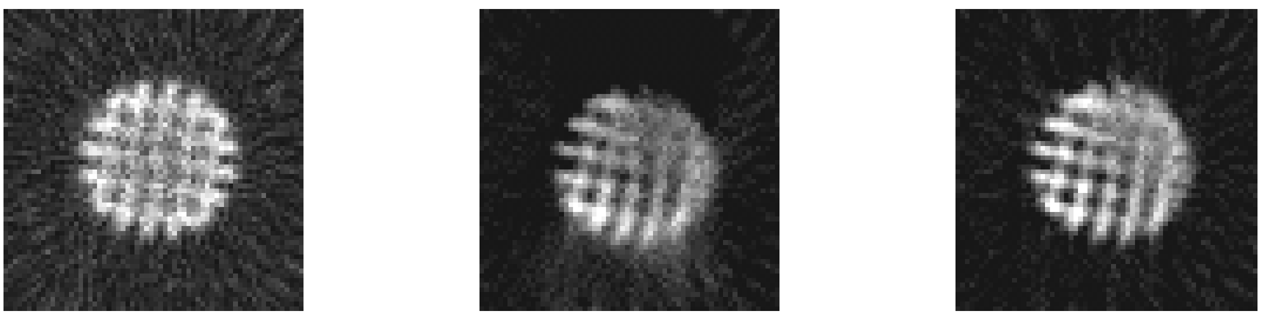
(a) (b) (c)

Legenda na pg. 196

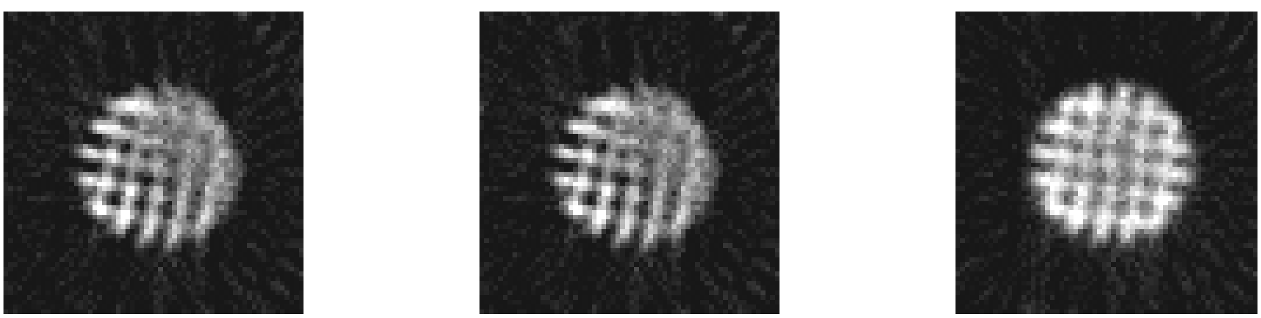


(d) (e) (f)

Figura 6.48 – (a) Corte reconstruído com programa *fbpw.m* e janela buttA (figura 6.40.d) . (b) Corte reconstruído com programa *fbpw.m* com buttA + *art.m* (figura 6.40.e). (c) Corte reconstruído com programa *fbpw.m* com buttA+ *artx.m* e *r (σ)* , σ = 0.25 . (d) Corte reconstruído com programa *fbpw.m* com buttA + *artx.m* e *r (σ)* , σ = 1 . (e) Corte reconstruído com programa *fbpw.m* com buttA + *artx.m* e *r (σ)* , σ = 4 . (f) Corte reconstruído com programa *fbpw.m* com buttA + *artx.m* e *r* =0.25.



(a) (b) (c)



(d) (e) (f)

Figura 6.49 – (a) Corte reconstruído com programa *fbpw.m* e janela buttA (figura 6.42.d) . (b) Corte reconstruído com programa *fbpw.m* com buttA + *art.m* (figura 6.42.e). (c) Corte reconstruído com programa *fbpw.m* com buttA+ *artx.m* e *r (σ)* , σ = 0.25 . (d) Corte reconstruído com programa *fbpw.m* com buttA + *artx.m* e *r (σ)* , σ = 1 . (e) Corte reconstruído com programa *fbpw.m* com buttA + *artx.m* e *r (σ)* , σ = 4 . (f) Corte reconstruído com programa *fbpw.m* com buttA + *artx.m* e *r* =0.25.

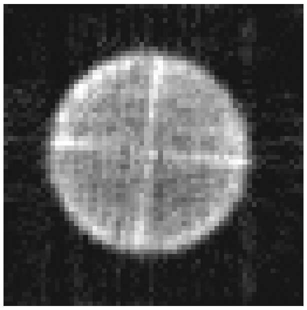
As figuras de 6.46 à 6.49 apresentam para comparação, além dos cortes reconstruídos com o coeficiente de relaxação fixo (r = 0.25) e $r = r(\sigma)$, os cortes reconstruídos com ***fbpw.m*** com janela buttA e ***art.m*** sem as alterações .

Observa-se no caso do algoritmo ART (figuras 6.46 e 6.47) que os cortes reconstruídos com $r = r(\sigma)$ apresentam melhores resultados que o reconstruído com r = 0.25, onde aparecem artefatos nas bordas da imagem. Dentre a imagens reconstruídas com $r = r(\sigma)$, as com $\sigma = 4$ apresentam melhor contraste, porém ainda artefatos (6.46.e) , as com $\sigma = 1$, apesar de não apresentarem artefatos ainda tem seu contraste degradado e, as com $\sigma = 0.25$, não apresentam um bom contraste.

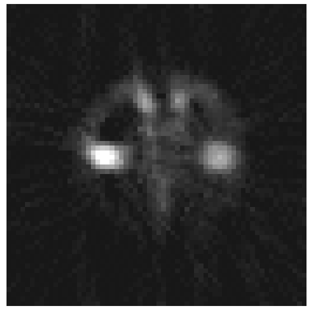
No caso do algoritmo híbrido (FBPW+ART), resultados apresentados nas figuras 6.48 e 6.49, os cortes reconstruídos com $r = r(\sigma)$ apresentam resultados piores que o reconstruído com r = 0.25, onde ainda aparecem artefatos nas imagens.

O comportamento acima descrito para os algoritmos explica-se pelo fato da imagem inicial das reconstruções ser diferente. No caso do algoritmo ART, a imagem inicial é dada pela equação 4.34, enquanto que no caso do híbrido pela imagem reconstruída pelo algoritmo FBP.

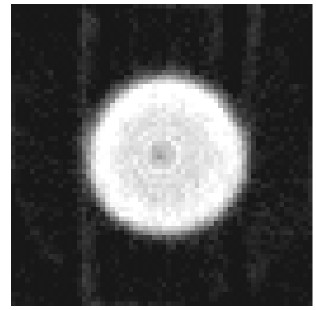
Além disso, verifica-se a necessidade de se testar um fator σ intermediário entre 1 e 4 para sanar os problemas ainda verificados com o algoritmo ART e confirmar o coeficiente de relaxação fixo como melhor opção para o algoritmo híbrido. Para tanto, reconstrói-se novamente os cortes das figuras 6.38, 6.40 e 6.42 com os algoritmos ART com $r = r(\sigma)$ para $\sigma = 2$ e híbrido com r = 0.25. Os resultados obtidos, apresentados na figura 6.50, confirmam as opções de modificação como corretas.



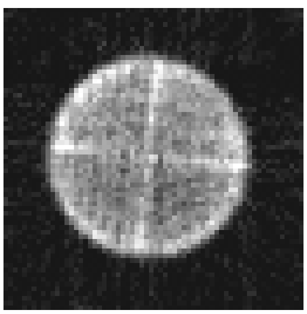
(a)



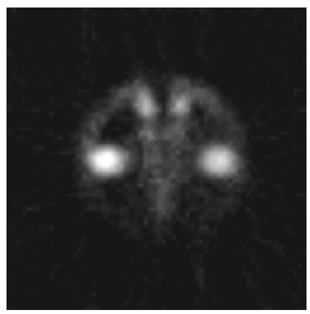
(b)



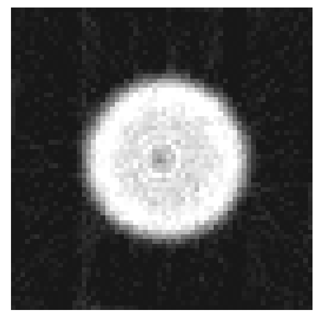
(c)



(d)



(e)



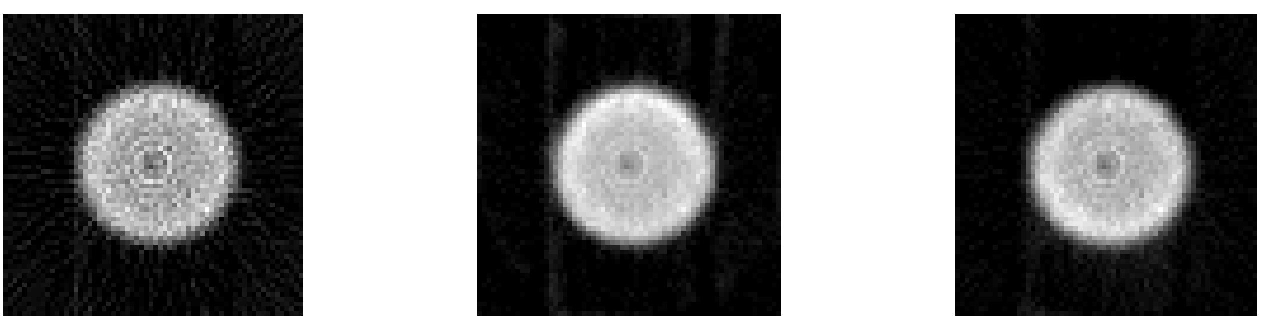
(f)

Figura 6.50 – (a) / (b) / (c) Cortes novamente reconstruídos com o programa *artx.m* e σ =2. (d) / (e) / (f) Cortes novamente reconstruídos com os programas *fbpw.m* com janela buttA + *artx.m* e *r* =0.25.

6.4.2– Estudo Quantitativo

Utilizando os cortes obtidos no ítem anterior com as correções necessárias para a eliminação de artefatos e os conceitos de Uniformidade tomográfica (equação 6.1) e Contraste (equação 6.2), faz-se uma análise quantitativa das imagens obtidas dos *phantoms* com os algoritmos. Realiza-se a análise das imagens sem qualquer filtragem pois, se deseja verificar a performance do algoritmo.

A análise da Uniformidade Tomográfica constará da verificação deste valor nas imagens dos cortes da linha 40 das projeções da figura 6.38.a , reconstruídos com os algoritmos FBPW apenas com filtro rampa (sem janela), ART e Híbrido, com as modificações descritas anteriormente, ou seja, com o programa ***artx.m***. Os cálculos seguem o exposto no ítem 6.1 deste Capítulo e, as imagens e os resultados, são apresentados a seguir – figuras 6.51 e 6.52 e Tabela 6.30, evidenciando uma melhor performance do algoritmo ART.



(a) (b) (c)

Figura 6.51 – (a) Corte da linha 40 das projeções de 6.38.a reconstruído com o programa *fbpw.m* . (b) Corte da linha 40 das projeções de 6.38.a reconstruído com o programa *artx.m* e *r* (σ) com σ = 2 . (c) Corte da linha 40 das projeções de 6.38.a reconstruído com os programas *fbpw.m* com janela buttA + *artx.m* e *r* =0.25 .

	X				Y				
Algoritmo	*Cmax*	*Cmin*	*Cmedia*	*U*	*Cmax*	*Cmin*	*Cmedia*	*U*	***Umedia***
FBPW	239,0	69,0	167,2	101,7	235,0	69,0	164,5	100,9	**101,3**
ART	231,0	125,0	191,2	55,4	252,0	93,0	188,2	84,5	**70,0**
HÍBRIDO	223,0	105,0	181,5	65,0	232,0	73,0	179,1	88,8	**76,9**

Tabela 6.30 – Valores de Uniformidade Tomográfica calculados para as imagens da figura 6.51.

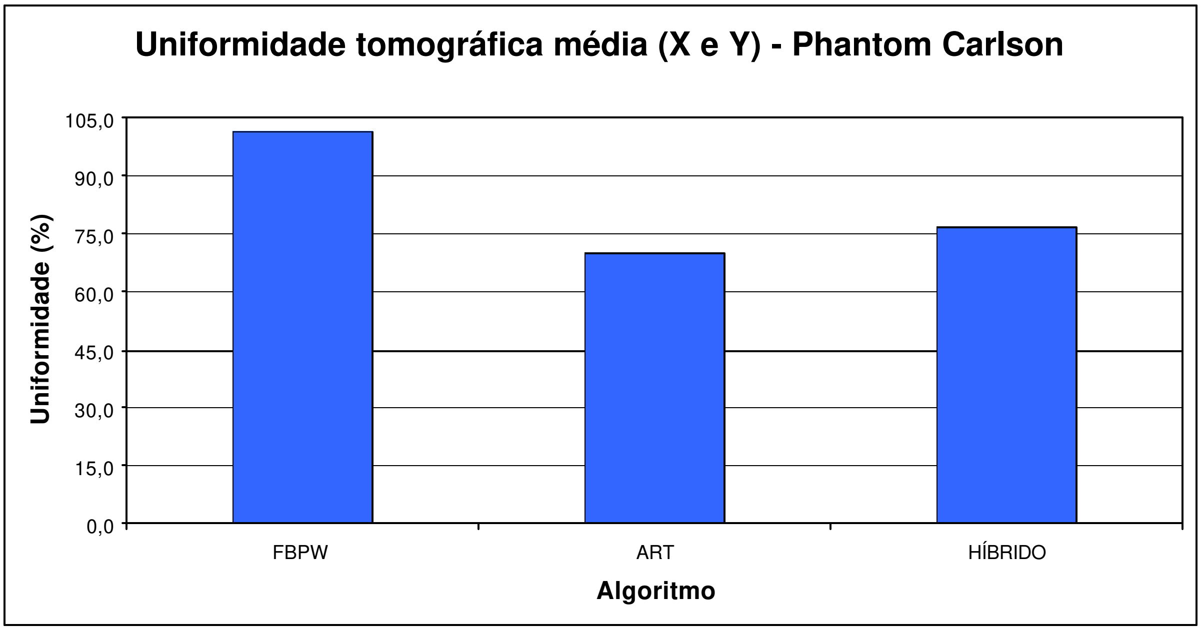
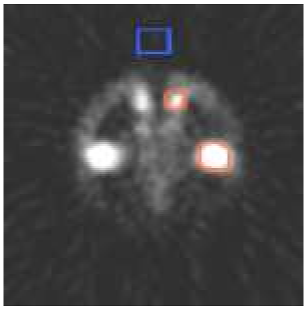
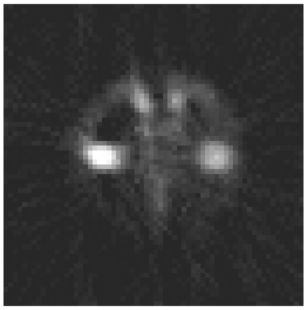
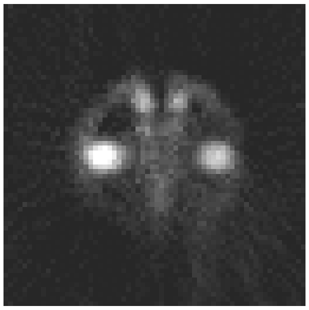


Figura 6.52 – Valores de Uniformidade Tomográfica apresentados na Tabela 6.30.

O contraste de objetos quentes é analisado a partir das reconstruções da linha 10 das projeções da figura 6.38.b para os mesmos algoritmos anteriormente citados. Os cálculos seguem o exposto no ítem 6.2 deste Capítulo e os objetos analisados, as imagens reconstruídas e os resultados são apresentados a seguir nas figuras 6.53 e 6.54 e Tabela 6.31, evidenciando uma melhor performance do algoritmo Híbrido.

(a) (b)

(c) (d)

Figura 6.53 – (a) ROIs (*Regions of Interest*) sobre o *insert* de lesões quentes – do alto para baixo : região fria (BG) , furo de 4 *pixels* de diâmetro, furo de 7 *pixels* de diâmetro. (b) Corte da linha 10 das projeções de 6.38.b reconstruído com o programa *fbpw.m* . (c) Corte da linha 10 das projeções de 6.38.b reconstruído com o programa *artx.m* e *r* (σ) com σ = 2 . (d)) Corte da linha 10 das projeções de 6.38.b reconstruído com os programas *fbpw.m* com janela buttA + *artx.m* e *r* =0.25.

Algoritmo	*F7*	*F4*	*BG*	***Contraste furo de diâmetro 7 pxs (%)***	***Contraste furo de diâmetro 4 pxs (%)***
FBPW	175,1	133,8	3,4	**96,2**	**95,0**
ART	101,5	85,7	1,5	**97,0**	**96,5**
HÍBRIDO	122,7	92,3	0,2	**99,7**	**99,6**

Tabela 6.31 – Valores de Contraste calculados para as imagens da figura 6.52.

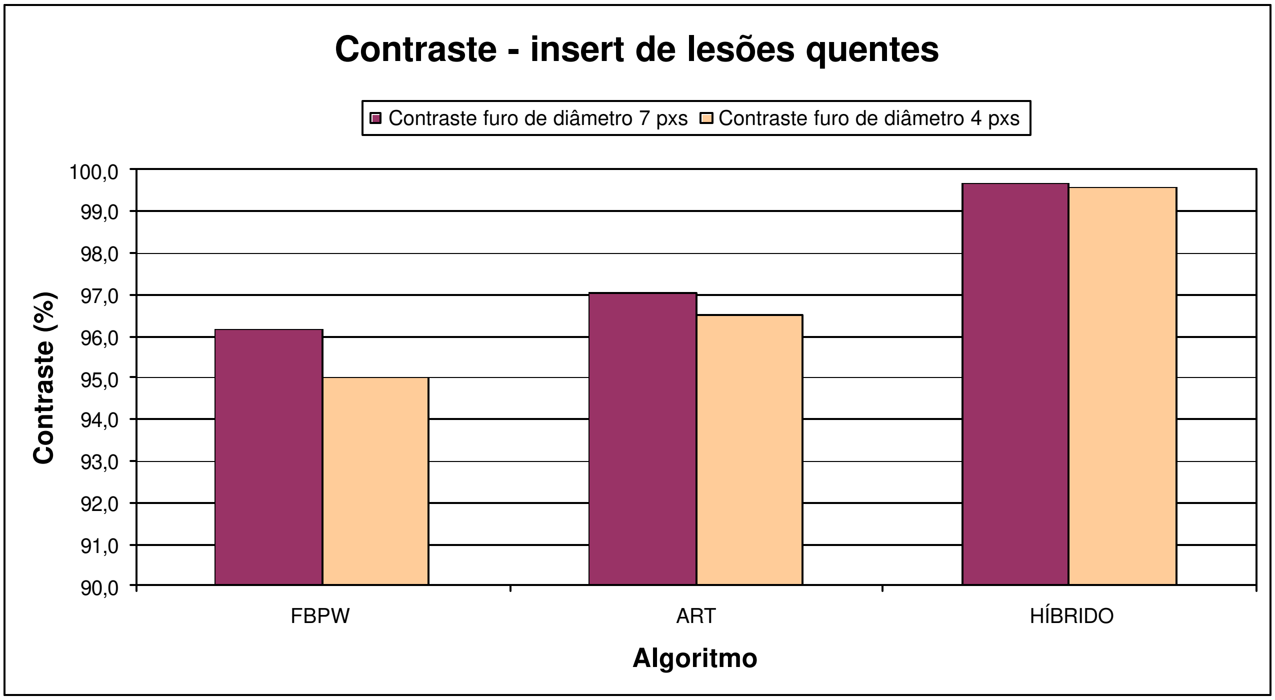


Figura 6.54 – Valores de Contraste apresentados na Tabela 6.31.

O contraste de objetos frios é analisado a partir das reconstruções da linha 27 das projeções da figura 6.55 para os mesmos algoritmos anteriormente citados. Os objetos são esferas frias (figura 6.56.a) inseridas em um meio quente (solução de água com ^{99m}Tc) em um *phantom* Jaszczak como da figura 6.38.a. Os cálculos seguem o exposto no ítem 6.2 deste Capítulo e os objetos analisados, as imagens reconstruídas e os resultados são apresentados a seguir nas figuras 6.56 e 6.57 e Tabela 6.32, evidenciando uma melhor performance do algoritmo FBPW.

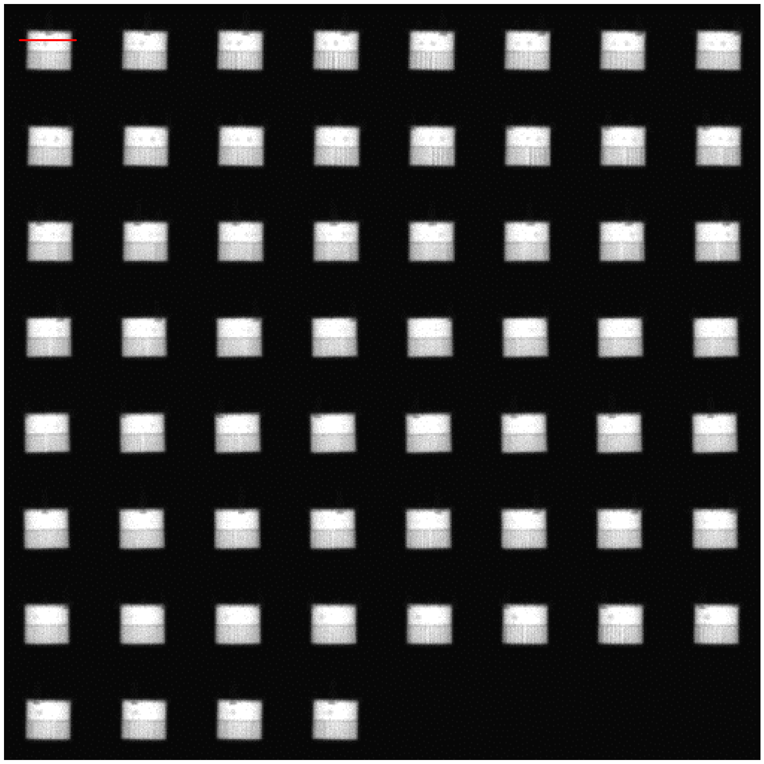
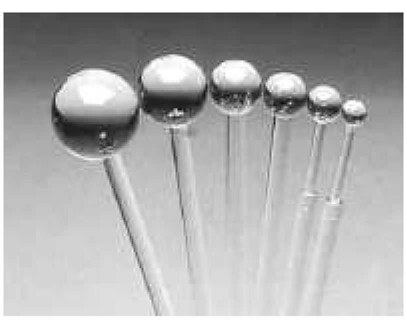
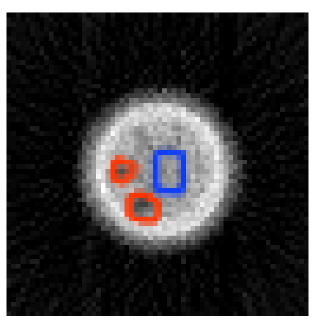


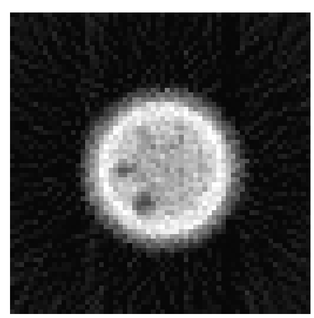
Figura 6.55 – Projeções para reconstruir os objetos frios da figura 6.56.a



(a)

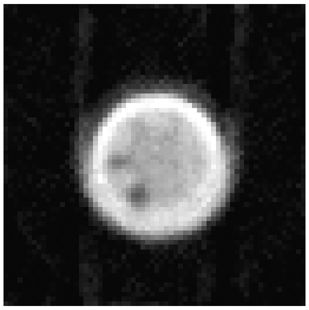


(b)

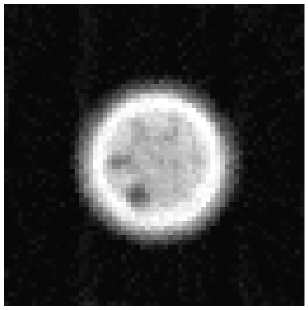


(c)

Legenda na pg. 204



(d)



(e)

Figura 6.56 – (a) Insetos – esferas frias (b) ROIs (*Regions of Interest*) sobre o *insert* – do alto para baixo : região fria (BG) , esfera de 4 *pixels* de diâmetro, esfera de 5 *pixels* de diâmetro. (c) Corte da linha 27 das projeções de 6.38.b reconstruído com o programa *fbpw.m* . (c) Corte da linha 10 das projeções de 6.38.b reconstruído com o programa *artx.m* e *r* (σ) com σ = 2 . (d)) Corte da linha 10 das projeções de 6.38.b reconstruído com os programas *fbpw.m* com janela buttA + *artx.m* e *r* =0.25.

Algoritmo	*E5*	*E4*	*BG*	***Contraste Esfera de diâmetro 5 pxs (%)***	***Contraste Esfera de diâmetro 4 pxs (%)***
FBPW	101,8	115,1	151,1	**19,5**	**13,5**
ART C/ FILTRO	110,6	130,5	153,4	**16,2**	**8,1**
HÍBRIDO	111,0	134,3	161,1	**18,4**	**9,1**

Tabela 6.32 – Valores de Contraste calculados para as imagens da figura 6.56.

Com os resultados dos cálculos de Uniformidade Tomográfica e Contrastes de objetos quentes e frios de reconstruções dos cortes de *phantoms* matemáticos e de *phantoms* apresentados neste Capítulo, já se poderia reconhecer um padrão do comportamento dos algoritmos de reconstrução. Porém, deve-se reconstruir as imagens clínicas de procedimentos de baixa estatística antes de se concluir tal padrão, objeto principal deste trabalho.

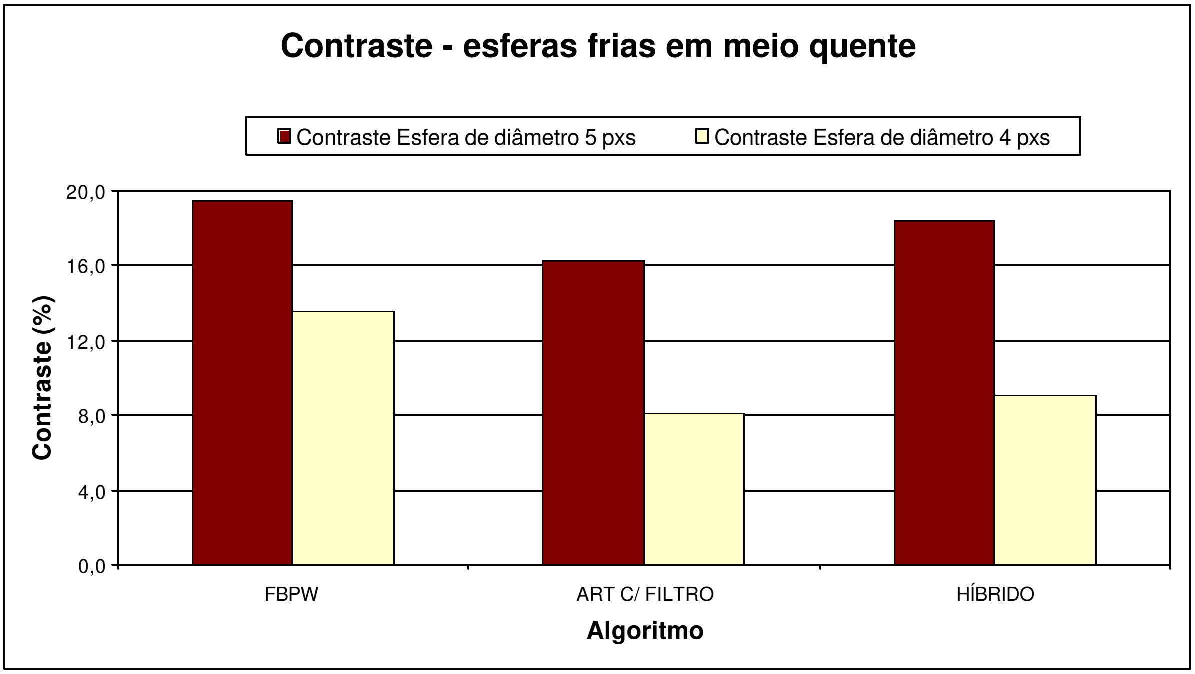


Figura 6.57 – Valores de Contraste apresentados na Tabela 6.32.

Capítulo 7 – Imagens Clínicas

Para encerrar a análise da performance dos algoritmos, faz-se a reconstrução de cortes tomográficos de estudos de ^{99m}Tc – MIBI, FDG (^{18}F) e ^{67}Ga. A comparação dos resultados das performances se dá através da verificação de perfis e cálculos de contrastes de estruturas nas imagens (equação 6.2), comparando-se os resultados, inclusive com os obtidos pelo algoritmo disponível nos equipamentos do SMN do HC–Unicamp, o qual gera cortes transaxiais, coronais e sagitais, apresentado-os em uma imagem denominada *report*. As imagens dos cortes transaxiais dos *reports* servirão como calibre quando comparadas com as reconstruídas com o algoritmo FBPW, pois, o método de reconstrução utilizado nestes equipamentos é o FBP.

Além disso, avalia-se as imagens com a aplicação de filtros pós processamento (ART e Híbrido), janelas (FBPW) e limiar[1] para a atenuação de ruídos, recursos utilizados no processamento cotidiano de serviços de Medicina Nuclear.

7.1 – ^{99m}Tc - MIBI

As imagens obtidas neste estudo utilizam 30 projeções das 31 adquiridas durante o exame (figura 7.1) para reconstruir os cortes tomográficos do coração. A primeira e última projeções são especulares, sendo portanto redundantes.

Uma das particularidades deste estudo é que as projeções são reorientadas segundo o eixo maior do ventrículo esquerdo para que a cavidade interna do órgão possa ser observada de um ponto perpendicular ao seu plano. Esta operação gera cortes oblíquos nos *reports* em contraponto aos transaxiais obtidos pelos algoritmos

[1] Técnica que modifica os valores associados a pixels abaixo ou acima de um valor limitante. Normalmente o novo valor associado é o próprio valor limitante.

implementados neste trabalho. A reorientação impossibilita a comparação direta dos cortes obtidos com os gerados nos *reports*.

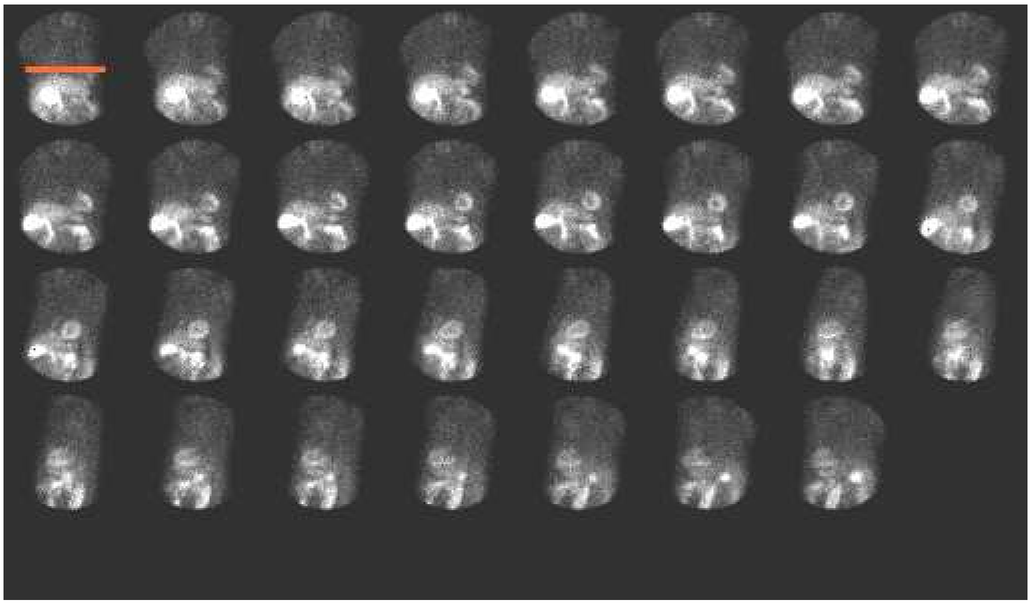


Figura 7.1 – Projeções de estudo MIBI (31) para obtenção de cortes tomográficos do coração. A linha vermelha indica a região das linhas 35 e 37 que geraram os cortes reconstruídos.

Às imagens reconstruídas com FBPW aplica-se a janela apresentada na figura 7.2 e nas imagens com o algoritmo Híbrido um filtro de pós processamento de borramento com formato Gaussiano para 2 *pixels* fornecido pelo *software* comercial *Corel Draw* 7.0. Os cortes reconstruídos (figura 7.3 e 7.4) contém apenas as imagens e resultados obtidos com os algoritmos implementados neste trabalho, já que os cortes dos *reports* não possuem a mesma orientação. Nota-se também que, como o número de projeções é insuficiente (M < N – caso discutido no Capítulo 4) as imagens obtidas com o algoritmo ART não convergem, não atingindo o valor de α para o Desvio padrão das projeções calculadas menor ou igual a 0.01, inclusive sendo necessária a interrupção quando da iteração de número 37.

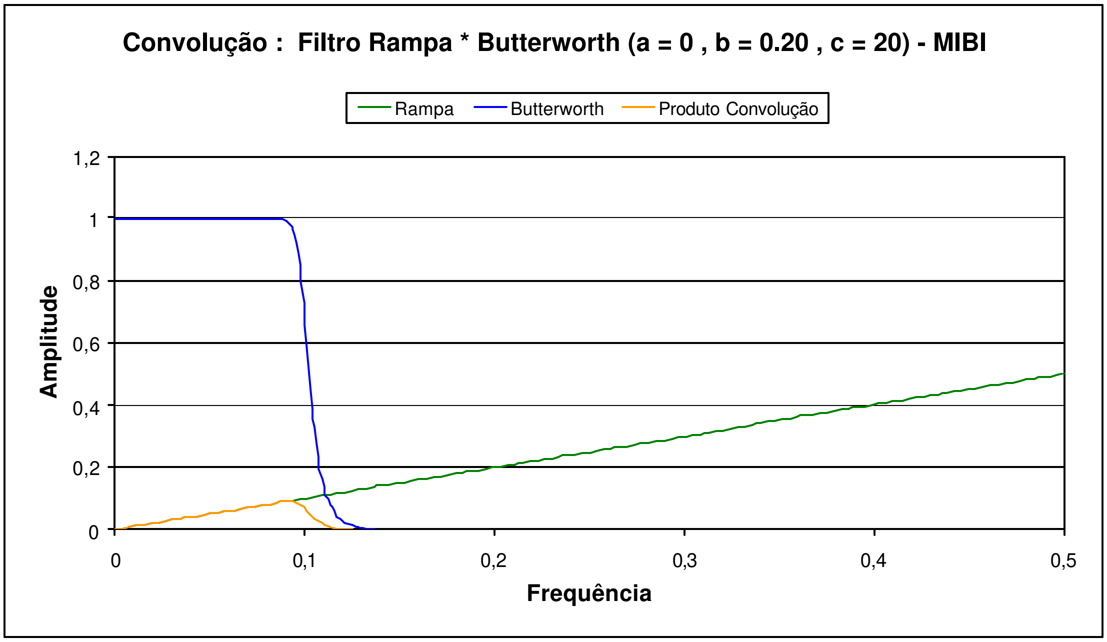


Figura 7.2 – Janela utilizada para a reconstrução dos cortes de estudo MIBI.

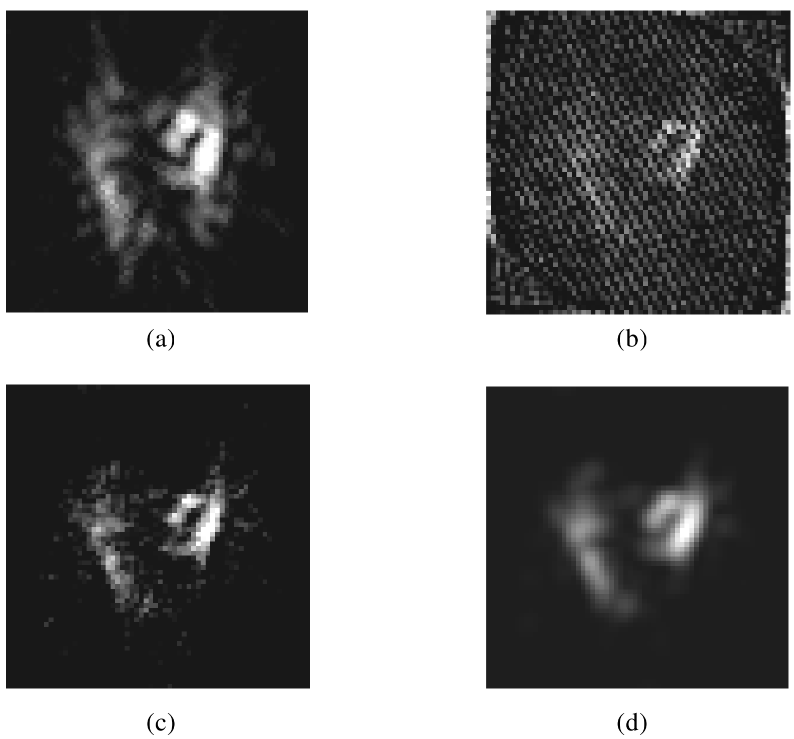


Figura 7.3 – (a) Corte da linha 35 reconstruído com *fbpw.m* + janela. (b) Corte da linha 35 reconstruído com *artx.m* com *r*(σ) para σ =2. (c) Corte da linha 35 reconstruído com híbrido *fbpw.m* + janela + artx.m com r =0.25 . (d) Imagem 7.3.c com aplicação de filtro pós processamento.

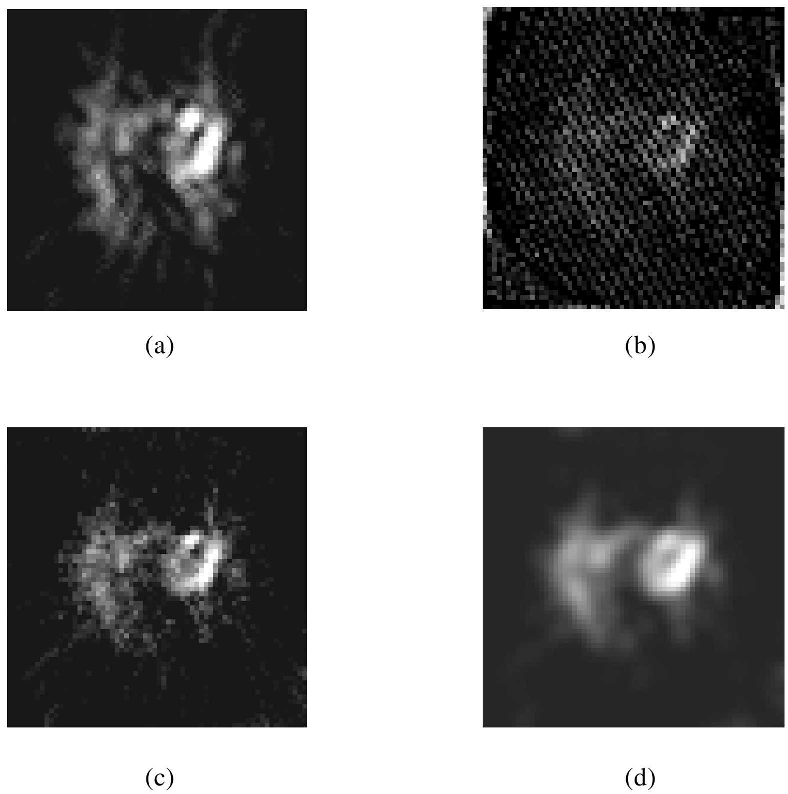
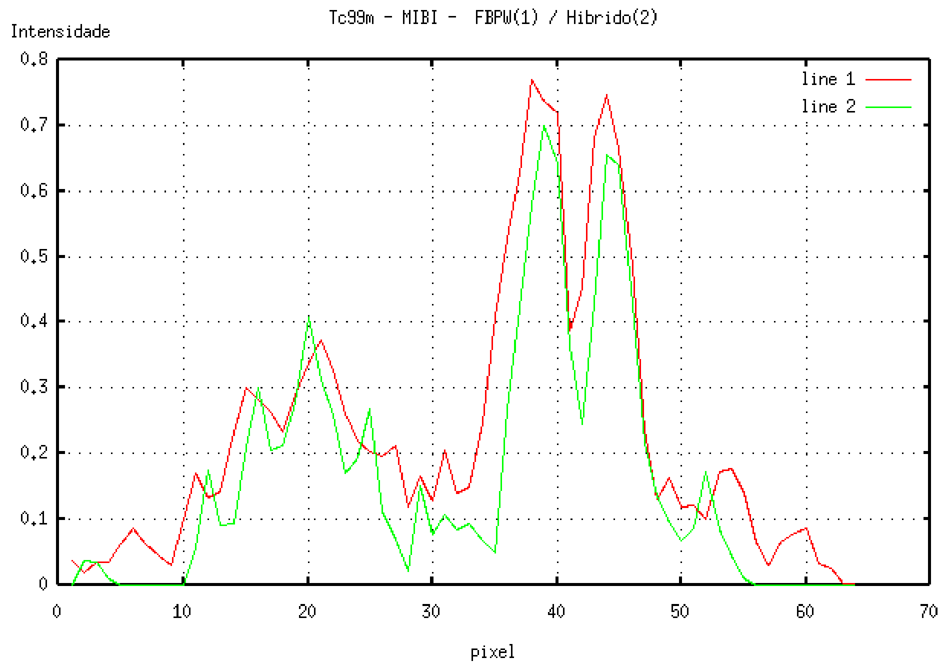
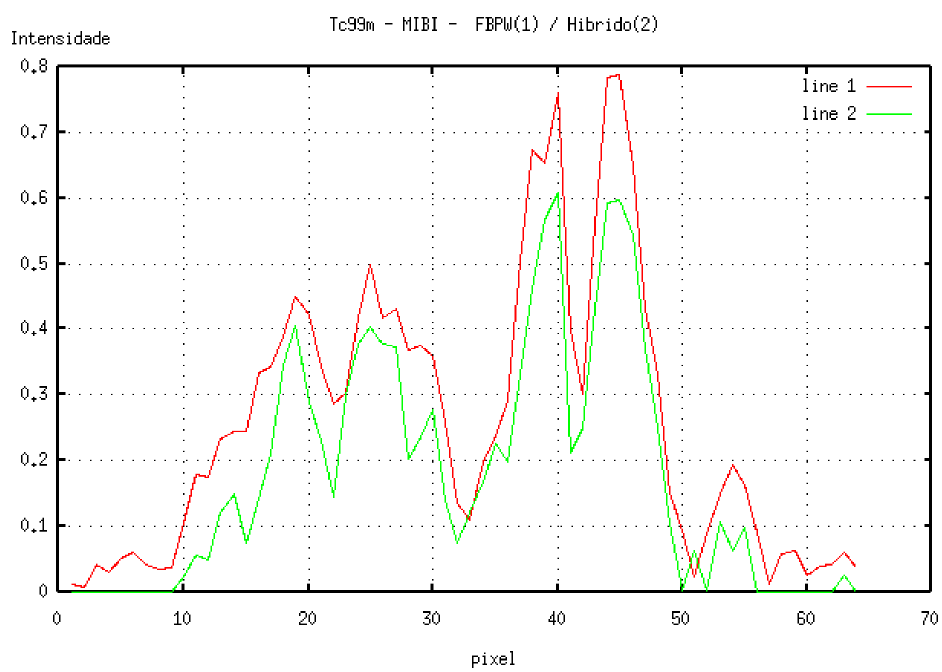


Figura 7.4 – (a) Corte da linha 37 reconstruído com *fbpw.m* + janela. (b) Corte da linha 37 reconstruído com *artx.m* com *r*(σ) para σ =2. (c) Corte da linha 37 reconstruído com híbrido *fbpw.m* + janela + *artx.m* com r =0.25 . (d) Imagem 7.4.c com aplicação de filtro pós processamento.

Para análise quantitativa dos cortes do estudo MIBI apresentados, limita-se ao levantamento do perfil da coluna 40 dos cortes 7.3.a , 7.3.c e 7.4.a , 7.4.c, verificando-se a coerência entra as reconstruções.



(a)



(b)

Figura 7.5 – (a) Perfis das colunas 40 das imagens 7.3.a e 7.3.c. (b) Perfis das colunas 40 das imagens 7.4.a e 7.4.c.

7.2 – FDG

Foram reconstruídos dois cortes de dois estudos com FDG (tórax superior e pélvis), tanto para as imagens com ^{18}F como para ^{99m}Tc [2] , perfazendo um total de oito cortes tomográficos.

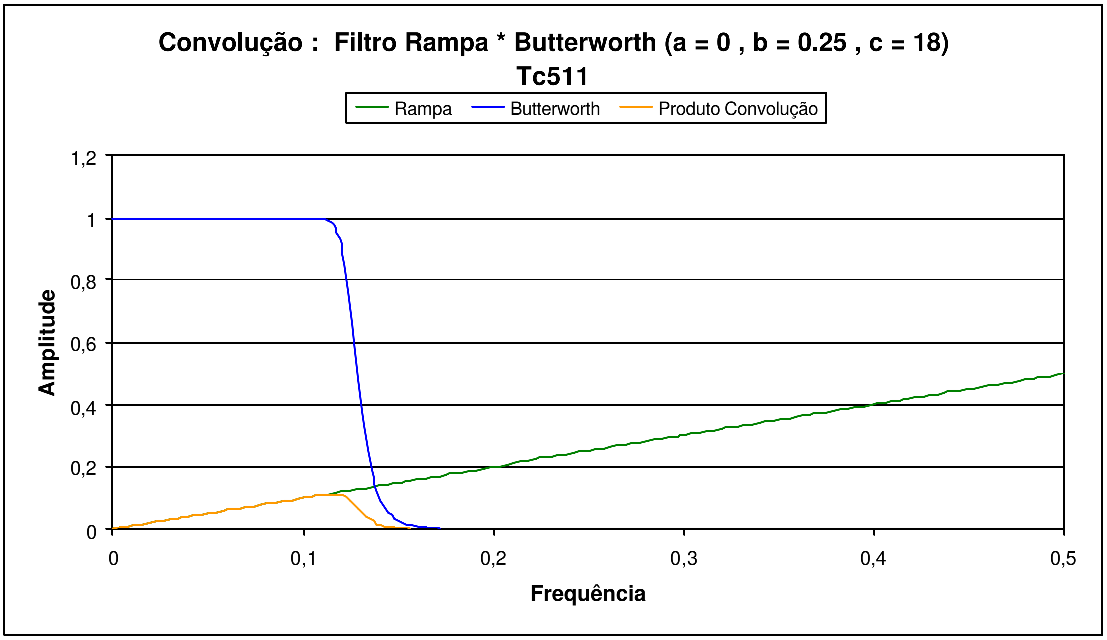
Os estudos apresentados nos *reports* (figuras 7.7 e 7.9) mostram cortes do tórax superior de um paciente, tanto para Tc511 como para ^{18}F. Por questões éticas, o nome e número de identificação no sistema de informações do HC – Unicamp foram omitidos.

A partir destas reconstruções não mais serão mostradas as projeções adquiridas pois, o *report* trás a projeção para $\theta = 0º$, sendo o suficiente para a verificação da linha de corte reconstruída.

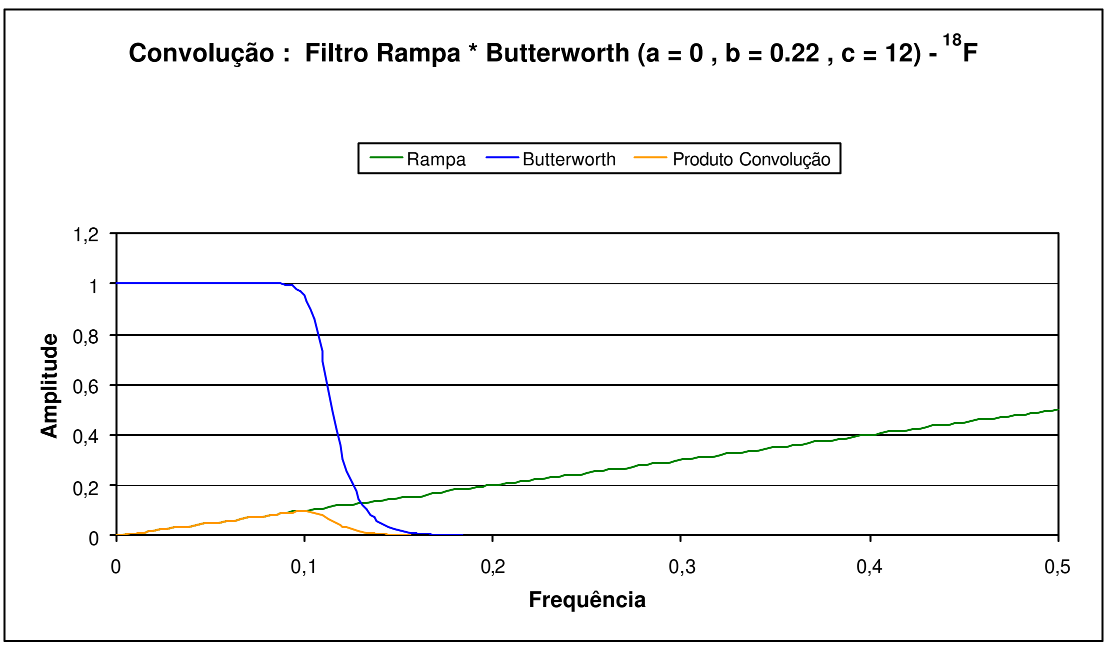
Inicia-e a avaliação dos cortes reconstruídos e os correspondentes nos *reports* para os corte 20 e 29 para o Tc511 e 20 e 27 para o ^{18}F, apresentados nas figuras 7.8, 7.9, 7.11 e 7.12, respectivamente. Os cortes com o algoritmo FBPW utilizam as janelas da figura 7.6.a para o Tc511 e 7.6.b para o ^{18}F. Além disso, nestas imagens o filtro pós processamento para os algoritmos ART e híbrido é o mesmo utilizado no caso do estudo MIBI .

Os cortes 29 – Tc511 , 27 – ^{18}F e os correspondentes reconstruídos com os algoritmos implementados, são avaliados em termos quantitativos com o levantamento dos perfis, onde as imagens feitas com os algoritmos iterativos não utilizaram os filtros pós processamento e, no caso do FBPW, mantém-e as janelas da figura 7.6, pois as imagens dos *reports* foram processadas com janelas. Além disso, algumas estruturas têm seu contraste avaliado, sendo os resultados apresentados nas figuras a seguir das reconstruções.

[2] Devido aos parâmetros utilizados na câmera cintiladora para a aquisição das projeções, chama–se esta aquisição de Tc de alta energia ou Tc511, sendo apenas uma convenção de nomenclatura.



(a)



(b)

Figura 7.6 – (a) Janela utilizada para a reconstrução dos cortes de estudo Tc511. (b) Janela utilizada para a reconstrução dos cortes de estudo ^{18}F.

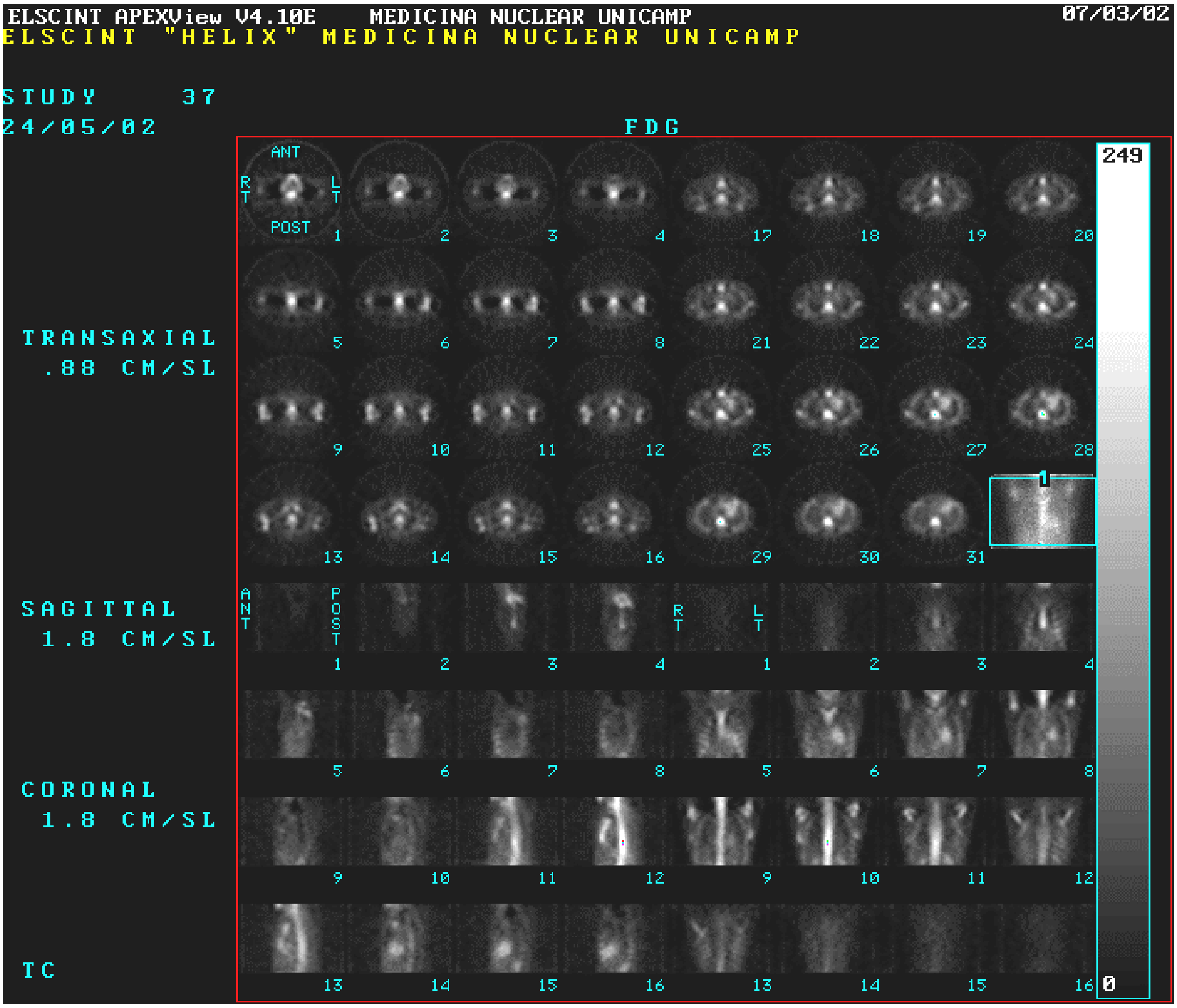
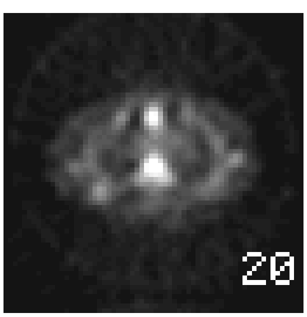
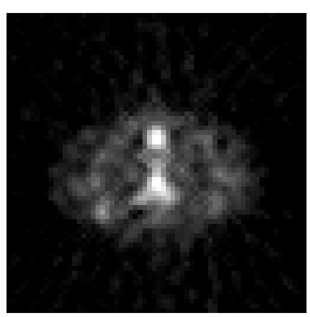


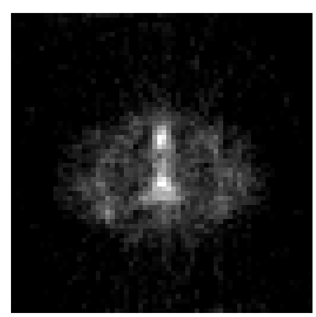
Figura 7.7 – *Report* – estudo FDG / Tc511 tórax .



(a)

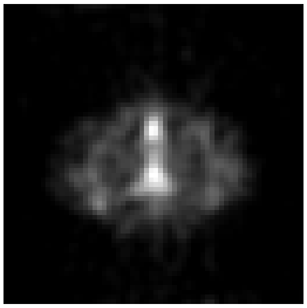
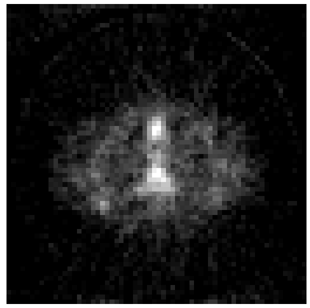
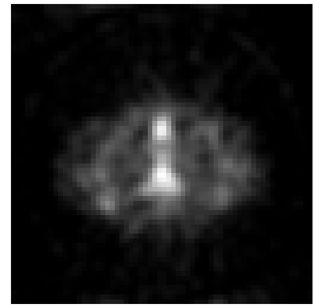


(b)



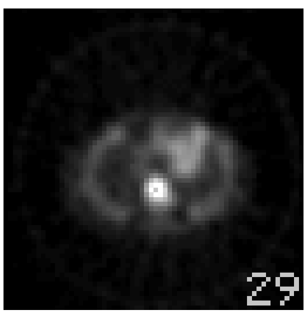
(c)

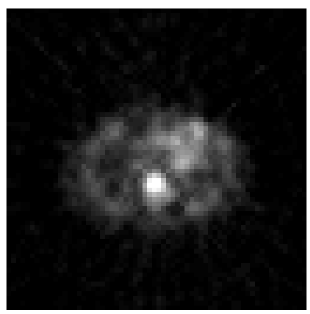
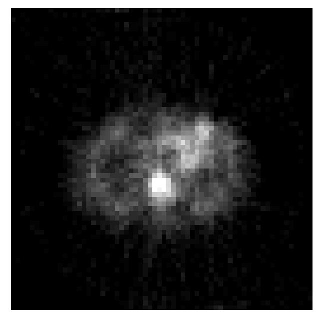
Legenda na pg. 214

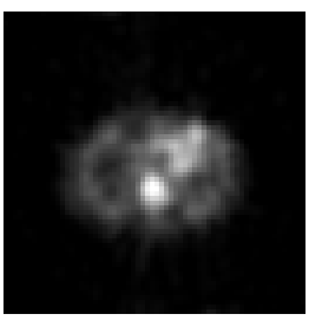
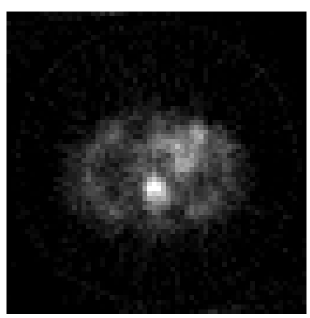
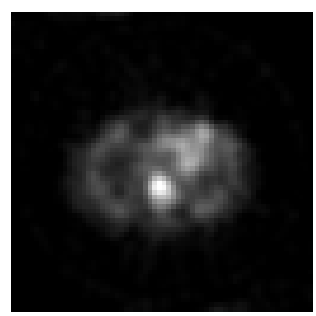
(d) (e) (f)

Figura 7.8 – (a) Corte 20 – *report* figura 7.7. (b) Reconstrução com *fbpw.m* + janela da figura 7.6.a. (c) Reconstrução com *artx.m* com *r (σ)* para σ =2. (d) Imagem 7.8.c com filtro pós processamento. (e) Reconstrução com híbrido com σ =0.25. (f) Imagem 7.8.e com filtro pós processamento.



(a) (b) (c)

(d) (e) (f)

Figura 7.9 – (a) Corte 29 – *report* figura 7.7. (b) Reconstrução com *fbpw.m* + janela da figura 7.6.a. (c) Reconstrução com *artx.m* com *r (σ)* para σ =2. (d) Imagem 7.9.c com filtro pós processamento. (e) Reconstrução com híbrido com σ =0.25. (f) Imagem 7.9.e com filtro pós processamento.

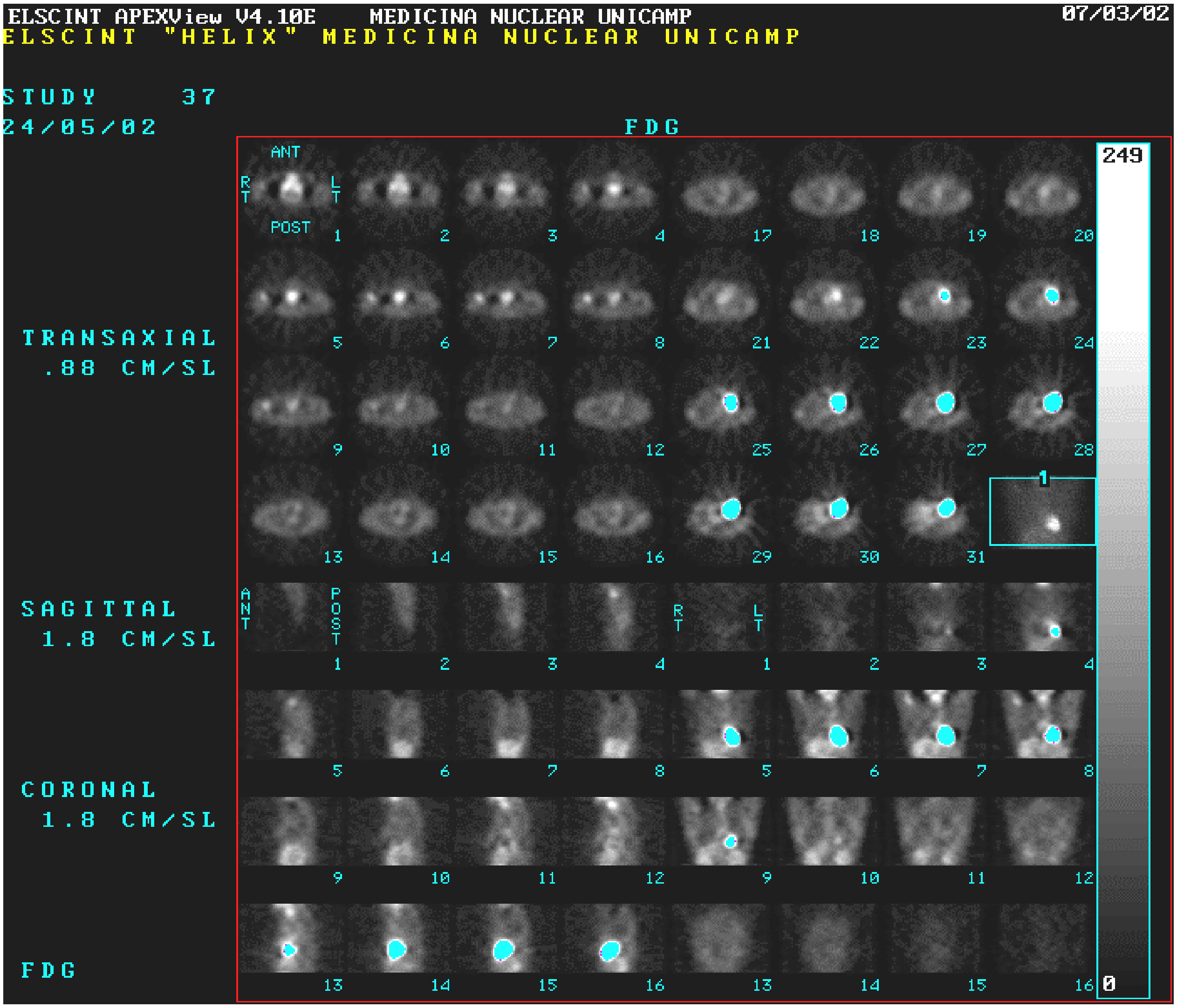
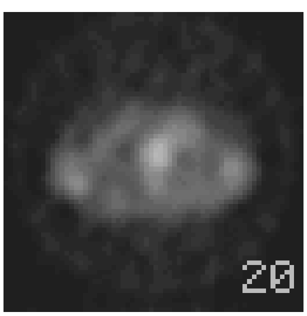
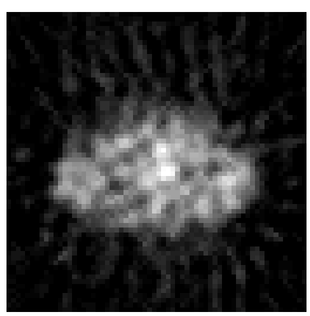


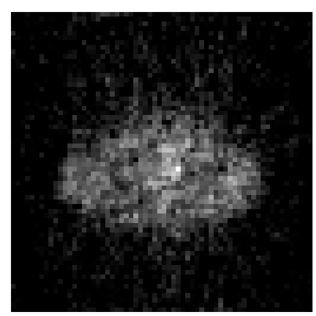
Figura 7.10 – *Report* – estudo FDG / ^{18}F tórax .



(a)

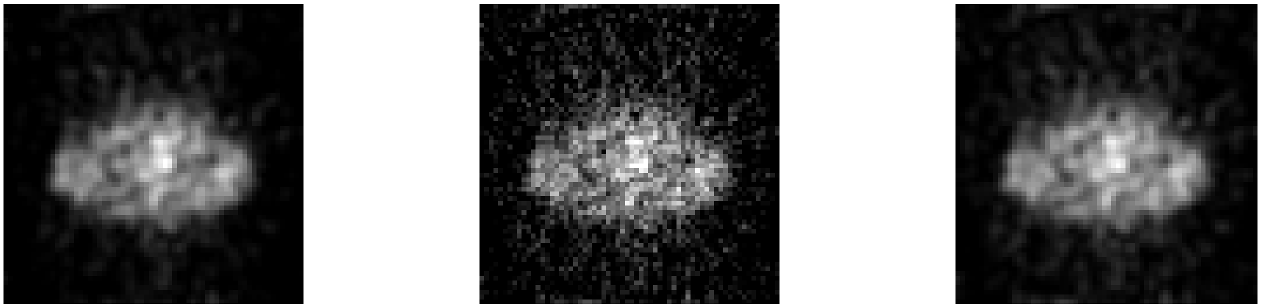


(b)



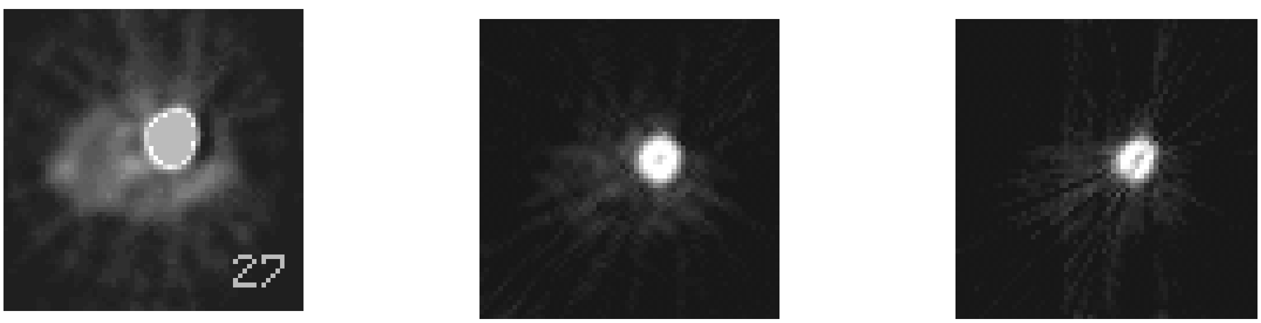
(c)

Legenda na pg. 216

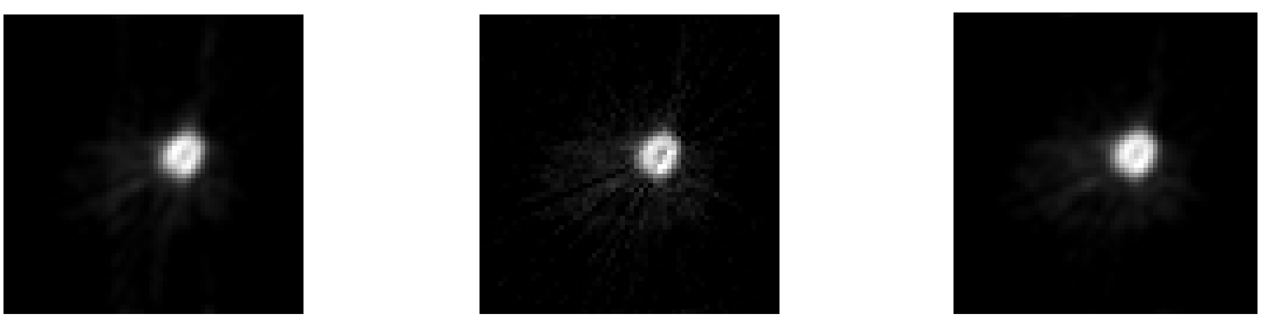


(d) (e) (f)

Figura 7.11 – (a) Corte 20 – *report* figura 7.10. (b) Reconstrução com *fbpw.m* + janela da figura 7.6.a. (c) Reconstrução com *artx.m* com *r (σ)* para σ =2. (d) Imagem 7.11.c com filtro pós processamento. (e) Reconstrução com híbrido com σ =0.25. (f) Imagem 7.11.e com filtro pós processamento.

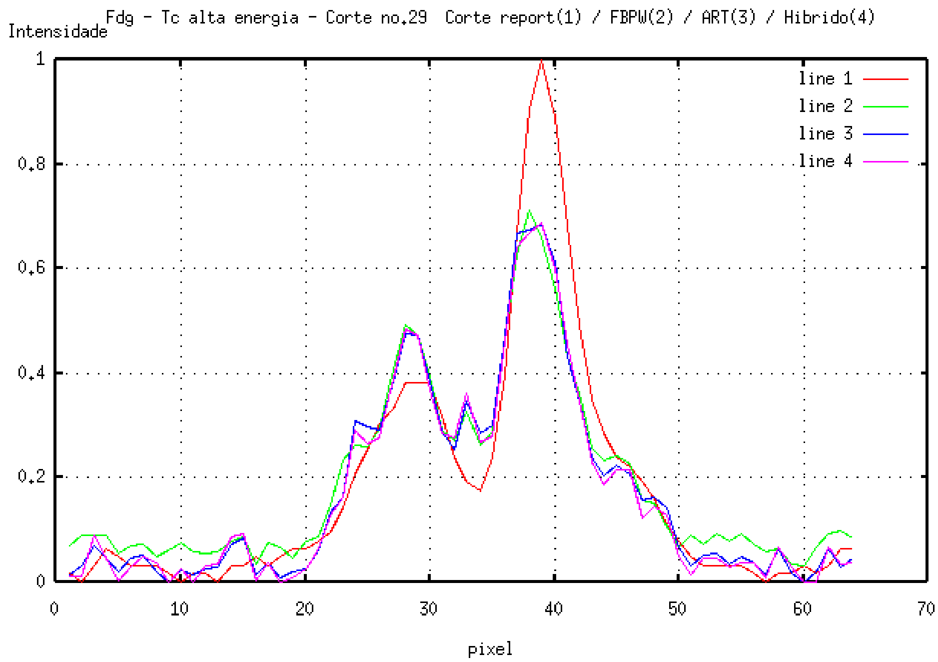


(a) (b) (c)

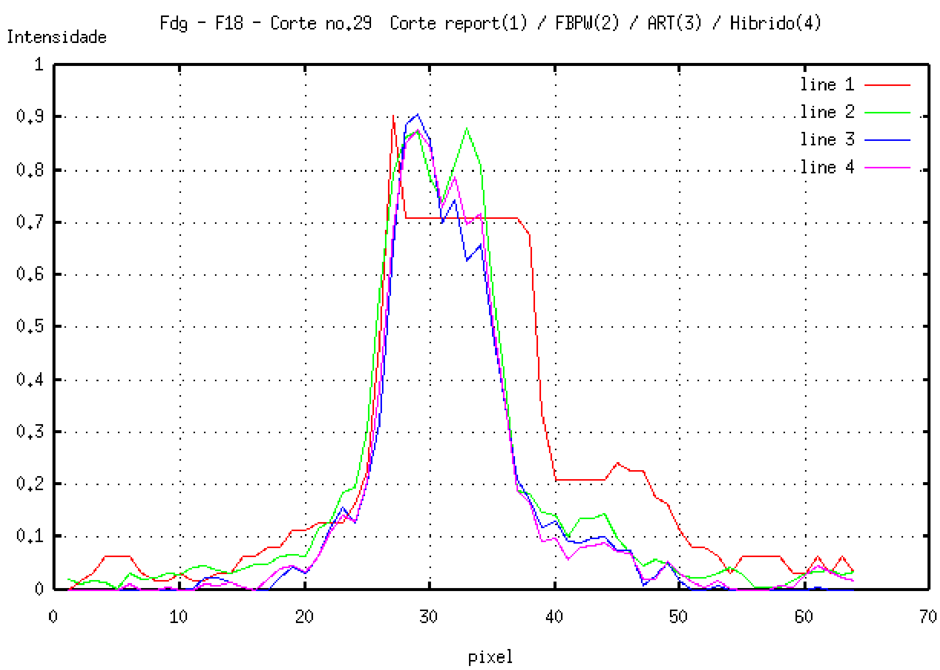


(d) (e) (f)

Figura 7.12 – (a) Corte 27 – *report* figura 7.10. (b) Reconstrução com *fbpw.m* + janela da figura 7.6.a. (c) Reconstrução com *artx.m* com *r (σ)* para σ =2. (d) Imagem 7.12.c com filtro pós processamento. (e) Reconstrução com híbrido com σ =0.25. (f) Imagem 7.12.e com filtro pós processamento.

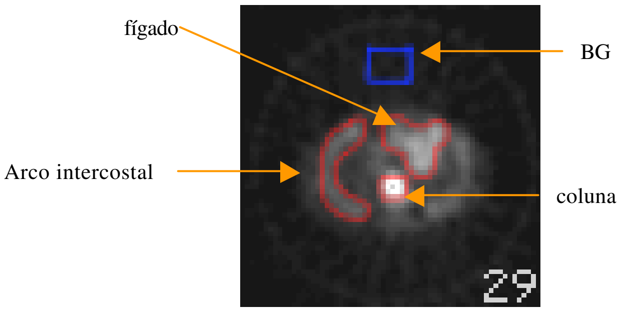


(a)

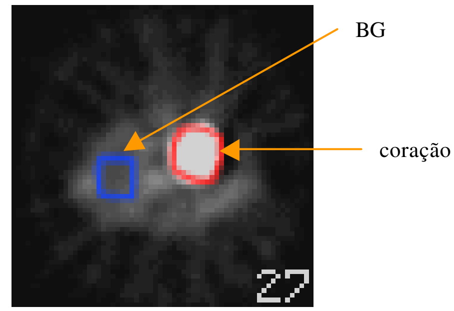


(b)

Figura 7.13 – (a) Perfil da coluna 34 do corte 29 da figura 7.9. (b) Perfil da coluna 38 do corte 27 da figura 7.12.



(a)



(b)

Figura 7.14 – (a) ROIs - Regiões quentes (vermelho) e fria (azul) utilizadas para o cálculo do contraste do corte 29 – *report* da figura 7.7. (b) ROIs - Regiões quente (vermelho) e fria (azul) utilizadas para o cálculo do contraste do corte 27 – *report* da figura 7.10.

	Corte 29						
	Valor médio				Contraste (%)		
Algoritmo	*Coluna*	*Fígado*	*Arco intercostal*	*BG*	*Coluna*	*Fígado*	*Arco intercostal*
REPORT	195,7	96,0	56,1	36,3	68,7	45,2	21,5
FBPW	195,4	111,3	68,5	40,1	65,9	47,0	26,1
ART	213,7	127,1	55,8	54,9	59,1	39,6	0,7
ART C/ FILTRO	208,6	128,8	61,5	49,3	61,8	44,6	11,0
HIBRIDO	195,4	111,3	55,8	40,1	65,9	47,0	16,3
HIBRIDO C/ FILTRO	194,3	114,9	56,3	42,6	64,1	45,9	13,9

Tabela 7.1 – Contrastes calculados para as estruturas do corte 29 da figura 7.14.a.

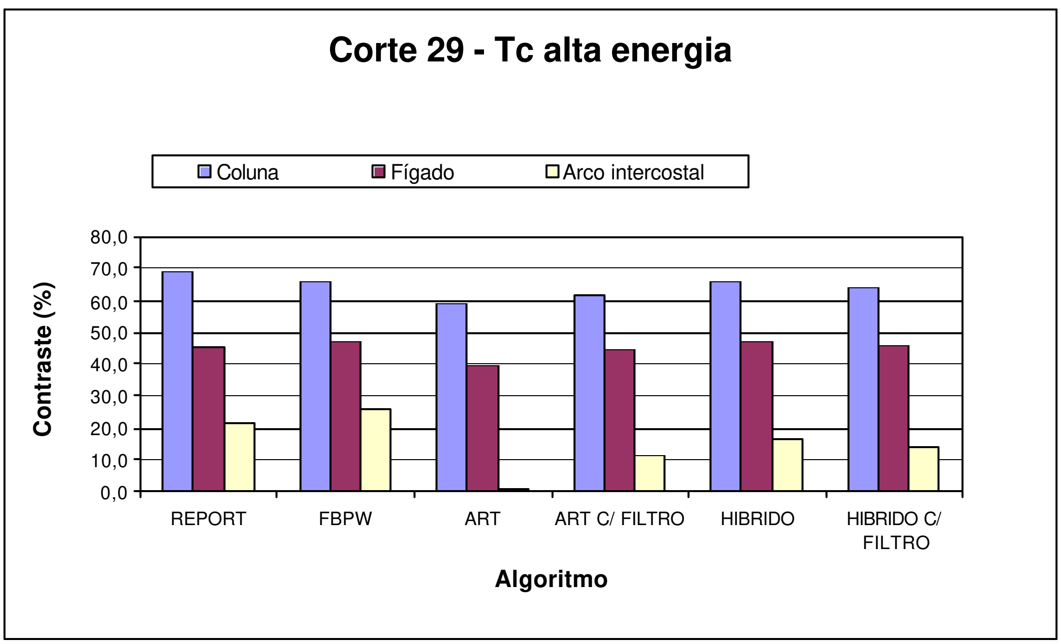


Figura 7.15 – Valores de contraste da Tabela 7.1.

	Corte 27		
	Valor médio		Contraste (%)
Algoritmo	*Coração*	*Cavidade*	*Coração / Cavidade*
REPORT	194,9	94,9	34,5
FBPW	201,0	19,9	82,0
ART	170,2	14,4	84,4
ART C/ FILTRO	188,2	14,9	85,3
HIBRIDO	176,9	15,9	83,5
HIBRIDO C/ FILTRO	194,9	16,0	84,8

Tabela 7.2 – Contrastes calculados para as estruturas do corte 27 da figura 7.14.b.

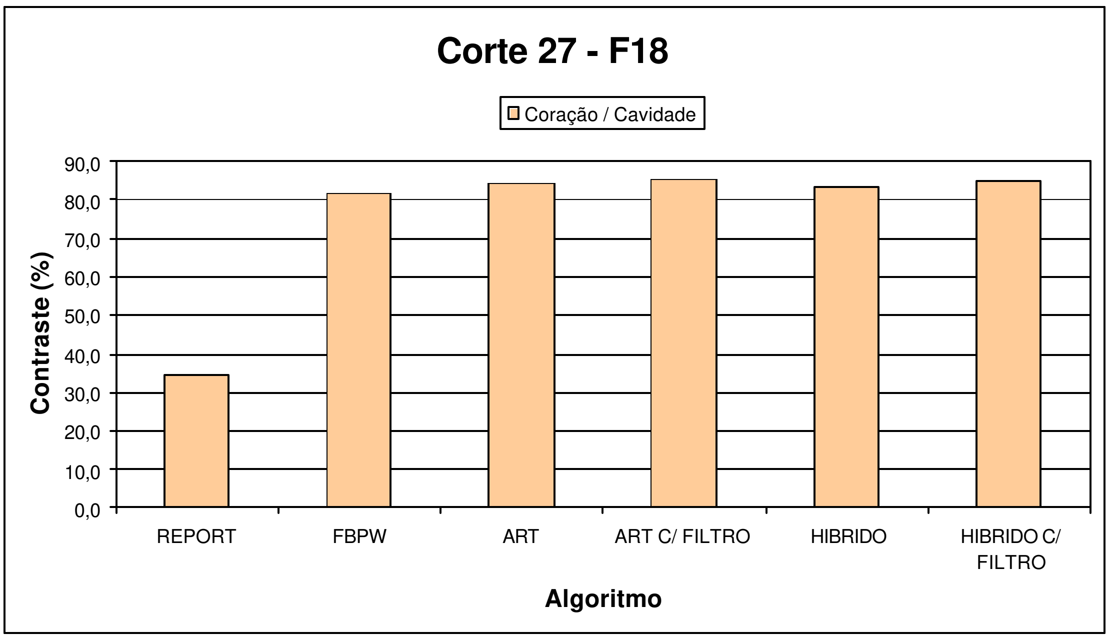


Figura 7.16 – Valores de contraste da Tabela 7.2.

Dentre os resultados quantitativos obtidos, verifica-se que o resultados obtidos entre os programas ***fbpw.m*** e o *software* dos equipamentos do SMN - HC Unicamp são equivalentes para o corte 29, o que não se pode avaliar para o corte 27. Devido a um recurso do *software* dos equipamentos do SMN - HC, a aplicação de um limiar automático às vezes ocorre um *overflow* dos valores dos *pixels*, o que se observa no perfil da figura 7.13.b. Também se verifica uma melhor performance dos algoritmos

FBP para estruturas menos captantes (arco intercostal – figura 7.15), não ocorrendo diferenças significativas entre os algoritmos para estruturas mais quentes.

O segundo estudo apresentado trata de um exame de pélvis, onde novamente dois cortes por radioisótopo foram reconstruídos e comparados com os obtidos em *report*. Para a apresentação dos resultados segue-se a mesma seqüência apresentada para o estudo de tórax, novamente omitindo-se os dados do paciente e utilizando-se as mesmas janelas e filtros pós processamento.

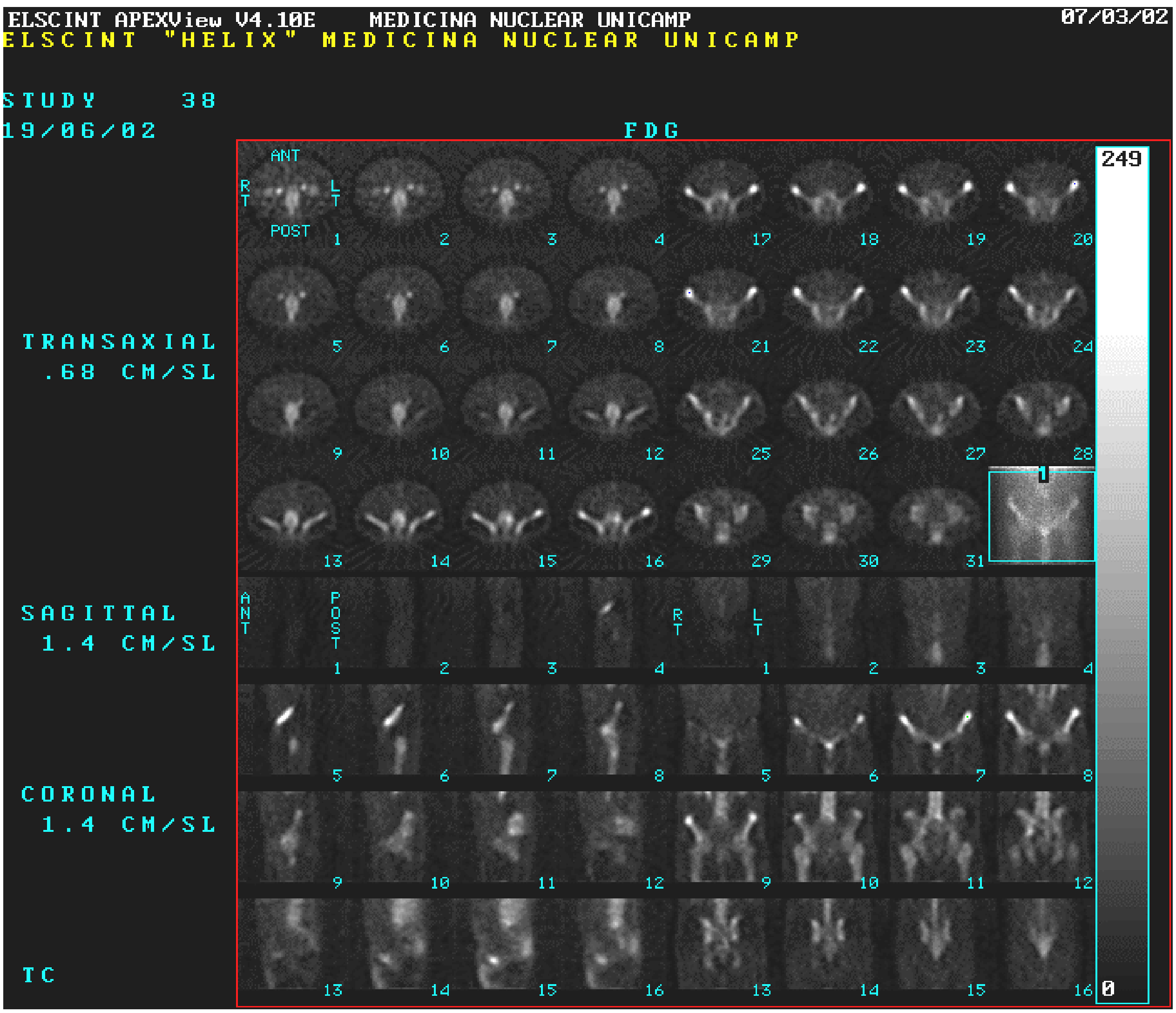
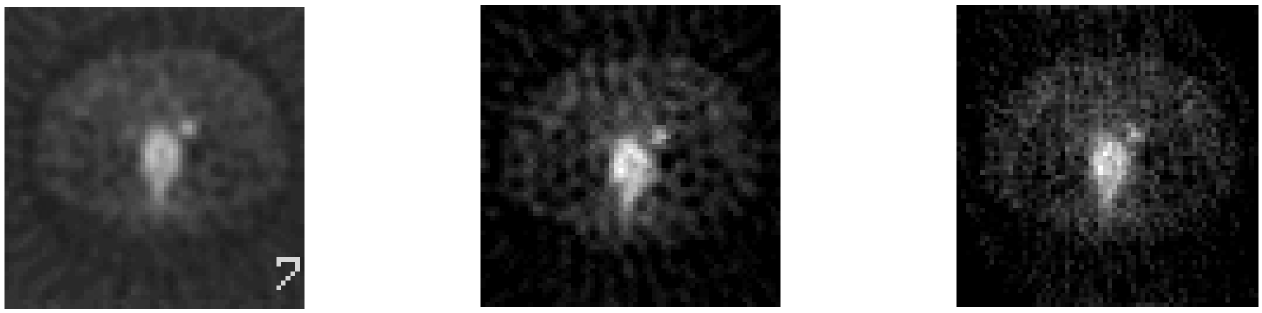
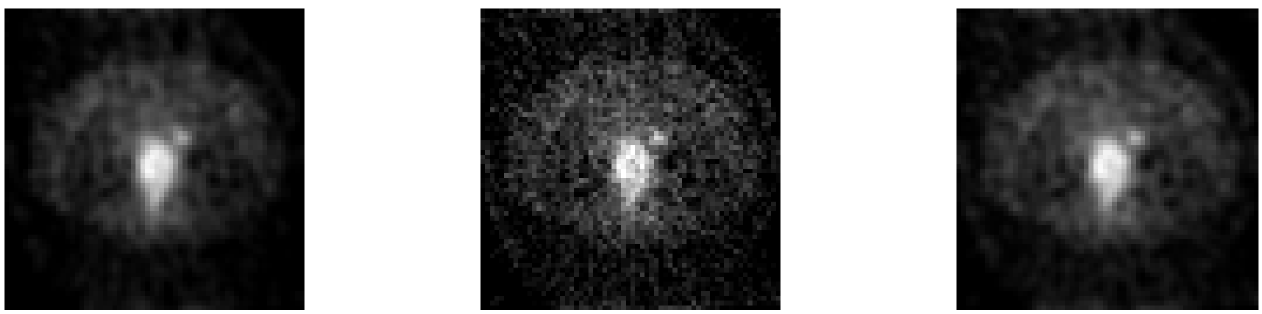


Figura 7.17 - *Report* – estudo FDG / Tc511 pélvis.

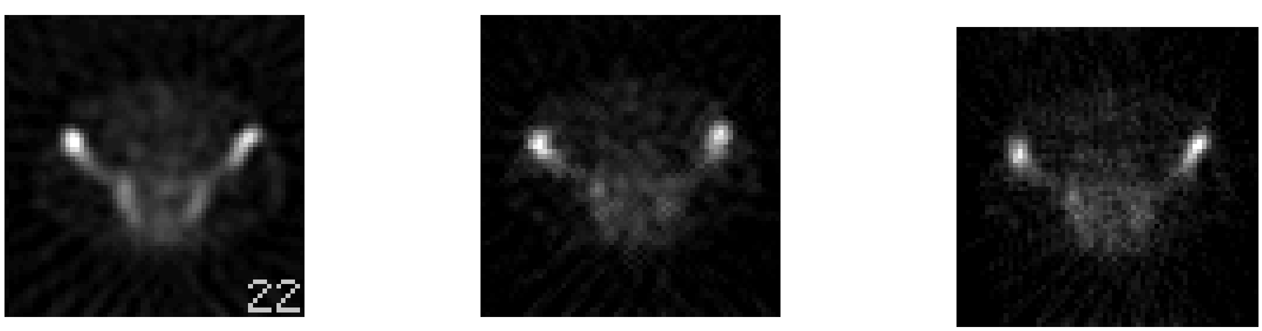


(a) (b) (c)



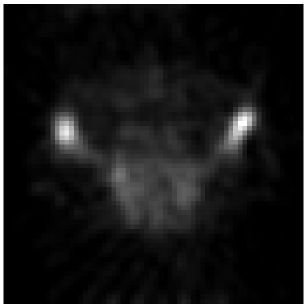
(d) (e) (f)

Figura 7.18 – (a) Corte 7 – *report* figura 7.17. (b) Reconstrução com *fbpw.m* + janela da figura 7.6.a. (c) Reconstrução com *artx.m* com *r (σ)* para σ =2. (d) Imagem 7.18.c com filtro pós processamento. (e) Reconstrução com híbrido com σ =0.25. (f) Imagem 7.18.e com filtro pós processamento.

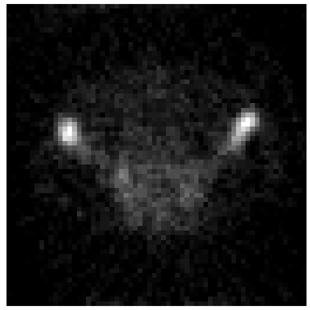


(a) (b) (c)

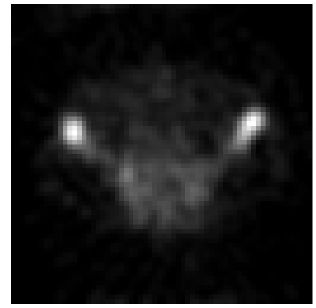
Legenda na pg. 222



(d)



(e)



(f)

Figura 7.19 – (a) Corte 22 – *report* figura 7.17. (b) Reconstrução com *fbpw.m* + janela da figura 7.6.a. (c) Reconstrução com *artx.m* com *r (σ)* para σ =2. (d) Imagem 7.19.c com filtro pós processamento. (e) Reconstrução com híbrido com σ =0.25. (f) Imagem 7.19.e com filtro pós processamento.

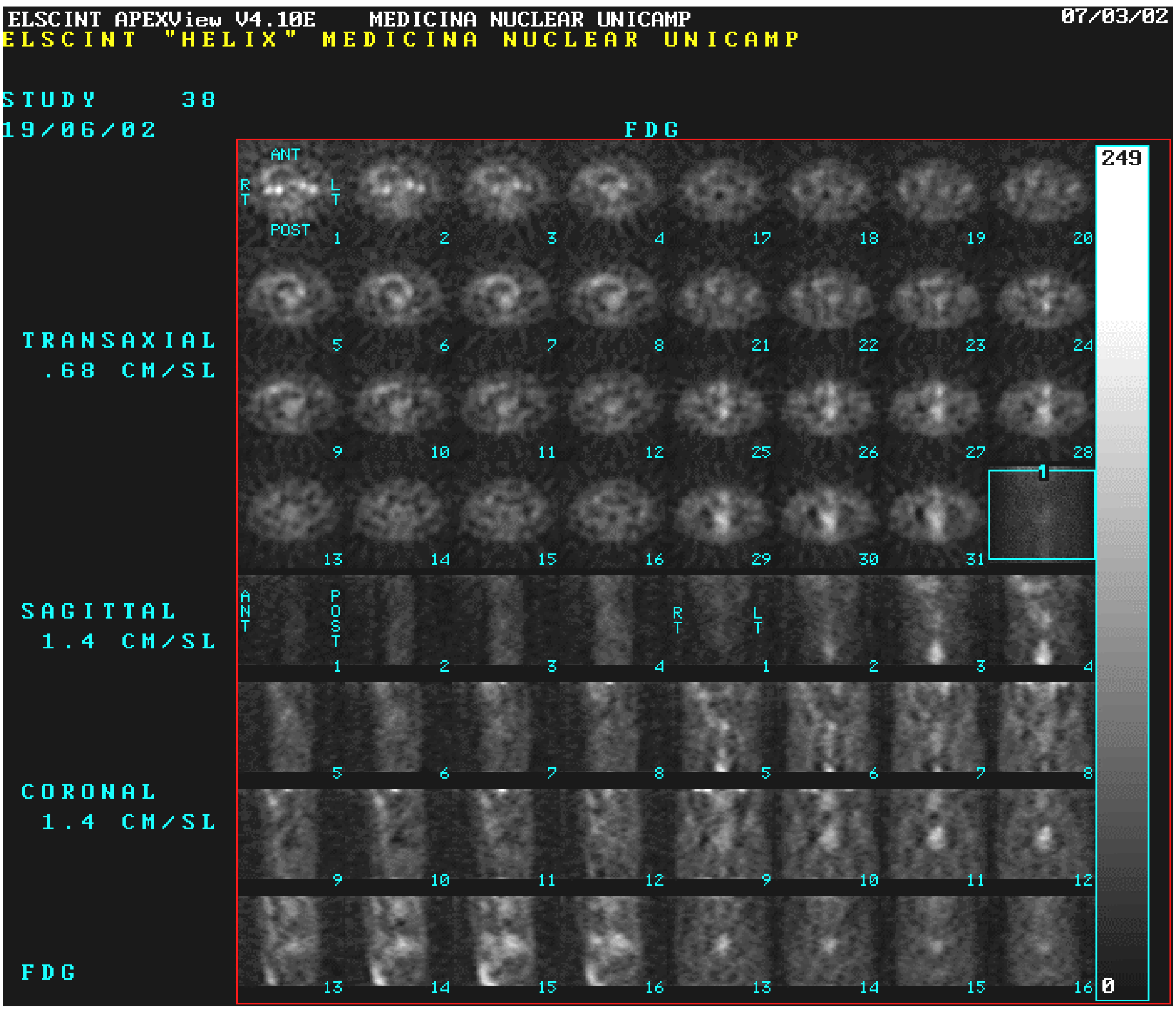
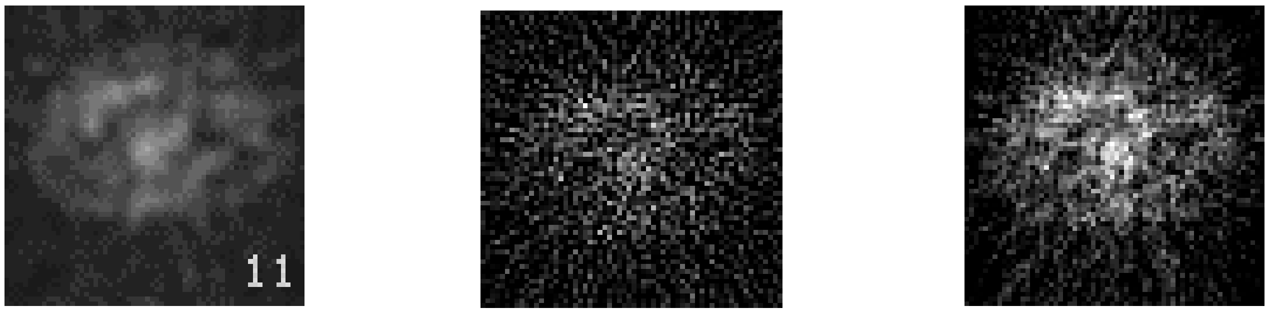
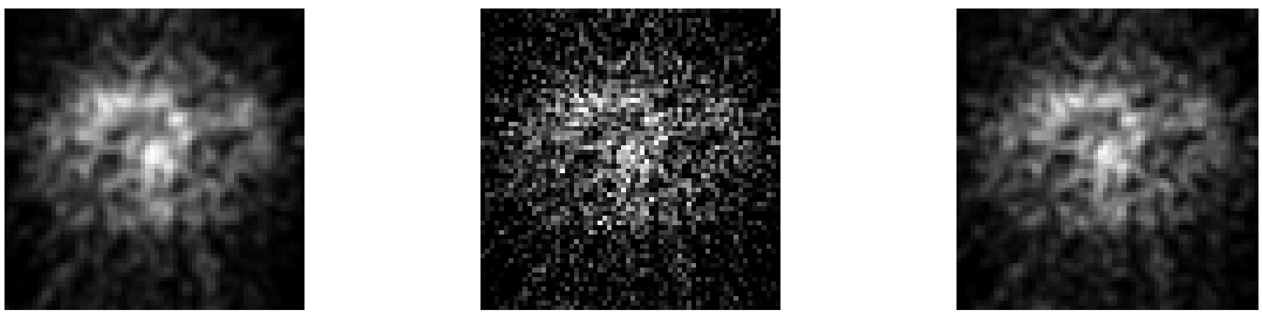


Figura 7.20 - *Report* – estudo FDG / ^{18}F pélvis.

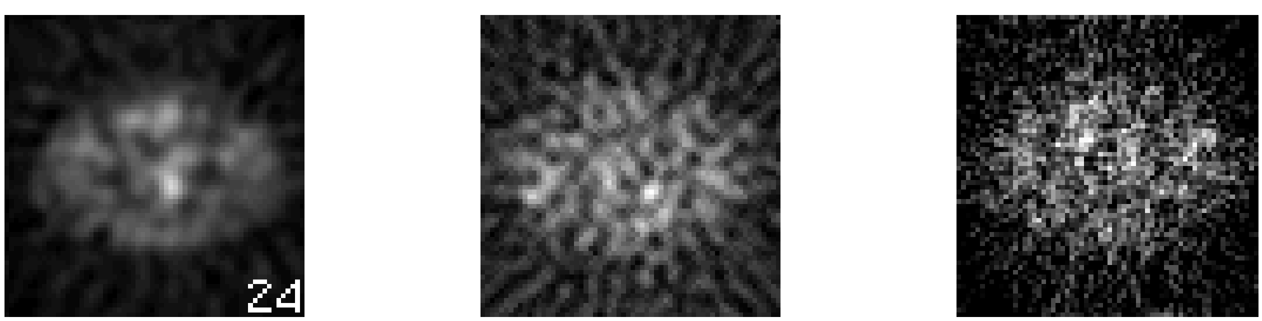


(a) (b) (c)



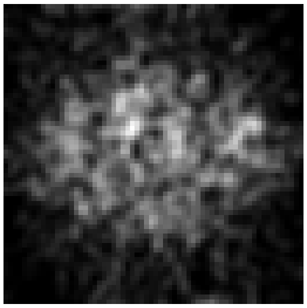
(d) (e) (f)

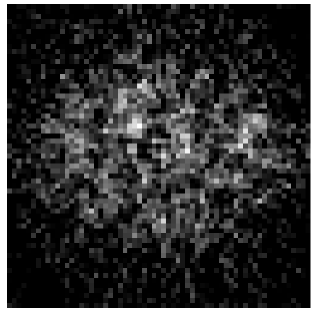
Figura 7.21 – (a) Corte 11 – *report* figura 7.20. (b) Reconstrução com *fbpw.m* + janela da figura 7.6.a. (c) Reconstrução com *artx.m* com *r (σ)* para σ =2. (d) Imagem 7.21.c com filtro pós processamento. (e) Reconstrução com híbrido com σ =0.25. (f) Imagem 7.21.e com filtro pós processamento.



(a) (b) (c)

Legenda na pg. 224


(d)


(e)

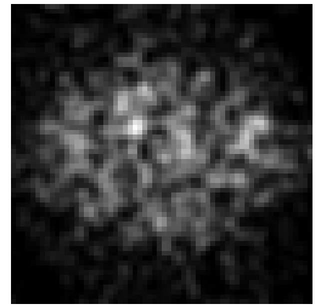

(f)

Figura 7.22 – (a) Corte 24 – *report* figura 7.20. (b) Reconstrução com *fbpw.m* + janela da figura 7.6.a. (c) Reconstrução com *artx.m* com *r (σ)* para σ =2. (d) Imagem 7.22.c com filtro pós processamento. (e) Reconstrução com híbrido com σ =0.25. (f) Imagem 7.22.e com filtro pós processamento.

Para as reconstruções do estudo de FDG-pélvis, verifica-se quantitativamente através do levantamento do perfil da linha 40 da imagem do *report* e das reconstruções o corte 22 do *report* de Tc511, usando os critérios descritos anteriormente. Também calcula-se o contraste das estruturas indicadas na figura 7.24 para o mesmo corte. Os resultados são apresentados nas figuras 7.23 e 7.25 e Tabela 7.3 .

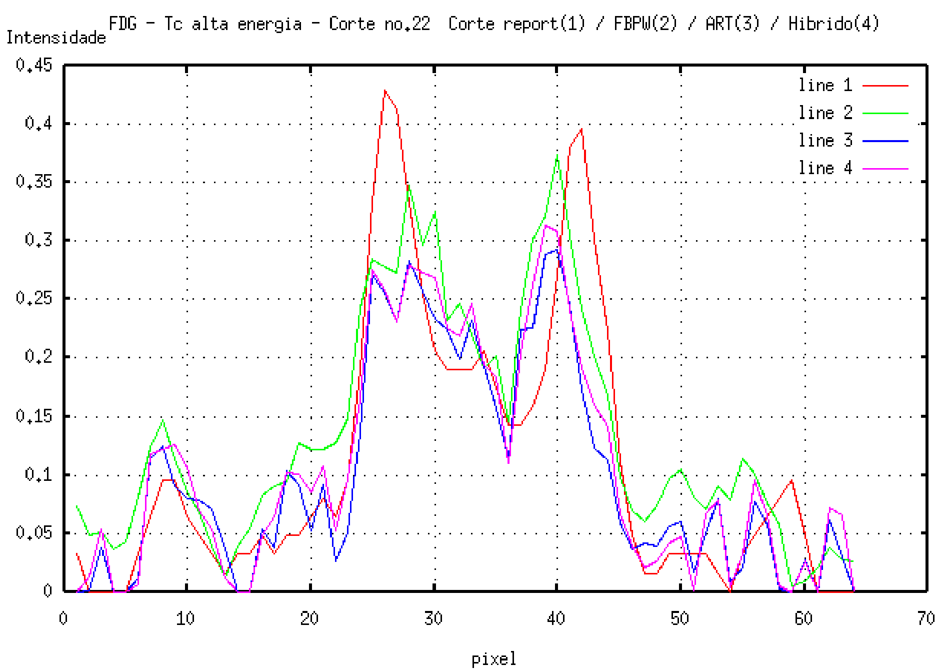


Figura 7.23 - Perfil da linha 40 do corte 22 da figura 7.17.

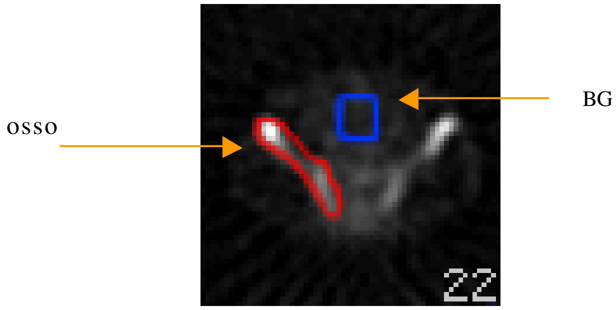


Figura 7.24 - ROIs - Regiões quente (vermelho) e fria (azul) utilizadas para o cálculo do contraste do corte 22 – *report* da figura 7.17.

	Corte 22		
	Valor médio		Contraste (%)
Algoritmo	*Osso*	*BG*	*Osso / BG*
REPORT	69,7	32,0	37,0
FBPW	59,6	19,7	50,4
ART	58,7	20,0	49,2
ART C/ FILTRO	68,0	21,9	51,3
HIBRIDO	64,6	24,1	45,6
HIBRIDO C/ FILTRO	72,3	25,8	47,3

Tabela 7.3 – Contrastes calculados para as estruturas do corte 22 da figura 7.24.

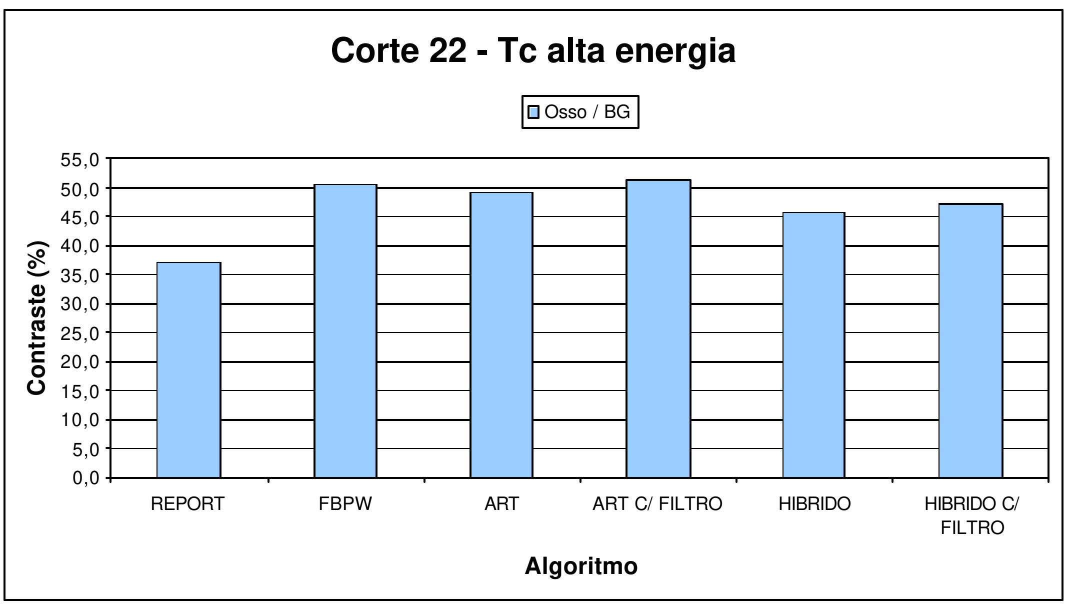


Figura 7.25 – Contrastes calculados para as estruturas do corte 22 da figura 7.24.

O conjunto de perfis apresentado na figura 7.23 mostra a coerência entre as imagens reconstruídas e o corte 22 do *report*. As diferenças de curvaturas notadas entre os perfis do corte do *report* e os obtidos com os algoritmos implementados está ligado a janela utilizada no *report* para a suavização da imagem, diferente ou ausente nas demais.

O contraste apresentado pela estrutura no corte 22 do *report* e o reconstruído pelo algoritmo ***fbpw.m*** , apresentam uma diferença significativa, tendo o segundo uma performance superior. Já entre os algoritmos implementados, a variação se mostra pequena.

7.3 – ^{67}Ga

Para as avaliações com ^{67}Ga foram reconstruídos dois cortes por estudo: um de tórax e um de cabeça, num total de 4 cortes.

O *report* referente ao estudo de tórax é apresentado na figura 7.26, seguido das imagens reconstruídas dos cortes 6 e 29 (figuras 7.28 e 7.29). Para a reconstrução dos cortes com o programa ***fbpw.m*** utilizou-se a janela apresentada na figura 7.27 e , para os algoritmos iterativos, o filtro pós processamento utilizado permanece o mesmo dos estudos MIBI.

Quantitativamente avalia-se o corte 29, conforme o conjunto de perfis traçados para a linha 23 (figura 7.30) e cálculo de contraste da estrutura identificada na figura 7.31. Os resultados dos cálculos são apresentados na tabela 7.4 e figura 7.32.

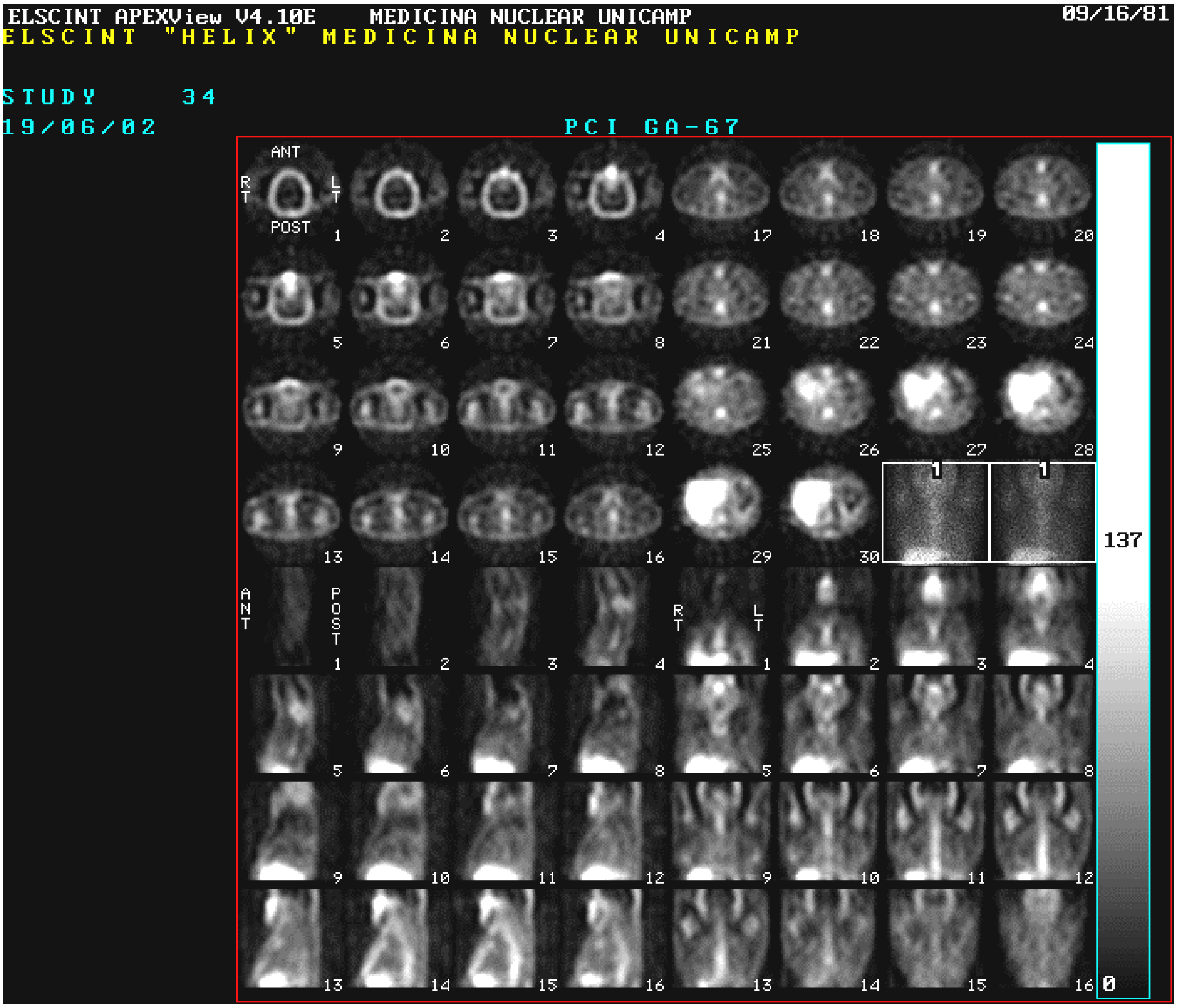


Figura 7.26 - ***Report*** **– estudo ^{67}Ga tórax .**

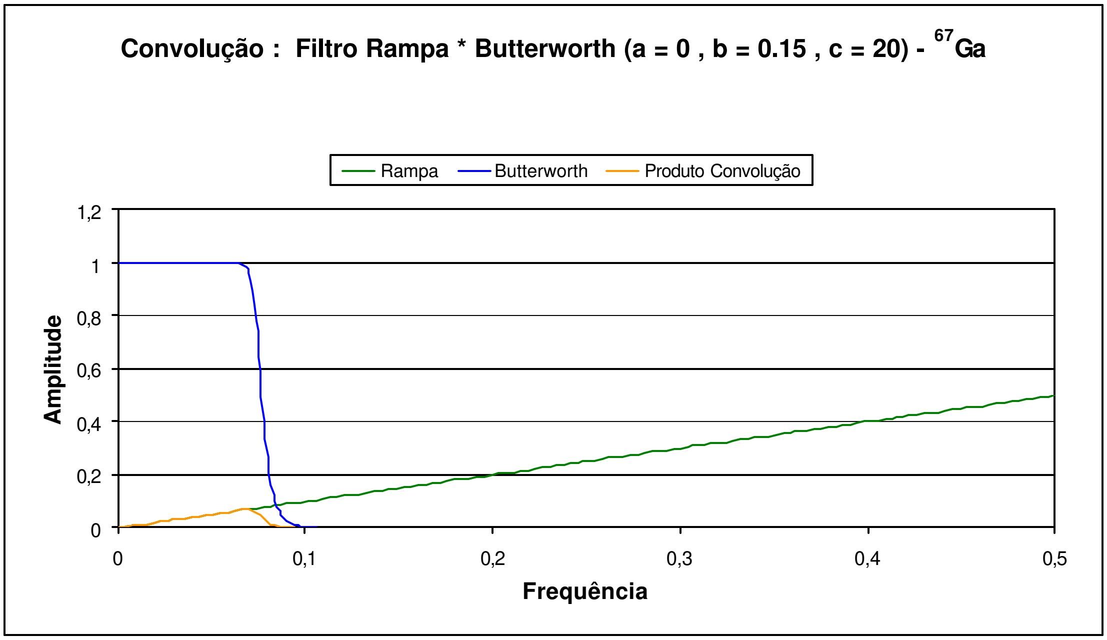


Figura 7.27 - Janela utilizada para a reconstrução dos cortes de estudo ^{67}Ga.

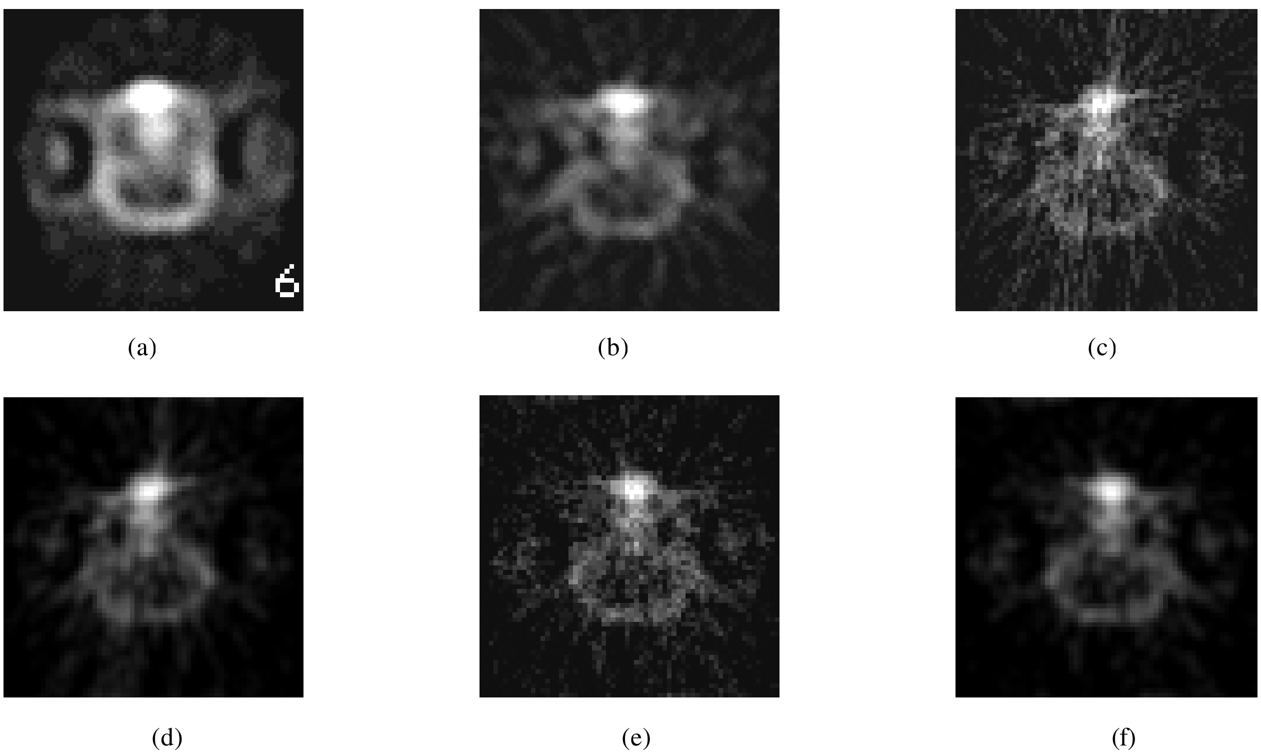
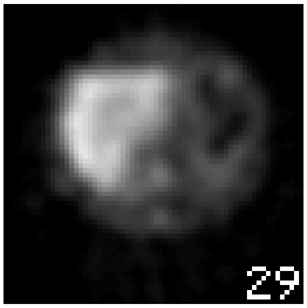
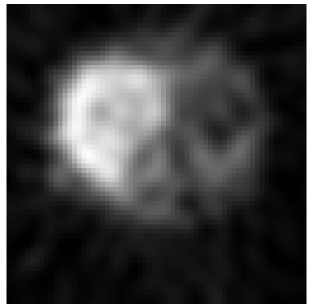


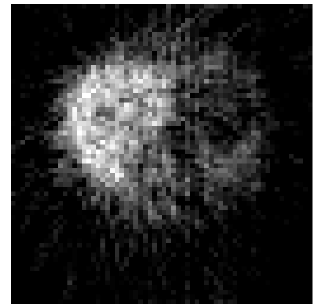
Figura 7.28 – (a) Corte 6 – *report* figura 7.26. (b) Reconstrução com *fbpw.m* + janela da figura 7.27.a. (c) Reconstrução com *artx.m* com *r (σ)* para σ =2. (d) Imagem 7.28.c com filtro pós processamento. (e) Reconstrução com híbrido com σ =0.25. (f) Imagem 7.28.e com filtro pós processamento.



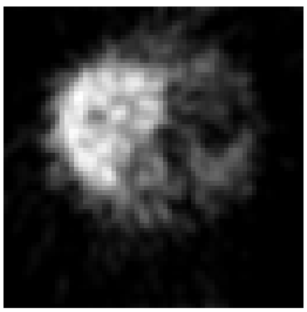
(a)



(b)

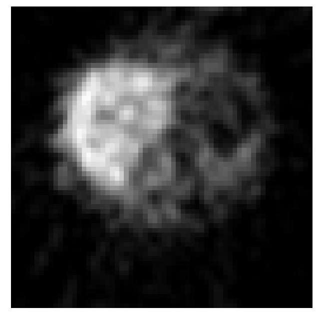


(c)



(d)

(e)



(f)

Figura 7.29 – (a) Corte 29 – *report* figura 7.26. (b) Reconstrução com *fbpw.m* + janela da figura 7.27.a. (c) Reconstrução com *artx.m* com *r (σ)* para σ =2. (d) Imagem 7.29.c com filtro pós processamento. (e) Reconstrução com híbrido com σ =0.25. (f) Imagem 7.29.e com filtro pós processamento.

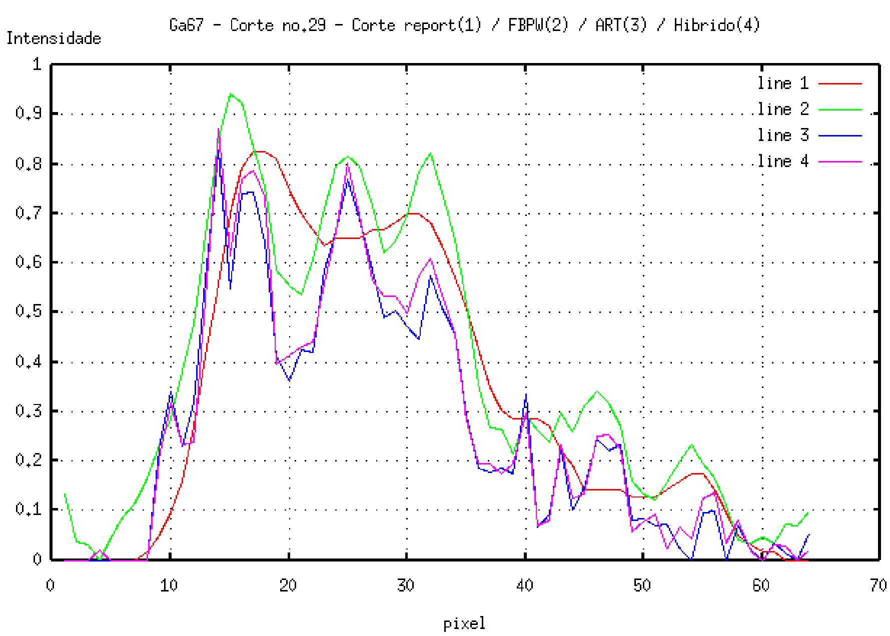


Figura 7.30 - Perfil da linha 23 do corte 29 da figura 7.28.

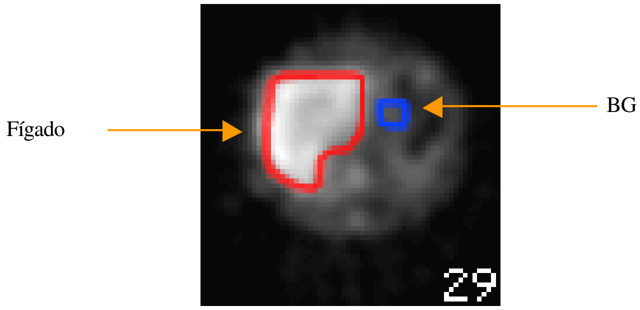


Figura 7.31 - ROIs - Regiões quente (vermelho) e fria (azul) utilizadas para o cálculo do contraste do corte 29 – *report* da figura 7.26.

	Corte 29		
	Valor médio		Contraste (%)
Algoritmo	*Fígado*	*BG*	*Fígado / BG*
REPORT	157,9	66,6	40,7
FBPW	184,1	66,0	47,2
ART	152,5	39,2	59,1
ART C/ FILTRO	170,7	41,2	61,1
HIBRIDO	147,1	40,9	56,5
HIBRIDO C/ FILTRO	166,0	45,4	57,0

Tabela 7.4 – Contrastes calculados para as estruturas do corte 29 da figura 7.30.

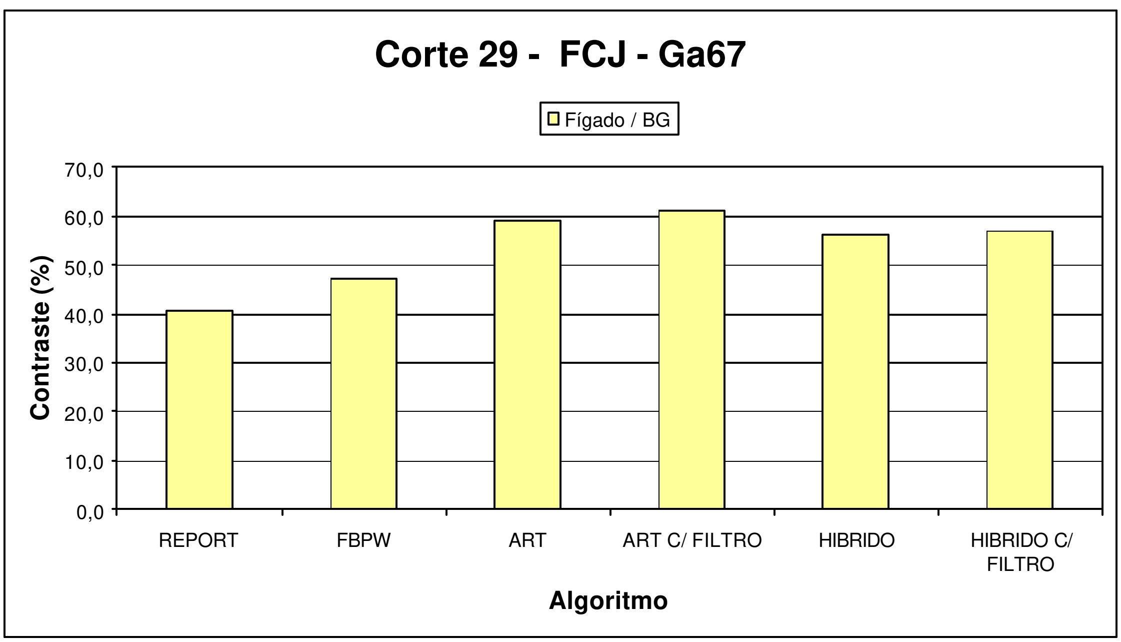


Figura 7.32 – Contrastes calculados para as estruturas do corte 29 da figura 7.30.

O conjunto de perfis traçados comporta-se como no estudo dos cortes de pélvis–FDG, ou seja , a diferença de curvatura entre estes se deve a uma diferença de janela utilizada ou a ausência desta, porém, os perfis possuem coerência. Quanto aos valores de contraste obtidos, verifica-se novamente uma melhor performance do algoritmo ***fbpw.m*** em relação ao utilizado para a produção do *report* e um melhor resultado dos algoritmos iterativos para estruturas quentes bastante captantes.

Repetem-se os procedimentos apresentados no estudo de tórax / ^{67}Ga para um estudo de cabeça do mesmo radioisótopo. Os resultados são apresentados a seguir para os cortes 11 e 15, com o qual se realiza os estudos quantitativos.

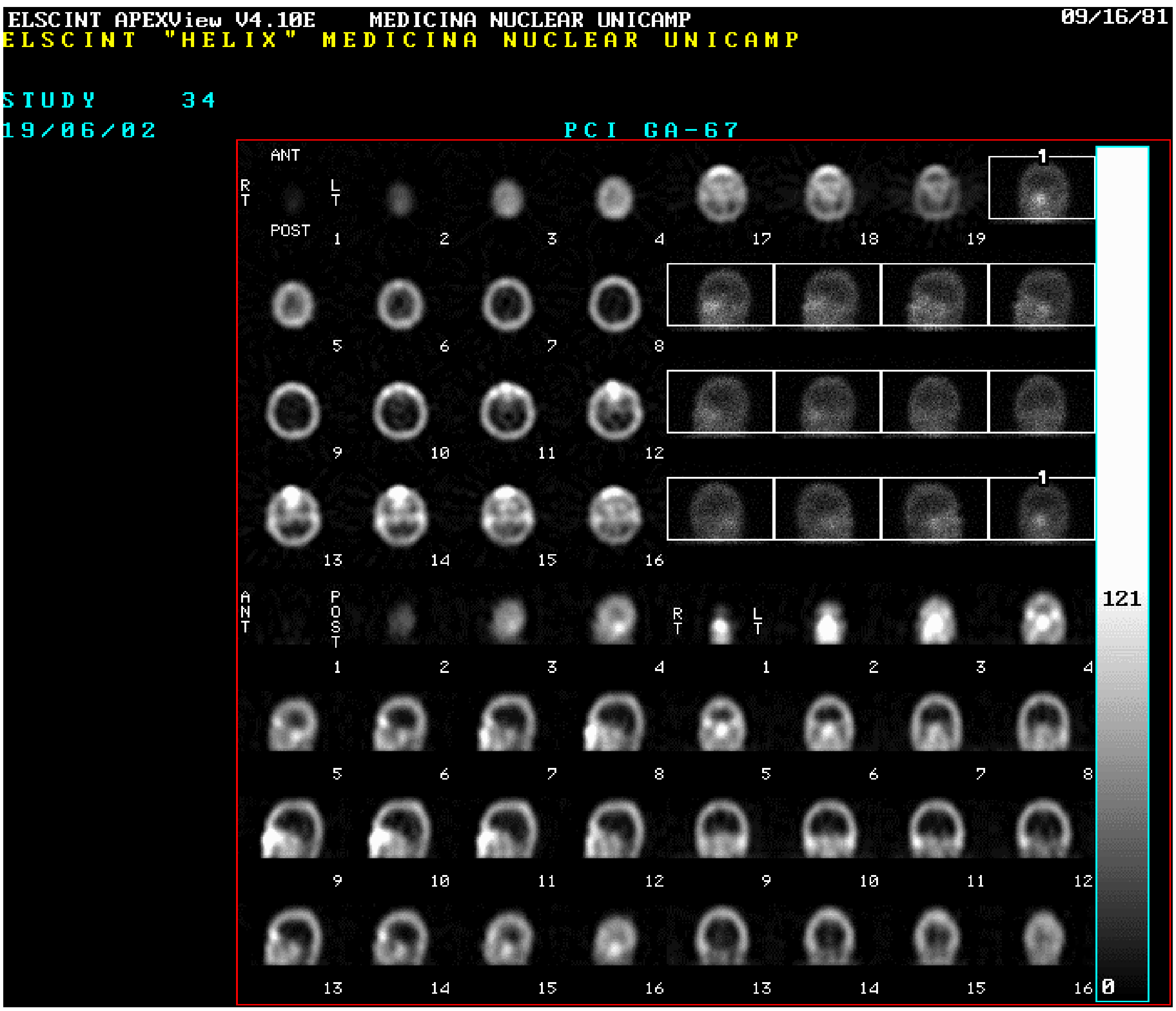
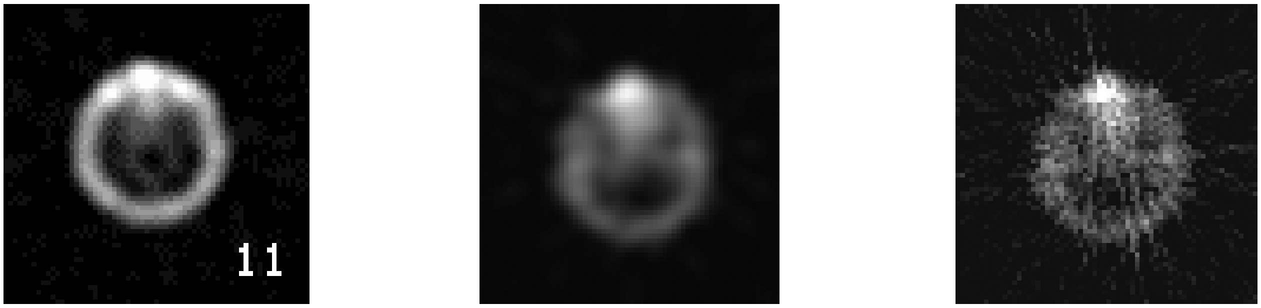
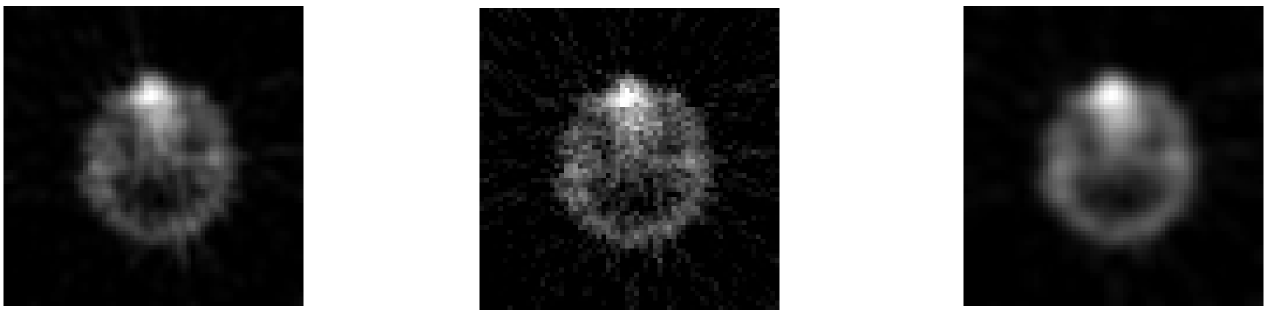


Figura 7.33 - *Report* – estudo ^{67}Ga cabeça.

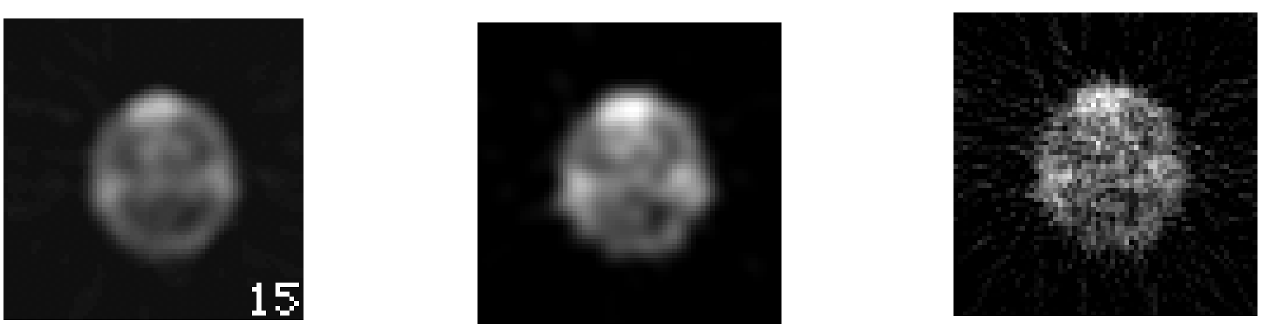


(a) (b) (c)



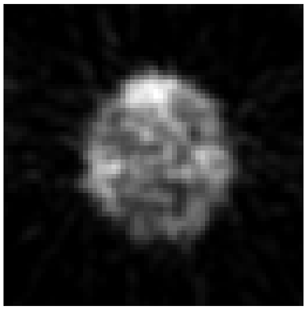
(d) (e) (f)

Figura 7.34 – (a) Corte 11 – *report* figura 7.32. (b) Reconstrução com *fbpw.m* + janela da figura 7.27.a. (c) Reconstrução com *artx.m* com *r (σ)* para σ =2. (d) Imagem 7.34.c com filtro pós processamento. (e) Reconstrução com híbrido com σ =0.25. (f) Imagem 7.34.e com filtro pós processamento.

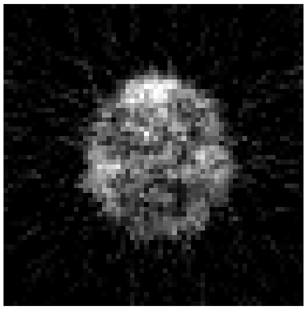


(a) (b) (c)

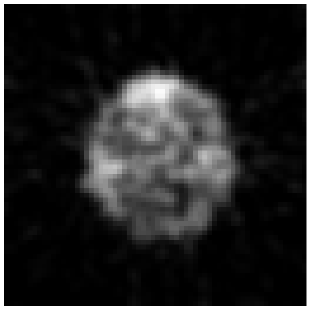
Legenda na pg. 233



(d)



(e)



(f)

Figura 7.35 – (a) Corte 15 – *report* figura 7.32. (b) Reconstrução com *fbpw.m* + janela da figura 7.27.a. (c) Reconstrução com *artx.m* com *r (σ)* para σ =2. (d) Imagem 7.35.c com filtro pós processamento. (e) Reconstrução com híbrido com σ =0.25. (f) Imagem 7.35.e com filtro pós processamento.

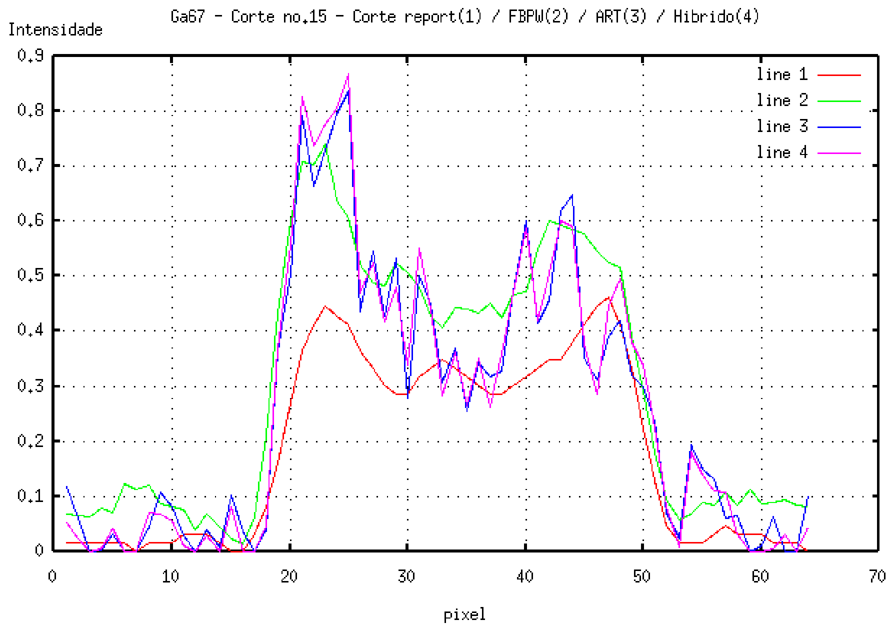


Figura 7.36 - Perfil da linha 35 do corte 15 da figura 7.35.

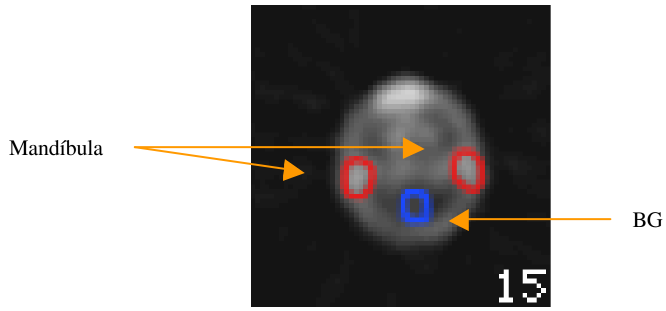


Figura 7.37 - ROIs - Regiões quentes (vermelho) e fria (azul) utilizadas para o cálculo do contraste do corte 15 – *report* da figura 7.33.

	Corte 15		
	Valor médio		Contraste (%)
Algoritmo	*Mandíbula*	*BG*	*Mandíbula / BG*
REPORT	85,3	40,2	35,9
FBPW	133,3	73,3	29,1
ART	114,9	71,6	23,2
ART C/ FILTRO	131,2	78,8	25,0
HIBRIDO	124,3	73,0	26,0
HIBRIDO C/ FILTRO	124,7	71,4	27,2

Tabela 7.5 – Contrastes calculados para as estruturas do corte15 da figura 7.37.

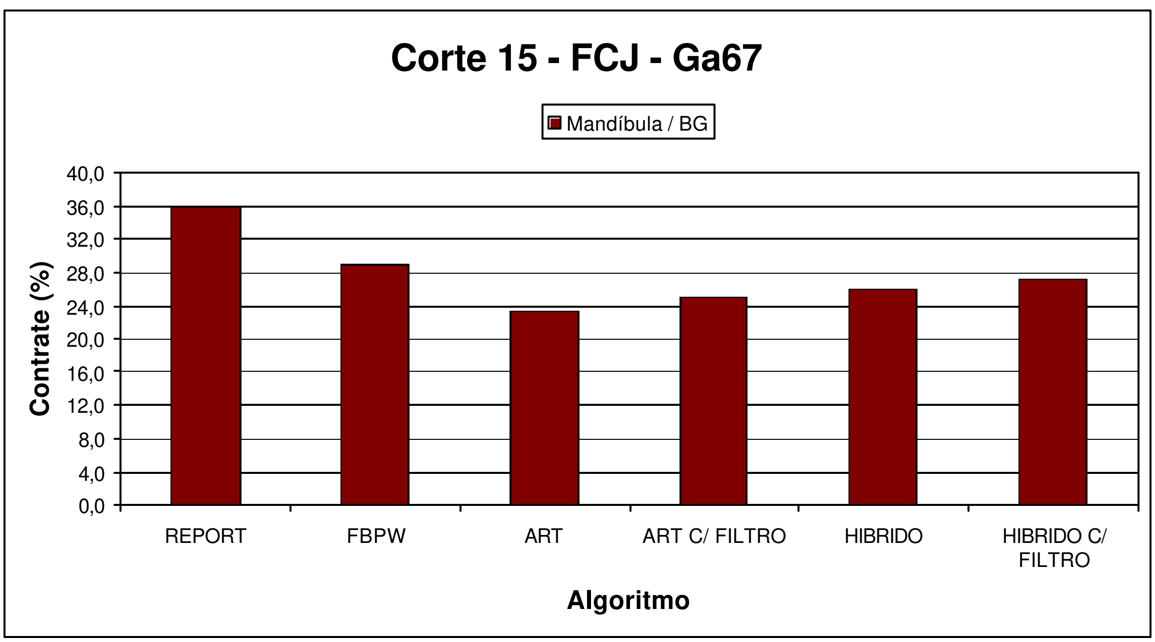


Figura 7.38 – Contrastes calculados para as estruturas do corte 15 da figura 7.37.

Novamente o comportamento no conjunto de perfis se repete devido a diferença de janelas utilizadas ou ausência desta, porém, verifica-se que o padrão dos perfis das imagens reconstruídas com os algoritmos implementados confere com o obtido da imagem do *report*. Observa-se também uma melhor performance dos algoritmos FBP para o contraste da estrutura estuda (mandíbula), com freqüência espacial maior que o fígado anteriormente verificado.

7.4 – Análise Clínica

Como último teste sobre as imagens clínicas, realiza-se uma avaliação visual e subjetiva por parte de 6 especialistas em Medicina Nuclear do SMN – HC Unicamp (todos com o mesmo tempo de experiência) sobre alguns dos cortes reconstruídos.

O critério adotado para a avaliação foi , observando-se as imagens dos *reports* e as reconstruídas com os algoritmos ***fbpw.m*** com janela e os iterativos ***artx.m*** e híbrido com filtro de borramento Gaussiano , identifica-se:

- qual ou quais das imagens apresentavam melhor definição das estruturas e
- qual ou quais das imagens apresentavam menos artefatos .

Atribuindo nota 5 as de melhor performance para cada critério em separado, as demais imagens deveriam receber notas no máximo igual a 5, caso tivessem a mesma performance, ou inferiores (de 4 a 1), tornando o critério de avaliação normalizado.

As notas de cada avaliador, para cada imagem avaliada foram multiplicadas, gerando um indicador. Ao final, os indicadores de cada avaliador para cada imagem foram somados. Dentro de cada imagem a nota final foi normalizada, obtendo o valor 1 a reconstrução do algoritmo com a melhor performance. A Tabela 7.6 resume, em ordem decrescente por imagem, os resultados das avaliações realizadas.

Radioisotopo	*Corte*	*Figura*	*Algoritmo*	*Nota Final*	***Nota Final normalizada***
Tc511	**20**	**7.8**	*art+filtro*	110	**1,00**
			report	99	**0,90**
			hibrido+filtro	72	**0,65**
			fbpw+janela	64	**0,58**
Tc511	**29**	**7.9**	*report*	117	**1,00**
			art+filtro	100	**0,85**
			hibrido+filtro	85	**0,73**
			fbpw+janela	54	**0,46**
F18	**20**	**7.11**	*report*	107	**1,00**
			art+filtro	101	**0,94**
			hibrido+filtro	76	**0,71**
			fbpw+janela	59	**0,55**
F18	**27**	**7.12**	*hibrido+filtro*	111	**1,00**
			art+filtro	98	**0,88**
			report	70	**0,63**
			fbpw+janela	51	**0,46**
Tc511	**7**	**7.18**	*art+filtro*	140	**1,00**
			hibrido+filtro	123	**0,88**
			report	93	**0,66**
			fbpw+janela	81	**0,58**
Tc511	**22**	**7.19**	*report*	140	**1,00**
			art+filtro	104	**0,74**
			hibrido+filtro	101	**0,72**
			fbpw+janela	57	**0,41**
Ga67	**29**	**7.29**	*report*	137	**1,00**
			fbpw+janela	82	**0,60**
			art+filtro	54	**0,39**
			hibrido+filtro	54	**0,39**
Ga67	**15**	**7.35**	*report*	118	**1,00**
			hibrido+filtro	95	**0,81**
			fbpw+janela	73	**0,62**
			art+filtro	33	**0,28**

Tabela 7.6 – Resumo dos resultados das avaliações do corpo clínico do SMN – HC Unicamp sobre as imagens reconstruídas.

Observando os resultados apresentados na Tabela 7.6, verifica-se a necessidade de se organizar as informações para uma melhor avaliação. Somando-se as notas normalizadas para cada radioisótopo sobre cada algoritmo e, calculando-se a porcentagem da performance do aspecto visual de cada um pode-se verificar as tendências nas avaliações (figuras 7.39 à 7.41).

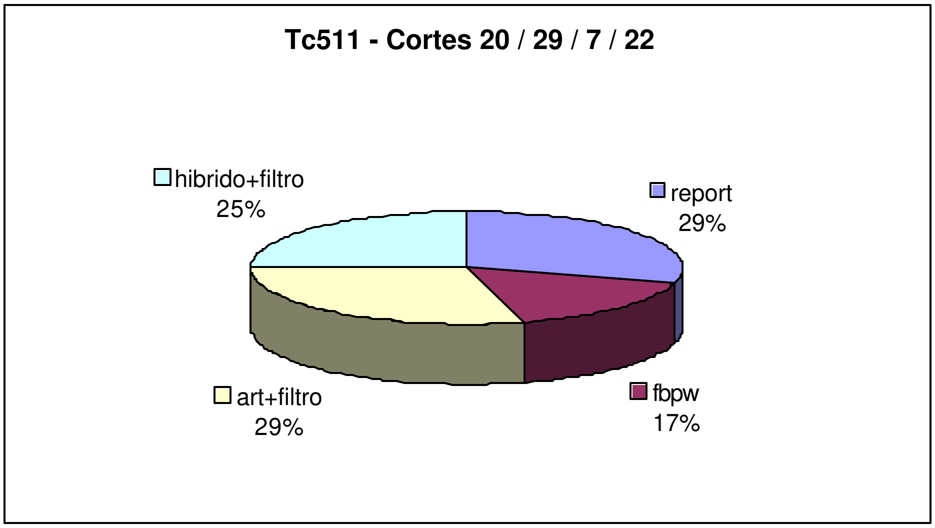


Figura 7.39 – Performance visual do algoritmos para o Tc511.

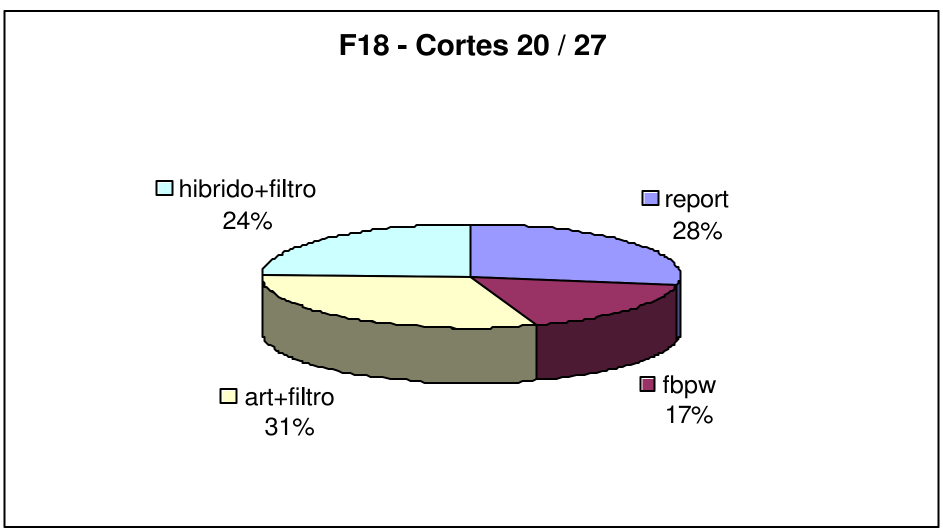


Figura 7.40 – Performance visual do algoritmos para o radioisótopo ^{18}F.

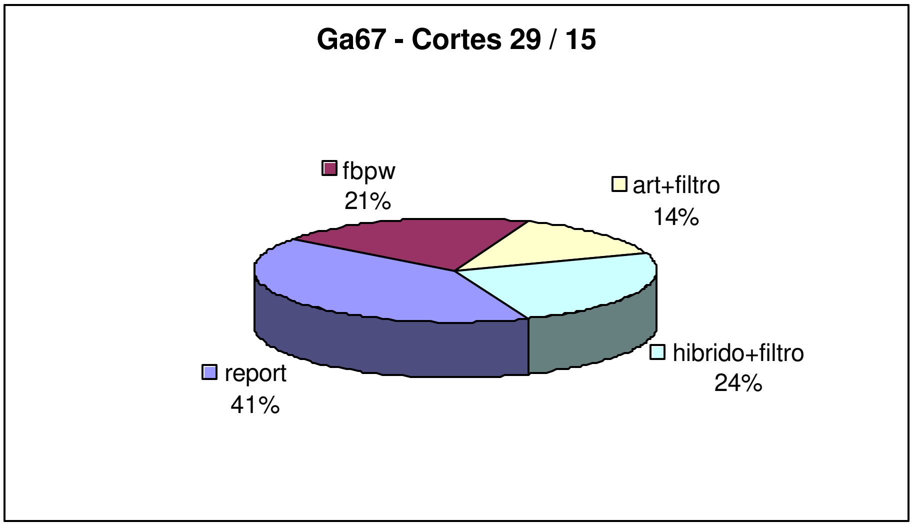


Figura 7.41 – Performance visual do algoritmos para o radioisótopo ^{67}Ga.

O gráfico da figura 7.39 indica que há uma uniformidade quanto ao critério visual dos cortes reconstruídos com as variadas técnicas para o Tc511. Isto aponta para uma equalização nos resultados alcançados com os filtros suavizadores usados.

Para o radioisótopo ^{18}F , a figura 7.40 apresenta resultados semelhantes à 7.39, indicando as mesmas conclusões.

No caso das imagens com ^{67}Ga , há uma tendência a preferência pelos cortes dos *reports*. Isto indica que os filtros suavizadores utilizados com os algoritmos implementados não estão otimizados para o protocolo clínico. Também devido as característica dos *reports* utilizados correntemente na prática clínica, não se está efetuando os cortes na mesma altura das projeções, ocorrendo uma variação de mais ou menos 01 *pixel*.

Capítulo 8 – Conclusões

A investigação sobre a performance dos algoritmos FBP, Iterativos (ART e SIRT) e as composições híbridas, trouxeram um vasto material, com muitos resultados, dos quais os principais foram destacados neste Capítulo.

Com referência aos algoritmos ART com correção multiplicativa e SIRT com correções aditiva e multiplicativa, acredito que os mesmos possam ainda ser explorados. Para tanto, fazem-se necessárias modificações na precisão de truncamento das matrizes peso e a aplicação de filtros suavizadores otimizados.

Dentre os mais importantes resultados alcançados, a utilização de um coeficiente de relaxação variável com Δp (equação 6.5) merece destaque. O resultado obtido neste trabalho deve ser explorado, tanto na otimização de valores de σ, como por exemplo os obtidos por cálculos que se valham dos valores de ID das imagens, como novas funções para $r(\sigma)$.

Com os resultados quantitativos obtidos nos Capítulos 6 e 7, pode-se definir um padrão de comportamento dos algoritmos testados mais exaustivamente : FBP com pesos (***fbpw.m***), ART com correção aditiva e coeficiente de relaxação variável (***artx.m***) e híbrido com coeficiente de relaxação fixo (r = 0.25). Para tanto , utilizam-se os resultados obtidos com os *phantoms* matemáticos com a adição de ruído, onde as simulações se aproximam mais da realidade quando comparadas com aquelas realizadas sem ruído, com *phantoms* físicos e imagens clínicas.

Os valores de Uniformidade tomográfica do *phantom* matemático com a adição de ruído, resumidos nas figuras 6.26 ,6.27 e 6.29, indicam que o algoritmo FBP terá resultados melhores que os Iterativos quando aplicados em projeções provenientes de equipamentos melhor calibrados porém, resultados muito próximos são alcançados entre os algoritmos em condições mais rotineiras de calibração de equipamentos (uniformidade plana = 4%) . Isto ocorre porque o algoritmo FBP utiliza um principio aditivo para a reconstrução, portanto, quanto maior a flutuação nas projeções, maior será a não uniformidade no corte reconstruído. Além disso, devido a aplicação do filtro rampa, há uma valorização do ruído, contribuindo para o aumento da não uniformidade.

Com o estudo sobre o contraste de objetos do *phantom* matemático, resumido nas figuras 6.30, 6.32 e 6.34, verifica-se uma performance praticamente igual dos algoritmos quando resultados para objetos de menor freqüência são avaliados, sendo estes quentes ou frios . Porém, devido a aplicação do filtro rampa, objetos com freqüências próximas ou igual a de Nyquist, têm seu contraste melhorado com o algoritmo FBP, principalmente no caso de objetos frios. (figura 8.1).

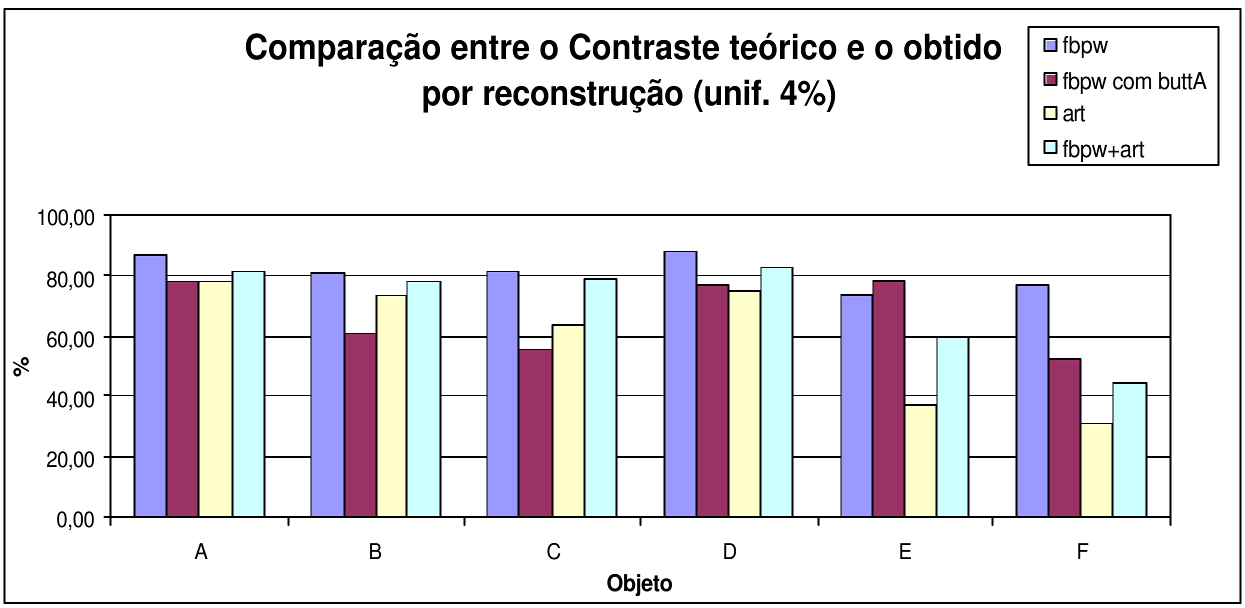


Figura 8.1 – Figura 6.34 repetida para a verificação dos contrastes dos objetos de alta freqüência (E e F) do *phantom* matemático, onde se verifica uma melhor performance do algoritmo FBP, principalmente para o objeto frio (F).

Os estudos quantitativos com *phantoms* (ítem 6.4.2) apontam uma maior definição do padrão de comportamento dos algoritmos.

O teste de Uniformidade tomográfica com as projeções do *phantom* Carlson mostram melhores resultados alcançados pelos iterativos (figura 6.51) para esta grandeza, resolvendo o impasse criado com os testes realizados com o *phantom* matemático.

Na avaliação do contraste de objetos quentes dos *phantoms* (figuras 6.52 e 6.53) encontra-se uma discreta superação de performance por parte dos algoritmos Iterativos sobre o FBP. Já na avaliação de objetos frios (figuras 6.55 e 6.56) verifica-se uma performance praticamente igual para o objeto de maior diâmetro, porém, ocorrendo uma melhora discreta com o algoritmo FBP para o objeto menor, novamente devido aplicação do filtro rampa. Os resultados alcançados na avaliação de contraste com *phantoms* tomográficos concordam com os obtido com os *phantoms* matemáticos.

O Capítulo 7 apresenta nos resultados quantitativos das análises feitas sobre as imagens clínicas uma confirmação daquilo que se obteve com os testes realizados com *phantoms*. Para estruturas quentes, ou seja, bastante captantes, os valores de contraste obtidos ou são praticamente iguais ou os algoritmos iterativos são discretamente melhores. Porém, para estruturas mais frias, ou seja, menos captantes, o algoritmo FBP novamente apresenta melhores resultados. As figuras 8.2 e 8.3 repetem as figuras 7.15 e 7.30, respectivamente para ilustração.

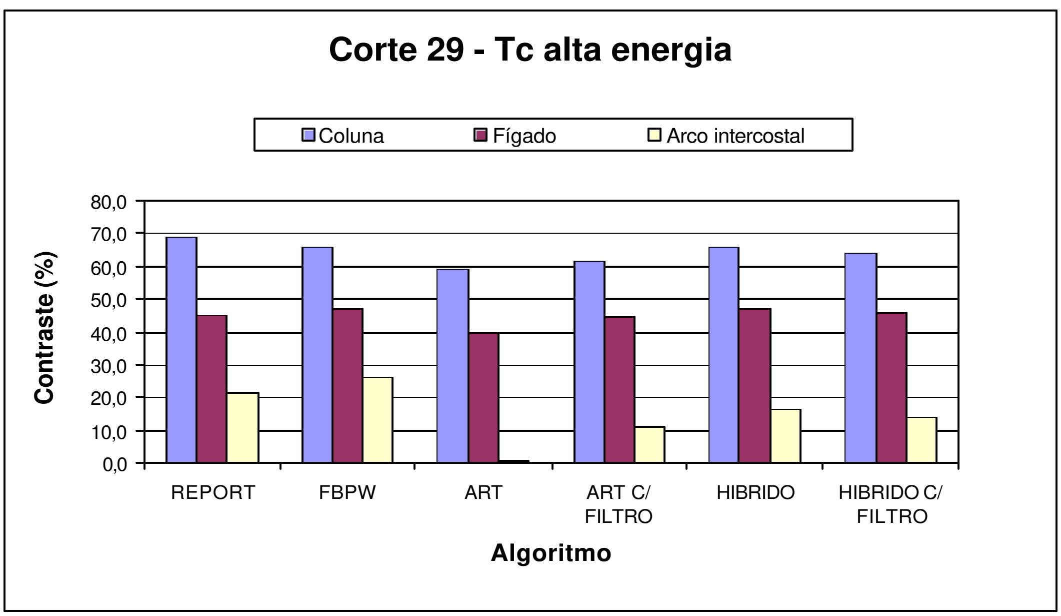


Figura 8.2 – figura 7.15.

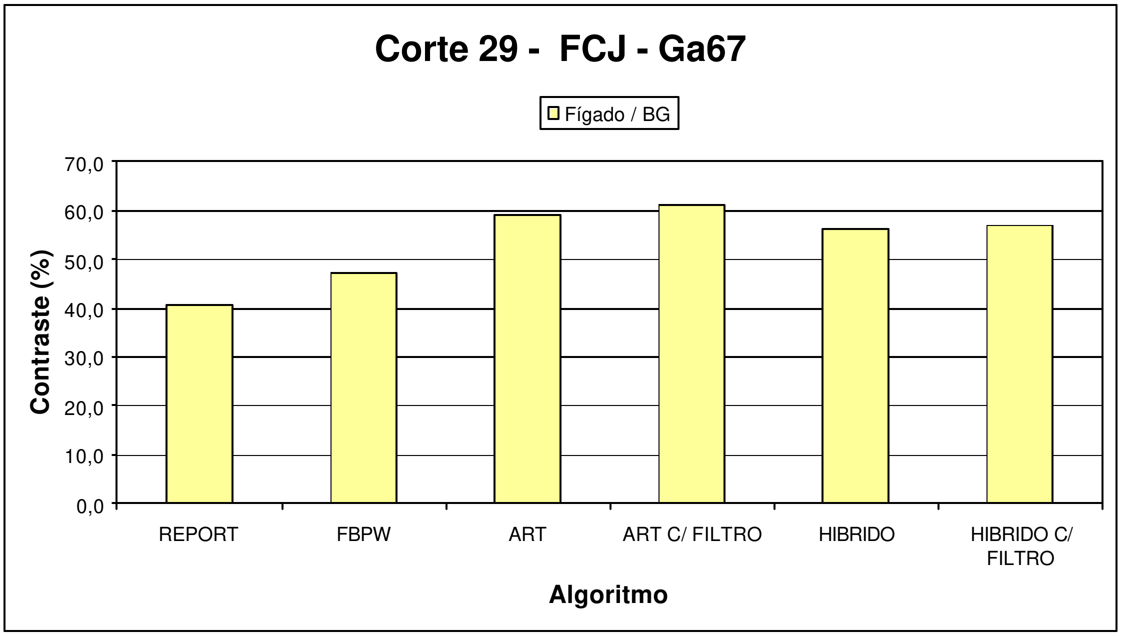


Figura 8.3 – figura 7.30.

Ao fim deste trabalho concluo, em linhas gerais, que os algoritmos Iterativos implementados como o foram, alcançam resultados discretamente superiores aos obtidos com o FBP, em se tratando de objetos quentes , sendo acentuadas as diferenças em favor deste último no caso de objetos frios, ainda não compensando o gasto computacional empregado para as reconstruções .

Porém, acredito que se possam alcançar resultados melhores com os Iterativos que com o FBP. Para tanto, deve–se estender os estudos sobre a eficácia dos coeficientes de relaxação com o valor de σ definido pela ID da imagem e na busca de filtros suavizadores otimizados para estes na aplicação de imagens clínicas – confirmado pela análise visual da equipe do SMN - HC Unicamp. Com estas mudanças adicionais a serem implementadas, a utilização de algoritmos Iterativos terá com certeza uma relação custo / benefício que compensa sua utilização nos atuais 7 equipamentos, ao invés do FBP.

Anexos

Anexo A – Sistemas Lineares

Seja um sinal de entrada $w_{in}(u)$ e outro de saída $w_{out}(u)$ em um sistema de medição, onde *u* é um conjunto de parâmetros (tempo, espaço etc.) do sistema.

Diz –se que um sistema é linear , quando :

1) o valor da soma das leituras de dois sinais diferentes ($w_{out}(u)$) implicar no mesmo valor de leitura da soma de dois sinais de entrada $w_{in}(u)$ distintos, ou seja : $w_{in}(u) = w_{in}^{(1)}(u) + w_{in}^{(2)}(u) \Rightarrow w_{out}(u) = w_{out}^{(1)}(u) + w_{out}^{(2)}(u)$;

2) a multiplicação do sinal de entrada $w_{in}(u)$ por um escalar $\alpha \in \Re$ não altera nenhuma característica do sinal de saída , além de sua amplitude : $\alpha . w_{in}(u) \Rightarrow \alpha . w_{out}(u)$ se $w_{in}(u) \Rightarrow . w_{out}(u)$.

Para se definir a relação entre um sinal de entrada e saída em um sistema linear, deve –se saber que estes não possuem comportamento local. Isto significa , por exemplo em sistemas onde *u* são coordenadas espaciais (câmeras cintiladoras) que $w_{out}(u)$ não é resultado apenas do valor de *u* correspondente, mesmo com a presença de colimadores, mas também de outros objetos em diferentes *u´*s. Portanto, $w_{out}(u)$ é resultado do sinal de vários objetos em diferentes *u´*s , evidenciando um comportamento não local.

Desse modo , $w_{out}(u)$ pode ser descrito como uma superposição de $w_{in}(u)$ para diferentes valores de *u*. Como *u* é normalmente contínuo :

$$w_{out}(u) = \int_{-\infty}^{\infty} p(u,u')w_{in}(u')du' \qquad \textbf{(A .1)}$$

A função *p(u,u´)* na equação A.1 é chamada de ***função resposta do sistema***, definida quando um pulso (δ) é aplicado na entrada do mesmo:

$$w_{in}(u) = \delta\,(u' - u_0) \qquad \textbf{(A.2)}$$

$$w_{out} = \int_{-\infty}^{\infty} p(u,u').\delta\,(u' - u_0)\,du' = p(u,u_0) \qquad \textbf{(A.3)}$$

Dessa forma, vê –se pela equação A3 que a função resposta de um sistema é formato do sinal de saída quando o sinal de entrada é um pulso. Em sistemas de imagem, a função p(u,u´) é chamada de função de espalhamento de ponto (PSF – Point Spread Function), a qual caracteriza totalmente o sistema.

Anexo B – Transformadas de Fourier ($\mathfrak{F}$) (46)

Definição :

$$g(w) = \Im\{f(x)\} = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} f(x) e^{-iwx} dx \qquad \textbf{(B.1)}$$

$$f(x) = \Im^{-1}\{g(w)\} = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} g(w) e^{iwx} dw \qquad \textbf{(B.2)}$$

Propriedades:

1) Linearidade : $\mathfrak{F}\{a.f + b.g\} = a.\mathfrak{F}\{f\} + b.\mathfrak{F}\{g\}$
2) Derivadas : $\mathfrak{F}\{f'(x)\} = iw\, \mathfrak{F}\{f(x)\}$

   $\mathfrak{F}\{f''(x)\} = -w^2\, \mathfrak{F}\{f(x)\}$
3) Convolução :

   Definição : $(f * g)(x) = \int_{-\infty}^{\infty} f(x).g(x-x')dx'$

   Sendo : $\mathfrak{F}(f * g)(x) = \mathfrak{F}\{f(x)\} . \mathfrak{F}\{g(x)\}$ **(B.3)**

Funções e suas Transformadas

1) Função δ

F {δ(x)} = 1 **(B.4)**

2) Função *rect*

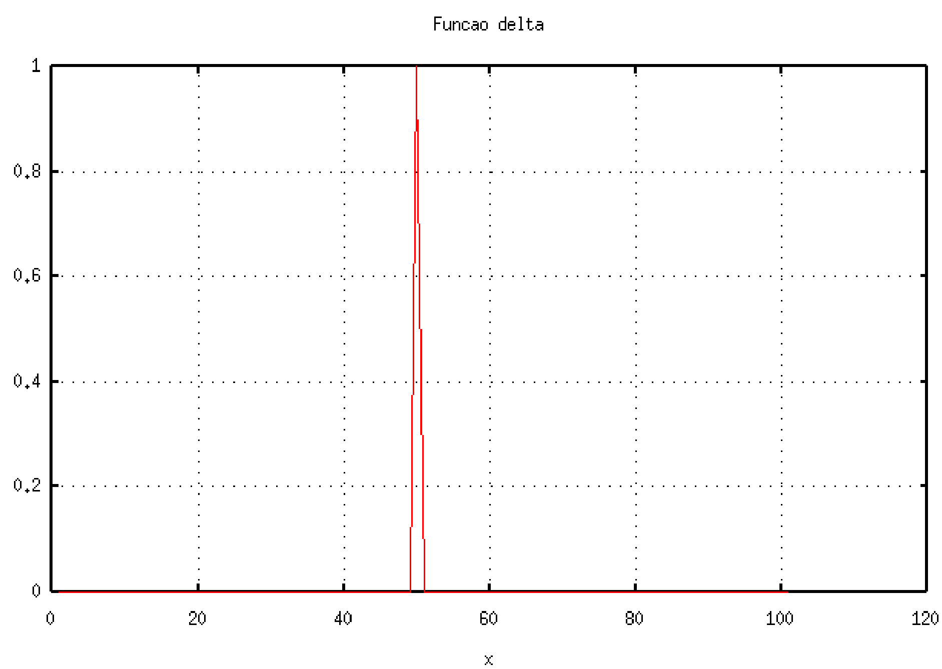
Sendo *rect* = 1 se | x | < 1/2

0 se | x | > 1/2

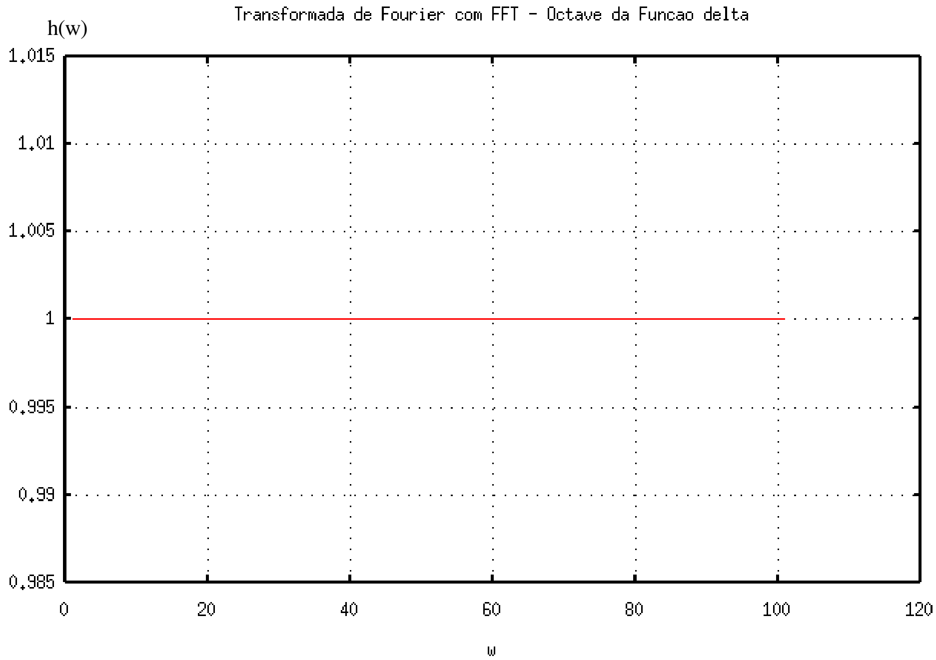
$\mathfrak{F}\{rect(x)\} = Sen(\pi w) / \pi w = Sinc(w)$ **(B.5)**

Anexo C – Octave – Pacote de FFT e FFT inversa – Testes

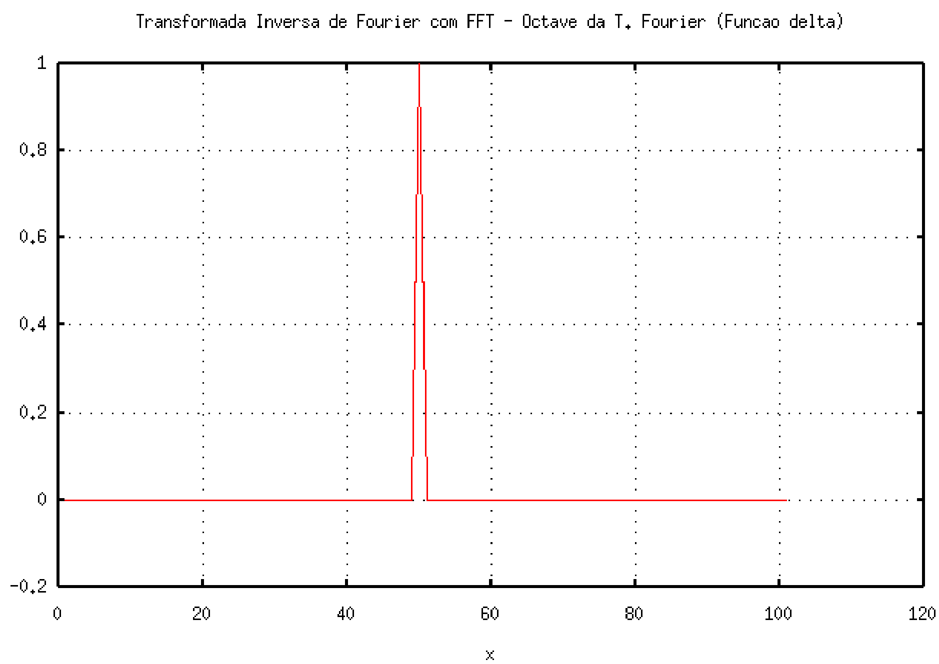
Para a comprovação do correto funcionamento do pacote FFT do programa Octave, realizam –se as Transformadas de Fourier e Transformadas Inversas de Fourier de funções conhecidas, tais como as das equações B.4 e B.5. Os resultado são apresentados a seguir, onde se evidencia o correto funcionamento dos pacotes *fft* e *ifft* do aplicativo Octave.



(a)

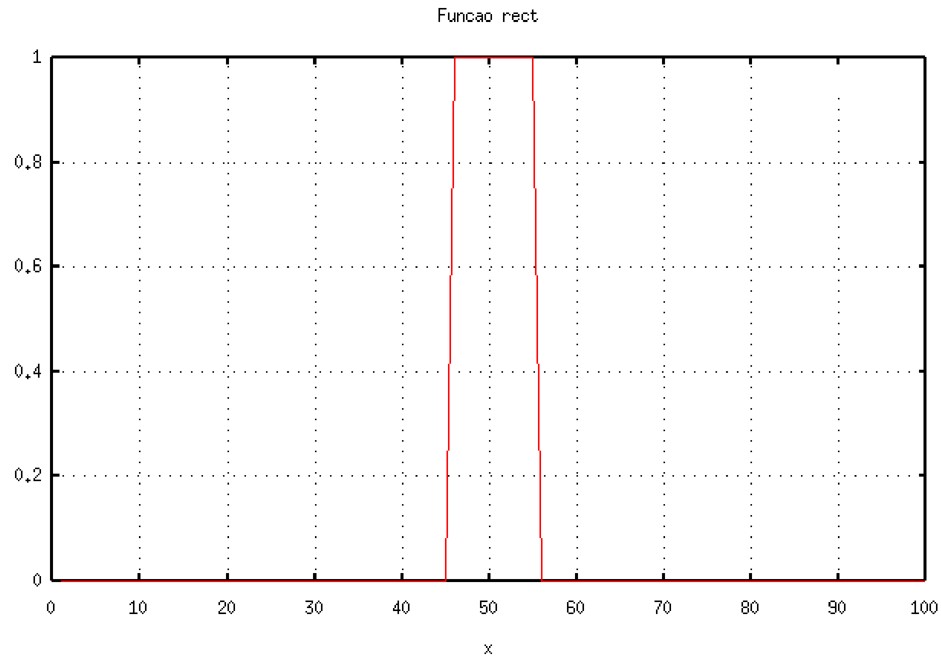


(b)

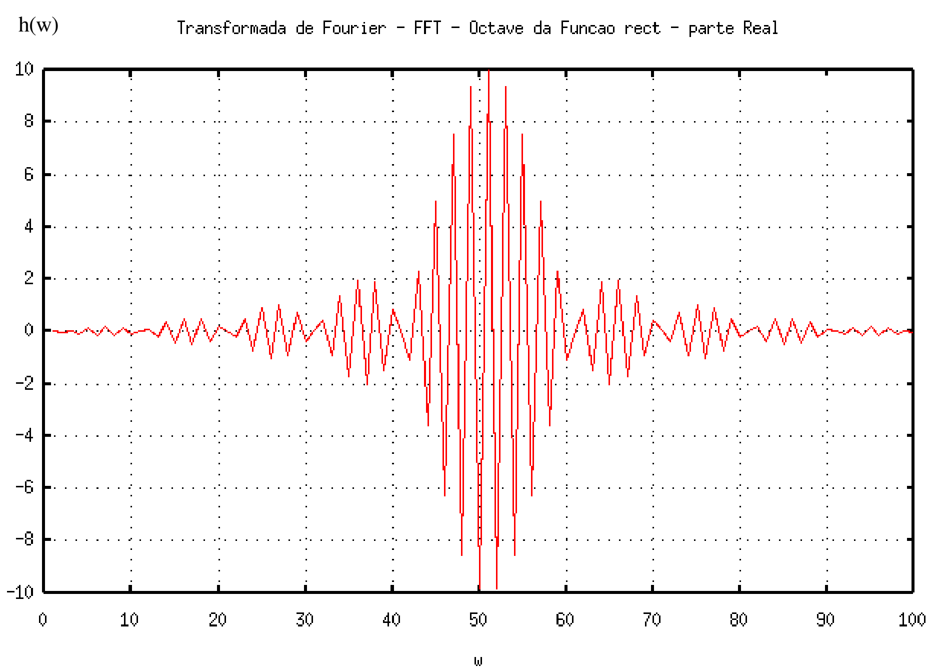


(c)

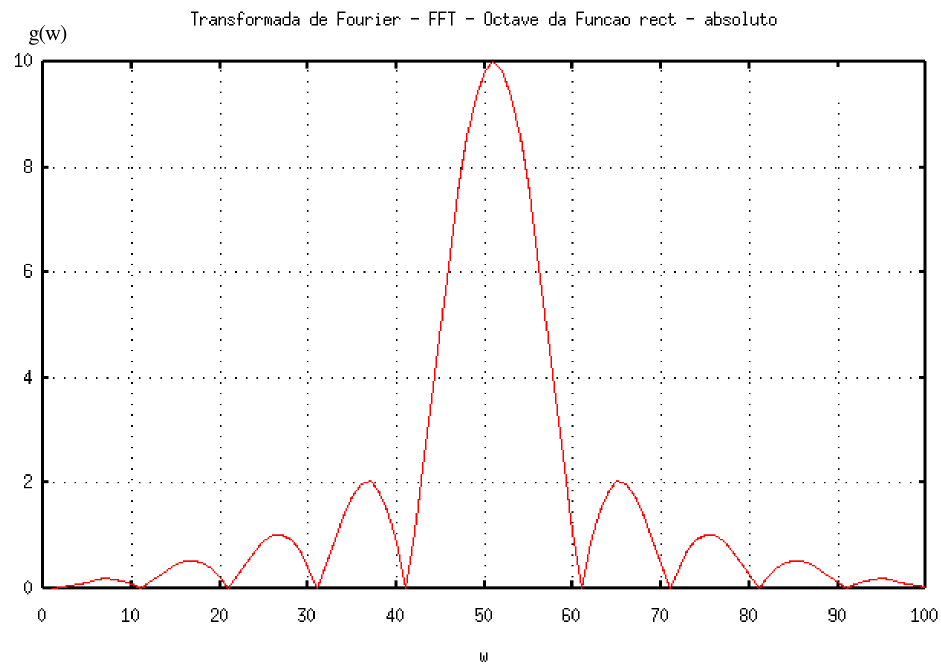
Figura C.1 – (a) Função $\delta(x)$. (b) $h(w) = \mathfrak{F}\{\delta(x)\}$ obtida com o pacote fft do Octave. (c) $\mathfrak{F}^{-1}\{h(w)\}$ obtida com o pacote ifft do Octave, retornando a função inicial $\delta(x)$.


Funcao rect
1
0.8
0.6
0.4
0.2
0
0 10 20 30 40 50 60 70 80 90 100
x

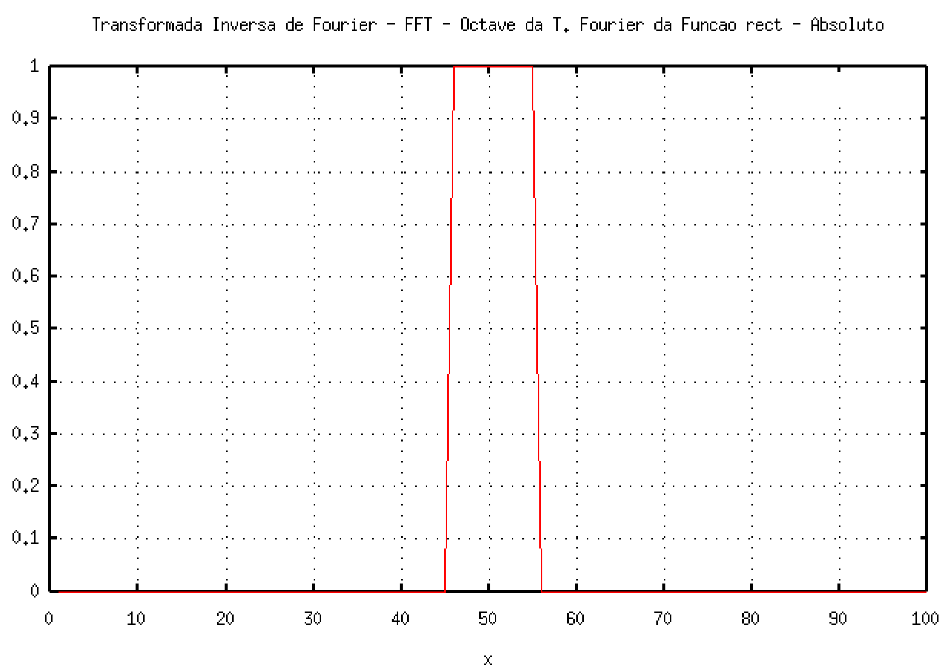
(a)


h(w)
Transformada de Fourier - FFT - Octave da Funcao rect - parte Real
10
8
6
4
2
0
-2
-4
-6
-8
-10
0 10 20 30 40 50 60 70 80 90 100
w

(b)



(c)



(d)

Figura C.2 – (a) Função rect(x) . (b) h(w) = Re $\mathfrak{F}$\{rect(x)\}) = sinc(w) obtida com o pacote fft do Octave. (c) g(w) = Abs (h(w)). (d) Abs ($\mathfrak{F}^{-1}$ \{h(w)\}) obtida com o pacote ifft do Octave, retornando a função inicial rect(x).

Anexo D1 – *framecut.m*

```
function framecut (arquivo,maxpx,maxfr)

        [r b g] = imread(arquivo);

        count = 1;
        L = 0;
        c = 0;

        while count <= maxfr
                 scount = int2str(count);
                 arq = ["fr",scount];
                 FRAMES = fopen(arq,'w+');
                 for i = 1:64
                     for j = 1:64
                     sino (count,j) = maxpx * r(32+L*64,j+c*64);
                     lino (i,count) = maxpx * r(i+L*64,32+c*64);
                     fprintf (FRAMES,"%4.3d ",maxpx*r(i+L*64,j+c*64));
                     endfor
                fprintf(FRAMES,"\n");
                endfor
                fclose (FRAMES);
                count++;

                if c < 7
                   c = c +1;
                else
                   c = 0;
                   L = L + 1;
                endif
        endwhile

        save -ascii  sino sino;
        save -ascii lino lino;
        imagesc (sino);
        imagesc (lino);

endfunction
```

Anexo D2 – *matrizW.c*

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

double A[131][2];

main()

{

void ordem (int left, int right);

const float pi = 3.14159265359;
int TAM,tam,steps,k,n,m,j,K,nn,mm,t,h,H,jm;
float Fiinicial,Fifinal,Fi;
int ni,mi;
double W[65][65][65];
FILE *matriz ;
char s[4] ;

K = 1 ;

printf (" Digite o valor do tamanho da matriz cubica (int) :");
scanf("\n%d",&TAM);
printf (" Digite o valor de fi inicial (rad):");
scanf("\n%f",&Fiinicial);
printf (" Digite o valor de fi final (rad):");
scanf("\n%f",&Fifinal);
printf (" Digite o numero de steps (30 para 6 graus e 60 para 3 graus):");
scanf("\n%d",&steps);
Fi = Fiinicial;
tam=TAM/2;


do {

        if (Fi == 0){
                for (n=1;n<=TAM;n++){
                        for (m=1;m<=TAM;m++){
                                for (j=1;j<=TAM;j++){
                                        if (m==j)
                                             W[n][m][j]= 1.;
                                        else
                                             W[n][m][j]= 0.;}}}

                sprintf (s,"W%d",K) ;
                matriz = fopen(s,"w+");
                for (n=1;n<=TAM;n++) {
                        for (m=1;m<=TAM;m++){
                                for (j=1;j<=TAM;j++){
                                        fprintf (matriz,"%f ",W[n][m][j]);}}
```

```
                        fprintf(matriz,"\n");}
                fclose (matriz);
                printf("%d\n",K) ;
                k = K + steps ;
                sprintf (s,"W%d",k) ;
                matriz = fopen(s,"w+");
                for (n=1;n<=TAM;n++) {
                        for (m=1;m<=TAM;m++){
                                for (j=1;j<=TAM;j++){
                                        fprintf (matriz,"%f ",W[n][m][TAM+1-
j]);}}
                        fprintf(matriz,"\n");}
                fclose (matriz);
                printf("%d\n",k) ;
                K++;
                Fi = Fi + pi/steps;}

        if (Fi > 0 && Fi < pi/2 ){
                for (n=1;n<=TAM;n++){
                        for (m=1;m<=TAM;m++){
                                for (j=1;j<=TAM;j++){
                                             W[n][m][j]= 0;}}}
               for (j=1;j<=tam;j++){
                        for (nn=1;nn<=TAM+1;nn++){
                                n = tam + 1- nn ;
                                A[nn-1][0] = n;
                                A[nn-1][1] = ((j-0.5)+ n*sin(Fi))/cos(Fi);}
                        for (mm=1;mm<=TAM+1;mm++){
                                m = tam +1-mm ;
                                A[TAM+mm][1] = m;
                                A[TAM+mm][0] = (-(j-0.5) +
m*cos(Fi))/sin(Fi);}
                        ordem (0,2*TAM+1);

                        for (t=0;t<=2*TAM+1;t++){
                                if (A[t][0] <=tam & A[t][0]>=-tam &
A[t+1][0]<=tam & A[t+1][0]>=-tam){
                                if (A[t][1] <=tam & A[t][1]>=-tam &
A[t+1][1]<=tam & A[t+1][1]>=-tam){
                                        if (A[t][0]>0 & A[t+1][0] >=0 &
A[t][1]>0 & A[t+1][1]>0){
                                        ni = floor((A[t][0]+A[t+1][0])/2) + 1;
                                        mi = floor((A[t][1]+A[t+1][1])/2) + 1;
                                        W[tam+1-ni][tam+mi][tam+j] =
sqrt(pow(A[t][0]-A[t+1][0],2)+pow(A[t][1]-A[t+1][1],2));}
                                        if (A[t][0]>=0 & A[t+1][0]<0 &
A[t][1]>0 & A[t+1][1]>=0){
                                        ni = ceil((A[t][0]+A[t+1][0])/2) - 1;
                                        mi = floor((A[t][1]+A[t+1][1])/2) + 1;
                                        W[tam-ni][tam+mi][tam+j] =
sqrt(pow(A[t][0]-A[t+1][0],2)+pow(A[t][1]-A[t+1][1],2));}
                                        if (A[t][0]<0 & A[t+1][0]<0 &
A[t][1]>0 & A[t+1][1]>=0){
                                        ni = ceil((A[t][0]+A[t+1][0])/2) - 1;
                                        mi = floor((A[t][1]+A[t+1][1])/2) + 1;
                                        W[tam-ni][tam+mi][tam+j] =
sqrt(pow(A[t][0]-A[t+1][0],2)+pow(A[t][1]-A[t+1][1],2));}
```

```
                                        if (A[t][0]<0 & A[t+1][0]<0 &
A[t][1]>=0 & A[t+1][1]<0){
                                        ni = ceil((A[t][0]+A[t+1][0])/2) - 1;
                                        mi = ceil((A[t][1]+A[t+1][1])/2) - 1;
                                        W[tam-ni][tam+1+mi][tam+j] =
sqrt(pow(A[t][0]-A[t+1][0],2)+pow(A[t][1]-A[t+1][1],2));}
                                        if (A[t][0]<0 & A[t+1][0] <0 &
A[t][1]<0 & A[t+1][1]<0){
                                        ni = ceil((A[t][0]+A[t+1][0])/2) - 1;
                                        mi = ceil((A[t][1]+A[t+1][1])/2) - 1;
                                        W[tam-ni][tam+1+mi][tam+j] =
sqrt(pow(A[t][0]-A[t+1][0],2)+pow(A[t][1]-A[t+1][1],2));}}}}}


               for (j=1;j<=tam;j++){
                        jm = -j;
                        for (nn=1;nn<=TAM+1;nn++){
                                n = tam + 1- nn ;
                                A[nn-1][0] = n;
                                A[nn-1][1] = ((jm+0.5)+ n*sin(Fi))/cos(Fi);}
                        for (mm=1;mm<=TAM+1;mm++){
                                m = tam +1-mm ;
                                A[TAM+mm][1] = m;
                                A[TAM+mm][0] = (-(jm+0.5) +
m*cos(Fi))/sin(Fi);}
                        ordem (0,2*TAM+1);

                        for (t=0;t<=2*TAM+1;t++){
                                if (A[t][0] <=tam & A[t][0]>=-tam &
A[t+1][0]<=tam & A[t+1][0]>=-tam){
                                if (A[t][1] <=tam & A[t][1]>=-tam &
A[t+1][1]<=tam & A[t+1][1]>=-tam){
                                        if (A[t][0]>0 & A[t+1][0] >0 &
A[t][1]>0 & A[t+1][1]>=0){
                                        ni = floor((A[t][0]+A[t+1][0])/2) + 1;
                                        mi = floor((A[t][1]+A[t+1][1])/2) + 1;
                                        W[tam+1-ni][tam+mi][tam+jm+1] =
sqrt(pow(A[t][0]-A[t+1][0],2)+pow(A[t][1]-A[t+1][1],2));}
                                        if (A[t][0]>0 & A[t+1][0]>=0 &
A[t][1]>=0 & A[t+1][1]<0){
                                        ni = floor((A[t][0]+A[t+1][0])/2) + 1;
                                        mi = ceil((A[t][1]+A[t+1][1])/2) - 1;
                                        W[tam+1-ni][tam+1+mi][tam+jm+1] =
sqrt(pow(A[t][0]-A[t+1][0],2)+pow(A[t][1]-A[t+1][1],2));}
                                        if (A[t][0]>0 & A[t+1][0]>=0 &
A[t][1]<=0 & A[t+1][1]<0){
                                        ni = floor((A[t][0]+A[t+1][0])/2) + 1;
                                        mi = ceil((A[t][1]+A[t+1][1])/2) - 1;
                                        W[tam+1-ni][tam+1+mi][tam+jm+1] =
sqrt(pow(A[t][0]-A[t+1][0],2)+pow(A[t][1]-A[t+1][1],2));}
                                        if (A[t][0]>=0 & A[t+1][0]<0 &
A[t][1]<0 & A[t+1][1]<0){
                                        ni = ceil((A[t][0]+A[t+1][0])/2) - 1;
                                        mi = ceil((A[t][1]+A[t+1][1])/2) - 1;
                                        W[tam-ni][tam+mi+1][tam+jm+1] =
sqrt(pow(A[t][0]-A[t+1][0],2)+pow(A[t][1]-A[t+1][1],2));}
```

```
                                        if (A[t][0]<0 & A[t+1][0] <0 &
A[t][1]<0 & A[t+1][1]<0){
                                        ni = ceil((A[t][0]+A[t+1][0])/2) - 1;
                                        mi = ceil((A[t][1]+A[t+1][1])/2) - 1;
                                        W[tam-ni][tam+1+mi][tam+jm+1] =
sqrt(pow(A[t][0]-A[t+1][0],2)+pow(A[t][1]-A[t+1][1],2));}}}}}


                sprintf(s,"W%d",K);
                matriz = fopen(s,"w+");
                for (n=1;n<=TAM;n++){
                        for (m=1;m<=TAM;m++){
                                for (j=1;j<=TAM;j++){
                                        fprintf (matriz,"%lf ",W[n][m][j]);}}
                        fprintf(matriz,"\n");}
                fclose (matriz);
                printf("%d\n",K);
                k = K + steps ;
                sprintf (s,"W%d",k) ;
                matriz = fopen(s,"w+");
                for (n=1;n<=TAM;n++) {
                        for (m=1;m<=TAM;m++){
                                for (j=1;j<=TAM;j++){
                                        fprintf (matriz,"%lf ",W[TAM+1-
n][TAM+1-m][TAM+1-j]);}}
                        fprintf(matriz,"\n");}
                fclose (matriz);
                printf("%d\n",k) ;
                H = steps + 2 - K ;
                sprintf (s,"W%d",H) ;
                matriz = fopen(s,"w+");
                for (n=1;n<=TAM;n++) {
                        for (m=1;m<=TAM;m++){
                                for (j=1;j<=TAM;j++){
                                        fprintf (matriz,"%lf ",W[n][TAM+1-
m][j]);}}
                        fprintf(matriz,"\n");}
                fclose (matriz);
                printf("%d\n",H) ;
                h = H + steps ;
                sprintf (s,"W%d",h) ;
                matriz = fopen(s,"w+");
                for (n=1;n<=TAM;n++) {
                        for (m=1;m<=TAM;m++){
                                for (j=1;j<=TAM;j++){
                                        fprintf (matriz,"%lf ",W[TAM+1-
n][m][TAM+1-j]);}}
                        fprintf(matriz,"\n");}
                fclose (matriz);
                printf("%d\n",h) ;
                K++;
                Fi = Fi + pi/steps;}

        if (Fi == pi/2){
                for (n=1;n<=TAM;n++){
                        for (m=1;m<=TAM;m++){
                                for (j=1;j<=TAM;j++){
```

```
                                        if (n==j)
                                             W[n][m][j]= 1.;
                                        else
                                             W[n][m][j]= 0.;}}}

                sprintf (s,"W%d",K) ;
                matriz = fopen(s,"w+");
                for (n=1;n<=TAM;n++) {
                        for (m=1;m<=TAM;m++){
                                for (j=1;j<=TAM;j++){
                                        fprintf (matriz,"%f ",W[n][m][j]);}}
                        fprintf(matriz,"\n");}
                fclose (matriz);
                printf("%d\n",K) ;
                k = K + steps ;
                sprintf (s,"W%d",k) ;
                matriz = fopen(s,"w+");
                for (n=1;n<=TAM;n++) {
                        for (m=1;m<=TAM;m++){
                                for (j=1;j<=TAM;j++){
                                        fprintf (matriz,"%f ",W[TAM+1-
n][m][TAM+1-j]);}}
                        fprintf(matriz,"\n");}
                fclose (matriz);
                printf("%d\n",k) ;
                K++;
                Fi = Fi + pi/steps;}


            }

while (Fi < Fifinal);

}


void ordem (int left,int right)
{
int dp , particao(int,int);
if (left<right){
        dp=particao(left,right);
        ordem(left,dp-1);
        ordem(dp+1,right);}
}

int particao(int left,int right)
{
int lo,hi;
double pivot0,pivot1;

pivot0=A[left][0];
pivot1=A[left][1];
lo=left;
hi=right+1;

do {
        do
```

```
           hi=hi-1;
        while (lo<hi && A[hi][0]<=pivot0);
        if (lo<hi){{
                A[lo][0]=A[hi][0];
                A[lo][1]=A[hi][1];}
                do
                   lo=lo+1;
                while (lo<hi && A[lo][0]>=pivot0);
                if (lo<hi){
                        A[hi][0]=A[lo][0];
                        A[hi][1]=A[lo][1];}
                }
        }
while(lo<hi);
A[lo][0]=pivot0;
A[lo][1]=pivot1;
return lo;
}
```

Anexo D3 – *fbpw.m*

```
function fbpw (count,maxfr,tam,filtro,a,b,c)


Img = zeros(tam);
I = zeros (tam);
rud=100;
st =0.5/(2*tam);


% "filtro rampa simetrico em matriz coluna

        h=zeros(2*tam,1);

        h(tam/2 +1)=1/4;
        for n=1:16
                h(tam/2+2+2*(n-1))= -1./((2*n-1)*3.1416).^2;
                h(tam/2-2*(n-1))= -1./((2*n-1)*3.1416).^2;
        endfor
        rampa=fft(h);

% "filtro janela com os parametros a,b,c

      %butterworth
      if (filtro == "but")
      for f = 0:2*tam-1
            w(f+1) = (1+a*(f*st))/(sqrt(1+((f*st)/(b*0.5))^(2*c)));
      endfor
      endif


      % sem filtro janela
      if (filtro == " ")
      for f = 1:2*tam
            w(f)=1;
      endfor
      endif



% "sinograma - leitura/confeccao

      load -f sino


while count <= maxfr

      ZP = zeros(1,2*tam);
      A = zeros(2*tam,1);
      Q = zeros(1,2*tam);
      proj=zeros(1,tam);
```

```
      for i= 1:tam
        proj(i)=sinophjl38(count,i);
      endfor


%"ZP


      for j = 1:tam
              ZP(j)=proj(j);
      endfor


% "filtragem rampa /janela

      S = fft(ZP);
      for k = 1:2*tam
          A(k) = S(k) * rampa(k) * w(k) ;
      endfor

      Q = ifft(A);


% "retroprojecao  de Q(theta) com pesos

scount = int2str(count);
      arq = ["W",scount];
      matriz = fopen(arq,'r');
      for n = 1:tam
          for mj = 1:tam*tam
              W(n,mj) = fscanf(matriz,"%lf",'tam,tam*tam');
          endfor
      endfor
      fclose(matriz);

      if(count<=30)
      for mj = 1:tam
         for n = 1:tam
                for j = 1:tam
                    l = mj + (j-1)*tam;
                    if(W(n,l)!=0)
                    I(n,j) = Q(mj+tam/2)*W(n,l);
                    endif
                endfor
         endfor
         Img=Img+I;
         I=zeros(tam);
      endfor
      endif

      if(count>30)
      for mj = 1:tam
         for n = 1:tam
                for j = 1:tam
```

```
                    l = (tam-mj+1) + (j-1)*tam;
                    if(W(n,l)!=0)
                    I(n,j) = Q(mj+tam/2)*W(n,l);
                    endif
                endfor
         endfor
         Img=Img+I;
         I=zeros(tam);
      endfor
      endif

      mem(count)=count;
      plot(mem)

      count++;



endwhile



%" formação da imagem

for n=1:tam
        for m=1:tam
                if (real(Img(n,m))<0)
                        Img(n,m)=imag(Img(n,m));
                endif
        endfor
endfor

Img=abs(Img);
Img(1,1)=0;


imgphjl38but00523=Img;


imagesc(Img);
save -ascii imgphjl38but00523 imgphjl38but00523

endfunction
```

Anexo D4 – *art.m*

```
function art (iter,tcor,count,maxfr,tam,alfa,critconv,h,coef)


ccount=count;
soma=0;
p=zeros(maxfr-count+1,tam);
pc=zeros(1,tam);
somaw=zeros(1,tam);
Dp=zeros(1,tam);

%"sinograma- leitura
load -f sinog
p=sinog;




%"matriz cinza"


for n = count:maxfr
    for m = 1:tam
        soma = soma .+ p(n,m);
    endfor
media = soma/((maxfr-count+1)*tam);
endfor

% "a) art puro

if(h=="art")


for n = 1:tam
    for m = 1:tam
        if (r==1)
        f(n,m) = media;
        else
        f(n,m)=0;
        endif
    endfor
endfor
endif


% "b) hibrido - fbpw"

if(h=="hib")
load -f Imgaf
mat=Imgaf;
max =0;
```

```
for n=1:64
       for m=1:64
              if(mat(n,m)>max)
                    max=mat(n,m);
              endif
       endfor
endfor

f=mat/max;
endif



% "reconstrução ART c/ no. inteiro de iterações"

if (iter != 0 && critconv == 0)

c=1;
Sa = 0;
Sb = 0;
siga = 0;
sigb = 0;

while c <= iter;


sig = zeros(1,tam);
countt=count;
Sa = Sb;
siga=sigb;
sigb = 0;

while countt<=maxfr

scountt = int2str(countt);
      arq = ["W",scountt];
      matriz = fopen(arq,'r');
      for n = 1:tam
          for mj = 1:tam*tam
              W(n,mj) = fscanf(matriz,"%lf",'tam,tam*tam');
          endfor
      endfor
      fclose(matriz);


if (tcor == 'a')

if (countt<=30)

for j = 1:tam
        for n = 1:tam
                for m = 1:tam
                        l = j  + (m - 1) * tam;
                        if (W(n,l)!=0)
                        pc(j) = f(n,m).*W(n,l).+pc(j);
                        somaw(j) = somaw(j).+ W(n,l).^2;
```

```
                        endif
                endfor
        endfor
        Dp(j) = p(countt,j) - pc(j);
        sig(j) = Dp(j).^2 .+ sig(j);
endfor



for j = 1:tam
        for n = 1:tam
                for m = 1:tam
                        l = j + (m - 1) * tam;
                        if(W(n,l)!=0)
                        f(n,m) = f(n,m) + coef*(W(n,l)*Dp(j))/somaw(j);
                        endif
                endfor
        endfor
endfor

endif

if (countt>30)

for j = 1:tam
        for n = 1:tam
                for m = 1:tam
                        l = (tam-j+1)  + (m - 1) * tam;
                        if (W(n,l)!=0)
                        pc(j) = f(n,m).*W(n,l).+pc(j);
                        somaw(j) = somaw(j).+ W(n,l).^2;
                        endif
                endfor
        endfor
        Dp(j) = p(countt,j) - pc(j);
        sig(j) = Dp(j).^2 .+ sig(j);
endfor

for j = 1:tam
        for n = 1:tam
                for m = 1:tam
                        l = (tam-j+1) + (m - 1) * tam;
                        if(W(n,l)!=0)
                        f(n,m) = f(n,m) + coef*(W(n,l)*Dp(j))/somaw(j);
                        endif
                endfor
        endfor
endfor

endif


pc=zeros(1,tam);
somaw=zeros(1,tam);
Dp=zeros(1,tam);

endif
```

```
if (tcor == 'm')

if (countt<=30)

for j = 1:tam
        for n = 1:tam
                for m = 1:tam
                        l = j  + (m - 1) * tam;
                        if (W(n,l)!=0)
                        pc(j) = f(n,m).*W(n,l).+pc(j);
                        endif
                endfor
        endfor
        Dp(j) = p(countt,j) - pc(j);
        sig(j) = Dp(j).^2 .+ sig(j);
endfor

for j = 1:tam
        for n = 1:tam
                for m = 1:tam
                        l = j + (m - 1) * tam;
                        if (pc(j) == 0 || W(n,l)==0)
                        f(n,m) = f(n,m) ;
                        else
                        f(n,m) = f(n,m) + f(n,m)*Dp(j)/pc(j) ;
                        endif
                endfor
        endfor
endfor

endif

if (countt>30)

for j = 1:tam
        for n = 1:tam
                for m = 1:tam
                        l = (tam-j+1)  + (m - 1) * tam;
                        if (W(n,l)!=0)
                        pc(j) = f(n,m).*W(n,l).+pc(j);
                        endif
                endfor
        endfor
        Dp(j) = p(countt,j) - pc(j);
        sig(j) = Dp(j).^2 .+ sig(j);
endfor

for j = 1:tam
        for n = 1:tam
                for m = 1:tam
                        l = (tam-j+1) + (m - 1) * tam;
                        if (pc(j) == 0 || W(n,l)==0)
                        f(n,m) = f(n,m) ;
                        else
                        f(n,m) = f(n,m) + f(n,m)*Dp(j)/pc(j) ;
```

```
                        endif
                endfor
        endfor
endfor

endif


pc = zeros(1,tam);
Dp = zeros(1,tam);

endif

countt++;

endwhile

for n = 1:tam
        for m = 1:tam
                if (f(n,m)!=0)
                Sb = - (1/log(tam*tam))*(f(n,m)/media)*log(f(n,m)/media) + Sb;
                endif
        endfor
endfor

critconvent = abs(Sa-Sb)

for j = 1:tam
sig(j) = sqrt(sig(j)/(maxfr - count + 1));
if (sig(j)>sigb)
sigb=sig(j);
endif
endfor

critconvsig = abs(siga-sigb)

alfaS(c) = critconvent/abs(Sb);
alfaSIG(c) = critconvsig/abs(sigb);

save -ascii alfaS
save -ascii alfaSIG


z(c)=c;
plot(z)

%"gravacao de matrizes intermediarias

q=f;

for n = 1:tam
        for m = 1:tam
                if (q(n,m)<0)
                q(n,m)=0;
                endif
        endfor
endfor
```

```
scount = int2str(c);
arq = ["fphjgr",scount];
FRAMES = fopen(arq,'w+');
        for i = 1:64
                for j = 1:64
                     fprintf (FRAMES,"%lf ",q(i,j));
                     endfor
                fprintf(FRAMES,"\n");
        endfor
fclose (FRAMES);


c++;

endwhile

endif

%"reconstrução ART c/ criterio de convergencia - entropia"

if (iter == 0 && critconv == "ent")

c=0;
Sa = 0;
Sb = 0;
critconvent=1;
siga = 0;
sigb = 0;


while critconvent >= alfa*abs(Sa)

countt=count;
c++;
sig = zeros(1,tam);
Sa=Sb;
siga=sigb;
sigb = 0;

while countt<=maxfr

scountt = int2str(countt);
      arq = ["W",scountt];
      matriz = fopen(arq,'r');
      for n = 1:tam
          for mj = 1:tam*tam
              W(n,mj) = fscanf(matriz,"%lf",'tam,tam*tam');
          endfor
      endfor
      fclose(matriz);

if (tcor == 'a')

if (countt<=30)
```

```
for j = 1:tam
        for n = 1:tam
                for m = 1:tam
                        l = j  + (m - 1) * tam;
                        if (W(n,l)!=0)
                        pc(j) = f(n,m).*W(n,l).+pc(j);
                        somaw(j) = somaw(j).+ W(n,l).^2;
                        endif
                endfor
        endfor
        Dp(j) = p(countt,j) - pc(j);
        sig(j) = Dp(j).^2 .+ sig(j);
endfor

for j = 1:tam
        for n = 1:tam
                for m = 1:tam
                        l = j + (m - 1) * tam;
                        if(W(n,l)!=0)
                        f(n,m) = f(n,m) + coef*(W(n,l)*Dp(j))/somaw(j);
                        endif
                endfor
        endfor
endfor

endif

if (countt>30)

for j = 1:tam
        for n = 1:tam
                for m = 1:tam
                        l = (tam-j+1)  + (m - 1) * tam;
                        if (W(n,l)!=0)
                        pc(j) = f(n,m).*W(n,l).+pc(j);
                        somaw(j) = somaw(j).+ W(n,l).^2;
                        endif
                endfor
        endfor
        Dp(j) = p(countt,j) - pc(j);
        sig(j) = Dp(j).^2 .+ sig(j);
endfor

for j = 1:tam
        for n = 1:tam
                for m = 1:tam
                        l = (tam-j+1) + (m - 1) * tam;
                        if(W(n,l)!=0)
                        f(n,m) = f(n,m) + coef*(W(n,l)*Dp(j))/somaw(j);
                        endif
                endfor
        endfor
endfor

endif
```

```
pc=zeros(1,tam);
somaw=zeros(1,tam);
Dp=zeros(1,tam);

endif


if (tcor == 'm')

if (countt<=30)

for j = 1:tam
        for n = 1:tam
                for m = 1:tam
                        l = j  + (m - 1) * tam;
                        if (W(n,l)!=0)
                        pc(j) = f(n,m).*W(n,l).+pc(j);
                        endif
                endfor
        endfor
        Dp(j) = p(countt,j) - pc(j);
        sig(j) = Dp(j).^2 .+ sig(j);
endfor

for j = 1:tam
        for n = 1:tam
                for m = 1:tam
                        l = j + (m - 1) * tam;
                        if (pc(j) == 0 || W(n,l)==0)
                        f(n,m) = f(n,m) ;
                        else
                        f(n,m) = f(n,m) + f(n,m)*Dp(j)/pc(j) ;
                        endif
                endfor
        endfor
endfor

endif

if (countt>30)

for j = 1:tam
        for n = 1:tam
                for m = 1:tam
                        l = (tam-j+1)  + (m - 1) * tam;
                        if (W(n,l)!=0)
                        pc(j) = f(n,m).*W(n,l).+pc(j);
                        endif
                endfor
        endfor
        Dp(j) = p(countt,j) - pc(j);
        sig(j) = Dp(j).^2 .+ sig(j);
endfor

for j = 1:tam
        for n = 1:tam
                for m = 1:tam
```

```
                        l = (tam-j+1) + (m - 1) * tam;
                        if (pc(j) == 0 || W(n,l)==0)
                        f(n,m) = f(n,m) ;
                        else
                        f(n,m) = f(n,m) + f(n,m)*Dp(j)/pc(j) ;
                        endif
                endfor
        endfor
endfor

endif


pc = zeros(1,tam);
Dp = zeros(1,tam);

endif

countt++;


endwhile

for n = 1:tam
        for m = 1:tam
                if (f(n,m)!=0)
                Sb = - (1/log(tam*tam))*(f(n,m)/media)*log(f(n,m)/media) + Sb;
                endif
        endfor
endfor

critconvent = abs(Sa-Sb)

for j = 1:tam
sig(j) = sqrt(sig(j)/(maxfr - count + 1));
if (sig(j)>sigb)
sigb=sig(j);
endif
endfor

alfaS(c) = critconvent/abs(Sb);
alfaSIG(c) = critconvsig/abs(sigb);

save -ascii alfaS
save -ascii alfaSIG


z(c)=c;
plot(z)


endwhile

%"gravacao de matrizes intermediarias

q=f;
```

```
for n = 1:tam
        for m = 1:tam
                if (q(n,m)<0)
                q(n,m)=0;
                endif
        endfor
endfor


scount = int2str(c);
arq = ["fphjgr",scount];
FRAMES = fopen(arq,'w+');
        for i = 1:64
                for j = 1:64
                     fprintf (FRAMES,"%lf ",q(i,j));
                     endfor
                fprintf(FRAMES,"\n");
        endfor
fclose (FRAMES);


c

endif

%"reconstrução ART c/ criterio de convergencia - sigma"

if (iter == 0 && critconv == "sig")

c=0;
Sa = 0;
Sb = 0;
critconvsig=1;
sig = zeros(1,tam);
siga = 0;
sigb = 0;

while critconvsig >= alfa*siga

countt=count;
sig = zeros(1,tam);
siga = sigb;
sigb = 0;
Sa = Sb;
c++;

while countt<=maxfr

scountt = int2str(countt);
      arq = ["W",scountt];
      matriz = fopen(arq,'r');
      for n = 1:tam
          for mj = 1:tam*tam
              W(n,mj) = fscanf(matriz,"%lf",'tam,tam*tam');
          endfor
      endfor
      fclose(matriz);
```

```
if (tcor == 'a')

if (countt<=30)

for j = 1:tam
        for n = 1:tam
                for m = 1:tam
                        l = j  + (m - 1) * tam;
                        if (W(n,l)!=0)
                        pc(j) = f(n,m).*W(n,l).+pc(j);
                        somaw(j) = somaw(j).+ W(n,l).^2;
                        endif
                endfor
        endfor
        Dp(j) = p(countt,j) - pc(j);
        sig(j) = Dp(j).^2 .+ sig(j);
endfor

for j = 1:tam
        for n = 1:tam
                for m = 1:tam
                        l = j + (m - 1) * tam;
                        if(W(n,l)!=0)
                        f(n,m) = f(n,m) + coef*(W(n,l)*Dp(j))/somaw(j);
                        endif
                endfor
        endfor
endfor

endif

if (countt>30)

for j = 1:tam
        for n = 1:tam
                for m = 1:tam
                        l = (tam-j+1)  + (m - 1) * tam;
                        if (W(n,l)!=0)
                        pc(j) = f(n,m).*W(n,l).+pc(j);
                        somaw(j) = somaw(j).+ W(n,l).^2;
                        endif
                endfor
        endfor
        Dp(j) = p(countt,j) - pc(j);
        sig(j) = Dp(j).^2 .+ sig(j);
endfor

for j = 1:tam
        for n = 1:tam
                for m = 1:tam
                        l = (tam-j+1) + (m - 1) * tam;
                        if(W(n,l)!=0)
                        f(n,m) = f(n,m) + coef*(W(n,l)*Dp(j))/somaw(j);
                        endif
                endfor
        endfor
```

```
endfor

endif


pc=zeros(1,tam);
somaw=zeros(1,tam);
Dp=zeros(1,tam);

endif


if (tcor == 'm')

if (countt<=30)

for j = 1:tam
        for n = 1:tam
                for m = 1:tam
                        l = j  + (m - 1) * tam;
                        if (W(n,l)!=0)
                        pc(j) = f(n,m).*W(n,l).+pc(j);
                        endif
                endfor
        endfor
        Dp(j) = p(countt,j) - pc(j);
        sig(j) = Dp(j).^2 .+ sig(j);
endfor

for j = 1:tam
        for n = 1:tam
                for m = 1:tam
                        l = j + (m - 1) * tam;
                        if (pc(j) == 0 || W(n,l)==0)
                        f(n,m) = f(n,m) ;
                        else
                        f(n,m) = f(n,m) + f(n,m)*Dp(j)/pc(j) ;
                        endif
                endfor
        endfor
endfor

endif

if (countt>30)

for j = 1:tam
        for n = 1:tam
                for m = 1:tam
                        l = (tam-j+1)  + (m - 1) * tam;
                        if (W(n,l)!=0)
                        pc(j) = f(n,m).*W(n,l).+pc(j);
                        endif
                endfor
        endfor
        Dp(j) = p(countt,j) - pc(j);
        sig(j) = Dp(j).^2 .+ sig(j);
```

```
endfor

for j = 1:tam
        for n = 1:tam
                for m = 1:tam
                        l = (tam-j+1) + (m - 1) * tam;
                        if (pc(j) == 0 || W(n,l)==0)
                        f(n,m) = f(n,m) ;
                        else
                        f(n,m) = f(n,m) + f(n,m)*Dp(j)/pc(j) ;
                        endif
                endfor
        endfor
endfor

endif


pc = zeros(1,tam);
Dp = zeros(1,tam);

endif

countt++;



endwhile

for n = 1:tam
        for m = 1:tam
                if (f(n,m)!=0)
                Sb = - (1/log(tam*tam))*(f(n,m)/media)*log(f(n,m)/media) + Sb;
                endif
        endfor
endfor

critconvent = abs(Sa-Sb)

for j = 1:tam
sig(j) = sqrt(sig(j)/(maxfr - count + 1));
if (sig(j)>sigb)
sigb=sig(j);
endif
endfor

alfaS(c) = critconvent/abs(Sb);
alfaSIG(c) = critconvsig/abs(sigb);

save -ascii alfaS
save -ascii alfaSIG


z(c)=c;
plot(z)
```

```
%"gravacao de matrizes intermediarias

q=f;

for n = 1:tam
        for m = 1:tam
                if (q(n,m)<0)
                q(n,m)=0;
                endif
        endfor
endfor


scount = int2str(c);
arq = ["fphjgr",scount];
FRAMES = fopen(arq,'w+');
        for i = 1:64
                for j = 1:64
                     fprintf (FRAMES,"%lf ",q(i,j));
                     endfor
                fprintf(FRAMES,"\n");
        endfor
fclose (FRAMES);


endwhile



c

endif


for n = 1:tam
        for m = 1:tam
                if (f(n,m)<0)
                f(n,m)=0;
                endif
        endfor
endfor


plot([alfaS,alfaSIG])


imagesc(f);



endfunction
```

Anexo D5 – *sirt.m*

```
function sirt (corte,iter,tcor,count,maxfr,tam,alfa,critconv)


ccount=count;
soma=0;
p=zeros(maxfr-count+1,tam);
pc=zeros(maxfr-count+1,tam);


%sinograma leitura

load sino
p=sino;

%matriz  "cinza"

for n = count:maxfr
    for m = 1:tam
        soma = soma .+ p(n,m);
    endfor
media = soma/((maxfr-count+1)*tam);
endfor


%for n = 1:tam
 %for m = 1:tam
  %      f(n,m) = media;
   %endfor
%endfor

% leitura da matriz de Peso W1 a W30
load -f W

load -f Lj
load -f Nj


% "reconstrução SIRT c/ no. inteiro de iterações"

if (iter != 0 && critconv == 0)


Sa = 0;
Sb = 0;
siga = 0;
sigb = 0;
somaa = 0;
somab = 0;
somac = 0;
somad = 0;
```

```
countt=count;

while countt <= iter;

sig = zeros(maxfr-count+1,tam);
pc=zeros(maxfr-count+1,tam);
Sa = Sb;
Sb=0;
siga=sigb;
sigb = 0;

for n = 1:tam
        for m = 1:tam
                for r = 1:(maxfr-count+1)/2
                        for j = 1:tam
                                l = j  + (m - 1) * tam + tam*tam*(r-1);
                                t = (tam-j+1) + (m-1)*tam + tam*tam*(r-1);
                                if (W(n,l)!=0 || W(n,t)!=0)
                                pc(r,j) = f(n,m).*W(n,l).+pc(r,j);
                                pc(r+(maxfr-count+1)/2,j) =
f(n,m).*W(n,t).+pc(r+(maxfr-count+1)/2,j);
                                endif
                                sig(r,j) = (p(r,j) - pc(r,j)).^2 +
(p(r+(maxfr-count+1)/2,j) - pc(r+(maxfr-count+1)/2,j)).^2;
                        endfor
                endfor

                for r = 1:(maxfr-count+1)/2
                        for j = 1:tam
                                l = j + (m - 1) * tam + tam*tam*(r-1);
                                t = (tam-j+1) + (m-1)*tam + tam*tam*(r-1);
                                if (W(n,l)!=0 || W(n,t)!=0)
                                somaa = p(r,j)*W(n,l) + p(r+(maxfr-
count+1)/2,j)*W(n,t) + somaa;
                                somab = pc(r,j)*W(n,l) + pc(r+(maxfr-
count+1)/2,j)*W(n,t) + somab;
                                somac = Lj(r,j)*W(n,l) + Lj(r+(maxfr-
count+1)/2,j)*W(n,t) + somac;
                                somad = Nj(r,j)*W(n,l) + Nj(r+(maxfr-
count+1)/2,j)*W(n,t) + somad;
                                endif
                        endfor
                endfor

                if (tcor == "a")
                        f(n,m) = f(n,m) + somaa/somac - somab/somad;
                endif

                if (tcor == "m")
                        if (somab!=0)
                                f(n,m) = f(n,m)*(somaa/somac)*(somad/somab);
                        endif
                endif

                somaa = 0;
                somab = 0;
                somac = 0;
```

```
                somad = 0;
                pc=zeros(maxfr-count+1,tam);

        endfor
endfor


for n = 1:tam
        for m = 1:tam
                if (f(n,m)!=0)
                Sb = - (1/log(tam*tam))*(f(n,m)/media)*log(f(n,m)/media) + Sb;
                endif
        endfor
endfor

critconvent = abs(Sa-Sb)

for r=1:maxfr-count+1
        for j = 1:tam
                sig(r,j) = sqrt(sig(r,j)/((maxfr-count+1)*tam));
                if (sig(r,j)>sigb)
                        sigb=sig(r,j);
                endif
        endfor
endfor

critconvsig = abs(siga-sigb)

S(countt) = Sb;
SIG(countt) = sigb;
Scrit(countt)=critconvent/abs(Sa);
SIGcrit(countt)=critconvsig/abs(siga);

save -ascii S
save -ascii SIG
save -ascii Scrit
save -ascii SIGcrit

z(countt)=countt;
plot(z)

countt++;

endwhile

endif

% "reconstrução SIRT c/ criterio de convergencia - entropia"

if (iter == 0 && critconv == "ent")

Sa = 0;
Sb = 0;
siga = 0;
sigb = 0;
somaa = 0;
somab = 0;
```

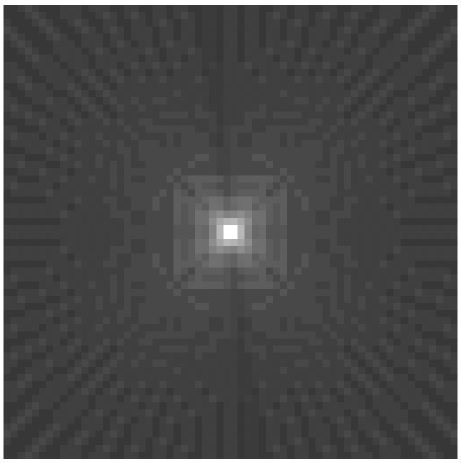
```
somac = 0;
somad = 0;
countt = 0;
critconvent = 1;

while critconvent >= alfa*abs(Sa)

sig = zeros(maxfr-count+1,tam);
Sa = Sb;
Sb=0;
siga=sigb;
sigb = 0;
countt++;

for n = 1:tam
        for m = 1:tam
                for r = 1:(maxfr-count+1)/2
                        for j = 1:tam
                                l = j  + (m - 1) * tam + tam*tam*(r-1);
                                t = (tam-j+1) + (m-1)*tam + tam*tam*(r-1);
                                if (W(n,l)!=0 || W(n,t)!=0)
                                pc(r,j) = f(n,m).*W(n,l).+pc(r,j);
                                pc(r+(maxfr-count+1)/2,j) =
f(n,m).*W(n,t).+pc(r+(maxfr-count+1)/2,j);
                                endif
                                sig(n,m) = (p(r,j) - pc(r,j)).^2 +
(p(r+(maxfr-count+1)/2,j) - pc(r+(maxfr-count+1)/2,j)).^2;
                        endfor
                endfor

                for r = 1:(maxfr-count+1)/2
                        for j = 1:tam
                                l = j + (m - 1) * tam + tam*tam*(r-1);
                                t = (tam-j+1) + (m-1)*tam + tam*tam*(r-1);
                                if (W(n,l)!=0 || W(n,t)!=0)
                                somaa = p(r,j)*W(n,l) + p(r+(maxfr-
count+1)/2,j)*W(n,t) + somaa;
                                somab = pc(r,j)*W(n,l) + pc(r+(maxfr-
count+1)/2,j)*W(n,t) + somab;
                                somac = Lj(r,j)*W(n,l) + Lj(r+(maxfr-
count+1)/2,j)*W(n,t) + somac;
                                somad = Nj(r,j)*W(n,l) + Nj(r+(maxfr-
count+1)/2,j)*W(n,t) + somad;
                                endif
                        endfor
                endfor

                if (tcor == "a")
                        f(n,m) = f(n,m) + somaa/somac - somab/somad;
                endif

                if (tcor == "m")
                        if (somab!=0)
                                f(n,m) = f(n,m)*(somaa/somac)*(somad/somab);
                        endif
                endif
```

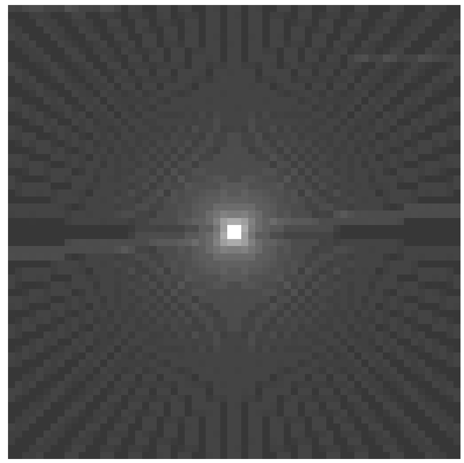
```
                somaa = 0;
                somab = 0;
                somac = 0;
                somad = 0;
                pc=zeros(maxfr-count+1,tam);

        endfor
endfor


for n = 1:tam
        for m = 1:tam
                if (f(n,m)!=0)
                Sb = - (1/log(tam*tam))*(f(n,m)/media)*log(f(n,m)/media) + Sb;
                endif
        endfor
endfor

critconvent = abs(Sa-Sb)

for r=1:maxfr-count+1
        for j = 1:tam
                sig(r,j) = sqrt(sig(r,j)/((maxfr - count + 1)*tam));
                if (sig(r,j)>sigb)
                        sigb=sig(r,j);
                endif
        endfor
endfor

critconvsig = abs(siga-sigb)

S(countt) = critconvent;
SIG(countt) = critconvsig;


endwhile

countt

endif

% "reconstrução SIRT c/ criterio de convergencia - sigma"

if (iter == 0 && critconv == "sig")

Sa = 0;
Sb = 0;
siga = 0;
sigb = 0;
somaa = 0;
somab = 0;
somac = 0;
somad = 0;
countt = 0;
critconvsig = 1;
sig = zeros(maxfr-count+1,tam);
```

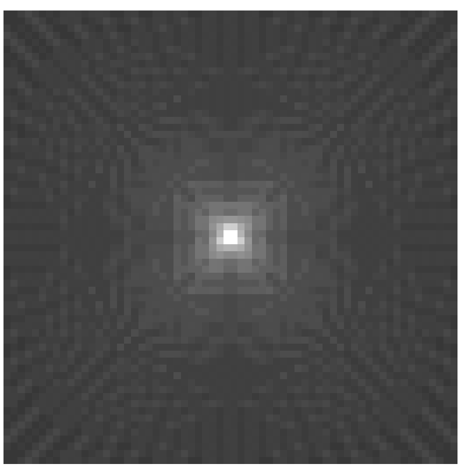
```
while critconvsig >= alfa*siga

sig = zeros(maxfr-count+1,tam);
Sa = Sb;
Sb=0;
siga = sigb;
sigb = 0;
countt++;

for n = 1:tam
        for m = 1:tam
                for r = 1:(maxfr-count+1)/2
                        for j = 1:tam
                                l = j  + (m - 1) * tam + tam*tam*(r-1);
                                t = (tam-j+1) + (m-1)*tam + tam*tam*(r-1);
                                if (W(n,l)!=0 || W(n,t)!=0)
                                pc(r,j) = f(n,m).*W(n,l).+pc(r,j);
                                pc(r+(maxfr-count+1)/2,j) =
f(n,m).*W(n,t).+pc(r+(maxfr-count+1)/2,j);
                                endif
                                sig(n,m) = (p(r,j) - pc(r,j)).^2 +
(p(r+(maxfr-count+1)/2,j) - pc(r+(maxfr-count+1)/2,j)).^2;
                        endfor
                endfor

                for r = 1:(maxfr-count+1)/2
                        for j = 1:tam
                                l = j + (m - 1) * tam + tam*tam*(r-1);
                                t = (tam-j+1) + (m-1)*tam + tam*tam*(r-1);
                                if (W(n,l)!=0 || W(n,t)!=0)
                                somaa = p(r,j)*W(n,l) + p(r+(maxfr-
count+1)/2,j)*W(n,t) + somaa;
                                somab = pc(r,j)*W(n,l) + pc(r+(maxfr-
count+1)/2,j)*W(n,t) + somab;
                                somac = Lj(r,j)*W(n,l) + Lj(r+(maxfr-
count+1)/2,j)*W(n,t) + somac;
                                somad = Nj(r,j)*W(n,l) + Nj(r+(maxfr-
count+1)/2,j)*W(n,t) + somad;
                                endif
                        endfor
                endfor

                if (tcor == "a")
                        f(n,m) = f(n,m) + somaa/somac - somab/somad;
                endif

                if (tcor == "m")
                        if (somab!=0)
                                f(n,m) = f(n,m)*(somaa/somac)*(somad/somab);
                        endif
                endif

                somaa = 0;
                somab = 0;
                somac = 0;
                somad = 0;
                pc=zeros(maxfr-count+1,tam);
```

```
        endfor
endfor


for n = 1:tam
        for m = 1:tam
                if (f(n,m)!=0)
                Sb = - (1/log(tam*tam))*(f(n,m)/media)*log(f(n,m)/media) + Sb;
                endif
        endfor
endfor

critconvent = abs(Sa-Sb)

for r=1:maxfr-count+1
        for j = 1:tam
                sig(r,j) = sqrt(sig(r,j)/((maxfr - count + 1)*tam));
                if (sig(r,j)>sigb)
                        sigb=sig(r,j);
                endif
        endfor
endfor

critconvsig = abs(siga-sigb)

S(countt) = critconvent;
SIG(countt) = critconvsig;

endwhile

countt

endif


for n = 1:tam
        for m = 1:tam
                if (f(n,m)<0)
                f(n,m)=0;
                endif
        endfor
endfor

save -ascii S S
save -ascii SIG SIG

plot([S,SIG])


imagesc(f);

save -ascii f f


endfunction
```

Anexo E – Exemplos de Reconstruções de Cortes Tomográficos com Matrizes *W* assimétricas







Anexo F – *artx.m*

```
function artx (iter,tcor,count,maxfr,tam,alfa,critconv,h,coef)


gset grid
ccount=count;
soma=0;
p=zeros(maxfr-count+1,tam);
pc=zeros(1,tam);
somaw=zeros(1,tam);
Dp=zeros(1,tam);

%"sinograma- leitura
load -f sino
p=sino;


%"matriz cinza"

maxp =0;

for n=count:maxfr
       for m=1:tam
              if(p(n,m)>maxp)
                    maxp=p(n,m);
              endif
       endfor
endfor

p=p/maxp;

for n =count:maxfr
    for m = 1:tam
        soma = soma .+ p(n,m);
    endfor
media = soma/((maxfr-count+1)*tam);
endfor



% "a) art puro

if(h=="art")

for n = 1:tam
    for m = 1:tam
        f(n,m) = media;
    endfor
endfor
endif
```

```
% "b) hibrido - fbpw"

if(h=="hib")
load -f imgefdg02l26but001512
mat=imgefdg02l26but001512;


max =0;

for n=1:64
       for m=1:64
              if(mat(n,m)>max)
                    max=mat(n,m);
              endif
       endfor
endfor

p=p*maxp;
f=mat/max;
endif



% "reconstrução ART c/ no. inteiro de iterações"

if (iter != 0 && critconv == 0)

c=1;
Sa = 0;
Sb = 0;
siga = 0;
sigb = 0;

while c <= iter;


sig = zeros(1,tam);
Sa = Sb;
siga=sigb;
sigb = 0;


for tt=1:15

for t =0:3

countt = tt + t*15;


scountt = int2str(countt);
      arq = ["W",scountt];
      matriz = fopen(arq,'r');
      for n = 1:tam
          for mj = 1:tam*tam
              W(n,mj) = fscanf(matriz,"%lf",'tam,tam*tam');
          endfor
```

```
      endfor
      fclose(matriz);


if (tcor == 'a')

if (countt<=30)

for j = 1:tam
        for n = 1:tam
                for m = 1:tam
                        l = j  + (m - 1) * tam;
                        if (W(n,l)!=0)
                        pc(j) = f(n,m).*W(n,l).+pc(j);
                        somaw(j) = somaw(j).+ W(n,l).^2;
                        endif
                endfor
        endfor
        Dp(j) = p(countt,j) - pc(j);
        sig(j) = Dp(j).^2 .+ sig(j);

endfor

 gset grid
 xx=(1- exp(-abs(Dp*coef)));
 plot([Dp,xx])

for j = 1:tam
        for n = 1:tam
                for m = 1:tam
                        l = j + (m - 1) * tam;
                        if(W(n,l)!=0)
                        if (h=="hib")
                        f(n,m) = f(n,m) + coef*(W(n,l)*Dp(j))/somaw(j);
                        else
                        f(n,m) = f(n,m) + (1- exp(-
abs(Dp(j)*coef)))*(W(n,l)*Dp(j))/somaw(j);
                        endif
                        endif
                endfor
        endfor
endfor
      imagesc(f);
endif

if (countt>30)

for j = 1:tam
        for n = 1:tam
                for m = 1:tam
                        l = (tam-j+1)  + (m - 1) * tam;
                        if (W(n,l)!=0)
                        pc(j) = f(n,m).*W(n,l).+pc(j);
                        somaw(j) = somaw(j).+ W(n,l).^2;
                        endif
                endfor
        endfor
```

```
        Dp(j) = p(countt,j) - pc(j);
        sig(j) = Dp(j).^2 .+ sig(j);
endfor

for j = 1:tam
        for n = 1:tam
                for m = 1:tam
                        l = (tam-j+1) + (m - 1) * tam;
                        if(W(n,l)!=0)
                        if (h=="hib")
                        f(n,m) = f(n,m) + coef*(W(n,l)*Dp(j))/somaw(j);
                        else
                        f(n,m) = f(n,m) + (1- exp(-
abs(Dp(j)*coef)))*(W(n,l)*Dp(j))/somaw(j);
                        endif
                        endif
                endfor
        endfor
endfor

endif


pc=zeros(1,tam);
somaw=zeros(1,tam);
Dp=zeros(1,tam);

endif


if (tcor == 'm')

if (countt<=30)

for j = 1:tam
        for n = 1:tam
                for m = 1:tam
                        l = j  + (m - 1) * tam;
                        if (W(n,l)!=0)
                        pc(j) = f(n,m).*W(n,l).+pc(j);
                        endif
                endfor
        endfor
        Dp(j) = p(countt,j) - pc(j);
        sig(j) = Dp(j).^2 .+ sig(j);
endfor

for j = 1:tam
        for n = 1:tam
                for m = 1:tam
                        l = j + (m - 1) * tam;
                        if (pc(j) == 0 || W(n,l)==0)
                        f(n,m) = f(n,m) ;
                        else
                        f(n,m) = f(n,m) + f(n,m)*Dp(j)/pc(j) ;
                        endif
                endfor
```

```
        endfor
endfor

endif

if (countt>30)

for j = 1:tam
        for n = 1:tam
                for m = 1:tam
                        l = (tam-j+1)  + (m - 1) * tam;
                        if (W(n,l)!=0)
                        pc(j) = f(n,m).*W(n,l).+pc(j);
                        endif
                endfor
        endfor
        Dp(j) = p(countt,j) - pc(j);
        sig(j) = Dp(j).^2 .+ sig(j);
endfor

for j = 1:tam
        for n = 1:tam
                for m = 1:tam
                        l = (tam-j+1) + (m - 1) * tam;
                        if (pc(j) == 0 || W(n,l)==0)
                        f(n,m) = f(n,m) ;
                        else
                        f(n,m) = f(n,m) + f(n,m)*Dp(j)/pc(j) ;
                        endif
                endfor
        endfor
endfor

endif


pc = zeros(1,tam);
Dp = zeros(1,tam);

endif


endfor

endfor

for n = 1:tam
        for m = 1:tam
                if (f(n,m)!=0)
                Sb = - (1/log(tam*tam))*(f(n,m)/media)*log(f(n,m)/media) + Sb;
                endif
        endfor
endfor

critconvent = abs(Sa-Sb)

for j = 1:tam
```

```
sig(j) = sqrt(sig(j)/(maxfr - count + 1));
if (sig(j)>sigb)
sigb=sig(j);
endif
endfor

critconvsig = abs(siga-sigb)

alfaS(c) = critconvent/abs(Sb);
alfaSIG(c) = critconvsig/abs(sigb);

save -ascii alfaS
save -ascii alfaSIG


z(c)=c;
plot(z)

%"gravacao de matrizes intermediarias

q=f;

for n = 1:tam
        for m = 1:tam
                if (q(n,m)<0)
                q(n,m)=0;
                endif
        endfor
endfor


scount = int2str(c);
arq = ["hteste",scount];
FRAMES = fopen(arq,'w+');
        for i = 1:64
                for j = 1:64
                     fprintf (FRAMES,"%f ",q(i,j));
                     endfor
                fprintf(FRAMES,"\n");
        endfor
fclose (FRAMES);


c++;

endwhile


endif

%"reconstrução ART c/ criterio de convergencia - entropia"

if (iter == 0 && critconv == "ent")

c=0;
Sa = 0;
Sb = 0;
```

```
critconvent=1;
siga = 0;
sigb = 0;


while critconvent >= alfa*abs(Sa)

countt=count;
c++;
sig = zeros(1,tam);
Sa=Sb;
siga=sigb;
sigb = 0;

for tt=1:15

for t =0:3

countt = tt + t*15;


scountt = int2str(countt);
      arq = ["W",scountt];
      matriz = fopen(arq,'r');
      for n = 1:tam
          for mj = 1:tam*tam
              W(n,mj) = fscanf(matriz,"%lf",'tam,tam*tam');
          endfor
      endfor
      fclose(matriz);

if (tcor == 'a')

if (countt<=30)

for j = 1:tam
        for n = 1:tam
                for m = 1:tam
                        l = j  + (m - 1) * tam;
                        if (W(n,l)!=0)
                        pc(j) = f(n,m).*W(n,l).+pc(j);
                        somaw(j) = somaw(j).+ W(n,l).^2;
                        endif
                endfor
        endfor
        Dp(j) = p(countt,j) - pc(j);
        sig(j) = Dp(j).^2 .+ sig(j);
endfor

for j = 1:tam
        for n = 1:tam
                for m = 1:tam
                        l = j + (m - 1) * tam;
                        if(W(n,l)!=0)
                        if (h=="hib")
                        f(n,m) = f(n,m) + coef*(W(n,l)*Dp(j))/somaw(j);
                        else
```

```
                        f(n,m) = f(n,m) + (1- exp(-
abs(Dp(j)*coef)))*(W(n,l)*Dp(j))/somaw(j);
                        endif
                        endif
                endfor
        endfor
endfor

endif

if (countt>30)

for j = 1:tam
        for n = 1:tam
                for m = 1:tam
                        l = (tam-j+1)  + (m - 1) * tam;
                        if (W(n,l)!=0)
                        pc(j) = f(n,m).*W(n,l).+pc(j);
                        somaw(j) = somaw(j).+ W(n,l).^2;
                        endif
                endfor
        endfor
        Dp(j) = p(countt,j) - pc(j);
        sig(j) = Dp(j).^2 .+ sig(j);
endfor

for j = 1:tam
        for n = 1:tam
                for m = 1:tam
                        l = (tam-j+1) + (m - 1) * tam;
                        if(W(n,l)!=0)
                        if (h=="hib")
                        f(n,m) = f(n,m) + coef*(W(n,l)*Dp(j))/somaw(j);
                        else
                        f(n,m) = f(n,m) + (1- exp(-
abs(Dp(j)*coef)))*(W(n,l)*Dp(j))/somaw(j);
                        endif
                        endif
                endfor
        endfor
endfor

endif


pc=zeros(1,tam);
somaw=zeros(1,tam);
Dp=zeros(1,tam);

endif


if (tcor == 'm')

if (countt<=30)

for j = 1:tam
```

```
        for n = 1:tam
                for m = 1:tam
                        l = j  + (m - 1) * tam;
                        if (W(n,l)!=0)
                        pc(j) = f(n,m).*W(n,l).+pc(j);
                        endif
                endfor
        endfor
        Dp(j) = p(countt,j) - pc(j);
        sig(j) = Dp(j).^2 .+ sig(j);
endfor

for j = 1:tam
        for n = 1:tam
                for m = 1:tam
                        l = j + (m - 1) * tam;
                        if (pc(j) == 0 || W(n,l)==0)
                        f(n,m) = f(n,m) ;
                        else
                        f(n,m) = f(n,m) + f(n,m)*Dp(j)/pc(j) ;
                        endif
                endfor
        endfor
endfor

endif

if (countt>30)

for j = 1:tam
        for n = 1:tam
                for m = 1:tam
                        l = (tam-j+1)  + (m - 1) * tam;
                        if (W(n,l)!=0)
                        pc(j) = f(n,m).*W(n,l).+pc(j);
                        endif
                endfor
        endfor
        Dp(j) = p(countt,j) - pc(j);
        sig(j) = Dp(j).^2 .+ sig(j);
endfor

for j = 1:tam
        for n = 1:tam
                for m = 1:tam
                        l = (tam-j+1) + (m - 1) * tam;
                        if (pc(j) == 0 || W(n,l)==0)
                        f(n,m) = f(n,m) ;
                        else
                        f(n,m) = f(n,m) + f(n,m)*Dp(j)/pc(j) ;
                        endif
                endfor
        endfor
endfor

endif
```

```
pc = zeros(1,tam);
Dp = zeros(1,tam);

endif

endfor

endfor

for n = 1:tam
        for m = 1:tam
                if (f(n,m)!=0)
                Sb = - (1/log(tam*tam))*(f(n,m)/media)*log(f(n,m)/media) + Sb;
                endif
        endfor
endfor

critconvent = abs(Sa-Sb)

for j = 1:tam
sig(j) = sqrt(sig(j)/(maxfr - count + 1));
if (sig(j)>sigb)
sigb=sig(j);
endif
endfor

critconvsig = abs(siga-sigb);

alfaS(c) = critconvent/abs(Sb);
alfaSIG(c) = critconvsig/abs(sigb);

save -ascii alfaS
save -ascii alfaSIG



plot(alfaS)


endwhile

%"gravacao de matrizes intermediarias

q=f;

for n = 1:tam
        for m = 1:tam
                if (q(n,m)<0)
                q(n,m)=0;
                endif
        endfor
endfor


scount = int2str(c);
arq = ["fphjgr",scount];
```

```
FRAMES = fopen(arq,'w+');
        for i = 1:64
                for j = 1:64
                     fprintf (FRAMES,"%f ",q(i,j));
                     endfor
                fprintf(FRAMES,"\n");
        endfor
fclose (FRAMES);


c

endif

%"reconstrução ART c/ criterio de convergencia - sigma"

if (iter == 0 && critconv == "sig")

c=0;
Sa = 0;
Sb = 0;
critconvsig=1;
sig = zeros(1,tam);
siga = 0;
sigb = 0;

while critconvsig >= alfa*siga

countt=count;
sig = zeros(1,tam);
siga = sigb;
sigb = 0;
Sa = Sb;
c++;


for tt=1:15

for t =0:3

countt = tt + t*15;


scountt = int2str(countt);
      arq = ["W",scountt];
      matriz = fopen(arq,'r');
      for n = 1:tam
          for mj = 1:tam*tam
              W(n,mj) = fscanf(matriz,"%lf",'tam,tam*tam');
          endfor
      endfor
      fclose(matriz);

if (tcor == 'a')

if (countt<=30)
```

```
for j = 1:tam
        for n = 1:tam
                for m = 1:tam
                        l = j  + (m - 1) * tam;
                        if (W(n,l)!=0)
                        pc(j) = f(n,m).*W(n,l).+pc(j);
                        somaw(j) = somaw(j).+ W(n,l).^2;
                        endif
                endfor
        endfor
        Dp(j) = p(countt,j) - pc(j);
        sig(j) = Dp(j).^2 .+ sig(j);
endfor

for j = 1:tam
        for n = 1:tam
                for m = 1:tam
                        l = j + (m - 1) * tam;
                        if(W(n,l)!=0)
                        if (h=="hib")
                        f(n,m) = f(n,m) + coef*(W(n,l)*Dp(j))/somaw(j);
                        else
                        f(n,m) = f(n,m) + (1- exp(-
abs(Dp(j)*coef)))*(W(n,l)*Dp(j))/somaw(j);
                        endif
                        endif
                endfor
        endfor
endfor

endif

if (countt>30)

for j = 1:tam
        for n = 1:tam
                for m = 1:tam
                        l = (tam-j+1)  + (m - 1) * tam;
                        if (W(n,l)!=0)
                        pc(j) = f(n,m).*W(n,l).+pc(j);
                        somaw(j) = somaw(j).+ W(n,l).^2;
                        endif
                endfor
        endfor
        Dp(j) = p(countt,j) - pc(j);
        sig(j) = Dp(j).^2 .+ sig(j);
endfor

for j = 1:tam
        for n = 1:tam
                for m = 1:tam
                        l = (tam-j+1) + (m - 1) * tam;
                        if(W(n,l)!=0)
                        if (h=="hib")
                        f(n,m) = f(n,m) + coef*(W(n,l)*Dp(j))/somaw(j);
                        else
```

```
                        f(n,m) = f(n,m) + (1- exp(-
abs(Dp(j)*coef)))*(W(n,l)*Dp(j))/somaw(j);
                        endif
                        endif
                endfor
        endfor
endfor

endif


pc=zeros(1,tam);
somaw=zeros(1,tam);
Dp=zeros(1,tam);

endif


if (tcor == 'm')

if (countt<=30)

for j = 1:tam
        for n = 1:tam
                for m = 1:tam
                        l = j  + (m - 1) * tam;
                        if (W(n,l)!=0)
                        pc(j) = f(n,m).*W(n,l).+pc(j);
                        endif
                endfor
        endfor
        Dp(j) = p(countt,j) - pc(j);
        sig(j) = Dp(j).^2 .+ sig(j);
endfor

for j = 1:tam
        for n = 1:tam
                for m = 1:tam
                        l = j + (m - 1) * tam;
                        if (pc(j) == 0 || W(n,l)==0)
                        f(n,m) = f(n,m) ;
                        else
                        f(n,m) = f(n,m) + f(n,m)*Dp(j)/pc(j) ;
                        endif
                endfor
        endfor
endfor

endif

if (countt>30)

for j = 1:tam
        for n = 1:tam
                for m = 1:tam
                        l = (tam-j+1)  + (m - 1) * tam;
                        if (W(n,l)!=0)
```

```
                        pc(j) = f(n,m).*W(n,l).+pc(j);
                        endif
                endfor
        endfor
        Dp(j) = p(countt,j) - pc(j);
        sig(j) = Dp(j).^2 .+ sig(j);
endfor

for j = 1:tam
        for n = 1:tam
                for m = 1:tam
                        l = (tam-j+1) + (m - 1) * tam;
                        if (pc(j) == 0 || W(n,l)==0)
                        f(n,m) = f(n,m) ;
                        else
                        f(n,m) = f(n,m) + f(n,m)*Dp(j)/pc(j) ;
                        endif
                endfor
        endfor
endfor

endif


pc = zeros(1,tam);
Dp = zeros(1,tam);

endif

endfor

endfor

for n = 1:tam
        for m = 1:tam
                if (f(n,m)!=0)
                Sb = - (1/log(tam*tam))*(f(n,m)/media)*log(f(n,m)/media) + Sb;
                endif
        endfor
endfor

critconvent = abs(Sa-Sb);

for j = 1:tam
sig(j) = sqrt(sig(j)/(maxfr - count + 1));
if (sig(j)>sigb)
sigb=sig(j);
endif
endfor


critconvsig = abs(siga-sigb);

alfaS(c) = critconvent/abs(Sb);
alfaSIG(c) = critconvsig/abs(sigb);

save -ascii alfaS
```

```
save -ascii alfaSIG



plot(alfaSIG)


%"gravacao de matrizes intermediarias

q=f;

for n = 1:tam
        for m = 1:tam
                if (q(n,m)<0)
                q(n,m)=0;
                endif
        endfor
endfor


scount = int2str(c);
arq = ["hibefdg02l26r025x",scount];
FRAMES = fopen(arq,'w+');
        for i = 1:64
                for j = 1:64
                     fprintf (FRAMES,"%f ",q(i,j));
                     endfor
                fprintf(FRAMES,"\n");
        endfor
fclose (FRAMES);


endwhile



c

endif


for n = 1:tam
        for m = 1:tam
                if (f(n,m)<0)
                f(n,m)=0;
                endif
        endfor
endfor


imagesc(f);



endfunction
```

Referências Bibliográficas

1. Cassen, B. , Curtis L., *Instrumentation for I131 in Medical Studies*, Clifton Reed and Raymond Libby Nucleonics, Vol. 9 , No. 2 , 1951, pg.46
2. Anger, H. O. , *Scintillation camera*, Rev. Sci. Instr. ,Vol.29,No. 27,1958
3. Anger, H.O. , *Gamma Ray and Positron Scintillation Cameras*, Nucleonics, Vol. 21, No. 10, 1963, pg. 56
4. Willians , L.E. , Nuclear Medical Physics – Vol. II , chp.4 , pg. 90
5. Barret, H.H. & Swindell, W. , *Radiological Imaging – Vol. II*, chp.6
6. Barret, H.H. & Swindell, W. , *Radiological Imaging – Vol. II*, chp.6 , pg.361
7. Sorenson, J.A. & Phelps,M.E., *Physics in Nuclear Medicine*,Chp.19, pg.404
8. Radon, J. – *Über die Bestimmung von Funktionen durch ihre integralwerte längs gewisser mannigfaltigkeiten*, Ber Saechsis Akad Wissensch, 69, 1917, pgs. 262 - 277
9. Bracewell, R.N. - Aust. J. Phys. 9, 1956, pg.198
10. DeRosier, D.J. e Klug, A. - Nature 217, 1968, pg.130
11. Hounsfield, G.N. - Br. J. Radiol. 46, 1973, pg. 1016
12. Liang, Z. - Med. Phys. Vol. 20, No. 4, 1993, pg. 1097
13. Chien-Min, K., Xiaochuan, P., Chin-Tu, C. - Med. Phys. Vol 25, No. 5, 1998, pg. 600
14. Glatting, G., Wuchenauer, M., Reske, S.N. - Med. Phys. Vol 26, No. 9, 1999, pg. 1838
15. Winchell, H.S. , Horst, W.D., Braun , L. , et al., *N-isotropil [^{123}I] – p – iodoamphetamine : single pass brain uptake and washout : binding to brain sinaptasomes ; and localization in dog and monkey brain*, J. Nucl, Med. , Vol. 21, 1980, pg. 947

16. Kunng, H.F. , Traposh, K.M. and Blau , M., *A new brain perfusion imaging agent : [I123] – HIPDM : N,N,N´- Trimetyl-N´-[2-Hidroxy-3Metyl-5-Iodobenzyl]-1,3-Propanediamine*, J. Nucl. Med. , Vol. 24 , 1983, pg. 66
17. Bushberg, J.T. , Seibert, J.A., Leidholdt, E.M. & Boone, J.M. , *The essential Physics of Medical Imaging*, Chp.17
18. Early, P.J. & Sodee, D.B., *Principles and Practice of Nuclear Medicine*, Chp.8
19. Sorenson, J.A. & Phelps,M.E., *Physics in Nuclear Medicine*, Chp.15, pg.301
20. IAEA – International Atomic Energy Agency , Tecdoc 602A
21. Knoll, G.F. , *Radiation Detection and Measurement*, Chp.8
22. Hofstadter, R., Phys. Rev. 74, 100, 1948
23. Sorenson, J.A. & Phelps,M.E., *Physics in Nuclear Medicine*, Chp.4, pg.74
24. Bushberg, J.T. , Seibert, J.A., Leidholdt, E.M. & Boone, J.M. , *The essential Physics of Medical Imaging,* Chp.17, pg. 531
25. Bushberg, J.T. , Seibert, J.A., Leidholdt, E.M. & Boone, J.M. , *The essential Physics of Medical Imaging,* Chp.17, pg.546
26. Bushberg, J.T. , Seibert, J.A., Leidholdt, E.M. & Boone, J.M. , *The essential Physics of Medical Imaging*, Chp.17, pg.545
27. Willians , L.E. , *Nuclear Medical Physics – Vol. II* , Chp.4, pg.108
28. Sorenson, J.A. & Phelps,M.E., *Physics in Nuclear Medicine*, Chp.18
29. Groch, M.W. & Erwin, W. D., *Spect in the Year 2000 – Basic Principles.html*, fig.4 – página da internet
30. Rosenfeld, A. & Kak , A.C., *Digital Picture Processing - Vol. I* , Chp.8
31. Cambridge University , *Numerical Recipes in C : The Art of Scientific Computing*, Chp.12
32. Rosenfeld, A. & Kak , A.C., *Digital Picture Processing - Vol. I* , Chp.8, pg.373
33. Rosenfeld, A. & Kak , A.C., *Digital Picture Processing - Vol. I* , Chp.8, pg.386
34. Smith, S.W., *The Scientist and Engineer's Guide to Digital Signal Processing*, Chp.18

35. Rosenfeld, A. & Kak , A.C., *Digital Picture Processing - Vol. I* , Chp.8, pg.358

36. Rosenfeld, A. & Kak , A.C., *Digital Picture Processing - Vol. I* , Chp.8, pg.376

37. Rosenfeld, A. & Kak , A.C., *Digital Picture Processing - Vol. I* , Chp.8, pg.379

38. Rosenfeld, A. & Kak , A.C., *Digital Picture Processing - Vol. I* , Chp.8, pg.380

39. Rosenfeld, A. & Kak , A.C., *Digital Picture Processing - Vol. I* , Chp.8, pg.381

40. Rosenfeld, A. & Kak , A.C., *Digital Picture Processing - Vol. I* , Chp.8, pg.382

41. Rosenfeld, A. & Kak , A.C., *Digital Picture Processing - Vol. I* , Chp.8, pg.415

42. Brooks, R. A. & Di Chiro, G , *Principles of Computer Assisted Tomography (CAT) in Radiographic and Radioisotopic Imaging*, – Phys. Med. Biol., Vol. 21, 1976, pg. 689

43. Barret, H.H. & Swindell, W. , *Radiological Imaging – Vol. II*, Chp.6

44. Figura de A. Todd-Pokropek, Medical Physics Departament University College London, London WC1E 6BT

45. Willians , L.E. , *Nuclear Medical Physics – Vol. III* , Chp.6

46. Barret, H.H. & Swindell, W. , *Radiological Imaging - Vol. I*, Apêndices A/B