



UNIVERSIDADE FEDERAL DE PERNAMBUCO
DEPARTAMENTO DE ENERGIA NUCLEAR
PROGRAMA DE PÓS-GRADUAÇÃO EM TECNOLOGIAS ENERGÉTICAS E NUCLEARES
(UFPE – DEN – PROTEN)

GEOVANE VITOR VASCONCELOS

PSIRT:

A Técnica Iterativa de Reconstrução Tomográfica Baseada em Sistemas de Partículas

(Recife, Março/2013)

GEOVANE VITOR VASCONCELOS

PSIRT:

A Técnica Iterativa de Reconstrução Tomográfica Baseada em Sistemas de Partículas

Tese de Doutorado submetida à Coordenação da Pós-Graduação em Tecnologias Energéticas e Nucleares, do Departamento de Energia Nuclear, como parte dos requisitos para obtenção de título de Doutor em Energia Nuclear, área de concentração Aplicações de Radioisótopos na Indústria e Medicina, subárea Química Industrial.

Orientador: Dr. Carlos Costa Dantas.

Co-orientador: Dr. Silvio de Barros Melo.

(Recife, Março/2013)

GEOVANE VITOR VASCONCELOS

PSIRT:

A Técnica Iterativa de Reconstrução Tomográfica Baseada em Sistemas de Partículas

Tese de Doutorado submetida à Coordenação da Pós-Graduação em Tecnologias Energéticas e Nucleares, do Departamento de Energia Nuclear, como parte dos requisitos para obtenção de título de Doutor em Energia Nuclear, área de concentração Aplicações de Radioisótopos na Indústria e Medicina, subárea Química Industrial.

Data da defesa: 08 de março de 2013.

Resultado: Aprovado. Reprovado.

BANCA EXAMINADORA:

Prof. Dr. Carlos Costa Dantas (DEN/UFPE).

Prof. Dr. Emerson Alexandre de Oliveira Lima (UPE).

Prof. Dr. José Wilson Vieira (IFPE e UPE).

Prof. Dr. Rajendra Narain (DEN/UFPE).

Prof. Dr. Cláudio Tadeu Cristino (DEINFO/UFPE).

À minha falecida mãe, Maria José (Zezé), razão de todos os meus valores positivos como ser humano e pensante.

AGRADECIMENTOS

Ao todo poderoso Deus.

Ao companheiro Geraldo Vitor, sem o qual provavelmente eu não teria conseguido, tio que fez dessa conquista um fardo particular, na mais sincera ajuda logística, moral e amiga. Idem em relação a sua esposa Neide, em muitos momentos exercendo o papel da mãe que eu não tenho mais.

Ao meu pai Genival Quintino, para seu orgulho e essência. Em extensão, à sua digníssima esposa Irene.

Aos meus filhos Vitor e Vitória, nítidos motivos de minha insistência, principalmente nos momentos nos quais a única saída era desistir. Em extensão, à minha esposa Patrícia.

Ao professor Silvio de Barros Melo, pela orientação decisiva, sábia, oportuna e amiga.

Ao professor Carlos Costa Dantas, pela orientação paciente, rigorosa, exigente e correta.

Ao professor Otávio, pela alegria e atenção tecnológica e administrativa que sempre prestou a todos os alunos do grupo, além de um bom papo no nosso cafezinho de cada dia.

Aos colegas do informal GREEN (Grupo de Estudo de Energia Nuclear): Eduardo, Cláudia, Moacir, Márcio e Patryk, pessoas especiais nessa minha jornada nobre em busca de evolução.

Aos meus irmãos 3G (Geane, Girleide e George) e seus conjugues, Samuel Araújo, Adriano Aragão e Deiseane Benigno, respectivamente, pelo apoio e incentivo.

Às minhas sobrinhas Géssica Vasconcelos, Ana Beatriz, Angélica Vasconcelos e Cecília Maria (esta que vai chegar agora em junho de 2013) e ao meu sobrinho Sávio, e em especial a Heloísa Maria (Loloca), que me fez aprender sobre muitos fenômenos nucleares ao pular constantemente em minha barriga quimicamente heterogênea.

À minha querida tia Maria Alice (Bocão), minha primeira segunda mãe, de todos os tempos e de todas as minhas verdades. Em extensão a seu esposo Espínola Sabino e meus primos Breno e Luciano e suas respectivas famílias.

À minha amada tia Iraci, fiel representante do carinho e amor alicerces do início de minhas conquistas.

A todos os colegas do Grupo de Radioquímica, em especial a Lendival e Eric.

A todos os meus outros familiares, que de alguma maneira serviram de inspiração para que eu vencesse esta etapa da minha vida.

Aos professores Josemar Viana, Antônio Berto Machado e Wellington Pereira, eternos exemplos para o meu segmento profissional docente.

À torcida do Santa Cruz Futebol Clube, simplesmente “A Mais Apaixonada do Brasil.

A todos os professores e professoras do DEN, que direta ou indiretamente me ajudaram nessa longa e difícil jornada.

A todos os funcionários e funcionárias do DEN, em especial a Zacarias (o Papai Cobra), Edivaldo, Elias, Lia, Josenildo (Carlinhos Bala) e Claudenice, com os quais formei uma sincera amizade.

“Minha impressão é que o mundo se dilui em entretenimento, ambição e guerras, enquanto muitos morrem de fome e de pestes pela falta de atenção da humanidade” (Dalai Lama).

PSIRT:

A Técnica Iterativa de Reconstrução Tomográfica Baseada em Sistemas de Partículas

RESUMO

A tomografia computadorizada tem se consolidado como uma alternativa bastante eficiente na investigação do processo industrial do refino de petróleo. O seu produto final, a imagem reconstruída, é objeto de estudo em uma área de intenso interesse, e inclui métodos de reconstrução advindos dos mais distintos campos de pesquisa. Esta tese apresenta um método de reconstrução iterativo baseado numa abstração chamada em computação gráfica de *sistemas de partículas*, o PSIRT. O método foi concebido originalmente para atuar em arranjos de tomógrafo com limitado número de trajetórias de raios gama, um requerimento de extrema importância para as aplicações industriais. O PSIRT representa fragmentos do objeto a ser reconstruído como partículas sujeitas a um campo de forças em torno das trajetórias. Tal campo é gerado pelas diferenças entre os dados de entrada de atenuação da trajetória e os dados computados numa dada iteração, baseados na quantidade de partículas presentes num envelope em torno da trajetória. O posicionamento do conjunto de partículas é iterativamente modificado para que essas diferenças atinjam um valor aceitável. Algumas iterações de ajustes de parâmetros são aplicadas de forma alternada com as iterações da dinâmica das partículas. Praticamente todos os diversos resultados da aplicação do PSIRT em dados simulados e também em dados experimentais sugerem uma superioridade qualitativa em relação a métodos algébricos consagrados. Os sinogramas simulados foram produzidos a partir de diferentes tipos de *fantomas*, incluindo duas configurações estacionárias observadas no estudo de fluidodinâmica, e as comparações foram feitas com o algoritmo ART e alguns de seus variantes, e incluiu uma métrica concebida para capturar formas geométricas. Uma das características mais marcantes do PSIRT é a ausência de ruídos de ordem global, um dos problemas enfrentados por outros métodos em número limitado de trajetórias. Os sinogramas de dados experimentais foram produzidos num tomógrafo de alta velocidade de raios gama (tomógrafo HSGT), como também num tomógrafo de 1ª geração. Embora os dados ruidosos tenham afetado sensivelmente o desempenho do PSIRT, verificou-se que este fato se deu menos no método do que nos métodos algébricos. Tais considerações presentes nesta tese conferem ao PSIRT a condição de um método bastante competitivo em aplicações de tomografia industrial.

Palavras-chave: Sistema de Partículas, Método Iterativo de Reconstrução de Imagem, Tomografia Industrial, Processo de Refino de Petróleo.

PSIRT:
The Iterative Reconstruction Technique Based on Particle Systems

ABSTRACT

Computed tomography has been established as a very efficient alternative in the investigation of the industrial process of petroleum refining. The final product, the reconstructed image, is an object of study in an area of intense interest, and includes reconstruction methods stemming from widely different fields of research. This thesis presents a reconstruction method based on an iterative abstraction known in computer graphics as *particle systems*, called *PSIRT*. The method was originally designed to operate in tomographic arrangements with limited amount of gamma-ray beams, a very important requirement for industrial applications. PSIRT represents object fragments as particles subject to a force field around the trajectories, and generated by the differences between the trajectory's attenuation data input and the corresponding data computed in a given iteration, based on the amount of particles present in the envelope around the trajectory. The positioning of the set of particles is iteratively modified so that these differences reach an acceptable value. Some iterations of parameter adjustments are applied alternately with the iterations of the particles dynamics. Virtually all of the various results of applying PSIRT to the simulated data, as well as to some experimental data suggest a qualitative superiority over some algebraic methods. The simulated sinograms were produced from different types of phantoms, including two stationary configurations observed in the study of fluid dynamics, and the comparisons were made against the ART algorithm and some of its variants, which included a novel metric designed to capture geometric shapes. One of the most striking features of the PSIRT is the absence of global noise, one of the problems faced by other methods in a limited number of trajectories. The experimental sinograms data were produced in a high-speed gamma-ray scanner (HSGT tomograph), as well as a 1st generation CT scanner. Although the noisy data have significantly affected the performance of PSIRT, it was found that this fact occurred less intensively than in the algebraic methods. Such considerations present in this thesis give the PSIRT the condition of a very competitive method in tomography for industrial applications.

Keywords: Particle System, Iterative Image Reconstruction, Industrial Tomography, Process of Oil Refining.

LISTA DE FIGURAS

Figura 1 – Riser do tipo FCC.	27
Figura 2 – Tomógrafo a Frio do Laboratório de Radioquímica da UFPE.	30
Figura 3 – Tomógrafo HSGT.	30
Figura 4 – Visão geral de reflexão, refração e espalhamento.	31
Figura 5 – Atenuação da radiação num material de espessura x .	32
Figura 6 – Riser, suas paredes e uma corda do raio gama atravessando substâncias.	33
Figura 7 – Rotação θ do sistema de coordenadas s para o sistema de coordenadas t .	36
Figura 8 – Discretização com uma matriz 10x10, destacando o pixel $a_{7,7}$ e um raio gama de projeção θ .	39
Figura 9 – Geração de um sinograma a partir do ponto (x', y') .	42
Figura 10 – Exemplo de uma projeção <i>fan bean</i> .	43
Figura 11 – Fantoma de uma imagem de 50x50 pixels	44
Figura 12 – Discretização destacando peso a , raio de espessura l , e as células f_j .	45
Figura 13 – Exemplo da aplicação do GC com dois hexágonos cada.	53
Figura 14 – Exemplo da aplicação do GC com três hexágonos cada.	54
Figura 15 – (a) Uma distribuição uniformemente aleatória de partículas na configuração 3x7, (b) o PSIRT atingindo o equilíbrio, (c) e o resultado final com a discretização em hexágonos.	60
Figura 16 – Conversão de uma partícula em pixel natural (discretização).	60
Figura 17 – Possíveis posições das partículas no PSIRT em equilíbrio.	62
Figura 18 – Reconstrução de imagem de partículas de posições “A” e “B”.	65
Figura 19 – Visão das regiões adjacentes dos raios que são cobertas pelo pixel natural.	66
Figura 20 – Partículas com expansão hexagonal na reconstrução de um objeto de formato irregular simulando ser uma região ocupada pelo catalisador.	67
Figura 21 – Fluxograma do funcionamento do PSRT.	70
Figura 22 – Tela Principal do PSIRT, aba “Main”.	71
Figura 23 – Tela Principal do PSIRT, aba “Status”.	71
Figura 24 – Tela Principal do PSIRT, aba “Input”.	72
Figura 25 – Distribuição de partículas para o status da figura 24.	72

Figura 26 – Posições finais das partículas após a execução da distribuição que consta na figura 24.	73
Figura 27 – Discretização em hexágonos do resultado que consta da figura 26.	73
Figura 28 – Tela principal do PSIRT para simular o tomógrafo HSGT.	74
Figura 29 – Tipos de partículas de acordo com a cor.	75
Figura 30 – Decaimento de cor do branco (255) para o preto (0).	76
Figura 31 – Exemplo visual de ruído global e local.	77
Figura 32 – Reconstrução de uma barra gerando dois ruídos locais em 28.652 iterações.	111
Figura 33 – Eliminação dos ruídos exemplificados na figura 32 em 35.876 iterações.	112
Figura 34 – Polipropileno com dois furos longitudinais.	120

LISTA DE QUADROS E TABELAS

Quadro 1 – Reconstrução binária em 3x7 da barra, anel, círculo e quadrados diagonais.	79
Quadro 2 – Reconstrução NÃO binária em 3x7 da barra, anel, círculo e quadrados.	82
Quadro 3 – Reconstrução binária em 3x15 da barra, anel, círculo e quadrados diagonais.	84
Quadro 4 – Reconstrução NÃO binária em 3x15 da Barra, Anel, Círculo e Quadrados.	87
Quadro 5 – Reconstrução binária em 3x21 da barra, anel, círculo e quadrados diagonais.	90
Quadro 6 – Reconstrução NÃO binária em 3x21 da barra, anel, círculo e quadrados.	92
Quadro 7 – Reconstrução binária em 3x31 da barra, anel, círculo e quadrados diagonais.	95
Quadro 8 – Reconstrução NÃO binária em 3x15 da barra, anel, círculo e quadrados.	97
Quadro 9 – Reconstrução da barra vertical direita em 3x7.	103
Quadro 10 – Reconstrução do anel em 3x7.	104
Quadro 11 – Reconstrução do círculo em 3x7.	105
Quadro 12 – Reconstrução dos quadrados diagonais em 3x7.	106
Quadro 13 – Reconstruções do fantoma quadrados diagonais com ruído local.	109
Quadro 14 – Exemplos de ruídos locais no PSIRT.	110
Quadro 15 – Momentos na reconstrução do fantoma Quadrados Diagonais.	113
Quadro 16 – Reconstrução da barra simples em quatro posições espaciais.	114
Quadro 17 – Reconstrução do Polipropileno de Dois Furos.	121
Quadro 18 – Melhores reconstruções do PSIRT simulando o tomógrafo HSGT com dados reais.	125
Quadro 19 – Classificação visual das imagens geradas pelo PSIRT.	126
Tabela 1 – RMSE e Δ GC das imagens binárias dos experimentos simulados em 3xN.	100
Tabela 2 – RMSE e Δ GC das imagens NÃO binárias dos experimentos simulados em 3xN.	101

LISTA DE SIGLAS

ART – Algebraic Reconstruction Technique.

CT – Computed Tomography.

FBP – Filtered Back-Projection.

FCC – Fluid Catalytic Cracking.

FFT – Fast Fourier Transform.

GC – Geometric Coherence.

HSGT – High Speed Gamma-ray Tomograph.

KL – Kullback-Leibler.

MART – Multiplicative Algebraic Reconstruction Technique.

PSIRT – *Particle System Iterative Reconstruction Technique*.

RMSE – Root Mean Squared Error.

SIRT – Simultaneous Iterative Reconstruction Technique.

SMART – Simultaneous Multiplicative Algebraic Reconstruction Technique.

TI – Tempo de Integração.

VR – Velocidade de Rotação.

SUMÁRIO

1. Introdução.....	17
1.1. Motivação e Justificativa	17
1.2. Caracterização do Problema.....	20
1.3. Contribuições	21
1.4. Objetivos	22
1.4.1. Objetivo Geral	22
1.4.2. Objetivos Específicos	22
1.5. Metodologia e Recursos Usados	23
1.6. Organização do Trabalho	24
2. Tomografia no Riser	26
2.1 Riser do Tipo FCC	26
2.2 Reconstrução Tomográfica no Riser	28
2.3 Tipos de Tomógrafos	29
2.3.1 Tomógrafo de 1ª Geração	29
2.3.2 Tomógrafo HSGT	30
2.4 Lei de Lambert-Beer	31
2.4.1 Fundamentação da Lei	31
2.4.2 Lei de Lambert-Beer Aplicada ao Riser	32
3. Métodos de Reconstrução de Imagem.....	35
3.1 Modelo Matemático Geral	35
3.2 Embasamento dos Métodos Algébricos	39
3.3 Problema do Sistema $f=Ax$	41
3.4 Conceitos Relacionados	42
3.4.1 Sinograma	42
3.4.2 <i>Fan Beam</i>	43
3.4.3 Fantoma	44
3.5 Métodos Algébricos	44
3.5.1 Método ART	47
3.5.2 Método MART	49
3.5.3 Método SIRT	50
3.5.4 Método SMART	50
3.5.5 Parâmetro de Relaxação	51
3.6 Métrica de Erro GC	52
4. Método de Reconstrução de Imagem PSIRT	56
4.1 Sistemas de Partículas	56
4.2 PSIRT – Apresentação Inicial	58
4.3 PSIRT – Parâmetros	59

4.4	PSIRT – Discretização	60
4.5	PSIRT – Equilíbrio (Estabilidade)	61
4.6	PSIRT – Posição das Partículas	62
4.7	PSIRT – Eficiência	64
4.8	PSIRT – Pixel Natural Abrangente	65
4.9	PSIRT – Configuração dos Parâmetros	68
4.10	PSIRT – Formalização Matemática	69
4.11	PSIRT – Telas	71
4.12	PSIRT – Simulação do Tomógrafo HSGT	74
4.13	PSIRT – Tratamento de Ruído.....	77
5.	Experimentos Simulados	78
5.1	Experimentos Simulados no Tomógrafo de 1ª Geração	78
5.1.1	Configuração 3x7	79
5.1.2	Configuração 3x15	84
5.1.3	Configuração 3x21	90
5.1.4	Configuração 3x31	95
5.2	Experimentos Simulados Exclusivamente em 3x7	102
5.3	Experimentos Simulados para Discutir Ruído	109
5.4	Experimentos Simulados no caso Especial da Barra Simples	114
5.5	Tempo de Reconstrução dos Métodos	115
5.6	Experimentos do PSIRT com dados reais do Tomógrafo HSGT	120
6.	Considerações Finais.....	130
6.1	Experimentos Simulados no Tomógrafo de 1ª Geração	130
6.2	Experimentos Simulados Exclusivamente em 3x7	131
6.3	Experimento Simulado no caso especial da Barra Simples	132
6.4	Tempos de Reconstrução dos Métodos.....	132
6.5	Experimentos do PSIRT com dados reais do Tomógrafo HSGT	133
6.6	Conclusão.....	134
6.7	Sugestões para Trabalhos Futuros	135
	Bibliografia.....	137
	Apêndice.....	143

Capítulo I

1. Introdução

Este capítulo tem a finalidade de discutir os tópicos preliminares desta tese (motivação e justificativa, caracterização do problema, contribuições, metodologia), bem como apresentar os assuntos que são explorados ao longo do trabalho. Ressalta-se que, para a concepção do PSIRT (*Particle System Iterative Reconstruction Technique*), um conjunto diversificado de assuntos de áreas do conhecimento foi investigado: refino de petróleo, riser do tipo FCC (*Fluid Catalytic Cracking*), tomografia industrial, lei de Lambert-Beer, métodos de reconstrução de imagem, sistemas de partículas, tomógrafo de 1ª geração e tomógrafo HSGT (*High Speed Gamma-ray Tomograph*). Tratando-se, portanto, de um trabalho multidisciplinar.

1.1. Motivação e Justificativa

A indústria de petróleo continua em constante ascensão. Embora haja uma preocupação mundial em se investir em outras fontes de energia, sobretudo devido aos cuidados com o Meio Ambiente, os combustíveis derivados do petróleo ainda são e serão uma alternativa amplamente usada em máquinas, motores de diversas finalidades, automóveis, fogões, aviões, além da vasta utilidade do petróleo na fabricação de inúmeros produtos, tais como pneus, botas, tintas, parafinas, óleos lubrificantes, materiais de plástico¹. Até a construção de estradas depende do petróleo.

Estudos apontam que a área de petróleo vai permanecer ativa por mais de 45 anos sem passar por crise ou colapso, conforme comenta Bruno Calzavara (2011), que fez um desses estudos sobre a produção mundial de petróleo. Bruno indica também que muitas reservas ainda vão ser descobertas, além das que já foram e estão em fase de estudos geológicos e econômicos para serem exploradas, ou estão apenas no início de suas explorações, ou ainda aquelas que são exploradas e não são contabilizadas no cenário geral.

¹ BRASIL.gov.br: "Petróleo e Derivados" (www.brasil.gov.br/sobre/economia/energia/petroleo-gas-natural-e-derivados).

O Pré-Sal brasileiro é um exemplo de uma imensa reserva, que teve sua primeira prospecção de petróleo em 02 de setembro de 2008, conforme informação de Guilherme Estrella, diretor de Exploração e Produção da Petrobrás.

A lentidão na adoção em massa de outras fontes de energia – tais como eólica, solar, nuclear, biocombustível –, impede que o petróleo seja substituído por tais fontes a médio e até longo prazo, em contextos onde essas fontes podem gerar energia tanto quanto é gerada com combustão, a exemplo do advento do carro elétrico, que se adotado em larga escala, certamente iria diminuir a demanda por gasolina. Enfim, o alto custo científico, tecnológico e operacional, para viabilizar novas fontes de energia, muitas vezes torna seus usos inviáveis, além da política de exploração praticada diferentemente pelos países (AGUIAR, 2004, pág. 10-15). Assim, a corrida pelo “ouro negro” continua em alta.

O Brasil, por exemplo, pretende no Pré-Sal atingir já em 2017 a produção de um milhão de barris por dia. Segundo a Petrobrás, o Pré-Sal é uma das descobertas mais importantes desses últimos anos na área de combustíveis. São mais de oito bilhões de barris de petróleo numa faixa de cerca de 800 km de extensão, que serão produzidos nos próximos 30 anos, das bacias do Rio de Janeiro (Campos) e Espírito Santo. A bacia de Santos é outra promissora realidade, onde foi descoberta uma reserva de petróleo leve (PETROBRÁS, 2012-a). Esses oito bilhões são apenas uma fatia do Pré-Sal, pois, segundo Bruno Rosa (2012), o Brasil tem potencial para explorar 55 bilhões de barris no total no Pré-Sal, projeção que colocará o país entre os 10 maiores produtores do mundo.

Direcionando a discussão ao petróleo no Brasil, a Petrobrás em 2011 obteve um lucro líquido de US\$33,3 bilhões (trinta e três bilhões e trezentos milhões de dólares), e ainda ofertou US\$ 7 bilhões (sete bilhões de dólares) de títulos ao mercado internacional (PETROBRÁS, 2012-a). É sem dúvidas uma das maiores empresas do mundo.

Atualmente, a Petrobrás possui 11 refinarias funcionando, produzindo aproximadamente 1,8 milhões de barris por dia. As maiores reservas de petróleo do Brasil estão sob plataformas continentais recentemente construídas, e em áreas descobertas onde ainda serão construídas plataformas. São e serão estações localizadas em águas profundas (até 2.000m de profundidade) e ultraprofundas (mais de 2.000m de profundidade). (PETROBRÁS, 2012-c).

A extração terrestre também é uma realidade. Apesar de já ter mais de 10 anos de exploração, as plataformas terrestres continuam produzindo com funcionamento satisfatório e com média de produção anual estável (PETROBRÁS, 2012-b).

A produção terrestre está concentrada, principalmente, nas regiões Norte e Nordeste e, em menor escala, no Sudeste, no Espírito Santo. A produção média dos campos terrestres é de 210 mil barris por dia de óleo e 17,9 milhões de m^3 de gás (PETROBRÁS, 2012-b).

É importante ressaltar que a Petrobrás explora e produz combustíveis e demais derivados de petróleo, conforme é exposto na missão da empresa em seu site oficial: “Nós pesquisamos, localizamos, identificamos, desenvolvemos, produzimos e incorporamos as reservas de óleo e gás natural” (PETROBRÁS, 2012-b).

O processo de refino é um elemento de extrema importância implícito na citação literária supracitada. Em outra de suas matérias divulgada no seu site oficial, a Petrobrás aponta que atualmente um dos maiores desafios encontrados em suas atividades é o de refinar o óleo bruto. A estatal afirma que o petróleo nacional tem a característica de ser mais pesado, aspecto que dificulta ainda mais a obtenção de produtos nobres. Assim, as produções de diesel, gás liquefeito, gasolina, lubrificantes, nafta, óleo combustível e querosene de aviação, dependem expressivamente da tecnologia de refinamento que, segundo a Petrobrás, precisa ainda de um pesado investimento (PETROBRÁS, 2012-c). De fato, o processo de refino do petróleo necessita evoluir para produzir combustíveis de forma mais eficiente e aproveitando o máximo o petróleo bruto.

A tecnologia de refino do petróleo enfrenta uma gama de múltiplos desafios: produzir economicamente e de modo seguro, processando matérias primas com qualidade cada vez mais baixa, para obter produtos de alta especificação e sob crescentes restrições ambientais. Isso explica porque as atuais refinarias foram ficando cada vez mais complexas e irreconhecíveis frente às primitivas destilarias (SZKLO, 2008, pág. VI).

Existem várias maneiras de se refinar o petróleo bruto. Duas importantes delas são a destilação² e usando catalisadores. O refino com catalisadores é de interesse particular deste trabalho de pesquisa, especificamente do refino no riser do tipo FCC.

² Processo de separação física de substâncias baseada nos pontos de ebulição dos componentes.

A reconstrução de seções do riser ao longo do processo de refino é uma questão científica desafiadora. Vários métodos já produzem essa reconstrução, cada um com suas características, grau de eficiência e limitações. Oferecer um novo método que, ao menos, seja uma alternativa interessante aos métodos já existentes, é uma grande conquista tecnológica.

Nos últimos 5 anos, o mundo do petróleo e, especialmente, o segmento de refino, chamaram a atenção dos especialistas e mídia em geral, não apenas devido o aumento de preço do óleo bruto, mas também devido à necessidade com que se deparam alguns países no mundo para expandir seu parque refinador. O Brasil está entre estes países (SZKLO, 2008, pág. IX).

Os procedimentos de refino praticados em refinarias espalhadas pelo mundo não são de domínio público, inclusive as que operam no Brasil, que muitas vezes são de origem de capital estrangeiro ou possuem política de terceirização na produção de combustíveis (THOMAS, 2004). Trata-se de um contexto próprio de “caixa-preta”. Em síntese, o momento do petróleo é propício e um dos segmentos relevantes da indústria petrolífera mundial é justamente o refino do petróleo, contexto em constante evolução. Assim, produzir um novo método de reconstrução tomográfica para investigar o processo de refino é uma questão altamente promissora.

1.2. Caracterização do Problema

Os métodos atuais usados para reconstruir seções do processo de refino de petróleo, que são os métodos algébricos da família ART, apresentam dois problemas graves no processo de reconstrução da imagem: (1º) o problema mal posto³ e (2º) a geração de ruído global³. O problema mal posto ocorre quando da sequência das execuções (iterações), onde os métodos algébricos produzem um sistema não invertível (caso dos problemas inversos), que implica diretamente no conjunto solução da reconstrução. Ou seja, a cada iteração o método introduz incorreções nas equações que constituem o sistema, devido às imprecisões que o modelo produz quando novas leituras são feitas sobre o objeto a ser reconstruído.

³ Esses assuntos serão abordados com mais detalhes nos capítulos III e IV, respectivamente.

O segundo problema, a geração de ruído do tipo global, é uma característica dos métodos algébricos. É um tipo de ruído⁴ que cobre uma região significativa além das fronteiras da imagem que está sendo reconstruída. Por ser uma falha geral, é um aspecto difícil de ser tratado.

1.3. Contribuições

A primeira contribuição é disponibilizar o PSIRT como alternativa à reconstrução tomográfica industrial, apostando nas distinções do método em relação aos métodos algébricos, a exemplo de não sofrer do problema mal posto e não produzir ruído global, aspectos sumariamente apresentados na seção anterior.

O PSIRT tipicamente não gera ruído do tipo global, mas gera do tipo local. Este é um tipo de ruído mais fácil de ser tratado, e o método faz isso naturalmente ou, nos casos mais graves, com exaustivos conjuntos de novas iterações, com cada conjunto alterando parâmetros do método até conseguir melhores resultados na eliminação de tais ruídos. O fato de o PSIRT produzir apenas ruídos do tipo local é também uma significativa contribuição ao contexto da tomografia gama industrial, já que serão produzidas imagens com artefatos mais fáceis de serem tratados do que em imagens com ruídos globais.

Outra contribuição é com relação a trazer alguns aspectos importantes dos sistemas de partículas para tal contexto tomográfico, uma vez que são sistemas eficientes na simulação de situações (objetos, características) similares à reconstrução de imagem tomográfica industrial. Os sistemas de partículas representam uma abordagem diferente, que preserva a característica no PSIRT de também ser iterativo, e ao mesmo tempo de ser um método capaz de oferecer recursos próprios na reconstrução em questão, com as vantagens de tais sistemas.

São recursos exclusivos do método, a exemplo da discretização da imagem reconstruída feita com polígonos (hexágonos principalmente), que são expandidos de acordo com o tipo da partícula: a discretização nos métodos algébricos é feita com pixels gerados a partir das leituras das trajetórias, situação que torna a reconstrução limitada com relação a não reconhecer o material em áreas adjacentes de cada pixel (trajetória) que não são cobertas por nenhum raio.

⁴ Ruídos são informações não úteis à reconstrução gráfica, ou seja, são informações que ficam fora da região do objeto a ser reconstruído.

Por fim, outra contribuição também importante é a de oferecer um método capaz de simular não apenas um tomógrafo de 1ª geração, mas também um tomógrafo avançado (tomógrafo HSGT). O PSIRT, enfim, mostrou-se eficiente nesse tipo de tomógrafo, ao simulá-lo usando dados reais com desempenho satisfatório.

1.4. Objetivos

1.4.1. Objetivo Geral

O objetivo geral traçado para esta tese é o de apresentar e avaliar o PSIRT, método iterativo de reconstrução tomográfica baseado em sistemas de partículas, que tem como propósito inicial investigar o processo de refino de petróleo no riser do tipo FCC.

1.4.2. Objetivos Específicos

Os objetivos específicos são os seguintes:

- Comparar o PSIRT com os métodos algébricos ART, MART, SIRT e SMART, com a finalidade de verificar o seu desempenho;
- Adotar os sistemas de partículas como uma alternativa às metodologias atuais de reconstrução de imagem, pois o PSIRT se baseia nesses tipos de sistemas;
- Realizar diversos experimentos simulados com o PSIRT, alternando configurações e parâmetros do método, para encontrar melhores resultados nas reconstruções das imagens, tanto em relação ao próprio método, quanto em relação aos métodos da família ART. Duas importantes características que vão ser avaliadas são a eficiência e a robustez do PSIRT;
- Utilizar as métricas de erro RMSE (*Root Mean Squared Error*) e GC (*Geometric Coherence*) para avaliar as reconstruções do PSIRT com as reconstruções dos métodos da família ART, e também comparar tais métricas com os resultados visuais das imagens geradas por cada método;
- Avaliar o desempenho do PSIRT na simulação do tomógrafo HSGT, um tipo de tomógrafo avançado no contexto da tomografia industrial;
- Validar o PSIRT como um método alternativo aos métodos de reconstrução de imagem já existentes.

1.5. Metodologia e Recursos Usados

Os experimentos simulados foram realizados considerando dois tipos de tomógrafos: o primeiro com um par *fonte-detector* (tomógrafo de 1ª geração), e o segundo com cinco conjuntos de fontes com 17 trajetórias em cada conjunto *fonte-detectores*, totalizando 85 raios gama (tomógrafo HSGT). Para isso, foram produzidas duas interfaces de entrada de dados para o PSIRT, uma para o primeiro tomógrafo e uma para o segundo, para confrontar as particularidades definidas para cada um deles.

As simulações no tomógrafo de 1ª geração foram feitas com imagens (*fantomas*) comumente usadas em avaliações dessa natureza (barras, anéis, círculos, quadrados). Nesse tipo de tomógrafo, o PSIRT é comparado aos métodos da família ART em configurações *MN* de projeções e trajetórias, fixando a quantidade de projeções *M* e, variando a quantidade de trajetórias *N*. O tipo e resolução dos fantomas originais reconstruídos e os parâmetros do método também foram alterados em busca de melhores resultados. As configurações *MN* adotadas foram as seguintes: 3x7, 3x15, 3x21 e 3x31, que são configurações baixas em termos de arranjo tomográfico industrial.

Para o tomógrafo HSGT as simulações usaram dados reais obtidos em Maad (2009). Foram mais de 75.000 quadros de imagem gerados experimentalmente e usados pelo PSIRT na simulação do tomógrafo em questão. Nesse tipo de tomógrafo de alta velocidade, os dados reais representam um fantoma cilíndrico feito de polipropileno com dois furos longitudinais. O objeto representa muito bem a simulação dinâmica de bolhas de gás imersas em óleo para dentro de um tubo, uma vez que o material de polipropileno apresenta densidades semelhantes às do óleo, aspecto de interesse particular deste trabalho. A execução do PSIRT simulando o tomógrafo HSGT variou em relação ao tempo de integração na geração de cada quadro de imagem, e em relação à velocidade de rotação do objeto dentro do tomógrafo.

Com relação à implementação do algoritmo do PSIRT, foi usada a linguagem C# do ambiente de desenvolvimento Microsoft Visual Studio 2010 (MVS 2010). O Apêndice no final do trabalho mostra detalhes desse ambiente, de como foi feita tal implementação e o código fonte do PSIRT. O funcionamento dos métodos ART, MART, SIRT e SMART, também foram implementados com o C# do MVS 2010, incorporados ao PSIRT em especial na simulação do tomógrafo de 1ª geração, para facilitar as execuções e comparações do PSIRT com esses métodos.

As métricas de erro RMSE e GC também foram implementadas usando o MVS 2010, mas em programas específicos para cada uma delas. Os dados resultantes dos cálculos dessas métricas foram tratados no Microsoft Excel 2010, sobretudo com relação à geração dos gráficos correspondentes. O Excel também foi usado na geração dos demais gráficos usados no trabalho.

1.6. Organização do Trabalho

O próximo capítulo, denominado “Tomografia no Riser”, trata dos assuntos relacionados à tomografia industrial no contexto do riser do tipo FCC. Esse capítulo aborda desde a apresentação do riser até a fundamentação sobre a radiação gama, tal como a lei de atenuação de Lambert-Beer. O propósito do Capítulo II é discorrer sobre a investigação com feixes gama do processo de refino de petróleo no riser. É também no Capítulo II que as informações dos tomógrafos de 1ª geração e HSGT são apresentadas.

No Capítulo III, os métodos de reconstrução de imagem são discutidos, especificamente os métodos algébricos ART, MART, SIRT e SMART, que são usados na comparação com o PSIRT. As métricas de erro RMSE e GC também são apresentadas nesse capítulo, destacando a métrica GC que, por não ser muito comum de ser usada em trabalhos dessa natureza, merece ser explicada para justificar seu uso.

O Capítulo IV mostra o PSIRT, apresentando inicialmente uma revisão literária sobre os sistemas de partículas e, em seguida, destacando aspectos do funcionamento do método, tais como discretização, parâmetros, iterações, tratamento de ruído local, formalização matemática.

No Capítulo V, os experimentos simulados realizados com o PSIRT são apresentados. Esses experimentos diferem em relação à resolução do fantoma de entrada, quantidade de trajetórias de raios gama, quantidade de partículas e de iterações, além dos parâmetros específicos do próprio PSIRT, a exemplo da quantidade de iterações para se atingir o equilíbrio. É no Capítulo V que o PSIRT é comparado com os métodos da família ART na simulação do tomógrafo de 1ª geração, e executa experimentos com dados reais na simulação do tomógrafo HSGT.

Ainda com relação ao Capítulo V, no caso particular do PSIRT no contexto do tomógrafo HSGT, o ponto alto desse experimento é a criação de *classes* de reconstrução de imagem, que são definidas de acordo com a análise visual dos fantasmas reconstruídos. Essas classes são importantes para avaliar a eficiência do PSIRT com relação ao tempo de integração (dado em *ms*), e velocidade de rotação (dada em *rad/s*) do objeto dentro do tomógrafo.

Na última parte do trabalho (Capítulo VI), as considerações finais são apresentadas, com a finalidade principal de qualificar o PSIRT – ao menos como uma alternativa para a reconstrução de imagem tomográfica do riser do tipo FCC, tendo como referência os experimentos feitos no Capítulo V. Para isso, o Capítulo VI não apenas conclui o trabalho, como também apresenta um extrato analítico sobre tais experimentos. Em seguida, o capítulo em questão é finalizado, apresentando sugestões para trabalhos futuros.

Capítulo II

2. Tomografia no Riser

2.1 Riser do Tipo FCC

A indústria petrolífera utiliza grandes⁵ tubos para o refino do petróleo bruto (óleo cru), conhecidos como *risers*. O refino é feito através de um processo fluidodinâmico, em que a separação dos elementos se dá pela ação química de catalisadores no interior do tubo. O fluxo no riser escoia de baixo para cima, num processo complexo de circulação sólida que envolve vários tipos de elementos: hidrocarbonetos com 84% de hidrogênio e 14% de carbono; de 1 a 3% de enxofre; nitrogênio, oxigênio, metais e sais, com aproximadamente 1% de cada. São números aproximados, que vão depender do tipo do petróleo encontrado na natureza, que por sua vez também depende da decomposição de matéria orgânica, uma vez que o petróleo é um tipo de combustível fóssil (SZKLO, 2008).

No riser a ação do catalisador é de extrema importância para a obtenção de produtos derivados aproveitando o máximo do petróleo bruto, e é esse o conceito de *refino*. Ou seja, refinar significa formar compostos úteis – tais como gasolina, querosene, óleo diesel, gás –, eliminando, como consequência, materiais que não são necessários na produção desses derivados (SZKLO, 2008). De uma maneira geral é relativamente fácil entender a eficiência do catalisador no refino: quanto mais abrangente e uniforme for à distribuição do catalisador no riser, melhor será a qualidade dos combustíveis produzidos (MELO, 2007, pág. 1).

Variações locais do catalisador no riser na razão da mistura de óleo podem levar a baixas conversões em algumas partes do tubo, enquanto em outras partes altas conversões irão produzir combustíveis indesejáveis. Assim, conhecer com relativa precisão a concentração sólida numa seção transversal do riser é de extrema importância para eficiência do processo de refino, uma vez que há uma relação direta entre concentração sólida e mistura matéria prima e catalisador (VASCONCELOS, 2003, pág. 84).

⁵ Um riser pode ter aproximadamente 1 metro de diâmetro e 60 metros de comprimento (altura).

Uma importante etapa do processo de refino do petróleo é feita com o uso de catalisadores. Essa etapa é chamada de “craqueamento catalítico” e ocorre em um riser do tipo FCC (SZKLO, 2008) (THOMAS, 2004). A Figura 1 mostra um esquema desse tipo de riser.

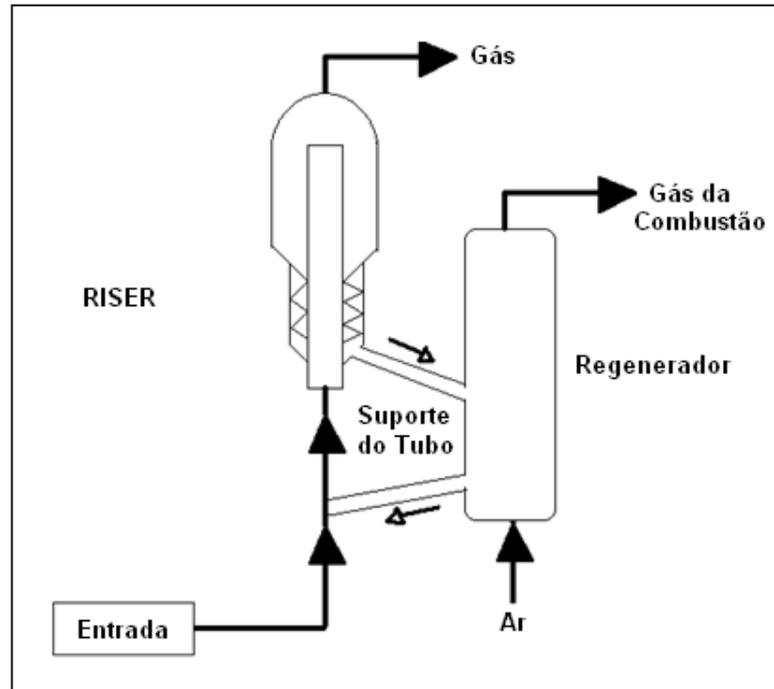


Figura 1 – Riser do tipo FCC.
Fonte da imagem: (AZZI, 1991, pág. 2).

Craquear significa dividir a matéria orgânica em partes menores com a finalidade de formar substâncias úteis. As fórmulas que representam os compostos após o craqueamento também podem variar com relação ao número de certos elementos, uma vez que a formação química desses compostos varia com o fracionamento adotado pela refinaria (THOMAS, 2004). Seguem três fórmulas que mostram a variação do composto para a gasolina, que é uma mistura de hidrocarbonetos da série dos alcanos ou parafinas (SZKLO, 2008):

- C_6H_{14} , fórmula com o número *mínimo* de cada elemento.
- $C_{12}H_{26}$, fórmula com o número *máximo* de cada elemento.
- C_8H_{18} , fórmula com o número *médio* de cada elemento.

Um grande problema que acontece no interior do riser é a formação de grandes vazios: o gás das partículas flui para baixo com os sólidos para preencher os vazios, que podem fluir para baixo ou para cima, ou permanecerem parados, dependendo das condições e dos tamanhos deles (dos vazios). Os vazios crescem lentamente, introduzindo perturbações na circulação do catalisador ou nas válvulas de regulação de fluxo. Assim, o processo provoca fenômenos que geralmente são muito difíceis de serem observados e, os únicos dados disponíveis são leituras de pressão, que são dados instantâneos que têm uma média numa base de volume. Medidas locais seriam mais úteis e, para isso, a investigação do riser com radiação gama passa a ser uma alternativa bastante interessante (DANTAS, 2007).

2.2 Reconstrução Tomográfica no Riser

A distribuição da densidade catalítica no interior do riser do tipo FCC pode ser deduzida de uma maneira não invasiva, através da tomografia gama computadorizada (CT – *Computed Tomography*). A CT consiste em usar a atenuação de feixes paralelos (projeções gama) atravessando a seção do riser, o qual é relacionado com a densidade das substâncias internas, respeitando a lei de Lambert-Beer. A investigação é naturalmente viável, pois a radiação gama não causa perturbação no fluxo interno do riser (MELO, 2007, pág. 2). Uma precisa reconstrução da distribuição da densidade do catalisador no interior do riser pode ser feita através da transformada de Radon, de acordo com o Teorema do Corte Central, teoricamente usando uma quantidade infinita de projeções de raios gama (KAK, 2001).

O método FBP (*Filtered Back-Projection*), que é um dos mais populares dentre os vários métodos de natureza similar, é usado em situações onde o resultado satisfatório depende de uma quantidade muito alta de projeções e trajetórias, tal como em imagens médicas (KAK, 2001). Esse aspecto faz com que o método FBP seja pouco eficiente para o contexto da tomografia industrial, devido às particularidades da indústria petrolífera, principalmente no que diz respeito aos arranjos práticos da tomografia, que facilmente conduzem a custos muito altos. O FBP é implementado usando a Transformada Rápida de Fourier (*Fast Fourier Transform – FFT*) no núcleo, o que, neste caso, para uma quantidade pequena de trajetórias por projeção produz uma imagem bastante ruidosa (KAK, 2001).

Um grupo mais apropriado de algoritmos para os propósitos industriais são os métodos iterativos, os quais tendem a produzir melhores resultados com quantidades pequenas de projeções e trajetórias. Neles, estão incluídos os métodos algébricos, tais como (KAK, 2001):

- ART (*Algebraic Reconstruction Technique*);
- MART (*Multiplicative ART*);
- SIRT (*Simultaneous Iterative Reconstruction Technique*);
- SMART (*Simultaneous MART*).

Com característica familiar aos métodos supracitados – por também ser um método iterativo –, o PSIRT é uma alternativa interessante e inovadora para a reconstrução tomográfica do riser. O PSIRT é baseado nos princípios e propriedades dos sistemas de partículas. Estes sistemas viabilizam métodos baseados nas leis da Física, que podem ser usados na Computação Gráfica para modelos de animação realista de objetos sintéticos, particularmente em cenários onde existem objetos *fuzzy*⁶. Esses métodos também são apropriados para a modelagem de fenômenos naturais ou objetos de formas geométricas variadas, tais como nuvem, fumaça, fogo, água. Para as formas geométricas assumidas pelo catalisador no processo de refino de petróleo no riser, a modelagem com partículas é viável.

2.3 Tipos de Tomógrafos

2.3.1 Tomógrafo de 1ª Geração

O tomógrafo de 1ª geração é constituído de uma fonte e de um detector, e o objeto a ser analisado (varrido) é colocado entre eles. Nesse tipo de tomógrafo, o par *fonte-detector* é deslocado uniformemente tantas vezes quantas forem às trajetórias definidas para ler o objeto a ser reconstruído graficamente. Assim, tem-se, um conjunto de raios paralelos para certa projeção θ . Para obter dados de um ângulo diferente, o conjunto é girado no novo valor θ desejado.

⁶ *Fuzzy*: objetos de formato irregular.

O Laboratório de Radioquímica da UFPE possui um tomógrafo desse tipo, usado para experimentos, chamado de “Tomógrafo a Frio”. A parte (a) da Figura 2 mostra a sala e o tomógrafo, e a parte (b) apenas o tomógrafo, onde são destacados a fonte (1) e o detector (2).

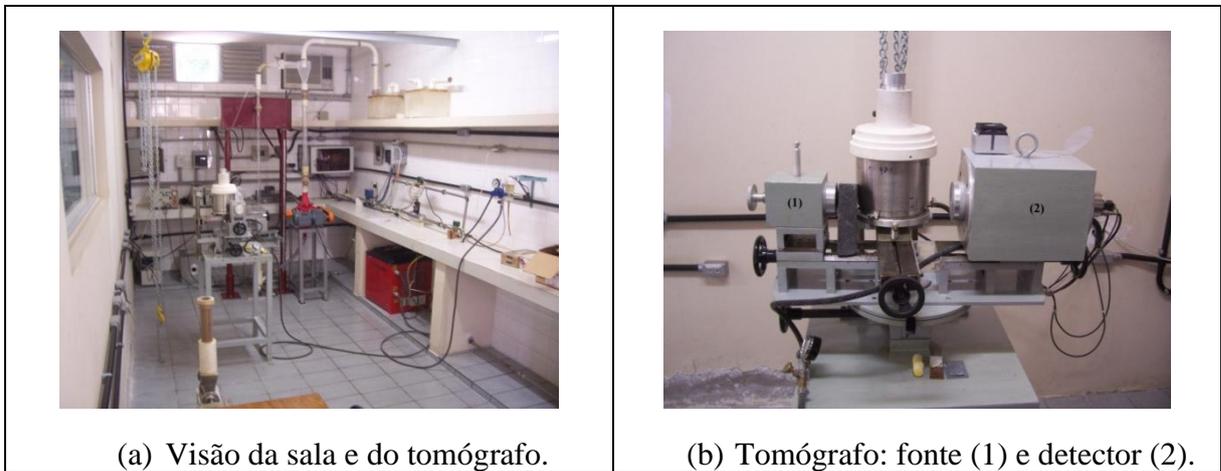


Figura 2 – Tomógrafo a Frio do Laboratório de Radioquímica da UFPE.

2.3.2 Tomógrafo HSGT

Um esquema de uma seção do tomógrafo HSGT pode ser visto na Figura 3.

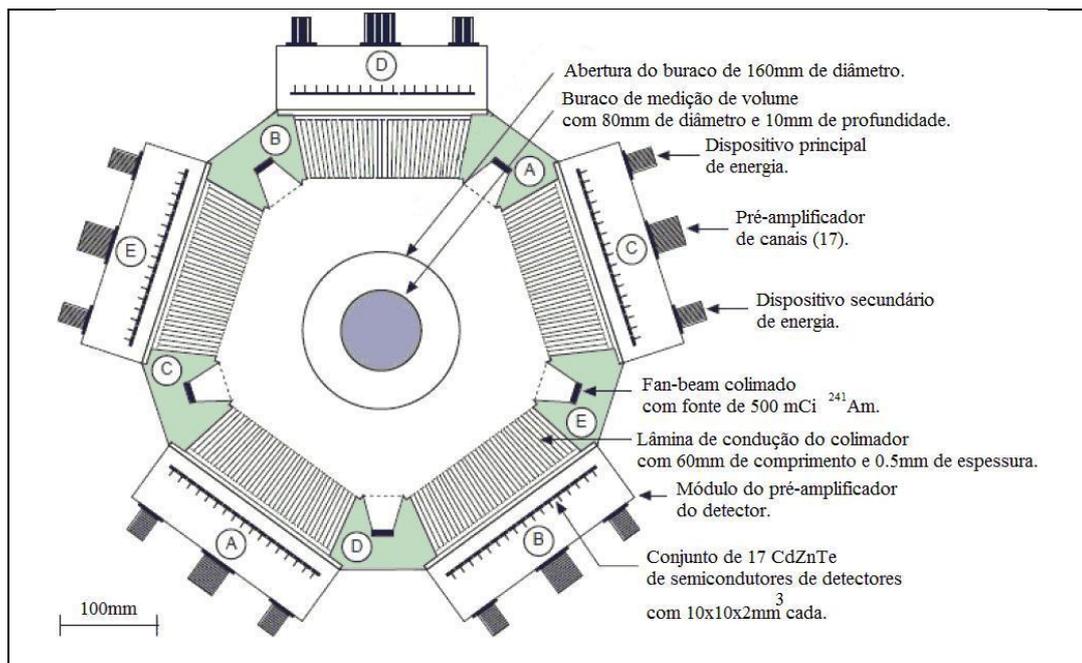


Figura 3 – Tomógrafo HSGT.
Fonte da Imagem: (MAAD, 2009, pág. 46).

O tomógrafo HSGT é um tipo de equipamento considerado complexo e de alta velocidade, devido à própria natureza de sua geometria tomográfica: cinco fontes, cada uma projetando 17 raios no formato de um leque (*fan beam*). A Figura 3 mostra a fonte “A” apontando para os detectores “A” e, assim, para os demais pares *fonte-detectores*.

Uma importante característica do tomógrafo HSGT é o tempo de integração, que representa o tempo que os detectores precisam para contabilizar dados suficientes para gerar um quadro de imagem do objeto que estar sendo reconstruído. Essa característica é de interesse particular deste trabalho e será considerada no Capítulo V, na seção dos experimentos sobre esse tipo de tomógrafo.

2.4 Lei de Lambert-Beer

2.4.1 Fundamentação da Lei

A lei que esclarece o fenômeno pelo qual a radiação eletromagnética sofre atenuação ao atravessar materiais é chamada de lei de Lambert-Beer. O nome da lei é dada em homenagem aos dois cientistas que a descobriram: o físico francês radicado na Alemanha Johann Heinrich LAMBERT (1728-1777) – que estudou a transmissão de luz através de materiais homogêneos; e o matemático alemão August BEER (1825-1863). Beer estudou os trabalhos de Lambert sobre a transmissão de luz em sólidos e, com seus próprios experimentos, concluiu a referida lei, a qual data de 1852⁷.

Na lei de Lambert-Beer a radiação ao atravessar certa substância é relacionada com a concentração do material existente nessa substância. Trata-se de uma relação de absorção da radiação pelo material (DANTAS, 2007). Além da absorção, a luz ao incidir certo material pode sofrer também reflexão, refração e espalhamento, como ilustrado na Figura 4.



Figura 4 – Visão geral de reflexão, refração e espalhamento.

⁷ A lei foi descoberta independentemente (e de diferentes maneiras) por Pierre Bouguer em 1729, Johann Heinrich LAMBERT em 1760 e August BEER em 1852, sendo que Beer concluiu a lei a partir do trabalho de Lambert.

Na lei de Lambert-Beer, os efeitos de reflexão, refração e espalhamento são considerados desprezíveis para certos contextos, por serem demasiadamente ínfimos. Assim, a absorção é o aspecto mais importante descrito na lei. A lei de Lambert-Beer também impõe que a radiação deve possuir apenas um comprimento de onda. A Figura 5 ilustra a transmissão da radiação.

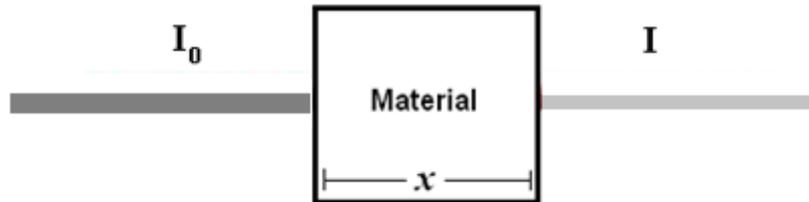


Figura 5 – Atenuação da radiação num material de espessura x .

A forma geral da lei de Lambert-Beer é dada pela seguinte equação,

$$I = I_0 e^{-\mu_L x} \quad (1)$$

na qual I_0 é a intensidade da radiação incidente, I a intensidade da radiação final, μ_L o coeficiente de atenuação linear e x a espessura do material.

2.4.2 Lei de Lambert-Beer Aplicada ao Riser

Para a aplicação da lei de Lambert-Beer no contexto do riser do tipo FCC, uma medida de interesse é a densidade ρ das substâncias existentes no riser, que são os materiais do petróleo bruto e do catalisador. A densidade ρ em função da espessura x do material fornece a massa por unidade de área, que representa uma porção bem definida da massa global do referido material. A densidade tem relação direta com o coeficiente de atenuação linear, conforme pode ser observado nos elementos da equação (1):

$$\begin{aligned} \mu &\Rightarrow \text{cm}^2 / \text{g ou m}^2 / \text{kg} \\ \rho &\Rightarrow \text{g} / \text{cm}^3 \text{ ou kg} / \text{m}^3 \\ x &\Rightarrow \text{cm ou m} \end{aligned} \quad \text{com} \quad \frac{\mu}{\rho} = \mu_L$$

A investigação no riser feita com radiação gama gera o seguinte procedimento: o raio gama emitido da fonte incide na parede do riser, atravessa o interior do tubo, incide novamente na parede (do outro lado), e é recebido no detector.

Tanto a parede do riser – a qual é atravessada pelo raio por duas vezes –, quanto as substâncias internas existentes no tubo, causam atenuação no raio, de maneira que sua intensidade inicial decai *exponencialmente* no percurso da fonte ao receptor. O principal responsável pelo decaimento são as substâncias internas do riser ao longo da distância x percorrida pela radiação. A Figura 6 destaca os elementos do riser que a radiação atravessa.

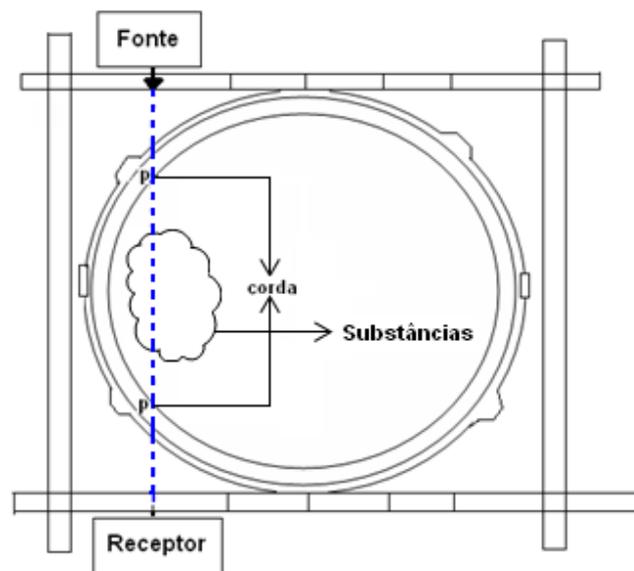


Figura 6 – Riser, suas paredes e uma corda do raio gama atravessando substâncias.

Fonte da imagem: (DANTAS, 2007).

Em detalhes, a análise na atenuação é feita através de medidas de absorção da radiação eletromagnética. O processo de absorção ocorre ao nível molecular. Ou seja, cada molécula possui níveis de energia quantizados, os quais podem ser ocupados pelos elétrons das moléculas. Em paralelo, a radiação possui energia que depende diretamente do comprimento de onda da referida radiação (MENDES, 2012).

Convergingo esses conceitos, a absorção da radiação acontece quando a energia da radiação é igual à diferença entre dois níveis de energia da molécula. Quando isso ocorre, então a energia da radiação é transferida para a molécula. Esse é o conceito elementar para *absorção* (MENDES, 2012). Uma vez que as moléculas de substâncias diferentes possuem níveis de energia diferentes, a radiação é absorvida por cada substância de maneira também

específica. Enfim, uma substância absorverá o comprimento de onda da radiação de uma maneira, enquanto outras substâncias absorverão de outras maneiras. Ou seja, cada substância absorverá à sua maneira. Nesse caso, se há dados referentes à intensidade de luz absorvida por uma substância em função do comprimento de onda da radiação, pode-se ter o “espectro de absorção da substância”.

O espectro fornece uma curva característica na relação substância *versus* radiação. Para identificar uma substância desconhecida, por exemplo, basta comparar sua curva de absorção com as curvas de substâncias conhecidas. O espectro de absorção da substância pode também determinar a quantidade em que a substância se encontra numa certa solução. Isso é feito medindo a intensidade da luz que atravessa a amostra do material analisado. Essa medida também é obtida com a equação (1). No *riser*, a concentração do material também é um dos grandes responsáveis pelo decaimento exponencial da intensidade da radiação transmitida.

Quando vários fótons (feixe) se deslocam através da matéria, a taxa global que determina a porção desses fótons que interage com um material específico é um dado importante, para se determinar o efeito da interação da radiação com a matéria⁸. Alguns fótons podem interagir com o material e outros podem apenas atravessá-lo. O interesse no *riser* é com os fótons que interagem com as substâncias internas.

Os raios que atravessam o material vão ser recebidos no detector, que é colimado com a fonte para tal propósito. As leituras do detector vão fornecer os dados para reconstrução do objeto que estar sendo analisado com radiação gama, daí a importância de se entender como ocorre à atenuação sofrida pela intensidade da radiação. Em síntese, à medida que o raio vai atravessando o interior do tomográfico, as informações do espaço percorrido vão sendo somadas, num procedimento que se baseia em um conceito chamado de “modelo aditivo”. A contagem dessa soma é feita no detector para cada raio, e é a partir dessa informação que a reconstrução da imagem do objeto é feita.

⁸ Efeitos fotoelétrico, Compton e produção de pares.

Capítulo III

3. Métodos de Reconstrução de Imagem

3.1 Modelo Matemático Geral

Os métodos de reconstrução gráfica seguem um modelo matemático com equações lineares. Trata-se de um modelo geral, que serve como base para os principais métodos algébricos. Para entender o modelo matemático de reconstrução de imagem, é preciso considerar inicialmente os seguintes aspectos:

- O cenário espacial analisado corresponde a uma circunferência num plano bidimensional (x, y) , cuja origem $(0, 0)$ coincide com o centro da circunferência.
- Os raios e as cordas da circunferência representam segmentos de raios gama, que cortam (atravessam) a circunferência em dois pontos distintos.
- Os ângulos θ de inclinação dos raios gamas são formados por esses raios em relação ao diâmetro horizontal da circunferência, fixado como eixo principal para a orientação e valor inicial de cada ângulo. Assim, existem três grupos de ângulos:
 - ✓ Ângulos de $0^{\circ} < \theta < 90^{\circ}$ (inclinações do tipo “/”);
 - ✓ Ângulos de $90^{\circ} < \theta < 180^{\circ}$ (inclinações do tipo “\”);
 - ✓ Ângulos horizontal 0° e vertical 90° .
- O objeto a ser reconstruído se localiza exclusivamente na parte interna da circunferência.
- A função f^9 calcula a atenuação sofrida pelos raios gama ao atravessarem a circunferência. Essas atenuações são os dados de base usados na reconstrução gráfica do objeto.
- Uma projeção de raios corresponde a um ângulo θ , também chamada de *vista*.
- Cada projeção de ângulo θ possui várias trajetórias de raios gama.

⁹ A função f será vista com mais detalhes na seção 3.2.

O plano em questão é retangular, denominado aqui de “plano s ”, cujas coordenadas são x e y (plano original). Para projetar outros conjuntos de raios gama de ângulos θ diferentes, é preciso realizar rotações no sistema para obter novas coordenadas (x' e y' , por exemplo) e, assim, criar novos sistemas de coordenadas, sendo o primeiro deles denominado aqui de “plano t ”. Então, têm-se no mínimo dois sistemas de coordenadas para a obtenção dos dados para a reconstrução tomográfica: o plano s e o plano t .

O procedimento de rotação é apenas para viabilizar conjuntos de raios de ângulos diferentes para, assim, cobrir a circunferência em várias *visões* (projeções). Quanto mais projeções são realizadas, mais informações são obtidas do objeto dentro da circunferência. Entretanto, a posição do objeto dentro da circunferência é teoricamente fixa para um dado momento. Dessa forma, as rotações devem existir, mas os pontos dos diversos sistemas devem ser convertidos para o sistema original s . A conversão é feita com o seguinte sistema de equações.

$$\begin{aligned} x &= x' \cos \theta - y' \sin \theta \\ y &= x' \sin \theta + y' \cos \theta \end{aligned} \quad (3)$$

Os elementos da equação (3) podem ser vistos na Figura 7.

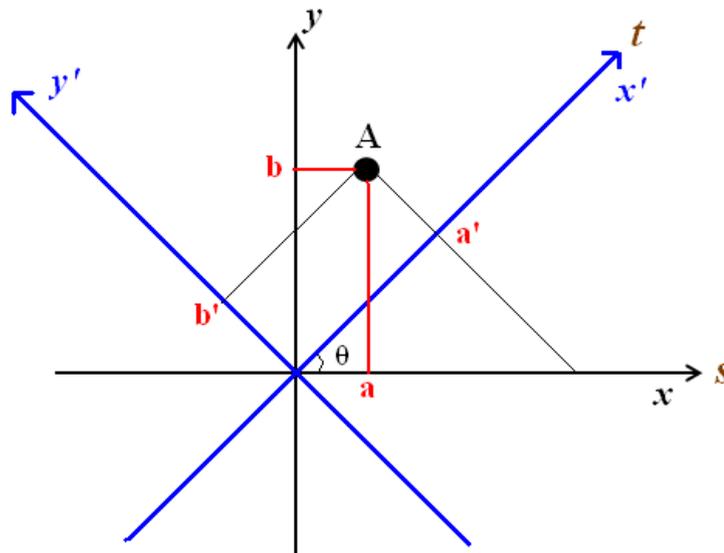


Figura 7 – Rotação θ do sistema de coordenadas s para o sistema de coordenadas t .

Observa-se na Figura 7 que, para cada valor x' no sistema de coordenadas t , existe uma única linha que é paralela ao eixo y' , deslocada a' do eixo y' . Adotando-se $f_\theta(x')$ como sendo a integral da função f , tem-se então (SALINA, 2001, pág. 28):

$$f_\theta(x') = \int_{-\infty}^{\infty} f dy' \quad (4)$$

A equação (4) representa a função de base para a Teoria da Tomografia Computadorizada, que usa os valores calculados com f para determinar as atenuações sofridas pelos raios gama ao atravessarem um objeto. A atenuação recebe influência direta do objeto que o raio atravessa. Cada objeto – uma substância química, por exemplo –, possui características físicas diferentes, que o leva a ter um coeficiente de atenuação também diferente de outro objeto.

Assim, pode-se associar a atenuação com a capacidade de absorção do objeto. Por exemplo, na tomografia de raios-X aplicada ao corpo humano, o tecido mole e os ossos sofrem atenuações de formas diferentes (SALINA, 2001, pág. 29). Nesse exemplo, sabe-se exatamente em qual desses dois materiais o raio atravessou. Esse é um dos principais princípios da Tomografia Computadorizada, que se faz presente nos diversos contextos da reconstrução tomográfica (KAK, 2001).

Na descrição do modelo matemático de reconstrução, duas importantes grandezas são necessárias para determinar a atenuação sofrida por cada raio, quais sejam: as intensidades inicial I_0 e final I do raio gama. Assim, a função $f_\theta(x')$ representa a atenuação em questão, computada pelo detector do feixe gama, registrando dois dados: o (1) ângulo θ correspondente à projeção atual, e o (2) deslocamento a' em relação ao y' do sistema de coordenadas que sofreu a rotação θ .

Em síntese, a função $f_{I_0}(x')$ representa a distribuição dos coeficientes de atenuação linear, que corresponde a integral de linha para o par de valores (θ, a') . O ângulo θ é também denominado de ângulo de visão ou de varredura, e a integral de linha usada no deslocamento a' é denominada de projeção (Transformada de Radon¹⁰), ou raio soma de f para (θ, a') .

¹⁰ Transformada que consiste da integral de uma função sobre linhas retas.

Assim, tem-se (SALINA, 2001, pág. 29):

$$f_{I_0}(x') \approx I_0 e^{-f_\theta(x')} \quad (5)$$

Uma importante variação da equação (5) pode ser escrita em função de $f_\theta(x')$, uma vez que I_0 e $f_{I_0}(x')$ são dados obtidos experimentalmente (SALINA, 2001, pág. 30):

$$f_\theta(x') = \ln \frac{I_0}{f_{I_0}(x')} \quad (6)$$

Ressalta-se que o objetivo da tomografia é reconstruir f a partir de $f_\theta(x')$. Isso é possível uma vez que o valor de f depende do conteúdo do objeto na posição espacial do sistema de coordenadas original, que no caso aqui é o plano s da Figura 7. Com isso, a função f pode ser usada para determinar a composição material do objeto em questão.

Em termos de detalhamento prático, os raios i e as projeções θ possuem uma quantidade finita. Em consequência, os deslocamentos a' também são limitados com relação à quantidade. Essas limitações são impostas pelo sistema real de tomografia. Portanto, são limitações físicas, tais com o tamanho do detector, a precisão do instrumento de medição dos feixes gama, a exposição do material à radiação, a energia dos raios gama, entre outros fatores relacionados com as características físicas do tomógrafo industrial (MUSIAL, 2010, pág. 52).

Apesar dessas limitações físicas, uma quantidade significativa de raios gama pode ser usada para reconstrução gráfica. As configurações vão depender do objeto que deve ser reconstruído, conforme já mencionado. Alguns contextos exigem uma quantidade grande de projeções e trajetórias de raios gama, a exemplo da tomografia médica, a qual impõe uma varredura mais ampla e precisa para obter imagens satisfatórias de partes do corpo humano.

Entretanto, outros contextos, a exemplo da tomografia industrial, possuem materiais com características que permitem ser reconhecidos por uma relativa aproximação espacial, sendo eficientes (as reconstruções) mesmo com um número pequeno de projeções e trajetórias. Em conformidade com essa realidade, os métodos algébricos possuem como base o modelo matemático discutido aqui e, devido a isso, esses métodos são adaptáveis a vários tipos de situações reais que podem ser investigadas com radiação gama. A distribuição da densidade catalítica no riser do tipo FCC, por exemplo, é uma delas.

3.2 Embasamento dos Métodos Algébricos

Ressaltando, na tomografia computadorizada a reconstrução da imagem do objeto analisado se dá através das atenuações sofridas pelos raios gama, que são emitidos considerando um conjunto de projeções de ângulos diferentes, a exemplos dos ângulos 0° , 60° , 120° . Cada projeção possui um conjunto de raios, que irão atravessar o riser e o objeto dentro dele. A discretização é feita com quadradinhos denominados *pixels*, que são numerados em forma de matriz, no formato padrão a_{ij} , conforme a Figura 8 (SALINA, 2001, pág. 31).

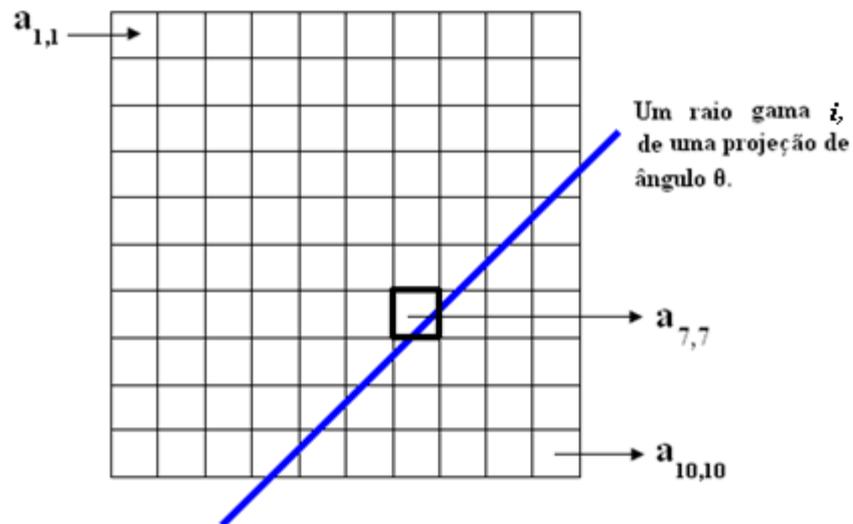


Figura 8 – Discretização com uma matriz 10x10, destacando o pixel $a_{7,7}$ e um raio gama de projeção θ .

A fonte e o detector do raio são modelados como sendo pontuais, enquanto os raios em si são modelados como linhas. O ângulo θ é formado pelo raio i e por um eixo (diâmetro) fixado para representar a origem do sistema (seção do riser). A matriz é montada a partir da quantidade de projeções e trajetórias por projeção. No caso hipotético da matriz da Figura 8, são 10 projeções, cada uma com 10 trajetórias. Essa configuração estabelece um limite de até 100 pixels, para garantir maior chance de solubilidade do sistema. Neste caso, o índice i representa o raio e o índice j o pixel.

Assim, a medida da atenuação física computada para cada raio i é representada pela integral de linha ao longo desse raio, através da função $f(f_i)$. Completando a formalização, adota-se para f^I uma integral de soma finita e, em consequência, o modelo passa a ser representado por um sistema de equações lineares, conforme expressão matemática a seguir (OLIVEIRA, 2011, pág. 17),

$$f_i = \sum_{j=1}^J x_j a_{ij} \quad (7)$$

na qual J é o último pixel, x é um valor constante, real e positivo e a é o comprimento do i -ésimo raio no j -ésimo pixel.

A partir da equação (7), deduz-se que f_i (com $f_i \in R$) representa o vetor de medidas, e x_j (com $x_j \in R$) representa o vetor imagem, e a_{ij} (de dimensão “IxJ”, com I sendo o último raio) representa a matriz de projeção, denominada aqui de matriz A . Logo, tem-se o seguinte sistema matricial (ANTON, 2006, pág. 131):

$$f = Ax \quad (8)$$

O sistema em (8) expressa à forma geral do funcionamento dos métodos de reconstrução, que têm a discretização em pixel como mecanismo de reconstrução da imagem do objeto analisado. Esse sistema pode ser estendido para atender contextos específicos da reconstrução tomográfica, a exemplo de ruído (η). Neste caso, o sistema em (8) passaria para a forma $f = Ax + \eta$. Os algoritmos algébricos são exemplos de métodos que usam esse tipo de discretização e que seguem o modelo do sistema em (8).

¹¹ Uma vez que parte do percurso da radiação ocorre fora da seção do riser, uma importante característica da função f é impor uma lógica para computar apenas informações internas, através de dois valores discretos (zero ou um): 0 (fora da seção) e 1 (dentro da seção).

3.3 Problema do sistema $f=Ax$

O modelo $f=Ax$, discutido na seção anterior, é um sistema mal posto¹², por não ser, em geral, invertível (problemas inversos). Normalmente, coloca-se uma quantidade maior de trajetórias do que pixels, fazendo com que o sistema fique super determinado (mais equações do que incógnitas), muitas vezes com equações incompatíveis, devido a ruídos e problemas numéricos (representação por pontos flutuantes no computador, por exemplo).

Do Teorema do Corte Central conclui-se que, quanto maior o número de projeções e trajetórias, menor o conjunto de soluções admissíveis, e quando o número de projeções e trajetórias se torna infinito, a solução é super determinada, que corresponde à seção do objeto a ser reconstruído (KAK, 2001). Assim, quando se tem uma quantidade limitada de trajetórias, o conjunto-solução é bem amplo, então o sistema $f=Ax$ admite infinitas soluções, apesar de o ruído afetar o posto do sistema (posto-linha cheio seria a solução única).

O mau condicionamento da matriz reflete num problema que um sistema pode possuir, que é ao se modificar ligeiramente a entrada, o problema pode fornecer um resultado extremamente distante do resultado anterior. Normalmente, o problema é causado por diferenças em magnitude nas entradas da matriz (por exemplo: um dos pivôs com valor 1 trilhão, enquanto outro pivô com valor 0,0000001).

A definição matemática do condicionamento da matriz A é o máximo de $\|Ax\|/\|x\|$, para todo x , onde $\|V\|$ é a norma do vetor V . Esse valor pode ser computado através da decomposição em valor singular, tomando o maior valor e dividindo-o pelo menor valor singular não nulo. Normalmente, a resolução do sistema dá como resposta o vetor singular associado ao menor valor singular. Assim, o sistema é mal condicionado quando o condicionamento é muito acima de 1 (10^{10} , por exemplo). A equação do condicionamento da matriz é a seguinte:

$$\|A\| = \max \frac{\|Ax\|}{\|x\|} \quad \forall x \in R^n \quad (9)$$

¹² Tipicamente ocorre quando o objeto a ser reconstruído é uniforme.

3.4 Conceitos Relacionados

O contexto dos métodos de reconstrução possuem inúmeros conceitos associados ao embasamento teórico desse assunto. Alguns conceitos existem historicamente, enquanto outros surgem constantemente ao longo do processo evolutivo desses métodos. Três importantes conceitos vão ser bastante úteis para um melhor entendimento do foco deste trabalho: *sinograma*, *fan beam* e *fantoma*.

3.4.1 Sinograma

O conceito de sinograma está diretamente associado às projeções de raios gama. Um conjunto de projeções representa um sinograma da imagem que estar sendo varrida por essas projeções. Adotando-se f como função de reconstrução, o sinograma de f possui os mesmos dados que a transformada de Radon de f , mas no novo sistema de coordenadas gerado após a rotação θ . A Figura 9 mostra um ponto específico (x', y') da transformada de Radon sobre uma circunferência, a qual corresponde a uma senóide do sinograma S da imagem (GAZZANI, 1999, pág. 19).

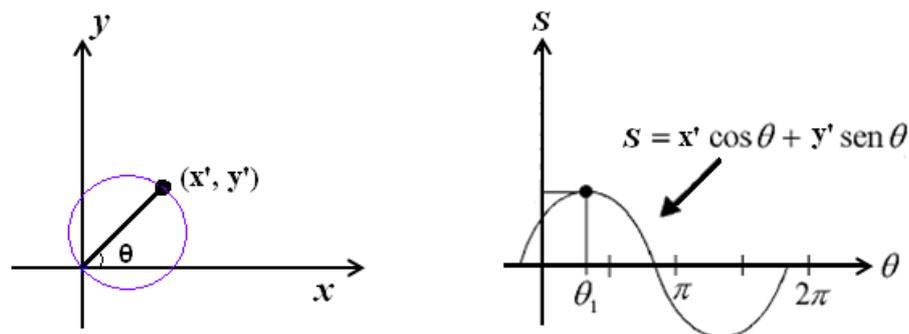


Figura 9 – Geração de um sinograma a partir do ponto (x', y') .

O sinograma diz respeito a um sistema *fonte-detector* fixo em cada projeção. Para uma nova projeção, o sistema é girado no ângulo θ desejado. Assim, quantas forem as projeções, também serão os novos sistemas de coordenadas criados. A aplicação da transformada de Radon nesses sistemas fornece o sinograma para o conjunto de projeções em questão (FRIDA, 2006). O sinograma, enfim, trata-se de uma figura bidimensional num plano (x, y) ,

por exemplo, onde os valores de x representam a detecção dos feixes gama, e os valores de y representam as inclinações θ desses feixes: posição angular do detector. Em síntese, o sinograma S de uma imagem é a representação da transformada de Radon da imagem original, conforme expressa a equação que se segue,

$$S(f) = \int_{a'} f dt \quad (10)$$

com t igual a medida de a' .

3.4.2 Fan Bean

No *fan bean* a projeção é obtida com fontes fixas e uma linha de detectores recebendo os raios de cada fonte. Esse formato geométrico do conjunto *fonte-detectores* dá uma ideia de “leque” (*fan bean*), e as integrais de linha são calculadas respeitando esse formato (GAZZANI, 1999, pág. 20). No *fan bean*, os raios não são paralelos, pois possuem aberturas diferentes uns dos outros, conforme mostra a Figura 10.

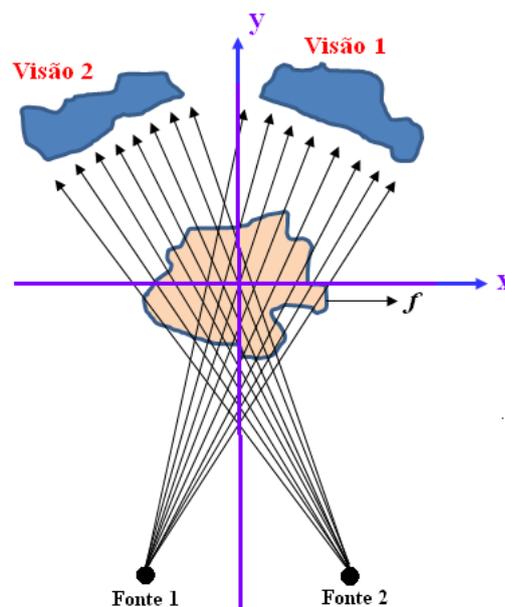


Figura 10 – Exemplo de uma projeção *fan bean*.
Fonte da imagem: (GAZZANI, 1999, pág 20).

3.4.3 Fantoma

Um fantoma ou *phantom* representa um objeto de teste com a finalidade de avaliar a performance da tomografia computadorizada, ou seja, serve para avaliar a precisão de um algoritmo de reconstrução de imagem. A base dos fantasmas são funções matemáticas que simulam as projeções tomográficas, que são as funções de reconstrução f e suas transformadas de Radon. Um fantoma possui valores numéricos bem definidos que promovem a discretização da imagem (os pixels). São valores que geralmente representam tons de cinza, que servem para mapear informações tais como densidade, coeficiente de atenuação (SUBBARAO, 1997, pág. 363). A Figura 11 mostra um exemplo de um fantoma para uma imagem *Spike*¹³ de 50x50 pixels.

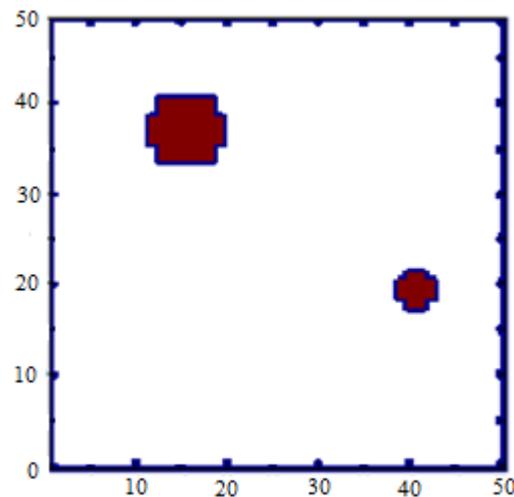


Figura 11 – Fantoma de uma imagem (*spike*) de 50x50 pixels.
Fonte da imagem: (VERHOEVEN, 1993).

3.5 Métodos Algébricos

Os métodos algébricos adotam, para a reconstrução da imagem tomográfica, que a seção transversal do riser é uma matriz desconhecida. Neste caso, várias equações algébricas são produzidas a partir das medidas dos dados projetados (SALINA, 2001, pág. 34). Na Figura 8, pode-se adotar a matriz 10x10 como sendo um *grid* (grade) retangular, e f como sendo uma imagem bidimensional de dimensões x e y , ou seja, $f = f(x, y)$. Assim, cada pixel do *grid* pode ser entendido como uma célula de $f(x, y)$. Assumindo-se que cada célula $f(x, y)$ é

¹³ Figura usada para testar a precisão de algoritmos de reconstrução, proposta por Verhoeven (1993).

constante, tem-se então que f_j seja o valor constante da j -ésima célula. Nessa abordagem, um raio passa a ser entendido como uma linha com certa espessura passando pelo plano (x, y) .

A espessura l do raio passa a ser importante, e a integral de linha é a soma da atenuação sofrida pelo raio dentro da seção do riser. Indexando cada raio i na projeção P , tem-se então P_i , como sendo a medida da soma do i -ésimo raio. Assim, tem-se na equação (11) a relação entre f_j e P_i .

$$P_i = \sum_{j=1}^N a_{ij} f_j \quad (11)$$

Na equação (11), N é a última célula e i pode variar até M . Assim, a forma MN indica uma configuração de M raios que produzem N células para reconstruir a imagem do objeto. O termo a_{ij} é denominado fator de peso, que representa o peso da j -ésima célula interceptada pelo i -ésimo raio. A Figura 12 mostra o cenário imposto pelo sistema em (11).

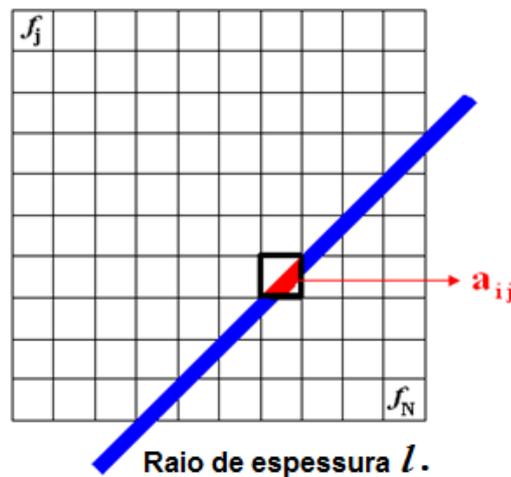


Figura 12 – Discretização destacando peso a , raio de espessura l , e as células f_j .

Os valores de M e N determinarão a forma a qual as equações do sistema em (11) irão ser escritas. Logo, tem-se:

$$\begin{aligned} a_{11} f_1 + a_{12} f_2 + a_{13} f_3 + \cdots + a_{1N} f_N &= P_1 \\ a_{21} f_1 + a_{22} f_2 + a_{23} f_3 + \cdots + a_{2N} f_N &= P_2 \\ &\vdots \\ a_{M1} f_1 + a_{M2} f_2 + a_{M3} f_3 + \cdots + a_{MN} f_N &= P_M \end{aligned} \quad (12)$$

O sistema de equações em (12) também pode ser expresso da seguinte maneira (ANTON, 2006, pág. 131):

$$\begin{bmatrix} a_{11} + a_{12} + a_{13} + \cdots + a_{1N} \\ a_{21} + a_{22} + a_{23} + \cdots + a_{2N} \\ \vdots \\ a_{M1} + a_{M2} + a_{M3} + \cdots + a_{MN} \end{bmatrix} \begin{bmatrix} f_1 \\ f_2 \\ \vdots \\ f_N \end{bmatrix} = \begin{bmatrix} P_1 \\ P_2 \\ \vdots \\ P_M \end{bmatrix} \quad (13)$$

Cada uma das equações de (12) pode ser considerada um *hiperplano* de um espaço N -dimensional. Quando existe apenas uma solução para (12), o ponto de interseção entre todos os hiperplanos representa essa solução. Para encontrar a solução de (12), o procedimento que deve ser seguido é o seguinte:

- (1) Considera-se uma solução inicial, projetando-a na primeira linha do sistema.
- (2) Reprojeta o ponto resultante de (1) na segunda linha.
- (3) O ponto resultante de (2) reprojeta na primeira linha.
- (4) O ponto resultante de (3) reprojeta na terceira linha.
- (5) O ponto resultante de (4) reprojeta na primeira linha.

Enfim, as soluções são projetadas linha a linha e, delas, com a primeira linha sempre. E, assim, sucessivamente.

A execução do procedimento recursivo descrito acima irá convergir para a solução do sistema, caso o mesmo tenha pelo menos uma solução. Adotando-se $\vec{f}^{(0)}$ como vetor solução inicial no espaço N -dimensional, esse valor inicial $\vec{f}^{(0)}$ aplicada ao hiperplano representado pela primeira equação de (12), resulta em $\vec{f}^{(1)}$. Assim, o valor $\vec{f}^{(1)}$ aplicado ao hiperplano representado pela segunda equação de (12), resulta em $\vec{f}^{(2)}$ e, assim, sucessivamente, até que $\vec{f}^{(i-1)}$ seja projetado no hiperplano representado pela i -ésima equação resultando em $\vec{f}^{(i)}$.

Esse algoritmo pode ser descrito da seguinte maneira, sendo também o que determina o procedimento de iteração dos métodos algébricos (SALINA, 2001, pág. 37):

$$\vec{f}^{(i)} = \vec{f}^{(i-1)} - \frac{\left(\vec{f}^{(i-1)} \times \vec{a}_i - P_i \right)}{\vec{a}_i \times \vec{a}_i} \times \vec{a}_i \quad (14)$$

3.5.1 Método ART

Os métodos algébricos de reconstrução de imagem buscam a solução do sistema de (12) pela projeção sequencial, a partir de um valor inicial $\vec{f}^{(0)}$. A Técnica de Reconstrução Algébrica (ART – *Algebraic Reconstruction Technique*) funciona dessa maneira. O ART é um método antigo, com indício de suas primeiras versões no final da década de 1930, a exemplo de sua citação em Kaczmarz (1937)¹⁴.

O ART foi usado em aplicações tomográficas computadorizadas em 1970 por Gordon e Herman (SUBBARAO, 1997). Naquele ano, o ART obteve êxito no contexto da tomografia industrial e, em consequência, serviu como base para vários outros métodos de reconstrução. O ART considera um conjunto de equações lineares, com cada uma delas representando um hiperplano, situação fiel ao que foi apresentado na seção anterior. O método calcula para cada raio gama as correções devidas, e as aplica nas células que correspondem ao referido raio.

O ART cria uma matriz imagem inicial com N elementos parametrizados em j . Em seguida, o método executa a primeira iteração k ($k=1$). Depois, são calculadas as projeções P_i para todos os raios gama, como expressa o sistema em (12). Esses são os passos gerais de funcionamento do algoritmo ART. O conjunto específico (mais detalhado) do funcionamento do método segue a seguinte lógica (OLIVEIRA, 2011, pág. 21–22):

- (a) É calculada para cada raio i a diferença entre a projeção original P_i e a projeção reconstruída P'_i :

$$\Delta P = P_i - P'_i \quad (15)$$

¹⁴ Stefan KACZMARZ: “Angentihrtte Auflosung von Systemen linearer Gleichunen”, 1937.

(b) É calculado o somatório dos pesos a_i ao longo de cada raio i :

$$a_i = \sum_{j=1}^N a'_{ij} \quad (16)$$

(c) É realizada a correção c_i para cada raio i :

$$c_i = \frac{\Delta P_i}{a_i} \quad (17)$$

(d) É aplicada a correção calculada em (c) para cada célula j ao longo do raio i , com λ representando o parâmetro de relaxação¹⁵, o qual é escolhido empiricamente num intervalo de $[0, 2]$:

$$f'_{k,j} = f'_{k-1,j} + \lambda \times a_{ij} \times c_i \quad (18)$$

(e) Então, uma nova projeção P_i é calculada, desta feita para o próximo raio i (P_2), e os passos de (a) a (d) são repetidos para essa nova projeção. Por fim, a iteração k é finalizada.

A última iteração é identificada como tal quando Δf atinge um valor pré-definido. A literatura científica representada pela maioria dos autores que dissertam sobre os métodos algébricos aponta o critério de $\Delta f \leq 0,01\%$ como um bom parâmetro para encerrar as iterações. Assim, quando Δf atinge este valor, nenhuma iteração será mais executada e, em consequência, k recebe seu último incremento ($k=n$).

Em Kak (2001) são discutidos os ruídos gerados na utilização do ART. São ruídos causados pelas inconsistências introduzidas no conjunto de equações, devido às aproximações impostas pelos pesos a_{ij} . Essa anomalia acontece porque cada pixel (célula f_j) ao longo de um raio i é alterado assim que a projeção é calculada. Neste caso, essa alteração é feita antes que outro raio i a faça.

¹⁵ As projeções j não necessitam estritamente que os vetores soluções representem as projeções exatas. Enfim, se aceita uma margem de tolerância que é quantificada no intervalo $[0, 2]$ (IUSEM, 1990, pág. 67).

O ART é um método muito importante da família dos algébricos. As próximas subseções irão apresentar os métodos MART, SIRT e SMART, que foram projetados para tratar aspectos específicos que não foram ou não são tratados pelo ART. Esses três métodos essencialmente possuem o mesmo embasamento matemático e lógica de funcionamento percorridos sobre os métodos algébricos. Assim, o destaque sobre eles é com relação aos aspectos relevantes que esses métodos diferem do ART.

3.5.2 Método MART

A Técnica de Reconstrução Algébrica com correção Multiplicativa (MART – *Multiplicative Algebraic Reconstruction Technique*) é considerada mais rápida e flexível do que o ART (VERHOEVEN, 1993). A rapidez está associada à velocidade na execução computacional, enquanto a flexibilidade está associada com a capacidade do método em atribuir mudanças dinâmicas à *lógica* de seu funcionamento, tal como mudanças que viabilizem uma convergência mais eficiente para a solução do sistema. Em Verhoeven (1993) é apresentado um estudo que indica que o MART é mais eficiente do que o ART em casos nos quais os dados são limitados, ou seja, quando os valores de M e N são pequenos.

De uma forma geral, o MART é bastante semelhante ao ART. A diferença mais notável é com relação ao fator de correção c . Para o MART, esse fator passa a ser feito de acordo com a seguinte equação:

$$c_i = \frac{\Delta P_i}{P_i \times a_i} \quad (19)$$

A diferença citada está ligada à presença da projeção P (P_i) no relacionamento que define a correção em questão. Neste caso, o elemento P_i na equação (19) promove uma correção mais rigorosa, matematicamente mais precisa, uma vez que a variação da projeção corrente no processo de iteração do método está recebendo como parâmetro relacional a própria projeção. Em termos práticos, isso é o mesmo que considerar a comparação da variação da imagem projetada com ela mesma. No ART, esta comparação não é feita, ou seja, as relações de correção no ART se restringem apenas aos pesos a .

3.5.3 Método SIRT

A Técnica de Reconstrução Algébrica Simultânea (SIRT – *Simultaneous Iterative Reconstruction Technique*) não foi essencialmente derivada do método ART, mas pode ser considerada uma variação dele, uma vez que ambas se assemelham muito. O SIRT foi desenvolvido independentemente do ART, em 1972 por Peter Gilbert (GILBERT, 1972). A diferença mais notável do SIRT para o ART é também com relação à correção c , sobretudo no que diz respeito à ordem em que essas correções são aplicadas. Tratam-se de correções aditivas e aplicadas simultaneamente, daí o adjetivo *simultaneous* atribuído ao método.

As correções no SIRT são implantadas ao final de cada iteração. No ART, as correções são aplicadas a cada célula f_j ao longo do raio i . No SIRT, as correções também são aplicadas a cada célula f_j , mas só que de uma única vez. Em suma, a correção definida para um pixel j no SIRT é feita de forma atômica, ou seja, sem considerar as correções que correspondem aos demais pixels do mesmo raio. Assim, o SIRT adota os mesmos passos do ART descritos de (a) a (d), procedendo fielmente os passos de (a) a (c) e, como mencionado, desenvolvendo o passo (d) com a distinção aqui discutida.

3.5.4 Método SMART

A Técnica de Reconstrução Algébrica Multiplicativa Simultânea (SMART – *Simultaneous Multiplicative Algebraic Reconstruction Technique*) é uma extensão do método MART. O método SMART é baseado na função de divergência conhecida como função de minimização da distância de Kullback-Leibler ou distância KL (YAMAOKA, 1978, pág. 166). É denominada nesses termos em homenagem aos cientistas que a formalizaram, os matemáticos americanos Solomon KULLBACK (1907-1994) e Richard LEIBLER (1914-2003)¹⁶.

A distância KL, também chamada de divergência KL, é a medida não simétrica da diferença entre duas distribuições de probabilidade. Uma delas supõe-se que seja a distribuição real, e a outra uma aproximação dessa distribuição real. A distância KL corresponde à diferença no número de bits necessário para codificar ambas as distribuições.

¹⁶ Hirotugu Akaike: “Akaike’s – An Information Criterion”, 1973.

Adotando-se a como distribuição real e b como a aproximação de a , a descrição geral da distância KL é expressa da seguinte maneira (BOZDOGAN, 2000, pág. 77):

$$D_{KL}(\bar{a}b) = a \times \log \frac{a}{b} + a - b \quad (20)$$

Assim, para o caso específico do SMART, a equação (11) é reescrita adicionando o valor D_{KL} , com o intuito de minimizar os efeitos desfavoráveis dos pesos a_{ij} atribuídos aos pixels f_j , no processo de reconstrução da imagem, conforme equação (20). Em cada iteração essa minimização é independente, podendo ser executada em paralelo (simultânea) em relação à outra iteração, daí também o adjetivo *simultaneous* atribuído ao método.

$$P_i = \sum_{j=1}^N D_{KL} a_{ij} f_j \quad (21)$$

3.5.5 Parâmetro de Relaxação

A eficiência no tratamento do ruído nos métodos algébricos também depende do parâmetro de relaxação adotado. São valores no intervalo $[0, 2]$. O valor é escolhido empiricamente e tem influência direta na taxa de convergência, o que, conseqüentemente, tem influência no desempenho desses métodos.

Em Oliveira (2011) é feito um estudo que mostra que não há um valor ideal de relaxação que vá contemplar igualmente todas as reconstruções de um mesmo método, pois mudanças tais como tipo do fantoma e quantidade de trajetórias, por exemplo, podem exigir um valor diferente do parâmetro para que o método reconstrua a imagem de forma mais eficiente possível. Trata-se de um estudo também de tomografia industrial, que usa fantasmas similares aos usados nesta tese, da mesma forma reconstruídos com os métodos da família ART. Tal pesquisa analisou diversas situações e concluiu sobre a incidência de valores que viabilizaram melhores resultados para cada método:

- ART e SIRT na ordem de 10^{-1} .
- MART na ordem de 10^{-2} .
- SMART na ordem de 10^{-3} .

Em Soto (2004) é apresentado um trabalho de otimização do ART usando diversos fantasmas, onde os valores de relaxação considerados ótimos estão em torno de 10^{-3} (SOTO, 2004, pág. 3), fato que novamente comprova que a atribuição do valor de relaxação depende do objeto analisado e das condições do estudo.

Os experimentos simulados realizados neste trabalho usaram os valores de relaxação determinados em Oliveira (2011), por serem consistentes com os melhores resultados alcançados pelos métodos algébricos e, principalmente, de isso ter sido realizado em fantasmas semelhantes aos analisados nesta pesquisa (barra, anel, círculo, quadrados).

3.6 Métrica de Erro GC

A taxa de erro calculada após a reconstrução da imagem é importante tanto para dimensionar a eficiência dos métodos, quanto para compará-los entre si. Embora haja outros parâmetros na reconstrução e as particularidades de cada método, o algoritmo que oferecer a menor taxa de erro – considerando as mesmas condições e configurações para o estudo –, teoricamente é o mais eficiente. Duas importantes métricas de erro são a RMSE e a GC. Ambas são usadas nesta tese de doutorado. A métrica RMSE é muito conhecida e amplamente discutida na literatura, além de ser bastante usada em trabalhos similares a este. A métrica GC não. Assim, esta subseção investe em apresentar a métrica GC e justificar seu uso nos experimentos desta tese.

Na métrica GC (*Geometric Coherence* – Coerência Geométrica) o cálculo de erro também é feito comparando a imagem reconstruída com o *fantoma* da imagem original. A diferença é que o GC usa os pixels em forma de polígonos, que representam a discretização da imagem de saída. São pixels denominados de pixels naturais¹⁷. O GC faz cálculo de aglomeração desses polígonos através de suas arestas, considerando a junção desses polígonos e a distância¹⁸ entre eles.

¹⁷ Maiores detalhes sobre o conceito de *pixel natural* são apresentados no próximo Capítulo.

¹⁸ Não há uma medida específica para a distância em questão, uma vez que o PSIRT foi concebido inicialmente para modelagem. Assim, qualquer situação em que dois polígonos não estejam colados, é considerada como sendo uma distância entre eles.

Adotando-se S como o sistema de cores usado no mapeamento de informações da reconstrução da imagem, e $P(g)$ como sendo o conjunto de pixels de tons de cinza g , tem-se:

$$S(g) = P(g) \cup P(g+1) \cup \dots \cup P(255)^{19} \quad (22)$$

Adotando-se $E(A)$ como sendo o conjunto de arestas externas (de fronteiras) do conjunto de pixels A , e $n(A)$ a cardinalidade do conjunto de pixels A , tem-se, então, a equação que determina o GC para um conjunto de pixels naturais (VASCONCELOS, 2011, pág. 7):

$$GC(g) = 1 - \frac{n[E(S(g))]}{6n(S(g))} \quad (23)$$

A Figura 13 mostra a aplicação do GC, propositadamente exemplificado com hexágonos, pois é o tipo predominante de polígono que representa a discretização do PSIRT.

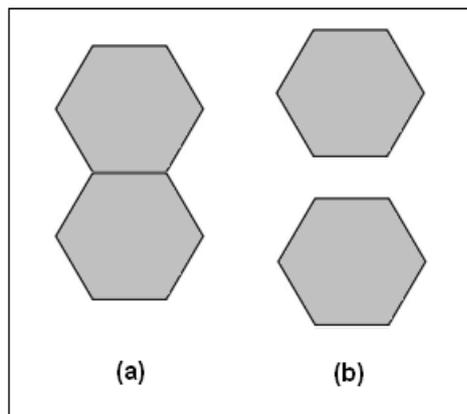


Figura 13 – Exemplo da aplicação do GC com dois hexágonos cada.

- Usando a equação (23) para os hexágonos adjacentes da parte (a) da Figura 13, tem-se, então:

$$GC(g) = 1 - \frac{11}{6 \times 2} \Rightarrow GC(g) = 0,0833333$$

¹⁹ Essa equação serve apenas para o padrão com 8 bits, que gera 256 (0 a 255) valores para representar os níveis de cinza.

- Usando a equação (23) para os hexágonos separados da parte (b) da Figura 13, tem-se, então:

$$GC(g) = 1 - \frac{12}{6 \times 2} \Rightarrow GC(g) = 0$$

- Aplicando o GC agora para três hexágonos, de acordo com Figura 14.

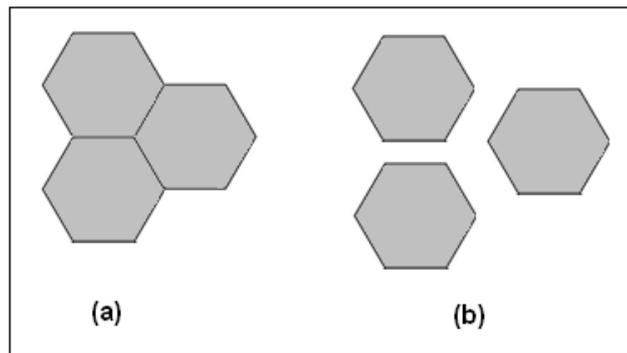


Figura 14 – Exemplo da aplicação do GC com três hexágonos cada.

- Para a parte (a) da Figura 14, tem-se, então:

$$GC(g) = 1 - \frac{15}{6 \times 3} \Rightarrow GC(g) = 0,166666,$$

- Para a parte (b) da Figura 14, tem-se, então:

$$GC(g) = 1 - \frac{18}{6 \times 3} \Rightarrow GC(g) = 0$$

Conclusão sobre o cálculo do GC com relação as figuras 13 e 14:

- Para polígonos separados (partes (b) das figuras), o valor é zero, que é o pior caso. Na prática, isso significa dizer que os polígonos vizinhos – mas não colados, não contribuem coletivamente para o mapeamento do objeto que estar sendo reconstruído. Contribuem apenas isoladamente, ou seja, cada um com seu peso.

- Já os polígonos vizinhos e adjacentes (partes (a) das figuras) contribuem coletivamente para a reconstrução do objeto. No exemplo, para dois polígonos, o valor do GC foi de aproximadamente 0,08, enquanto para três polígonos o valor foi de aproximadamente 0,16. Assim, quanto mais polígonos juntos, melhor é a reconstrução. Então, quanto mais o GC se aproxima de 1, é porque se tem cada vez mais polígonos juntos, que é a situação mais desejada.
- Entretanto, calcular o GC isoladamente não faz muito sentido, pois a reconstrução requer que a figura de entrada (fantoma) seja comparada com a figura de saída. Assim, se faz necessário estabelecer diferença entre o GC da figura de entrada com o GC da figura de saída. Adotando-se F_E como figura de entrada, e F_S como figura de saída, tem-se:

$$\Delta GC = GC(F_E) - GC(F_S) \quad (24)$$

Para (24) o desejado é que o valor de ΔGC se aproxima cada vez mais de zero, ou seja, se essa diferença se aproximar de zero, significa que a imagem reconstruída está se tornando idêntica a imagem original. Porém, essa situação não se aplica para os casos dos polígonos separados, pois não faz sentido, uma vez que não há contribuição *global* desses polígonos na reconstrução da imagem. Neste caso, o valor ΔGC só é calculado para GC maiores que zero. No caso do polígono usado no exemplo, o hexágono, o valor mínimo é de 0,083, calculado a partir da quantidade mínima de hexágonos: dois.

Por fim, embora a métrica GC não seja muito comum de ser usada em avaliações de natureza tomográfica, sua adoção neste trabalho é para dimensionar o quanto a aglomeração de hexágonos branco e cinza é um recurso eficiente no PSIRT. Ou seja, a discretização em hexágonos é uma vantagem do método em relação aos métodos da família ART, e uma maneira também de se analisar isso é usando a métrica GC.

Capítulo IV

4. Método de Reconstrução de Imagem PSIRT

4.1 Sistemas de Partículas

A base do PSIRT são os sistemas de partículas. Assim, antes da descrição propriamente dita do método, esta seção irá apresentar uma revisão bibliográfica destacando o Estado da Arte desses sistemas.

Os sistemas de partículas é um segmento promissor da computação gráfica, uma importante e evolutiva área da Ciência da Computação. Os sistemas de partículas têm sido usados há vários anos na modelagem e simulação de sistemas dinâmicos. Em Witkin (1997) é apresentada uma aplicação particular que usa os princípios de um sistema de partículas dinâmico. Nessa aplicação, uma partícula possui massa, posição, velocidade e responde à ação de uma força, mas não possui extensão espacial, característica esta que simplifica a dinâmica do sistema. E é esta simplificação que credencia métodos desse tipo a serem usados pela computação gráfica no estudo de uma enorme variedade de fenômenos naturais e objetos de formato e geometria pouco definidos (REEVES, 1983).

Em Blinn (1982) é visto um dos primeiros trabalhos dessa natureza, onde partículas são usadas para a simulação de nuvens e áreas áridas (deserto, tempestades de areia). Em Reeves (1983) é apresentada uma importante técnica com partículas para a modelagem de objetos irregulares ou ondulares. Em Miller (1989) é feita a simulação de fluidos viscosos com sistema de partículas. Em Sims (1990) uma animação é realizada usando computação paralela, através de um sistema para síntese e modelagem baseadas em partículas, tendo como principal finalidade a simulação e visualização de dados científicos (várias grandezas – variáveis independentes, multidimensionais: carga, temperatura, pressão, viscosidade).

Em Szeliski (1991) é feita uma modelagem de superfície de geometria variada, usando o sistema de partículas. Em Breen (1994) é proposto um sistema baseado em partículas para simular roupas de forma realista. Em Marks (1997) é apresentada uma aproximação para a calibragem de parâmetros para a computação gráfica e animação baseada em partículas.

Em Steigleder (1997) a reconstrução apresentada usa o sistema gráfico de partículas. Nessa reconstrução, os resultados alcançados comprovam o método com partículas como uma alternativa eficiente na modelagem de objetos *fuzzy*. Na modelagem de nuvem, por exemplo, toda gota de água suspensa no ar pode ser definida como sendo uma partícula. Assim, a configuração inicial da partícula pretende reproduzir o comportamento de uma nuvem sob certas condições físicas, incluindo o fenômeno chuva (STEIGLEDER, 1997).

Em Witkin (1997) outro importante trabalho sobre o assunto é apresentado. Esse trabalho é baseado na dinâmica do sistema de partículas usada na computação gráfica, basicamente de maneira semelhante ao que foi feito em Steigleder (1997). Um ponto importante em Witkin (1997) é a apresentação de um *código-fonte* (linguagem C²⁰) para definir o comportamento das partículas no sistema computacional. Este código reproduz a simplicidade de usar partículas para modelar sistemas dinâmicos. Trata-se de um código fácil de implementar e entender, não requerendo um avançado conhecimento em programação.

Em Hug (1999) é visto um importante trabalho sobre o uso de partículas para segmentação semiautomática, a qual trata a falta de informação na imagem (dados incompletos). O sistema de partícula é também usado em aplicações tais como animações de objetos, como pode ser visto em Amrani (2000) e novamente em Sims (1990).

De fato, a abordagem de sistemas de partícula tem sido usada eficientemente na computação gráfica, em contextos tais como simulação, modelagem e animação de diversos objetos e fenômenos naturais. A comunidade científica está atenta para esta realidade promissora, investindo em pesquisas e estudos sobre o assunto. Além disso, a área de reconstrução gráfica carece de métodos rápidos e eficientes e de alternativas que, ao menos, possam ser competitivas com as técnicas analíticas e algébricas atuais. O PSIRT é uma delas.

²⁰ Linguagem de programação universalmente usada na construção de sistemas científicos.

4.2 PSIRT – Apresentação Inicial

O método proposto nesta tese é do tipo iterativo e discreto, duas características também dos métodos algébricos discutidos no capítulo anterior. A técnica de reconstrução iterativa usando sistema de partículas, o PSIRT (*Particle System Iterative Reconstruction Technique*), considera cada pixel natural²¹ – que cobre na imagem uma área que representa material –, como sendo uma partícula com certas características físicas.

Uma partícula é submetida a um conjunto de forças, o qual pode produzir movimentos na mesma. Um pixel natural é a menor área (detalhes detectados) na imagem que pode ser detectada pela configuração escolhida de projeções e trajetórias. Uma partícula no PSIRT é associada ao pixel natural se, somente se, é suposto representar a presença do material que se pretende reconstruir graficamente. A partícula, para fins de dinâmica, pode ser descrita como um círculo diminuto em comparação ao espaço entre trajetórias adjacentes.

Então, é associada uma força (um campo de força) a uma dada trajetória de tal forma que sua direção (da força) é ortogonal à trajetória, com a orientação convergindo para a referida trajetória. A intensidade da força (do campo) é proporcional à diferença entre a atenuação sofrida pelo feixe gama na configuração atual, e a atenuação correspondente na leitura experimental para a projeção em questão.

A intensidade do campo é constante ao longo de direções paralelas à trajetória. A intenção é fazer com que as partículas se aproximem da trajetória, toda vez que a atenuação calculada na configuração atual seja inferior à atenuação experimental para essa trajetória. A intensidade F da força que atua na partícula é inversamente proporcional à distância ao quadrado da trajetória j à partícula p (a seção 4.10 mais a frente apresenta a formalização matemática do método).

Inicializando com certa quantidade de partículas, elas são distribuídas de maneira uniformemente aleatória ao longo da geometria que modela a seção do riser. Essa quantidade é proporcional às leituras experimentais de todas as projeções. O sistema computa para cada partícula todas as forças que agem sobre a mesma, produzindo assim, a resultante, a qual dá a direção e a orientação para mover as partículas. Executa-se, então, um passo (movimento atômico) para todas as partículas.

²¹ Pixel natural representa a unidade atômica (indivisível) de informação fornecida pela reconstrução. O pixel natural é delimitado pelas adjacências dos raios e seus envelopes de influência. O pixel natural no PSIRT é diferente do pixel nos métodos algébrico. Nesses métodos, os pixels são as informações computadas em cada célula da matriz formada pelas projeções e trajetórias de raios gama. Enfim, no PSIRT, o pixel é mais abrangente e flexível.

Toda vez que as partículas atingem uma trajetória, a atenuação computada para esta trajetória aumenta, o que acarreta na diminuição da intensidade da força na trajetória em questão para a próxima iteração. Quando a intensidade das forças de todas as trajetórias atingir um valor mínimo aceitável (zero ou próximo de zero), então o sistema é declarado em equilíbrio (estabilizado), e as posições finais das partículas – uma vez substituídas pelo pixel natural pintado –, torna-se a reconstrução oferecida pelo PSIRT.

4.3 PSIRT – Parâmetros

Alguns parâmetros necessitam ser calibrados para que o algoritmo do PSIRT possa trabalhar da forma mais eficiente possível. Como exemplo, pode-se definir a constante de proporcionalidade para a intensidade da força *versus* a diferença da atenuação entre a configuração atual e as leituras experimentais. Outro parâmetro é a quantidade total de partículas, a qual não necessita ser superiormente delimitada, e deve ser diretamente relacionada à quantidade de pixels naturais na imagem. Como apenas a informação disponível é a das projeções, então esta relação deve ser arbitrada.

Se acontecer daquele valor (quantidade de partículas) ser muito pequeno, o sistema não converge para uma solução. No entanto, se a quantidade de partículas for grande demais, então, ou as partículas excedentes são naturalmente eliminadas ou elas são aglutinadas pelo programa, de modo a formar um conjunto menor de partículas, que é utilizado para satisfazer interseções de trajetórias que possuem diferenças não nulas em relação às atenuações experimentais.

A vantagem de iniciar com um número relativamente grande de partículas é a facilidade com que o sistema alcança o equilíbrio. Mas se o número for grande demais, ressaltando, o sistema poderia levar um longo tempo para alcançar o equilíbrio, já que mais processos computacionais teriam que ser instanciados para a execução. Outros parâmetros são o raio de uma partícula e a tolerância para a força ao redor de cada trajetória. Como a intensidade diminui de acordo com a distância, a intensidade da força próxima à trajetória pode tornar-se muito grande. A solução do PSIRT para isto é estabelecer que a intensidade seja constante numa região próxima de uma trajetória.

4.4 PSIRT – Discretização

Com relação à discretização, o PSIRT inicialmente foi projetado para o modelo “M=projeções e N=trajetórias” na configuração 3x7, ou seja, três projeções e sete trajetórias por projeção, totalizando 21 raios gama. A Figura 15 mostra de forma geral como é a discretização no PSIRT para o modelo 3x7.

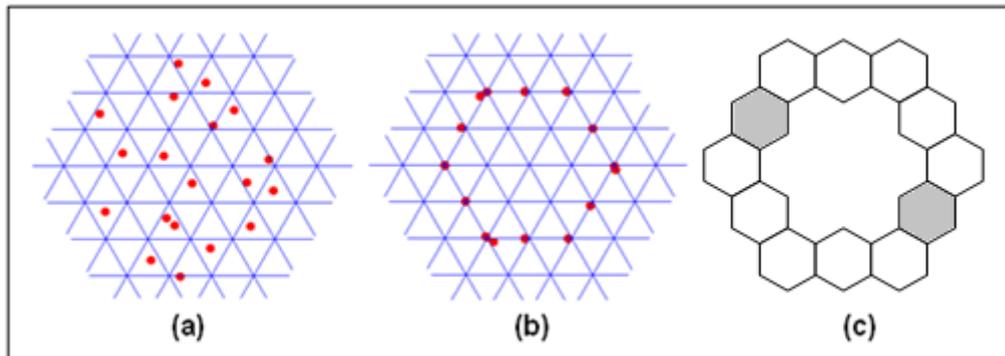


Figura 15 – (a) Uma distribuição uniformemente aleatória de partículas na configuração 3x7, (b) o PSIRT atingindo o equilíbrio, (c) e o resultado final com a discretização em hexágonos.

Para a simulação, é convertida a imagem em um conjunto de funcionalidades que atribui para cada trajetória uma quantidade de partículas necessárias para produzir a atenuação correspondente. Esse conjunto representa a informação de projeção para a simulação. Uma vez que o sistema finaliza a computação, é convertida cada partícula em um pixel natural, o qual na configuração 3x7 cobre uma região hexagonal dos pixels da imagem. A Figura 16 ilustra a discretização para uma partícula que cai numa intersecção tripla.

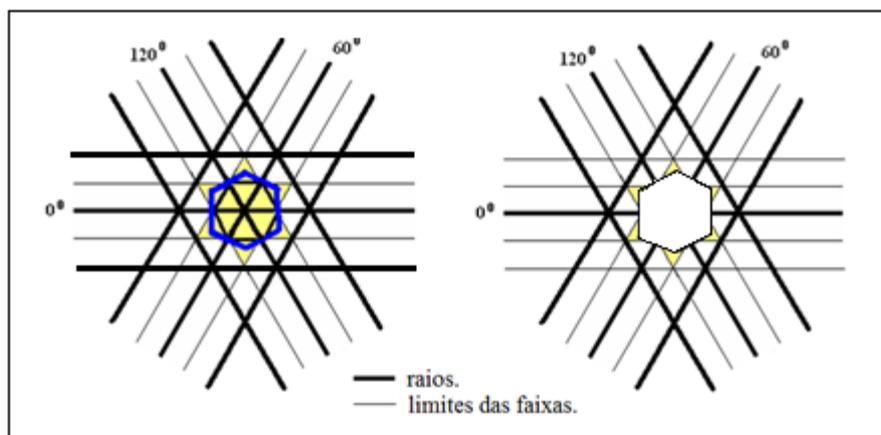


Figura 16 – Conversão de uma partícula em pixel natural (discretização).

Uma partícula pode finalizar seu deslocamento e ficar próxima a uma intersecção tripla, mas não exatamente nela. Este é o caso onde duas trajetórias detectam o material, mas a terceira não detecta. Nesse caso, somente metade do hexágono representa aquela partícula. Uma vez que existe uma ambiguidade com relação a qual metade do hexágono deve ser preenchida, então o hexágono é pintado todo com a cor cinza ao invés de branco, já que a cor branca é no caso da intersecção tripla. Uma partícula pode se localizar somente numa trajetória, mas neste caso não existe uma localização determinada ao longo da trajetória que vá essencialmente representar a presença do material.

4.5 PSIRT – Equilíbrio (Estabilidade)

Na implementação do PSIRT, o sistema remove as partículas e inicializa novamente, utilizando o último resultado como configuração de entrada. Entende-se que este é o caso em que existe uma quantidade excedente de partículas, e é possível que outras partículas – localizadas em bifurcações – possam ser relocadas para formar intersecções triplas para satisfazer a possível necessidade de uma trajetória. Se o sistema não alcançar o equilíbrio para um dado tempo de tolerância, então aquele excesso de partículas é considerado ruído.

Entender a finalização de uma execução como suficientemente satisfatória é uma questão nobre no PSIRT. Ou seja, atingir o equilíbrio para certo experimento não significa gerar uma solução definitiva nem mais exata, porque o fim de uma execução pode ser o cenário de entrada para começar outra execução, de uma maneira dinâmica. Nesse caso, o sistema produzirá um resultado mais eficiente, uma vez que os dados de entrada já são uma solução significativa para a reconstrução gráfica do objeto em questão.

Em síntese, novas execuções tentarão fazer com que as trajetórias atraiam as partículas para formarem intersecções triplas ou ao menos intersecções duplas, além de garantir que as intersecções triplas já formadas não sejam desfeitas. A melhor situação do sistema seria o caso em que todas as intersecções fossem triplas (todos os hexágonos seriam pintados de branco). Trata-se de uma situação exaustiva de ser alcançada e relativamente desnecessária, uma vez que as intersecções duplas também produzem dados úteis para a reconstrução do objeto, ou seja, também representam a presença do material.

O aspecto do PSIRT que leva ao equilíbrio é a intensidade das trajetórias. Enquanto houver trajetórias com algum valor de intensidade não nulo, o sistema continua executando a dinâmica das partículas, em busca de uma solução cada vez melhor (em busca de mais interseções triplas e duplas). Em contrapartida, quando o conjunto dos campos se aproximar de zero, a situação de equilíbrio é finalizada. Em consequência, as partículas que não atingiram nenhuma trajetória devem ser removidas.

Porém, não é apenas na última execução que a remoção de partículas deve ocorrer. Partículas podem ser removidas ao longo das iterações. Isso tipicamente acontece quando se atribui uma quantidade muito grande de partículas. Remover partículas desnecessárias numa certa execução ajuda o sistema a atingir o equilíbrio na próxima execução e, assim, sempre.

4.6 PSIRT – Posição das Partículas

Analisando com mais detalhes a lógica de funcionamento do PSIRT, a posição das partículas é uma informação decisiva para o sucesso da reconstrução da imagem. A Figura 17 mostra os diversos tipos de posicionamento das partículas quando o sistema atinge o equilíbrio, classificando-os de A até H.

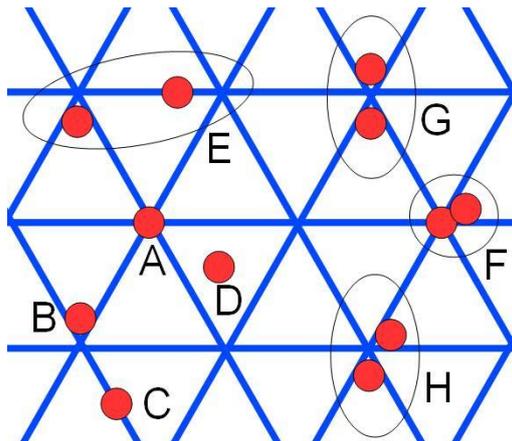


Figura 17 – Possíveis posições das partículas no PSIRT em equilíbrio.

O tipo “A” é caso mais desejado, que é quando as três trajetórias encontram o material (interseção tripla). O tipo “B” é formado por uma partícula perto de um cruzamento triplo, mas tocando apenas em duas trajetórias, o que significa que a terceira trajetória não *enxerga* o material da partícula correspondente. Em “B”, o problema da ambiguidade quanto a que lado à partícula está da trajetória que não a enxerga é resolvido pintado o hexágono (pixel natural) na cor cinza, conforme já mencionado.

No tipo “C” apenas uma trajetória enxerga o material. Neste caso, a informação não será útil para execução atual, pois se sabe que a trajetória possui uma partícula, mas não se sabe exatamente onde essa partícula está localizada na referida trajetória. Enfim, sua posição é desconhecida, mas sua existência garante a presença de material. Para tentar aproveitar essa partícula *solta* na próxima execução, ela é reposicionada (de maneira uniformemente aleatória) quando da reinicialização da nova configuração de entrada. Assim, espera-se que ao menos duas trajetórias atraiam a partícula solta na nova execução. Esse é um risco aceitável, pois pode acontecer que nenhuma das trajetórias encontre a partícula, nem mesmo a que a encontrou na execução anterior.

A partícula do tipo “D” não apenas não fornece nenhuma informação útil para a reconstrução, como também não contribui para o equilíbrio de forças após a execução, já que nenhuma trajetória a encontrou (atraiu) e, portanto, nenhuma trajetória diminuirá sua intensidade. Esse tipo de partícula deve ser removida, independente se ocorrerá ou não outra execução. Já o tipo “E” é um caso particular da ocorrência simultânea dos tipos “B” e “C” em trajetórias complementares. Como essas partículas estão muito próximas, essa situação deve ser convertida para interseção tripla, transformando (trazendo) “C” em “B”. Em síntese, é como substituir uma partícula “B” e uma “C” por uma “A”. Isso é feito inclusive sem afetar as forças, uma vez que essa substituição é equivalente fisicamente.

O tipo “F” é composto de um tipo “A” adjacente a um tipo “B”. É uma situação não interessante, pois o tipo “A” é suficiente para a interseção das três trajetórias. Enfim, é um desperdício. Nesse caso, a partícula do tipo “B” deve ser também reposicionada para a próxima execução, com os riscos já mencionados. Esse procedimento também é válido para partículas do tipo “G”, que é o caso em que a interseção possui duas partículas do tipo “B” adjacentes. Uma delas é reposicionada para a nova execução.

A partícula do tipo “H” é composta de duas partículas “B” adjacentes, mas que têm uma das trajetórias que as detectam diferente. Nesse caso, o mais interessante é substituir apenas uma delas por uma partícula do tipo “A”, e reposicionar a outra partícula.

Ressalta-se que, remover, substituir ou reposicionar partículas não irão causar grandes *danos* ao funcionamento do PSIRT, em relação à busca do equilíbrio em execuções seguintes. A remoção implica em quase nada, apenas na diminuição da quantidade de partículas que, naturalmente, ajudará no equilíbrio da próxima execução. Na substituição, a trajetória só será computada com relação a sua intensidade, respeitando o tipo da partícula que tomou o lugar da outra. No reposicionamento, que é uniformemente aleatório para todos os casos, as intensidades das trajetórias envolvidas não serão alteradas para a próxima execução.

4.7 PSIRT – Eficiência

Num cenário no qual partículas que podem garantir a presença de material são reposicionadas e as que não garantem são eliminadas – sendo esses dois aspectos associados ao fato de que as trajetórias mantêm suas necessidades de atração de partículas para a próxima iteração –, há uma situação de extrema propensão para atingir o sucesso na reconstrução do objeto. Trata-se de uma situação contínua de busca da solução ideal, sendo facilitada significativamente a cada nova iteração.

Com essas regras de funcionamento, certa iteração sempre fornecerá um resultado melhor do que a iteração anterior. Entender quando parar, ou seja, quando o algoritmo irá produzir uma execução que forneça uma reconstrução eficiente, é uma questão aberta, pois vai depender dos vários parâmetros das configurações de entrada, tais como formato do objeto, resolução da imagem, quantidade de projeções e trajetórias, quantidade de partículas, estabilidades satisfatórias, tolerância do tempo de execução (parcial e total), dentre outros.

Entretanto, na sequência operacional do PSIRT, uma vez finalizadas as execuções de um conjunto determinado dinamicamente de reinicializações, o sistema se encontra pronto para produzir a imagem. Vistas de formas isoladas, as posições finais desejadas para as partículas são dos tipos “A” e “B”. As demais posições, com exceção das do tipo “D” que serão excluídas, são combinações que de alguma maneira serão forçadas a se converterem para uma daquelas duas, pelo menos que se tornem do tipo “B”.

A partir das partículas de posições “A” e “B” é que são gerados os pixels naturais que, conforme já foi dito, são hexágonos formados pelas interseções das três trajetórias, que são pintados de branco quando as três trajetórias encontram o material (partícula “A”), e de cinza quando apenas duas trajetórias encontram o material (partícula “B”). A cor preta é quando não há material. A Figura 18 é uma versão ampliada das partes (b) e (c) da Figura 15.

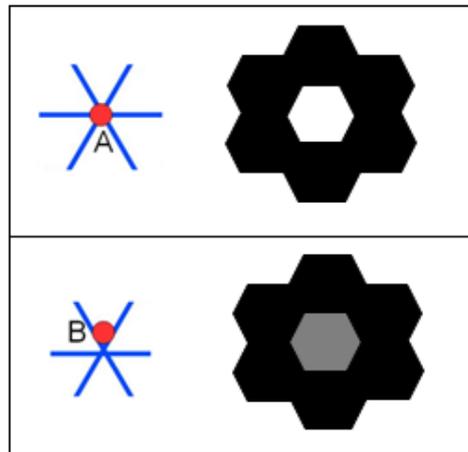


Figura 18 – Reconstrução de imagem de partículas de posições “A” e “B”.

A Figura 18 representa a reconstrução de imagem oferecida pelo PSIRT. A discretização em pixel natural é um recurso bastante interessante e já consagrado em outros métodos de natureza similar, a exemplo do trabalho visto em Azzi (1991), uma importante reconstrução e também uma das pioneiras no contexto do riser do tipo FCC. O pixel natural, devido sua possibilidade de expansão espacial, é um dos principais recursos que torna o PSIRT eficiente.

4.8 PSIRT – Pixel Natural Abrangente

O pixel natural no PSIRT cobre uma região hexagonal atômica (menor região possível) onde se pressupõe a existência de material. Essa região preenche o espaço ocupado por uma única partícula, de forma uniforme e em todas as direções. O pixel natural engloba não só os raios que encontraram o material como também as adjacências direita e esquerda, superior e inferior, desses raios. A Figura 19 dá uma visão mais precisa dessa informação.

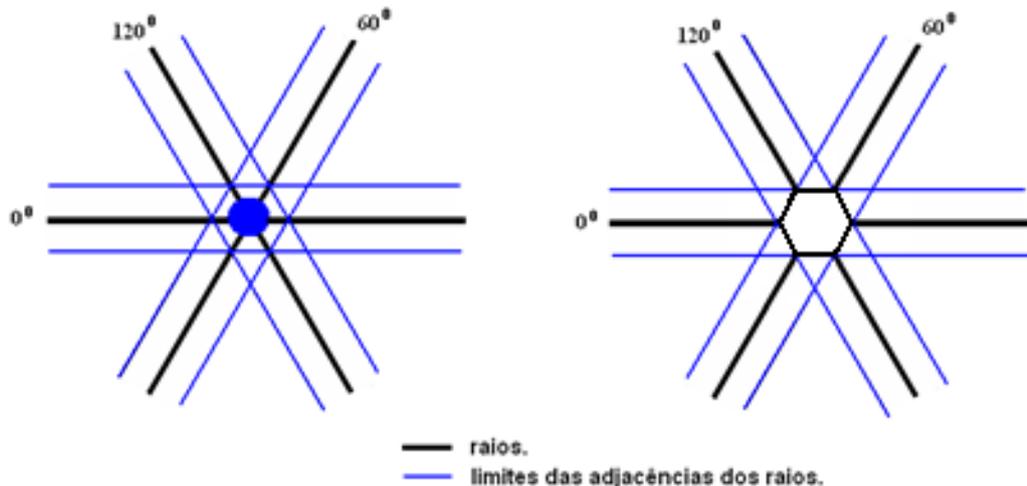


Figura 19 – Visão das regiões adjacentes dos raios que são cobertas pelo pixel natural.

A situação exposta na Figura 19 pressupõe que, uma vez que a partícula localizada na interseção – tendo sido encontrada por duas ou três trajetórias –, representa a presença de material e, considerando que esse material deve se fazer presente não apenas no espaço exato ocupado pela partícula, o método então trabalha com uma *expansão* “territorial” prudente, no sentido de cobrir uma área maior possível do material. Como isso irá ocorrer para todas as partículas “A” e “B”, o resultado é que a imagem final seja a mais próxima possível do objeto que estar sendo reconstruído.

Como é sabido, as partículas de um sistema de partículas mapeiam satisfatoriamente objetos de formatos irregulares. Assim, podendo ampliar a área de abrangência de cada partícula, numa distância curta o suficiente para não sair da região do material e, ao mesmo tempo, longa com limites para tentar cobrir ao máximo o material num espaço significativo, é um avanço considerável num método de reconstrução. E, no caso específico do riser tipo FCC, as adjacências dos raios confinadas nos hexágonos são regiões altamente propensas à existência do catalisador, por este possuir um formato relativamente irregular. A Figura 20 dá uma visão desse cenário, simulando uma situação hipotética da expansão de partículas como se estivessem reconstruindo uma região representando o catalisador.

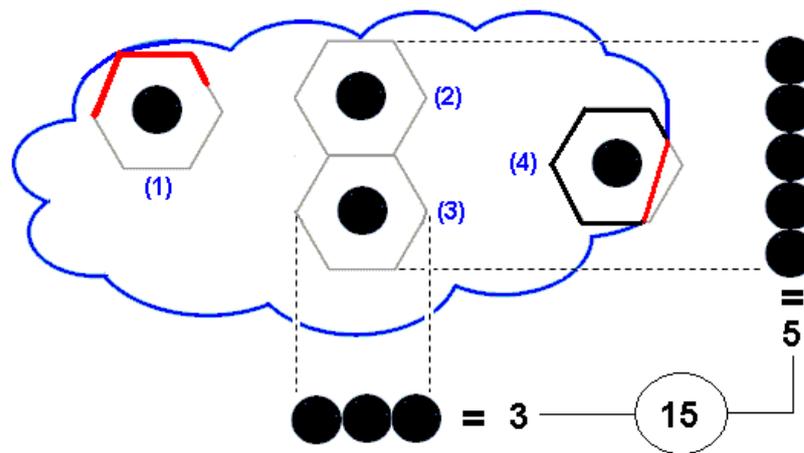


Figura 20 – Partículas com expansão hexagonal na reconstrução de um objeto de formato irregular simulando ser uma região ocupada pelo catalisador.

A expansão (1) é uma situação altamente desejada, pois a área hexagonal, cujos limites estão destacados de vermelho, se aproxima muito do contorno que representa uma das fronteiras do objeto. As expansões (2) e (3) são bastante comuns de serem geradas, pois no interior do objeto fatalmente ao menos duas trajetórias irão encontrar partículas. Elas (as expansões) estão ilustrando que, sem a expansão hexagonal, seriam necessárias quase 15 partículas para representar a mesma área do objeto (sendo três na base e cinco na altura). Essa é uma quantidade aproximada, pois a junção de hexágonos forma um retângulo imaginário incompleto, além das partículas cobrirem áreas circulares. Mesmo assim, a diferença é muito grande, que no exemplo da Figura 20 é de quinze contra dois.

A expansão (4) é uma situação em que o hexágono vaza uma das fronteiras do objeto (além da linha vermelha). Mesmo assim, as regiões do hexágono cujos limites estão destacados com linhas pretas, justificam a “invasão” desse pedaço fora do objeto. Uma questão importante nisso é que, como o catalisador possui um formato irregular e parecido com nuvens, as partículas detectadas no seu interior e próximas as suas fronteiras irão se aproximar de seu formato real e, assim, irão reproduzir com bastante eficiência a reconstrução catalítica em seções do riser.

4.9 PSIRT – Configuração dos Parâmetros

Os parâmetros físicos das partículas discutidos na seção 4.3 precisam ser calibrados para que o algoritmo funcione plenamente:

- **Constante de proporcionalidade:** ajusta a intensidade da força de acordo com a diferença entre a “atenuação computada para a partícula” e a “atenuação real atribuída a uma dada trajetória”. O valor padrão é um fator de 0,005 para leituras normalizadas e simuladas do detector.
- **Quantidade total de partículas:** a configuração inicial é feita com um valor proporcional, calculado de acordo com a atenuação global a partir das leituras experimentais, denominado “valor referência”. Se a quantidade inicial de partículas for demasiadamente pequena – bem menor que o valor referência, o sistema não converge para o equilíbrio, embora, normalmente, produza uma reconstrução que pode ser considerada uma aproximação para aquela quantidade inicial aquém de partículas. Entretanto, se a quantidade de partículas for bem maior do que o valor referência, as partículas excedentes são naturalmente removidas pelo próprio sistema. A base usada para o cálculo do valor referência é o fator de oito vezes os valores normalizados para cada trajetória. Esse cálculo é feito a partir do sinograma do fantoma original, que é uma maneira de representar as projeções. Por exemplo, para o caso de três projeções e sete trajetórias, a quantidade total de partículas pode chegar a 168 unidades ($M \times N \times 8 = 3 \times 7 \times 8$).
- **Tolerância para a força em torno de cada trajetória:** a intensidade é definida como constante ao longo da trajetória. É um processo que simplifica as configurações e ao mesmo tempo torna eficientes as execuções.
- **Aceleração das partículas:** é a quantidade fracionada de pixels por unidade quadrada de tempo. O valor de referência é de 0,1.
- **Atração e repulsão de partículas:** o nível de atração e repulsão de partículas é feito por um controlador ao longo de cada trajetória. Esse parâmetro também depende da quantidade inicial de partículas, uma vez que as trajetórias vão se comportar nesse sentido baseadas no que cada uma precisa para sua estabilidade individual.

4.10 PSIRT – Formalização Matemática

No PSIRT, a *força* emanada por uma trajetória atinge uma partícula, procedimento de um modelo matemático baseado na atração gravitacional ou de atração elétrica de partículas carregadas. A força em questão segue a seguinte equação,

$$F = K \frac{M_1 M_2}{d^2} \quad (25)$$

onde K é uma constante, M_1 a massa da partícula, M_2 a massa da trajetória, e d a distância entre partícula e trajetória. No caso de se considerar $M_1 = M_2 = 1$, e se deixar K como parâmetro, a equação (25) é reduzida para a seguinte equação, o que caracteriza que a força é ortogonal à trajetória:

$$F = \frac{K}{d^2} \quad (26)$$

As forças são ortogonais às trajetórias e atuam sobre as partículas vetorialmente. A resultante R é calculada, então, pela soma vetorial das forças, conforme a seguinte expressão:

$$R = F_1 + F_2 + \dots + F_n \quad (27)$$

A intensidade da força determina a velocidade com que a mesma se desloca, medida em pixels por unidade de tempo. Como $M_1 = 1$ e $F = M_1 a = a$, a aceleração da partícula vai sempre ser positiva. Assim, as partículas deslocarão com velocidades distintas. Por conta da relação de K com a aceleração, o usuário tem como opção o parâmetro “aceleração” para a constante K . Para evitar a explosão do valor de F quando $d \rightarrow 0$, é deixado um envelope em torno de cada trajetória onde F possui valor constante. São parâmetros do usuário.

A aceleração da partícula promove duas situações quando do movimento dela na execução do programa: (1) quando a aceleração é alta demais, a partícula pula algumas trajetórias, muitas das quais não tem interesse por ela (não possuem necessidade de partícula); (2) quando a aceleração é baixa demais, a partícula não consegue pular trajetórias e acaba ficando presa numa trajetória, a qual pode não estar precisando de tal partícula. A escolha de (1) ou (2) vai depender da reconstrução, daí a decisão de que o usuário deve configurar a aceleração das partículas. Na situação *default*, a aceleração é configurada com um valor considerado intermediário (0,1), também chamado de valor referência.

Outro importante aspecto na formalização do PSIRT é com relação ao seu funcionamento geral. O organograma da Figura 21 mostra um fluxograma que ilustra tal funcionamento.

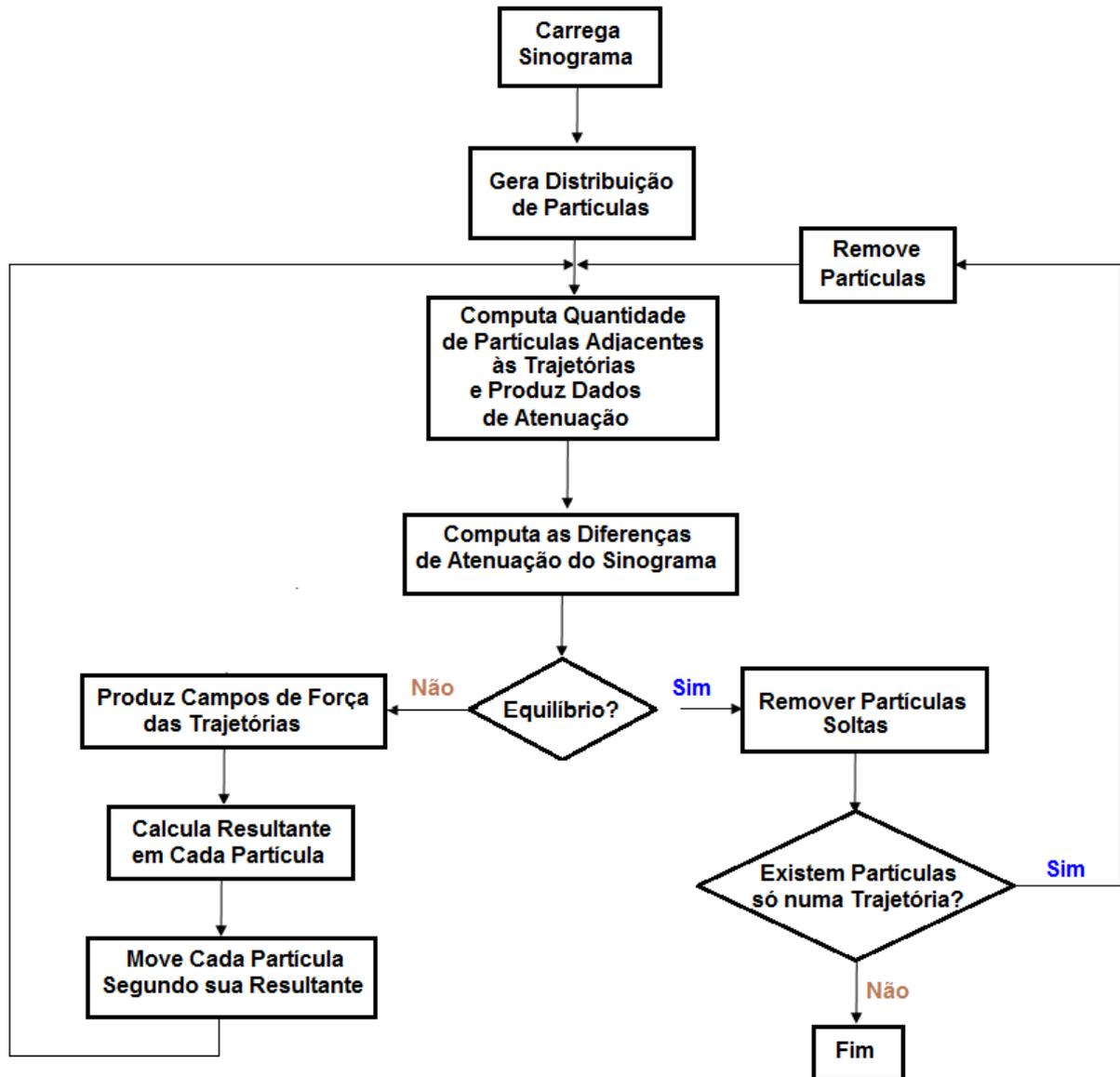


Figura 21 – Fluxograma do funcionamento do PSIRT.

4.11 PSIRT – Telas

As figuras de 22 a 23 mostram as principais telas do PSIRT para o tomógrafo de 1ª geração.

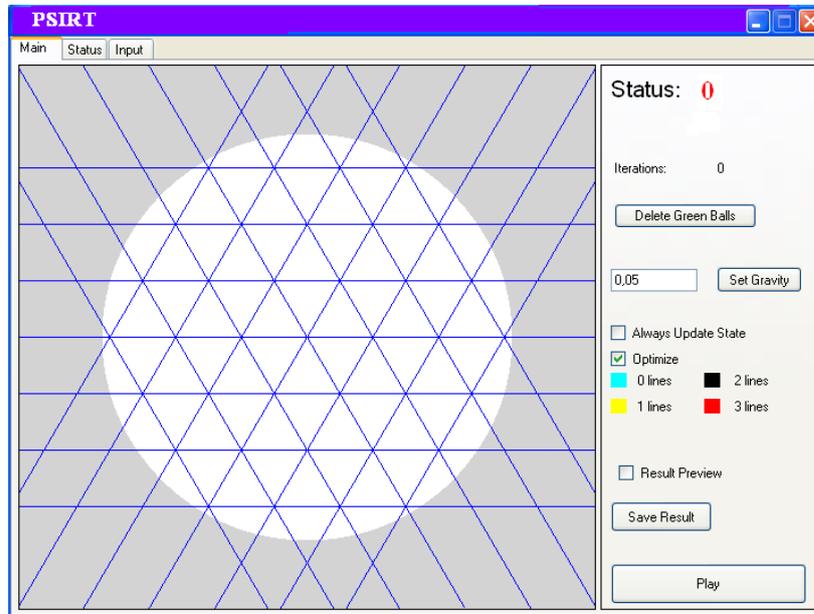


Figura 22 – Tela Principal do PSIRT, aba “Main”.

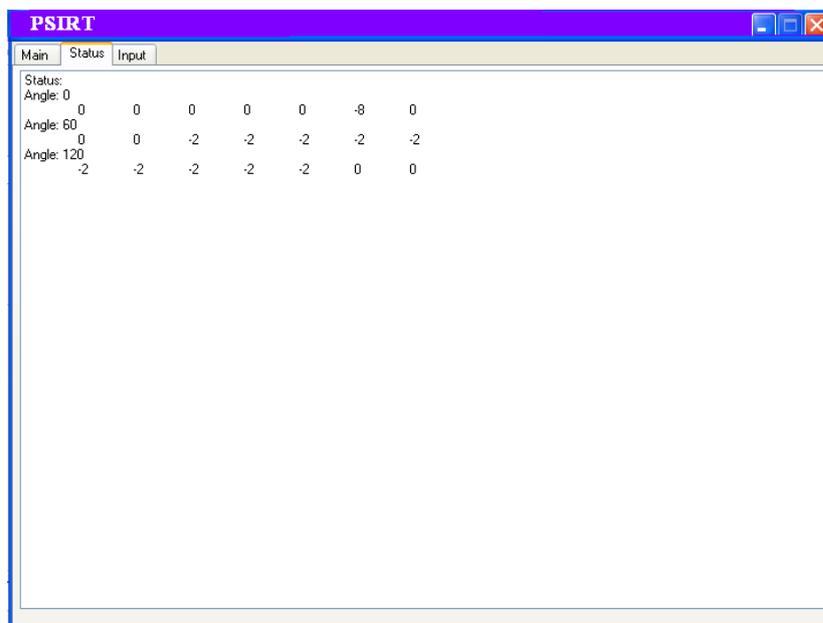


Figura 23 – Tela Principal do PSIRT, aba “Status”.

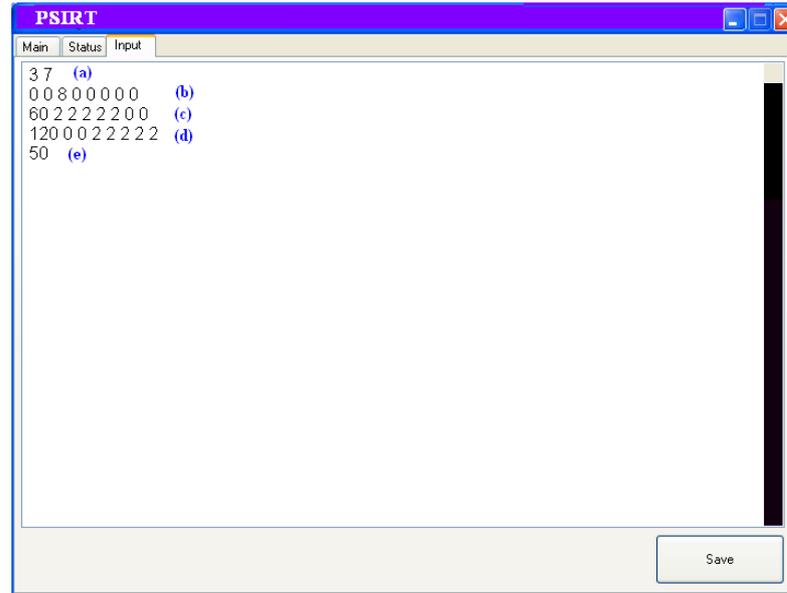


Figura 24 – Tela Principal do PSIRT, aba “Input”.

Na Figura 24 a linha (a) indica a quantidade de projeções (três) e trajetórias (sete). As linhas (b), (c) e (d) indicam como primeiro dado o ângulo de cada uma das 3 projeções que, no exemplo da figura, são 0° , 60° e 120° . Os demais dados dessas linhas indicam quantas partículas cada trajetória irá precisar para reconstruir a imagem. Por exemplo, em (b) apenas a segunda trajetória precisará de partículas (oito). A linha (e) indica a quantidade total de partículas que será usada na reconstrução. Somando as necessidades de (b), (c) e (d) dá um total de 28. Assim, 50 partículas é um valor coerente para essa reconstrução. As figuras 25, 26 e 27 mostram, respectivamente, a distribuição das partículas, as posições finais das partículas após as iterações e o resultado final em hexágonos, para esse exemplo de configuração da Figura 24.

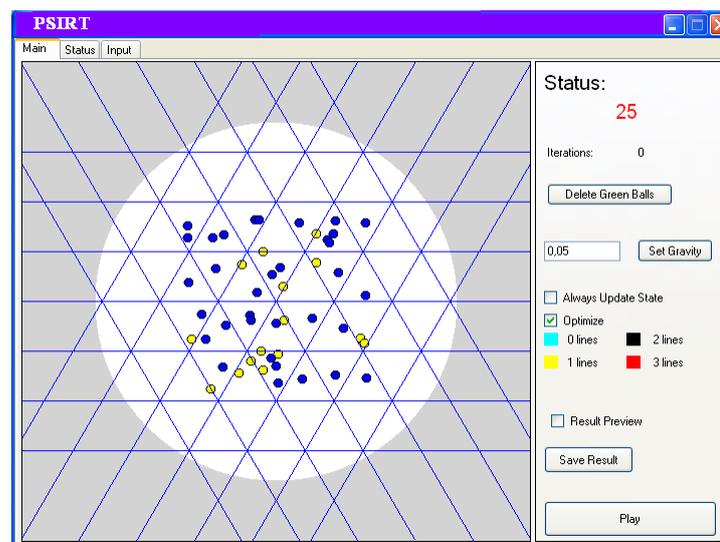


Figura 25 – Distribuição de partículas para o *input* da Figura 24.

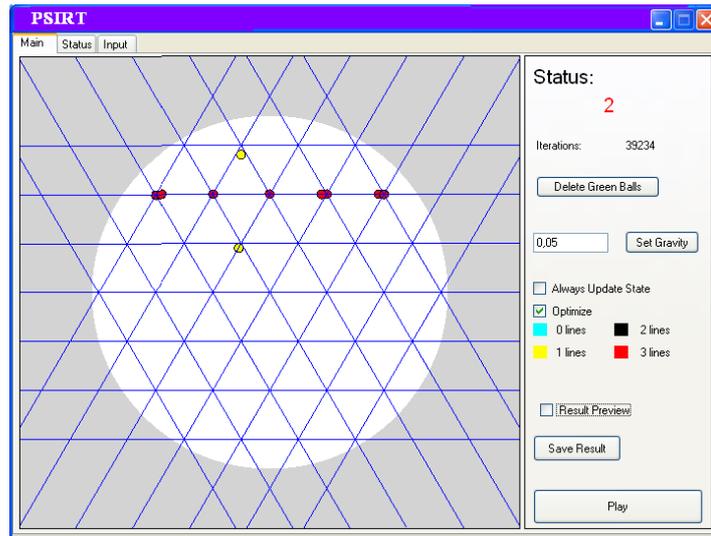


Figura 26 – Posições finais das partículas após a execução da distribuição que consta na Figura 24.

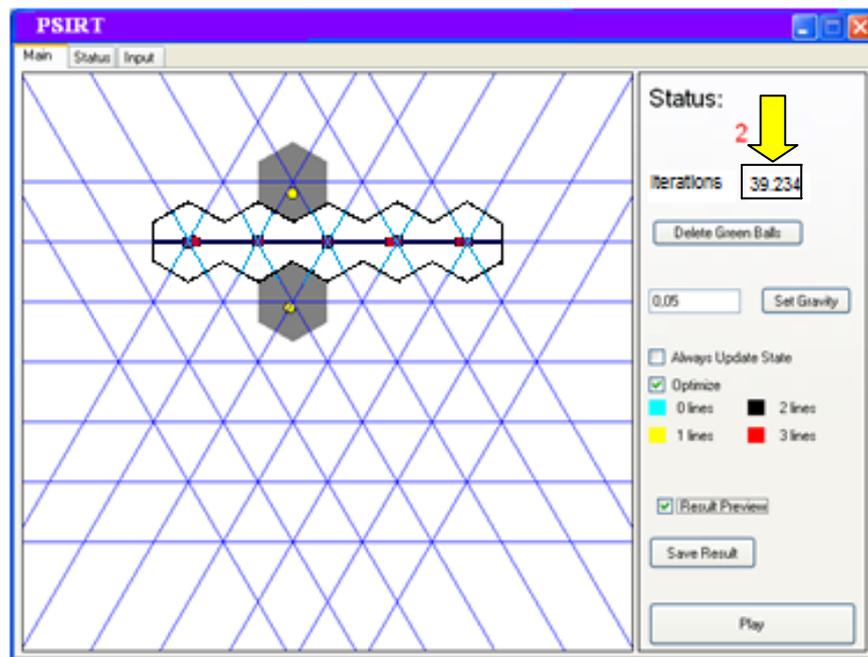


Figura 27 – Discretização em hexágonos do resultado que consta da Figura 26.

A reconstrução em questão é de uma barra horizontal superior. Conforme mostra a Figura 27, foram necessárias 39.234 iterações para o PSIRT atingir o equilíbrio.

4.12 PSIRT – Simulação do Tomógrafo HSGT

A Figura 28 mostra a tela principal do PSIRT para simular o tomógrafo HSGT.

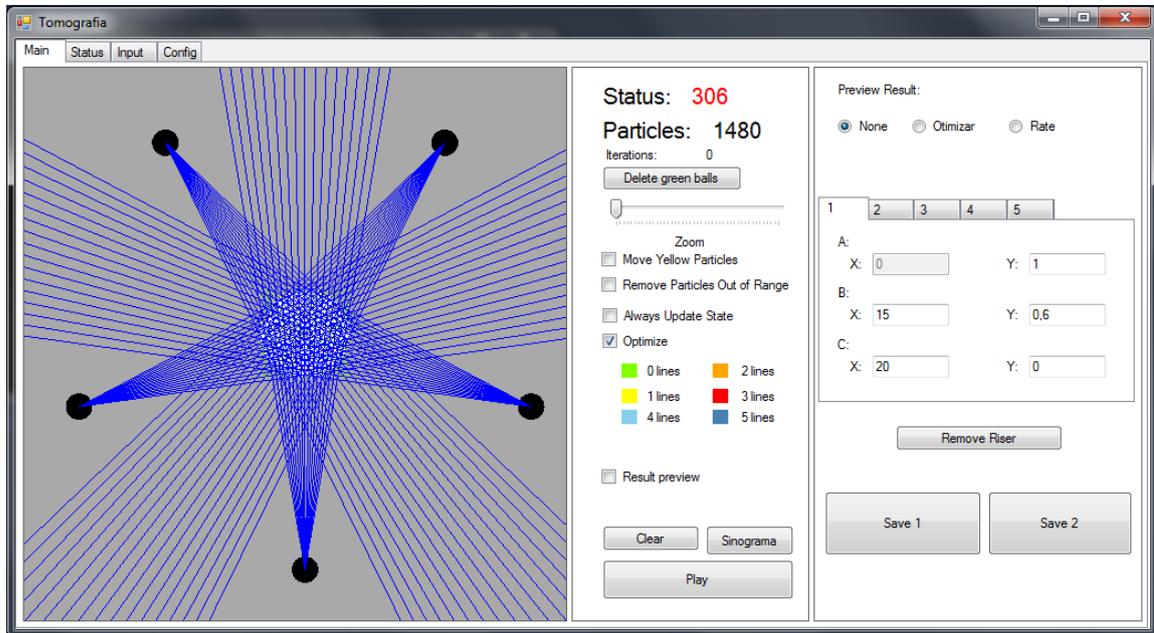


Figura 28 – Tela principal do PSIRT para simular o tomógrafo HSGT.

Na Figura 28 um aspecto importante são os tipos de intersecções, que podem ser 2, 3, 4 e 5, uma vez que o arranjo é constituído de cinco conjuntos de raios com 17 trajetórias em cada. Para fins de visualização do usuário, é possível ativar o modo gráfico do programa, onde uma animação é apresentada com a dinâmica das partículas, com estas sendo ilustradas com diferentes cores, representando as suas posições relativas às intersecções das trajetórias.

As cores é a maneira de diferenciar uma partícula de outra, uma espécie de coloração do “mapa”. Dessa maneira, por exemplo, todos os pontos (partículas) laranja vistos de forma geral na seção do riser (mapa), representam partículas encontradas por duas trajetórias. Isso é uma informação sobre o contexto visual da reconstrução, mas o importante mesmo é que tais partículas serão convertidas em elementos que indicam a presença do material. Segue o esquema de cores para cada tipo de intersecção:

- Partícula isolada (verde).
- Partícula atraída por UMA trajetória (amarela).
- Partícula atraída por DUAS trajetórias (laranja).
- Partícula atraída por TRÊS trajetórias (vermelha).
- Partícula atraída por QUATRO trajetórias (azul celeste).
- Partícula atraída por CINCO trajetórias (azul turquesa).

Além de atribuir uma cor baseada no tipo de interseção, o PSIRT para simular o tomógrafo HSGT define tipos de partículas “A”, “B” e “C”, através de parâmetros no plano cartesiano para cobrir áreas maiores a partir da cor mais ativa (mais forte) da respectiva partícula. As cores são normalizadas de [0, 255] para [0, 1], onde *zero* representa o mais escuro possível, e o *um* o mais branco possível, conforme Figura 29.

Nesse contexto, após o término de execução do algoritmo, as partículas são substituídas por representações de partes do material a ser reconstruído. Essa característica está vinculada ao tipo de partícula após o equilíbrio final. Se a partícula é “vista” por mais trajetórias, então ela deve ser retratada como um material mais *denso*, ou que produz mais atenuação. Os parâmetros “A”, “B” e “C” regulam como este vínculo é feito, ou seja, é uma forma de se incorporar mais informação a priori: se o objeto é delimitado por fronteiras de alta frequência, ou se é mais parecido com uma nuvem (lógica *fuzzy*).

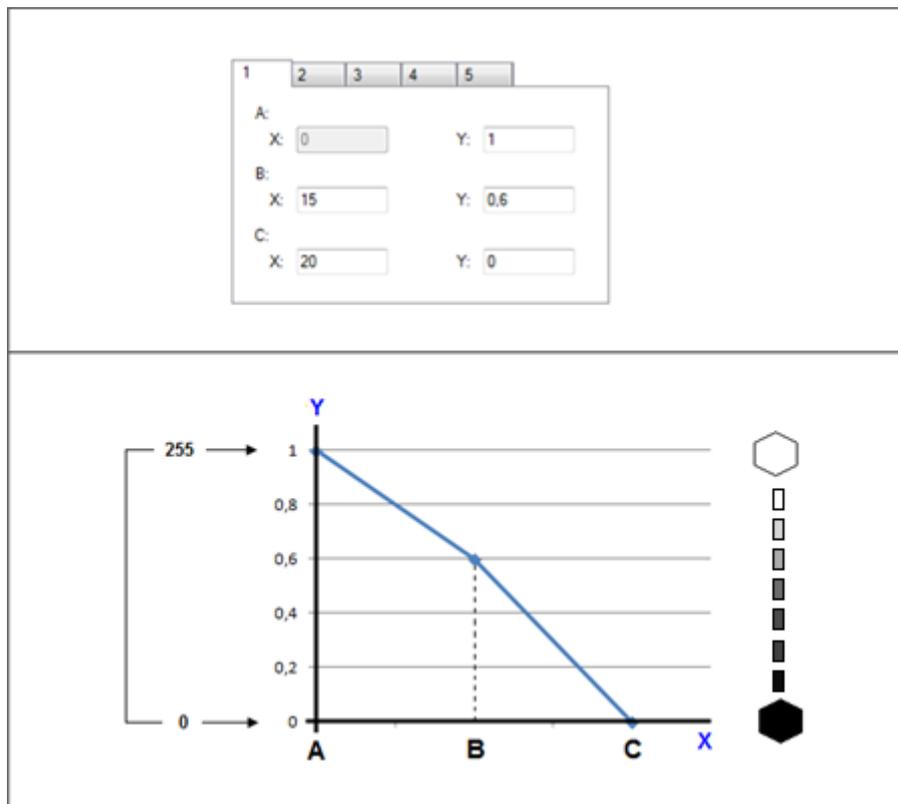


Figura 29 – Tipos de partículas de acordo com a cor.

Na parte de cima da Figura 29 estão os valores cartesianos para cada tipo de partícula. Na parte de baixo, está o gráfico que ilustra o decaimento de intensidade de cinza. As partículas do tipo “A”, por exemplo, são todas brancas, e decaem de cor (começam a escurecer) durante até 15 unidades (pixels) depois. Enquanto isso, as partículas do tipo “B” já começam com uma cor relativa (0,6) e decaem de cor durante 20 unidades depois. As partículas do tipo “C” são todas pretas, e não há nenhum decaimento para elas, pois seus valores são zero (valor mais escuro).

O valor $x=15$, por exemplo, indica que o decaimento de cor ocorrerá num raio de tamanho correspondente a 15 pixels naturais, que já são as partículas convertidas em hexágonos. Para o caso de partícula do tipo “A”, nesse exemplo, a área próxima a seu centro será o mais branco possível e, esse branco vai decaindo até o raio de 15 partículas. Esse decaimento ocorre da maneira ilustrada na Figura 30.

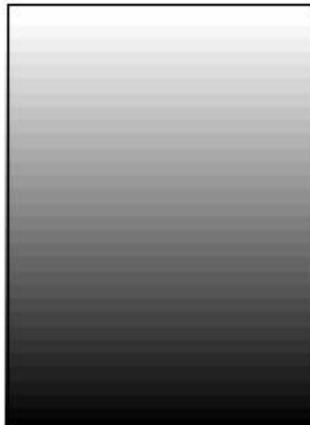


Figura 30 – Decaimento de cor do branco (255) para o preto (0).

O quanto vai ficar escuro é o valor de x atribuído à partícula. Por exemplo, se esse valor for 0,5, o decaimento vai ocorrer até metade, ou seja, até um valor que é “meio branco e meio preto”, uma espécie de cinza intermediário mais central possível. Dessa forma, até antes de 0,5, o decaimento é predominantemente branco, com relativa sutileza no escurecimento e, acima de 0,5, é o contrário.

4.13 PSIRT – Tratamento de Ruído

Um dos grandes problemas da reconstrução tomográfica é a existência de ruído, conforme já mencionado neste trabalho. Um ruído pode ser considerado global, caso ocorra numa região ampla na imagem reconstruída. Quando o problema é numa região específica, o ruído é considerado local. Os ruídos globais – naturalmente pelo fato de cobrir uma região demasiada da imagem reconstruída –, são mais difíceis de serem tratados. De forma contrária, os ruídos locais são situações mais brandas. A Figura 31 apresenta visualmente um exemplo de cada tipo de ruído, na reconstrução de dois quadrados diagonais.

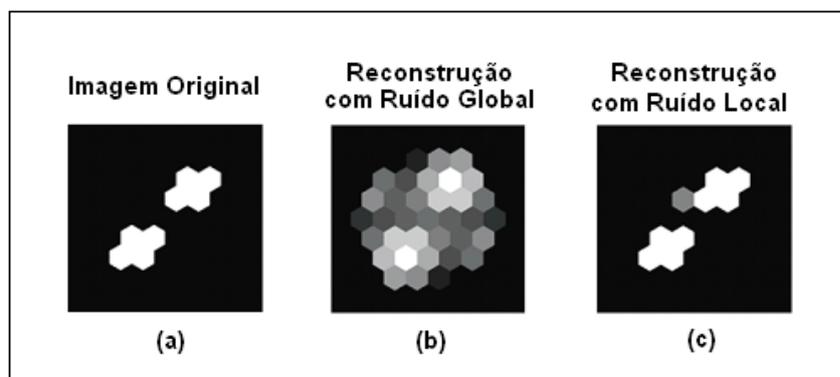


Figura 31 – Exemplo visual de ruído global e local.

O PSIRT tipicamente produz ruídos do tipo local, a exemplo do hexágono cinza isolado na Figura 31 (c), o que confere ao método uma vantagem aos métodos algébricos, que normalmente produzem ruídos globais, conforme aglomerado predominante de hexágonos de diversos tons de cinza (b), que estão inclusive fora da região da imagem que está sendo reconstruída: imagem original.

O aspecto que faz com que o PSIRT produza predominantemente ruído local é a eliminação de partículas que não representam material ao longo das iterações. Na busca do equilíbrio, o método efetua muitas eliminações de partículas do tipo “D”, deixando o mínimo possível. As que sobram, geralmente são partículas encontradas por uma única trajetória, que são partículas do tipo “C”. Estas partículas que resistem as iterações são justamente as propensas a candidatas a ruídos. Geralmente são ruídos. Assim, independente se o sistema ainda esteja buscando o equilíbrio ou tenha finalizado a reconstrução, a trajetória tanto garante a existência de tal partícula, como e, principalmente, fornece sua localização, o que impõe ao problema a condição de ser setorizado: ruído local.

5. Experimentos Simulados

5.1 Experimentos Simulados no Tomógrafo de 1ª Geração

Os experimentos apresentados nesta seção vão considerar os seguintes dados:

- **Tipo de imagem:** binária (preto e branco) e não binária (preto, branco e tons de cinza).
- **Resolução das imagens:** 7x7, 15x15, 21x21 e 31x31 no formato “jpg”.
- **Métodos comparados:** ART, MART, SIRT, SMART (métodos da “família ART”).
- **Fator de relaxação:** ART com 0,1 | MART com 0,01 | SIRT com 0,1 | SMART com 0,001, valores empíricos julgados em Oliveira (2011) como os mais adequados para experimentos usando fantasmas similares aos usados aqui.
- **Aceleração das partículas:** 0,01 (pixel por unidade quadrada de tempo).
- **Quantidade de partículas:** no mínimo zero e no máximo $M \times N \times 8$ partículas por trajetória.
- **Métrica de erro:** RMSE e GC.
- **Iterações do PSIRT:** definida pelo usuário com valores empíricos, até atingir a estabilidade em cada execução.
- **Fantomas:** barra horizontal superior, anel (também chamado de coroa), círculo (também chamado de núcleo) e quadrados diagonais.
- **Como os experimentos simulados foram realizados:**
 - ✓ O fantoma é construído de acordo com a resolução da imagem a ser simulada;
 - ✓ A partir do fantoma o programa constrói o sinograma da imagem;
 - ✓ A partir do sinograma as imagens são geradas. Os algoritmos ART, MART, SIRT e SMART foram incorporados ao programa que contém o PSIRT, para produzir todas as saídas dos métodos de uma única vez;

- ✓ Depois de geradas as saídas (imagens reconstruídas) os cálculos RMSE e GC são realizados;
- **Observação:** nas reconstruções NÃO binárias, um importante aspecto avaliado é o formato das imagens reconstruídas com relação aos tons de cinza dos hexágonos úteis. É definido um intervalo de tolerância, onde a tolerância mínima representa o menor valor de cinza (mais escuro) que pode ser atribuído a um hexágono para que o mesmo represente material, sendo a tolerância máxima justamente o contrário: maior valor de cinza (mais claro). Os valores são entre $[0, 1]$, e não existem, dentro desse intervalo, elementos para se produzir uma faixa específica para o RMSE e GC, por diversas razões, entre elas a forma distinta de se calcular ambas as taxas de erro.

5.1.1 Configuração 3x7

Reconstruções na Configuração 3x7

Quadro 1 – Reconstrução binária em 3x7 da barra, anel, círculo e quadrados diagonais.

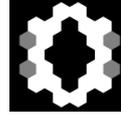
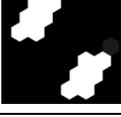
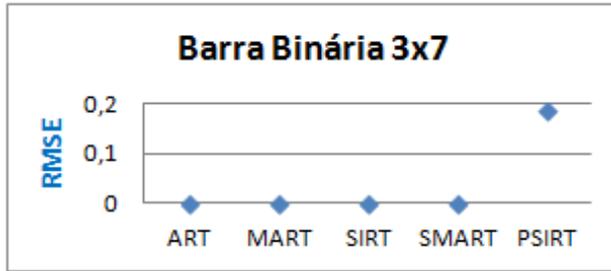
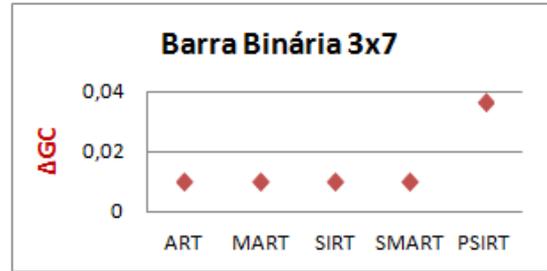
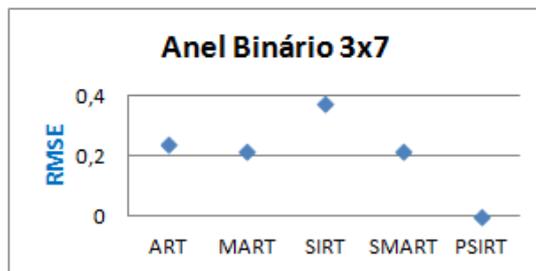
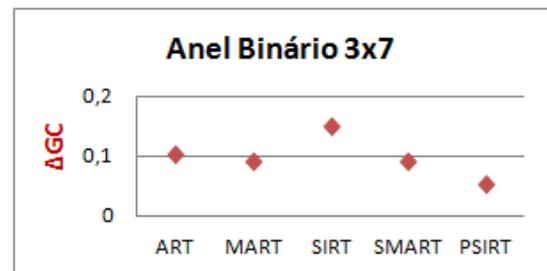
Fantoma	ART	MART	SIRT	SMART	PSIRT
					
					
					
					

Gráfico 1 – RMSE da barra binária 3x7.

Gráfico 2 – ΔGC da barra binária 3x7.

Na reconstrução da barra binária em 3x7, o RMSE e ΔGC estão rigorosamente consistentes com as imagens reconstruídas pelos métodos. Ou seja, os métodos algébricos reconstruíram perfeitamente²² a barra, enquanto que o PSIRT produziu uma barra incompleta. O RMSE (Gráfico 1) e ΔGC (Gráfico 2) refletem igualmente isso.

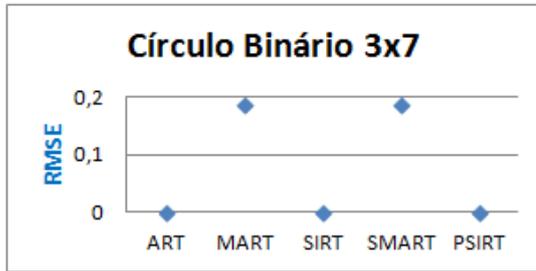
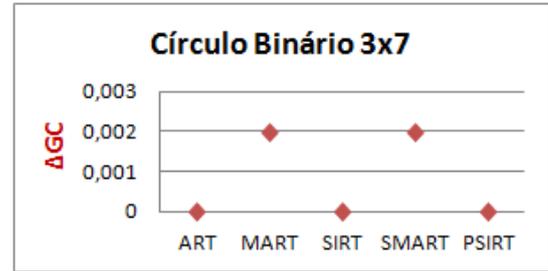
Gráfico 3 – RMSE do anel binário 3x7.

Gráfico 4 – ΔGC do anel binário 3x7.

Na reconstrução do anel binário em 3x7, o SIRT apesar de ter gerado hexágonos na melhor situação (todos brancos), produziu a espessura do anel um pouco maior do fantoma original. Os demais métodos foram regulares nessa reconstrução, com o PSIRT ligeiramente melhor, pelo fato dos outros métodos terem gerado alguns hexágonos cinza candidatos a ruídos (fora da área da imagem). O PSIRT também gerou hexágonos cinza (quatro), mas dentro do escopo da imagem (na borda). Os RMSE e ΔGC são consistentes com a inspeção visual das reconstruções dos métodos (gráficos 3 e 4).

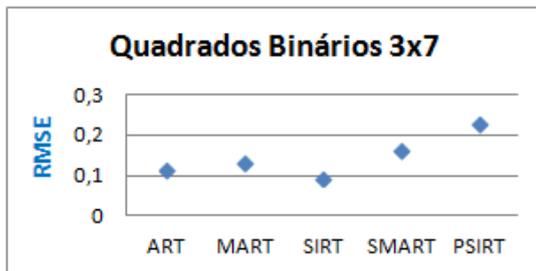
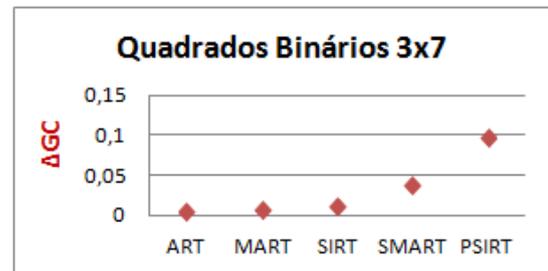
²² A reconstrução fiel gera um RMSE igual a zero, situação normal em experimentos simulados.

Gráfico 5 – RMSE do Círculo binário 3x7.

Gráfico 6 – ΔGC do Círculo binário 3x7.

Na reconstrução do círculo binário em 3x7, o ART, SIRT e PSIRT produziram uma reconstrução fiel ao fantoma original, fato consistente com os valores do RMSE (Gráfico 5) e refletidos igualmente no ΔGC (Gráfico 6). O MART e SMART geraram alguns hexágonos cinza na borda do círculo, mas sutilmente fora da região da imagem, o que confere um menor desempenho para esses dois métodos nesse fantoma, fato também registrado nos gráficos.

Gráfico 7 – RMSE dos Quadrados binários 3x7.

Gráfico 8 – ΔGC dos Quadrados binários 3x7.

Na reconstrução dos quadrados diagonais binários em 3x7, os métodos foram regulares, mas o PSIRT gerou dois hexágonos isolados que são ruídos (ruído local). Um aspecto diferente é que o SIRT é desfavorecido pelo ΔGC (Gráfico 8) em relação ao ART e MART (Gráfico 7). O fato é que os dois hexágonos cinza no SIRT são ruídos, enquanto que dos três hexágonos cinza no ART e MART pelo menos um deles não é ruído e, os dois que supostamente são, estão mais próximo um do outro do que no SIRT. Como o ΔGC considera, também, a proximidade entre hexágonos úteis (cinza e branco), tal desfavorecimento ao SIRT é coerente.

Quadro 2 – Reconstrução NÃO binária em 3x7 da barra, anel, círculo e quadrados.

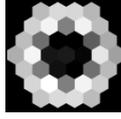
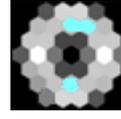
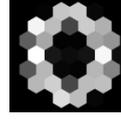
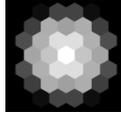
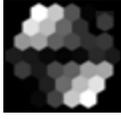
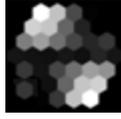
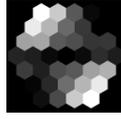
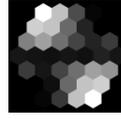
Fantoma	ART	MART	SIRT	SMART	PSIRT
					
					
					
					

Gráfico 9 – RMSE da barra NÃO binária 3x7.

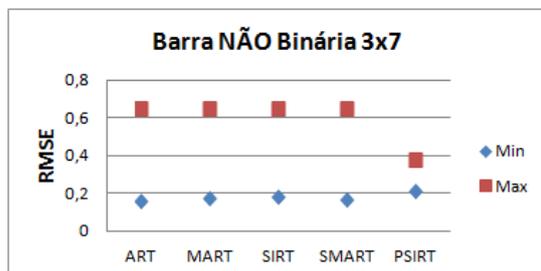
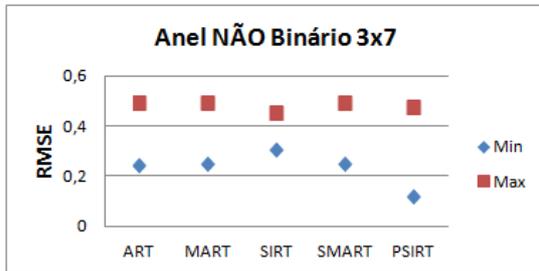
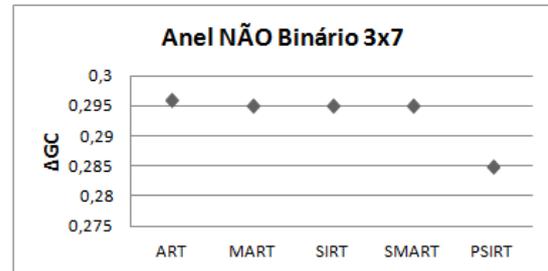


Gráfico 10 – ΔGC da barra NÃO binária 3x7.



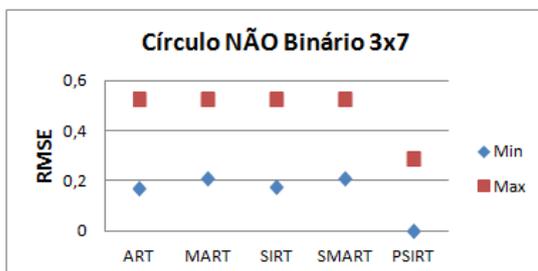
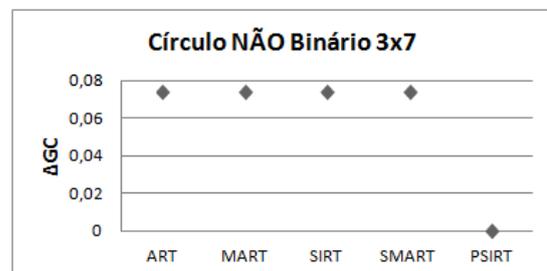
Na reconstrução da barra não binária em 3x7, apesar de o PSIRT ter produzido uma imagem incompleta, em comparação ao fantoma original, o método gerou apenas dois ruídos locais, enquanto os demais métodos geraram muito ruído global, não apenas nas bordas da barra, como também bem além das fronteiras da imagem (parte de baixo da imagem). Os métodos algébricos foram regulares em relação à geração dos tons de cinza, conforme expressam os limites do RMSE (Gráfico 9). Entretanto, o PSIRT só produziu hexágonos na região da imagem (com exceção daqueles dois), o que confere ao método melhor desempenho no ΔGC (Gráfico 10).

Gráfico 11 – RMSE do Anel NÃO binário 3x7.

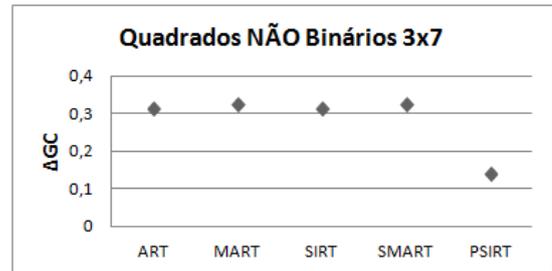
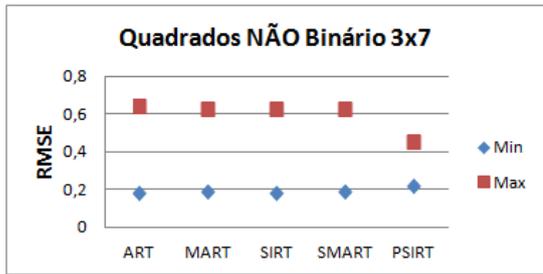
Gráfico 12 – ΔGC do Anel NÃO binário 3x7.

Na reconstrução do anel não binário em 3x7, a situação é muito parecida a da barra, em relação ao aspecto visual e geração de ruídos. Entretanto, o PSIRT foi um pouco mais sensível na geração de tons mais escuros de cinza na região da imagem e, na tolerância máxima, todos os métodos se mantiveram regulares, fatos consistentes com o RMSE (Gráfico 11). No ΔGC , o PSIRT apresentou uma sutil vantagem em relação aos demais métodos, pelo fato de ter produzido hexágonos úteis apenas na região da imagem. Por fim, o ART, MART e SMART, visualmente, produziram a mesma imagem, com ínfima vantagem para o SIRT.

Gráfico 13 – RMSE círculo NÃO binário 3x7.

Gráfico 14 – ΔGC do círculo NÃO binário 3x7.

Na reconstrução do círculo não binário em 3x7, os métodos algébricos foram bastante regulares, mas novamente apresentando muito ruído. Nota-se que o ART e SIRT produziram hexágonos um pouco mais escuros que o MART e SMART, o que é consistente com o RMSE (Gráfico 13). O ART foi um pouco melhor que os demais métodos algébricos, por ter produzido dois hexágonos ruidosos a menos. No geral, o PSIRT obteve melhor desempenho, novamente por ter gerado todos hexágonos úteis na região da imagem, e nenhum ruído local, fato consistente com ΔGC (Gráfico 14).

Gráfico 15 – RMSE dos quadrados NÃO binários 3x7. Gráfico 16 – ΔGC dos quadrados NÃO binários 3x7.

Na reconstrução dos quadrados não binários em 3x7, os métodos algébricos geraram muito ruído global, mas novamente preservaram a regularidade entre si na reconstrução. Também houve regularidade na geração de tons de cinza, fato confirmado com o RSME (Gráfico 15). O PSIRT gerou dois ruídos locais e ainda não reconstruiu os quadrados de forma completa. Apesar disso, o método obteve melhor desempenho para esse fantoma, justamente pelo excesso de ruído global dos métodos algébricos, sendo mais coerente (ΔGC) com a reconstrução (Gráfico 16).

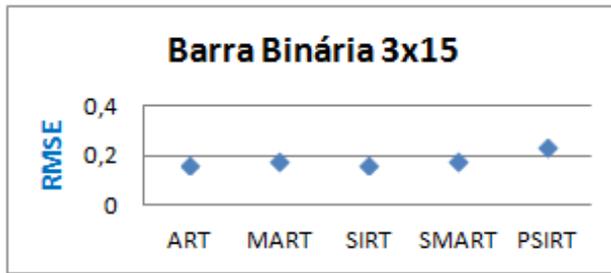
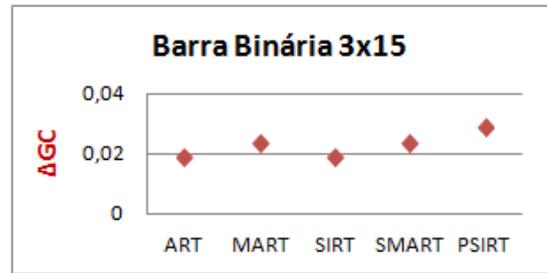
5.1.2 Configuração 3x15

Reconstruções na Configuração 3x15

Quadro 3 – Reconstrução binária em 3x15 da barra, anel, círculo e quadrados diagonais.

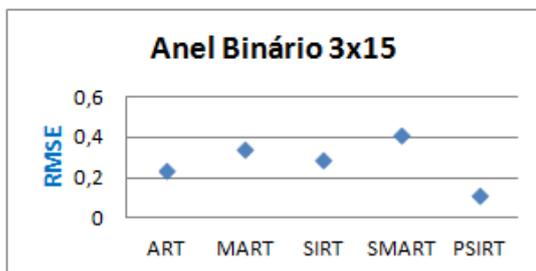
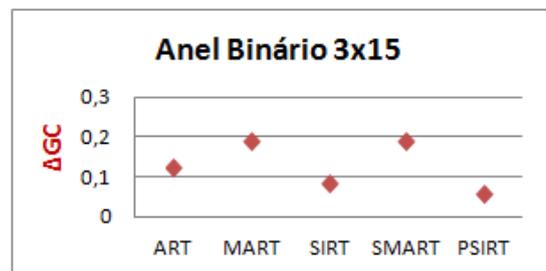
Fantoma	ART	MART	SIRT	SMART	PSIRT

Gráfico 17 – RMSE barra binária 3x15.

Gráfico 18 – Δ GC da barra binária 3x15.

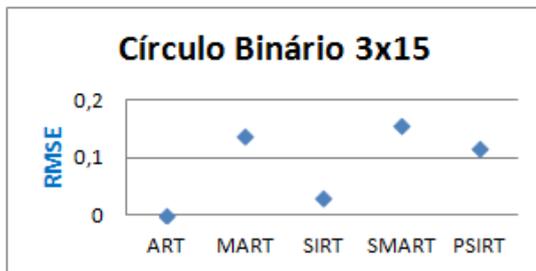
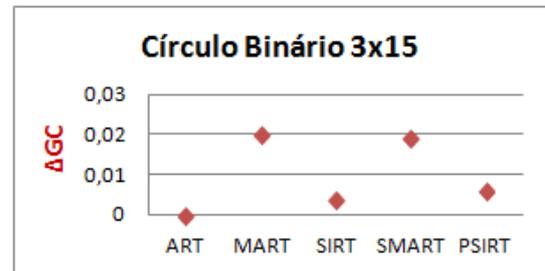
Na reconstrução da barra binária em 3x15, O ART e o SIRT praticamente reproduziram fielmente a barra original, enquanto o MART e SMART geraram alguns hexágonos ruidosos na borda superior da imagem. O PSIRT foi um pouco irregular, e também produziu hexágonos ruidosos. O Δ GC (Gráfico 18) reflete bem o que mostram as imagens e se faz coerente ao RMSE (Gráfico 17).

Gráfico 19 – RMSE do anel binário 3x15.

Gráfico 20 – Δ GC do anel binário 3x15.

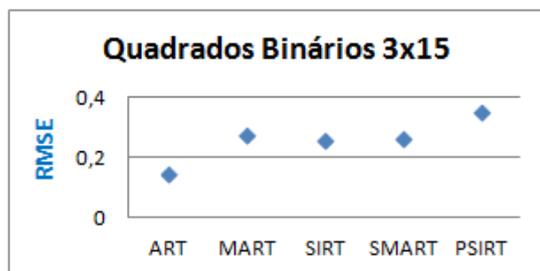
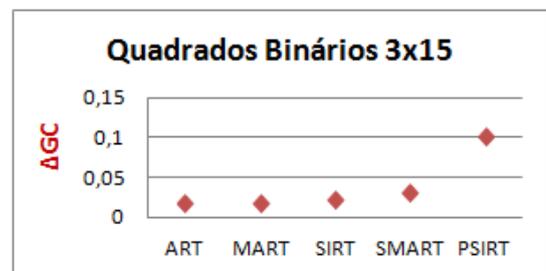
Na reconstrução do anel binário em 3x15, o MART e SMART praticamente não reconstruíram o núcleo do anel, gerando todos os hexágonos nessa região na cor cinza, fato refletido tanto no RMSE (Gráfico 19) quanto no Δ GC (Gráfico 20). Já o ART e SIRT reproduziram o núcleo de forma razoável. O PSIRT reconstruiu bem o núcleo e ainda preservou a espessura do anel bem mais que os outros métodos, o que conferiu ao método melhores valores nos gráficos.

Gráfico 21 – RMSE do círculo binário 3x15.

Gráfico 22 – ΔGC do círculo binário 3x15.

Na reconstrução do círculo binário em 3x15, o ART e o SIRT reproduziram melhor a imagem, com destaque para o ART, que praticamente reconstruiu fielmente o fantoma original, daí os valores zero conferidos ao ART tanto no RMSE (Gráfico 21) quanto no ΔGC (Gráfico 22). O SIRT também reproduziu fielmente a imagem, o que lhe conferiu valores próximos de zero para ambas as métricas. A sutil distinção foi que o SIRT gerou alguns hexágonos cinza, que são hexágonos com peso menor que os de cor branca. O MART e SMART produziram picos no círculo, dando uma ideia de estrela, e o PSIRT foi muito bom nessa reconstrução, fatos também refletidos nos gráficos.

Gráfico 23 – RMSE dos quadrados binários 3x15.

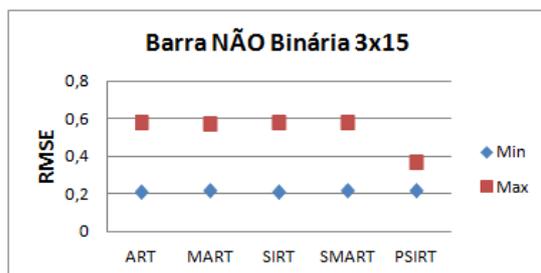
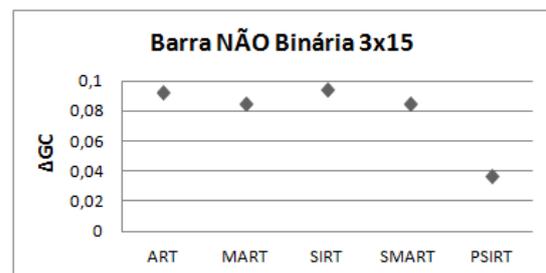
Gráfico 24 – ΔGC dos quadrados binários 3x15.

Na reconstrução dos quadrados binários em 3x15, os métodos algébricos geraram muito ruído, tanto no espaço que separa os dois quadrados, quanto além das fronteiras da imagem. Mesmo assim, os métodos algébricos tiveram melhor desempenho que o PSIRT que, apesar de não ter gerado hexágonos ruidosos, não preservou o tamanho original dos quadrados, fato que é igualmente refletido pelo RMSE (Gráfico 23) e ΔGC (Gráfico 24). Ainda sobre o ΔGC , a oscilação sutil desse valor nos métodos algébricos é devido à pequena diferença na quantidade de hexágonos ruidosos que um método algébrico teve do outro.

Quadro 4 – Reconstrução NÃO binária em 3x15 da barra, anel, círculo e quadrados.

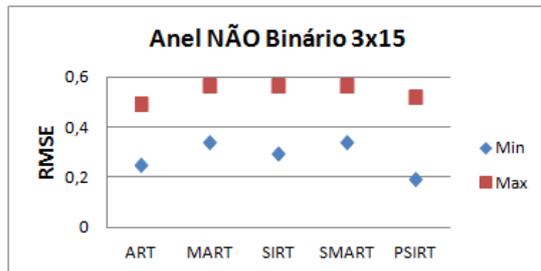
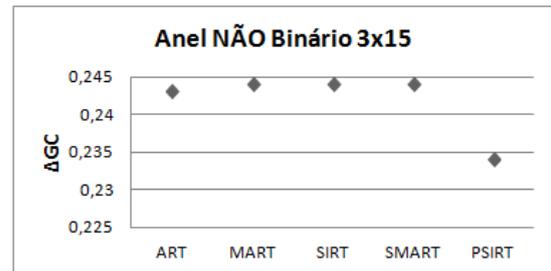
Fantoma	ART	MART	SIRT	SMART	PSIRT

Gráfico 25 – RMSE barra NÃO binária 3x15.

Gráfico 26 – ΔGC da barra NÃO binária 3x15.

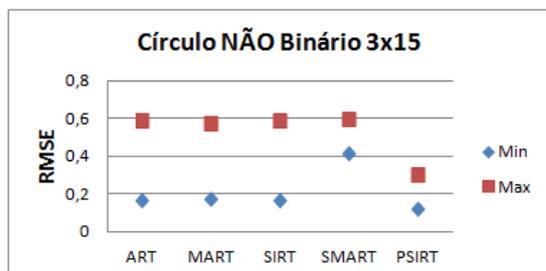
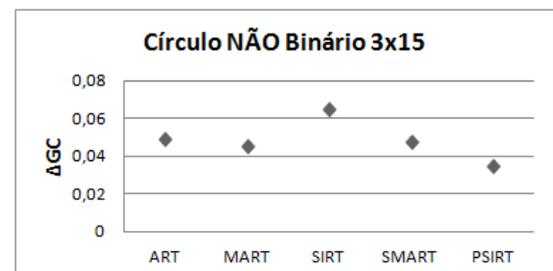
O melhor desempenho conferido ao PSIRT na reconstrução da barra não binária em 3x15, tanto no RMSE (Gráfico 25) quanto no ΔGC (Gráfico 26), deu-se pela quantidade demasiada e abrangente de ruídos globais produzidos pelos métodos algébricos. Pela inspeção visual dá para observar que os métodos algébricos preservaram o formato da imagem original, mas abaixo da fronteira da imagem geraram muito ruído global, mais ainda pelo ART. O PSIRT não preservou perfeitamente o formato da imagem, mas gerou pouquíssimo ruído, reforçado pelo fato de terem sido hexágonos ruidosos setorizados (ruído local). Novamente os métodos algébricos foram regulares na geração de tons de cinza, e o PSIRT aparentemente gerou hexágonos um pouco mais claros que os demais métodos, que é o que reflete a tolerância máxima do Gráfico 25.

Gráfico 27 – RMSE do anel NÃO binário 3x15.

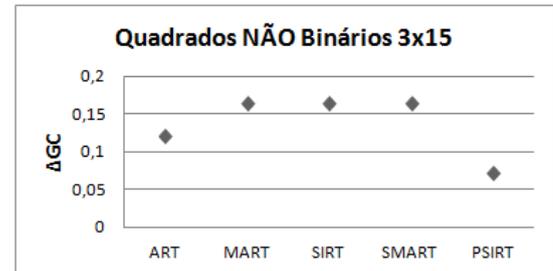
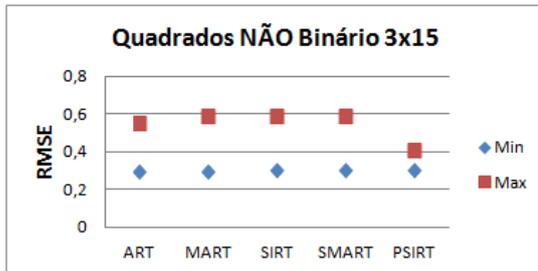
Gráfico 28 – ΔGC do anel NÃO binário 3x15.

Na reconstrução do anel não binário em 3x15, a situação foi estritamente semelhante a da barra não binária nessa mesma configuração, com excesso de ruído global por parte dos métodos algébricos, e pouco ruído local por parte do PSIRT (mais no núcleo do anel). A diferença é que o PSIRT foi mais fiel ao formato original do anel do que na barra. Da mesma forma, o RMSE (Gráfico 27) e ΔGC (Gráfico 28) refletiram os resultados visuais da reconstrução, com regularidade novamente na geração de tons de cinza.

Gráfico 29 – RMSE do círculo NÃO binário 3x15.

Gráfico 30 – ΔGC do círculo NÃO binário 3x15.

Na reconstrução do círculo não binário em 3x15, a situação é análoga ao da barra e anel, com distinção desfavorável ao SIRT, que produziu bem mais hexágonos ruidosos do que os outros métodos algébricos, fato melhor representado pelo ΔGC (Gráfico 30) do que pelo RMSE (Gráfico 29), já que são hexágonos ruidosos muito próximos ao contorno do anel e em todas as direções.

Gráfico 31 – RMSE dos quadrados NÃO binários 3x15. Gráfico 32 – ΔGC dos quadrados NÃO binários 3x15.

Na reconstrução dos quadrados não binários em 3x15, todos os métodos geraram muito ruído, novamente com os métodos algébricos gerando bem mais e de forma semelhante entre si. Entretanto, o PSIRT reproduziu quadrados bem menores que os originais, e teve um desempenho superior aos métodos algébricos. A questão é que os métodos algébricos também reproduziram quadrados bem menores: dá para observar isso visualmente, percebendo pequenas manchas brancas na parte superior e inferior da imagem, e as manchas que se seguem são cinza (hexágonos candidatos a ruidosos – a minoria; e hexágonos ruidosos – a maioria). Daí o favorecimento ao PSIRT no RMSE (Gráfico 31) e, principalmente, no ΔGC (Gráfico 32). A geração de tons de cinza foi regular, e o PSIRT produziu hexágonos mais claros na região da imagem, distinção expressa na tolerância máxima do Gráfico 31.

5.1.3 Configuração 3x21

Reconstruções na Configuração 3x21

Quadro 5 – Reconstrução binária em 3x21 da barra, anel, círculo e quadrados diagonais.

Fantoma	ART	MART	SIRT	SMART	PSIRT

Gráfico 33 – RMSE da barra binária 3x21.

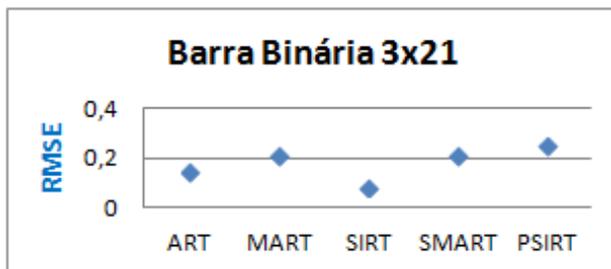
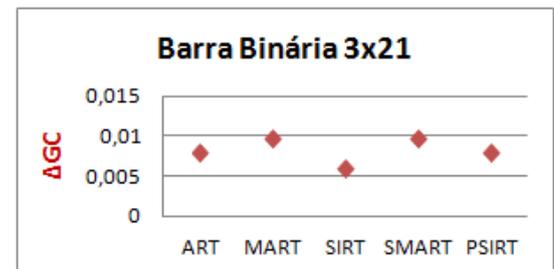
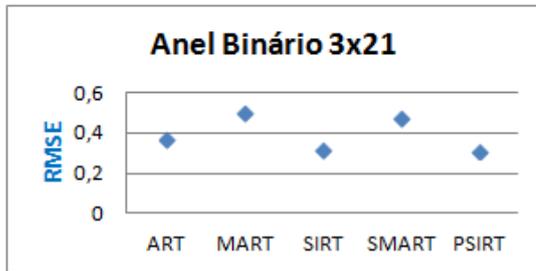
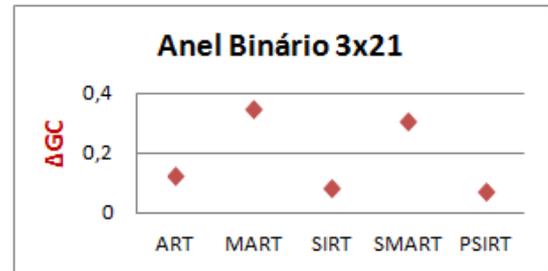


Gráfico 34 – ΔGC da barra binária 3x21.



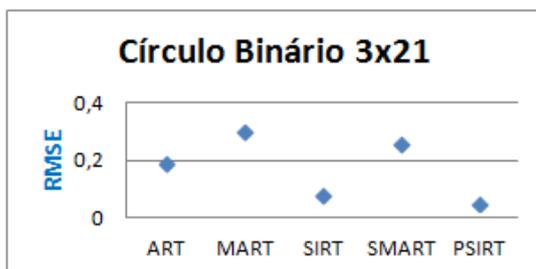
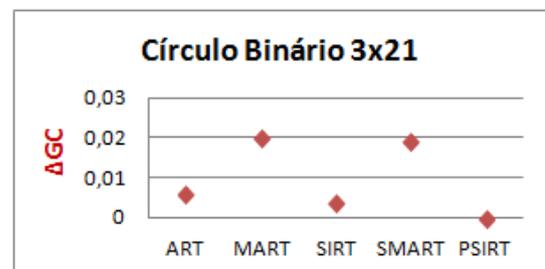
Na reconstrução da barra binária em 3x21, os métodos de uma maneira geral foram regulares, com pequeno destaque para o SIRT, que reproduziu a imagem fielmente sem nenhum ruído. Os demais métodos também reproduziram uma imagem fiel ao fantoma original, mas com alguns hexágonos ruidosos. O PSIRT produziu pouco ruído, mas a barra ficou incompleta. Tal regularidade entre os métodos algébricos é refletida no RMSE (Gráfico 33) e ΔGC (Gráfico 34). No caso do ΔGC , o PSIRT fica relativamente igual ao ART, por este ter gerado maior quantidade de ruído.

Gráfico 35 – RMSE do anel binário 3x21.

Gráfico 36 – ΔGC do anel binário 3x21.

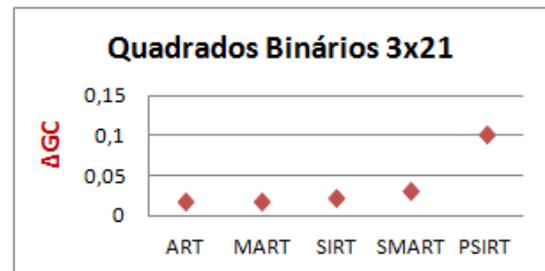
Na reconstrução do anel binário em 3x21, o MART e SMART não reconstruíram o núcleo do anel de forma satisfatória. Os demais métodos reconstruíram bem, não apenas o núcleo como também o formato do anel, com sutil destaque para o PSIRT, que praticamente reproduziu fielmente o fantoma original. O ART, SIRT e PSIRT foram muito regulares. Todas essas situações são consistentes com o RMSE (Gráfico 35) e ΔGC (Gráfico 36).

Gráfico 37 – RMSE do círculo binário 3x21.

Gráfico 38 – ΔGC do círculo binário 3x21.

Na reconstrução do círculo binário em 3x21, o ART e SIRT tiveram um bom desempenho, o MART e SMART produziram muito ruído, e o PSIRT reproduziu a imagem com extrema precisão. Situações refletidas tanto no RMSE (Gráfico 37) quanto no ΔGC (Gráfico 38).

Gráfico 39 – RMSE dos quadrados binários 3x21.

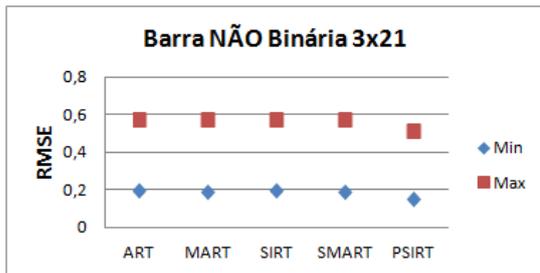
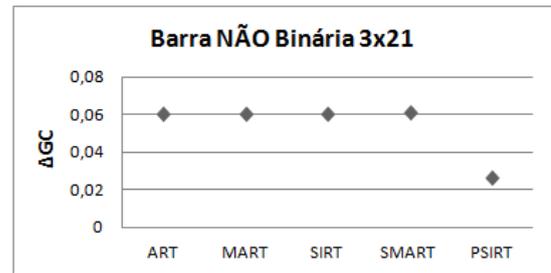
Gráfico 40 – ΔGC dos quadrados binários 3x21.

Na reconstrução dos quadrados binários em 3x21, todos os métodos produziram ruídos, com o PSIRT produzindo bem menos que os métodos algébricos. Essa distinção na quantidade de hexágonos ruidosos conferiu ao PSIRT um valor um pouco melhor no RSME (Gráfico 39), embora tal situação não se refletiu no ΔGC (Gráfico 40). A questão é que o PSIRT gerou uma imagem irregular e incompleta, sendo justamente os espaços sem hexágonos úteis que contribuíram para a queda no desempenho do PSIRT com relação à coerência geométrica: sem hexágonos cinza ou branco, o peso diminuiu no cálculo do ΔGC .

Quadro 6 – Reconstrução NÃO binária em 3x21 da barra, anel, círculo e quadrados.

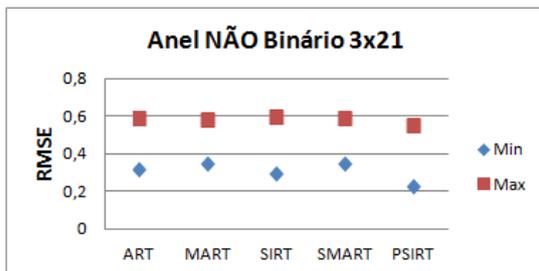
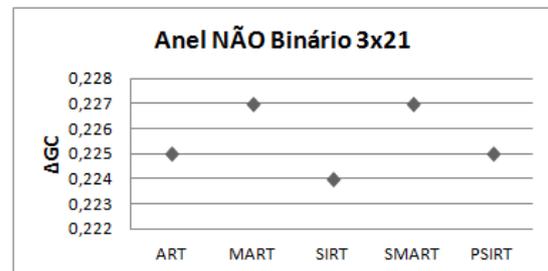
Fantoma	ART	MART	SIRT	SMART	PSIRT

Gráfico 41 – RMSE da barra NÃO binária 3x21.

Gráfico 42 – ΔGC da barra NÃO binária 3x21.

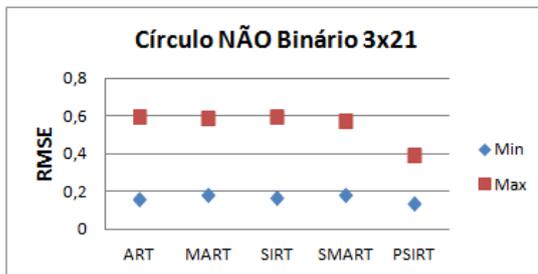
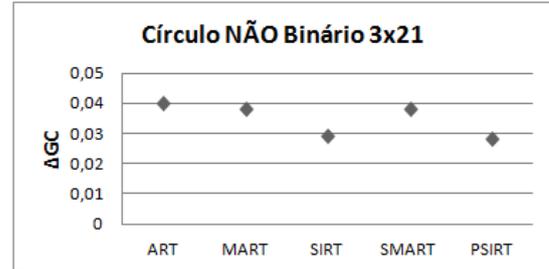
Na barra não binária em 3x21, o PSIRT produziu a espessura da imagem original um pouco menor, mas não gerou nenhum ruído. Os métodos algébricos novamente produziram muito ruído global. Por não ter produzido ruído, os pesos dos hexágonos úteis conferiram ao PSIRT um melhor desempenho no ΔGC (Gráfico 42). Os métodos algébricos reproduziram a imagem num formato um pouco mais fiel ao fantoma original do que o PSIRT, e esse fato fez com que o RMSE (Gráfico 41) ficasse relativamente regular entre todos os métodos: de fato, a distinção visual em relação à geração de tons de cinza é muito difícil de ser percebida para esse caso.

Gráfico 43 – RMSE do anel NÃO binário 3x21.

Gráfico 44 – ΔGC do anel NÃO binário 3x21.

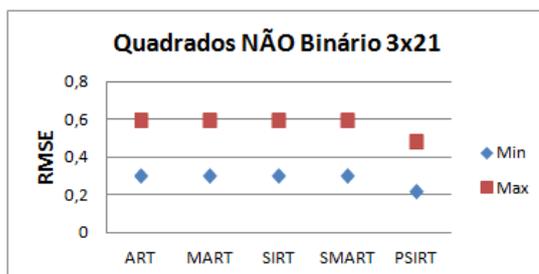
Na reconstrução do anel não binário em 3x21, novamente o MART e SMART não tiveram um bom desempenho, pela dificuldade desses métodos em reconstruir o núcleo do anel. Essa situação se repetiu um pouco no ART. O SIRT produziu no geral hexágonos mais escuros do que os outros métodos algébricos, o PSIRT também, fatos consistentes com a tolerância mínima do RMSE (Gráfico 43). O PSIRT produziu muita falha na reconstrução, apesar de ter preservado a espessura do anel um pouco melhor que os outros métodos. Mesmo assim, o ΔGC (Gráfico 44) foi mais favorável ao SIRT, por ter produzido uma imagem mais enxuta e com contornos geométricos bem mais definidos.

Gráfico 45 – RMSE do círculo NÃO binário 3x21.

Gráfico 46 – ΔGC do círculo NÃO binário 3x21.

Na reconstrução do círculo não binário em 3x21, a situação demonstra uma tendência que tem se confirmado nas outras configurações já discutidas: muito ruído global nas reconstruções dos métodos algébricos – mesmo preservando características importantes da imagem original –, e pouco ou quase nenhum ruído no PSIRT – mesmo que o método não preserve fielmente algumas características do formato original da imagem. A análise visual da reconstrução desse círculo reflete bem essa tendência, que novamente é consistente com o RMSE (Gráfico 45) e ΔGC (Gráfico 46). No limite mínimo, o PSIRT possui uma mancha vertical mais escura dentro da região da imagem, o que lhe confere uma pequena distinção na geração de tons de cinza.

Gráfico 47 – RMSE dos quadrados NÃO binários 3x21.

Gráfico 48 – ΔGC dos quadrados NÃO binários 3x21.

Na reconstrução dos quadrados não binários em 3x21, os métodos algébricos não apenas geraram muito ruído global, como também reproduziram a imagem um pouco menor do fantoma original, se igualando ao PSIRT neste último aspecto. Assim, as métricas novamente favoreceram ao PSIRT, conforme expressam o RMSE (Gráfico 47) e ΔGC (Gráfico 48).

5.1.4 Configuração 3x31

Reconstruções na Configuração 3x31

Quadro 7 – Reconstrução binária em 3x31 da barra, anel, círculo e quadrados diagonais.

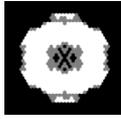
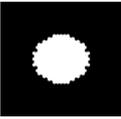
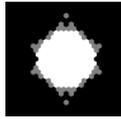
Fantoma	ART	MART	SIRT	SMART	PSIRT
					
					
					
					

Gráfico 49 – RMSE da barra binária em 3x31.

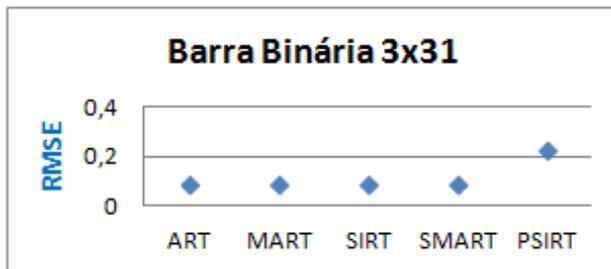
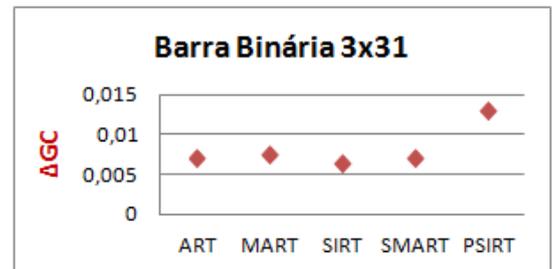
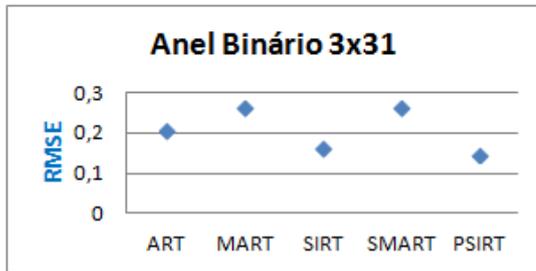
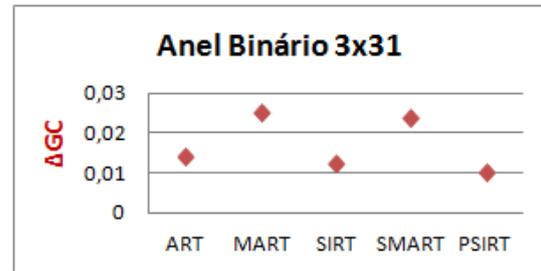


Gráfico 50 – ΔGC da barra binária em 3x31.



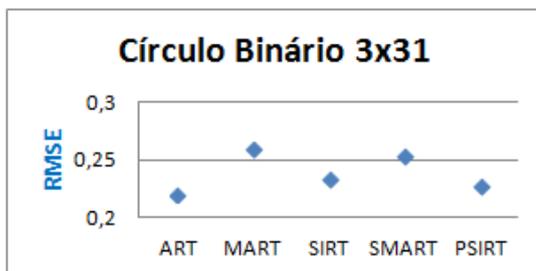
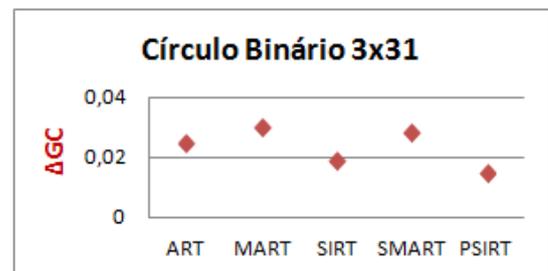
Na reconstrução da barra binária em 3x31, os métodos algébricos tiveram igualmente um ótimo desempenho, enquanto que o PSIRT gerou uma imagem muito irregular ao fantoma original, fatos consistentes com o RMSE (Gráfico 49). A reconstrução de um pequeno pedaço a mais nos cantos inferiores direito e esquerdo da barra, conferiram aos métodos algébricos uma pequena oscilação no ΔGC (Gráfico 50). Da mesma forma, a irregularidade na reconstrução do PSIRT, já citada, conferiu ao método um fraco desempenho nessa métrica.

Gráfico 51 – RMSE do anel binário em 3x31.

Gráfico 52 – ΔGC do anel binário em 3x31.

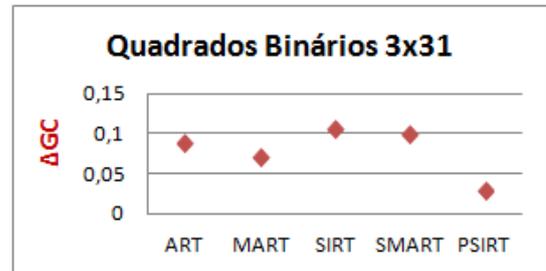
Na reconstrução do anel binário em 3x31, dá para observar visualmente que o PSIRT reconstruiu quase que fielmente a imagem original, da mesma forma o ART e SIRT, fatos consistentes com o RMSE (Gráfico 51). Novamente o MART e SMART tiveram dificuldade na reconstrução do núcleo do anel. Por ter reconstruído o anel com maior predominância de hexágonos branco (maior peso), o PSIRT foi ligeiramente mais favorecido no ΔGC do que o ART e SIRT, fato consistente com o Gráfico 52.

Gráfico 53 – RMSE do círculo binário em 3x31.

Gráfico 54 – ΔGC círculo binário em 3x31.

Na reconstrução do círculo binário em 3x31, o PSIRT ficou muito igual ao ART, com pequena diferença desfavorável, mesmo o ART tendo produzido ruídos ao redor do círculo e o PSIRT não ter gerado ruído nenhum. Novamente a questão é que o PSIRT produziu a imagem um pouco irregular. Entretanto, o PSIRT teve melhor desempenho do que, por exemplo, o SIRT, que manteve o formato do círculo, mas produziu muito ruído. Por fim, sobre essa reconstrução, os demais métodos novamente produziram picos ao redor do anel. Ambas as situações são consistentes com o RMSE (Gráfico 53) e ΔGC (Gráfico 54).

Gráfico 55 – RMSE dos quadrados binários em 3x31.

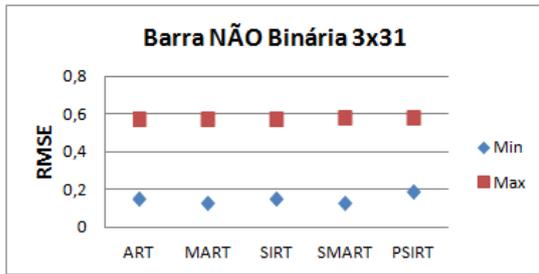
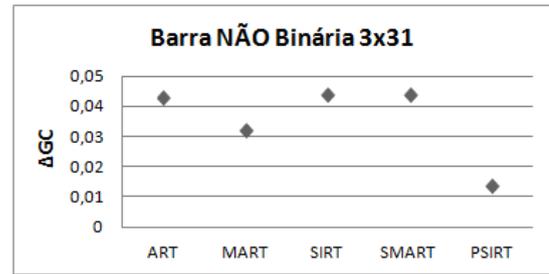
Gráfico 56 – ΔGC dos quadrados binários em 3x31.

Na reconstrução dos quadrados diagonais binários em 3x31, os métodos algébricos produziram muito ruído. O PISRT mesmo não preservando fielmente a imagem original, superou os outros métodos pelo fato de não produzir ruído, situação também consistente com os valores RSME (Gráfico 55) e ΔGC (Gráfico 56), e que reforça a tendência comentada quando da análise do Quadro 6.

Quadro 8 – Reconstrução NÃO binária em 3x31 da barra, anel, círculo e quadrados.

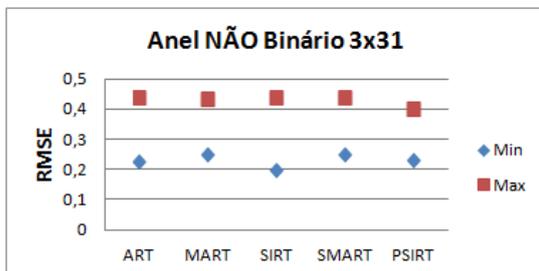
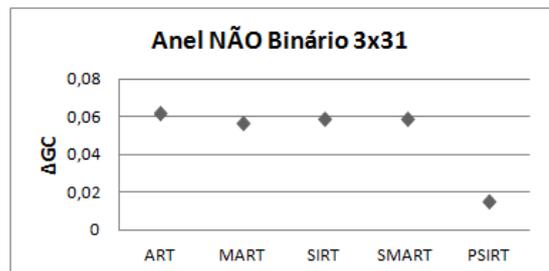
Fantoma	ART	MART	SIRT	SMART	PSIRT

Gráfico 57 – RMSE da Barra NÃO binária em 3x31.

Gráfico 58 – ΔGC da Barra NÃO binária em 3x31.

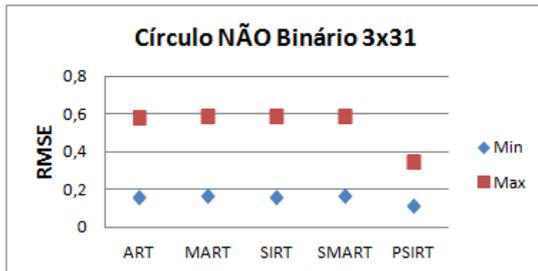
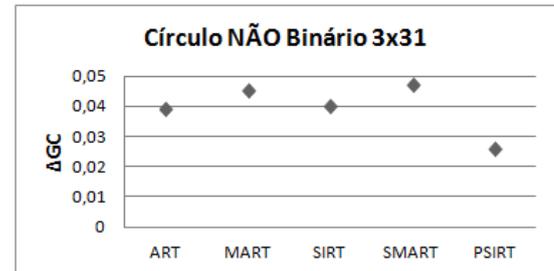
Nessa reconstrução da barra não binária em 3x31, o PSIRT também produziu ruído, porém bem menos que os métodos algébricos. Mesmo assim, os métodos algébricos foram regulares, e sutilmente melhores que o PSIRT no RMSE (Gráfico 57), justamente pela irregularidade da reconstrução do PSIRT. Em contrapartida, o PSIRT produziu a imagem com maior predominância de hexágonos de cor branca. Esse aspecto associado ao fato da grande quantidade de ruído global gerado pelos métodos algébricos nessa reconstrução conferiu ao PSIRT um ótimo desempenho no ΔGC (Gráfico 58).

Gráfico 59 – RMSE do anel NÃO binário em 3x31.

Gráfico 60 – ΔGC do anel NÃO binário em 3x31.

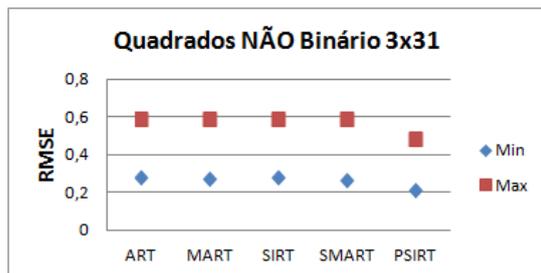
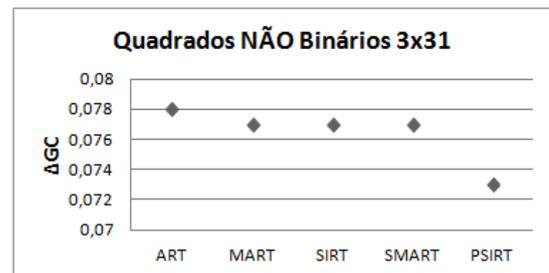
Na reconstrução do anel não binário em 3x31, a figura gerada pelo PSIRT parecer ser uma anomalia, pelo método ter produzido uma espessura do anel bem aquém da original, e mesmo assim ter sido favorecido pelo menos na coerência geométrica (ΔGC – Gráfico 60). Novamente, tal favorecimento está mais associado ao fato da geração da grande quantidade de ruído por parte dos métodos algébricos, do que pela falta de fidelidade do PSIRT com relação às dimensões da imagem, embora essa irregularidade no formato não seja predominante nas reconstruções do PSIRT, tendo acontecido em algumas situações, a exemplo da reconstrução desse anel. Os métodos algébricos produziram tons de cinza de forma regular, com o SIRT produzindo hexágonos um pouco mais escuros. O PSIRT novamente produziu hexágonos um pouco mais claros. Ambas as situações são consistentes com o RMSE (Gráfico 59).

Gráfico 61 – RMSE do círculo NÃO binário em 3x31.

Gráfico 62 – ΔGC do círculo NÃO binário 3x31.

Na reconstrução do círculo não binário em 3x31, o PSIRT é favorecido não apenas pelo fato de não ter produzido ruído, mas também de ter reconstruído a imagem com uma área muito próxima da imagem original, mesmo que o método não tenha preservado a forma arredondada do círculo. O ruído global mais uma vez diminuiu o desempenho dos métodos algébricos. Tais situações são consistentes com o RMSE (Gráfico 61) e ΔGC (Gráfico 62).

Gráfico 63 – RMSE quadrados NÃO binários 3x31.

Gráfico 64 – ΔGC dos quadrados NÃO binários 3x31.

Na reconstrução dos quadrados não binários em 3x31, a situação é idêntica a anterior (a do círculo não binário) com relação à geração de ruído global por parte dos métodos algébricos. A diferença é que o PSIRT produziu mais ruído do que na reconstrução dos quadrados nas configurações anteriores, principalmente na região que separa um quadrado do outro. Mesmo assim, essa quantidade de ruído foi bem aquém das dos métodos algébricos. Um aspecto a ser destacado nessa reconstrução é que, ao mesmo tempo, o PSIRT gerou hexágonos mais claros (tolerância máxima) na parte superior esquerda do quadrado de cima e na parte inferior direita do quadrado de baixo, e hexágonos mais escuros (tolerância mínima), no espaço diagonal entre os dois quadrados. Ambos os fatos são consistentes com o RMSE (Gráfico 63) e ΔGC (Gráfico 64).

Finalizando esta seção sobre os experimentos simulados em 3xN, as tabelas 1 e 2 faz um resumo de todos os valores RME e Δ GC de todos os experimentos realizados, tanto para imagens binárias quanto para imagens não binárias.

Tabela 1 – RMSE e Δ GC das imagens binárias dos experimentos simulados em 3xN.

		Imagens BINÁRIAS									
		ART		MART		SIRT		SMART		PSIRT	
Fantoma	MN	RMSE	Δ GC	RMSE	Δ GC	RMSE	Δ GC	RMSE	Δ GC	RMSE	Δ GC
Barra	3x7	0,000	0,010	0,000	0,010	0,000	0,010	0,000	0,010	0,185	0,036
	3x15	0,170	0,019	0,183	0,023	0,169	0,019	0,183	0,023	0,236	0,028
	3x21	0,146	0,008	0,215	0,009	0,087	0,006	0,216	0,009	0,250	0,008
	3x31	0,091	0,007	0,096	0,007	0,092	0,006	0,091	0,007	0,228	0,013
Anel	3x7	0,239	0,104	0,219	0,091	0,373	0,151	0,219	0,091	0,000	0,054
	3x15	0,239	0,124	0,351	0,192	0,297	0,088	0,414	0,192	0,115	0,060
	3x21	0,374	0,124	0,500	0,350	0,324	0,088	0,480	0,310	0,312	0,075
	3x31	0,209	0,014	0,265	0,025	0,164	0,012	0,265	0,023	0,150	0,010
Círculo	3x7	0,000	0,000	0,187	0,002	0,000	0,000	0,187	0,002	0,000	0,000
	3x15	0,000	0,000	0,139	0,020	0,030	0,003	0,155	0,019	0,117	0,006
	3x21	0,194	0,006	0,299	0,020	0,079	0,003	0,259	0,019	0,050	0,000
	3x31	0,221	0,025	0,260	0,030	0,234	0,019	0,255	0,028	0,228	0,015
Quadrados	3x7	0,115	0,004	0,132	0,007	0,094	0,011	0,162	0,037	0,226	0,097
	3x15	0,150	0,019	0,279	0,019	0,263	0,022	0,267	0,031	0,353	0,102
	3x21	0,279	0,019	0,289	0,019	0,266	0,022	0,300	0,031	0,242	0,102
	3x31	0,259	0,088	0,255	0,072	0,279	0,106	0,293	0,099	0,213	0,029

Tabela 2 – RMSE e Δ GC das imagens NÃO binárias dos experimentos simulados em 3xN.

		Imagens NÃO BINÁRIAS														
		ART			MART			SIRT			SMART			PSIRT		
Fantoma	MN	RMSE		Δ GC	RMSE		Δ GC	RMSE		Δ GC	RMSE		Δ GC	RMSE		Δ GC
		Min	Max		Min	Max		Min	Max		Min	Max		Min	Max	
Barra	3x7	0,158	0,646	0,224	0,168	0,646	0,235	0,177	0,646	0,235	0,166	0,646	0,225	0,209	0,373	0,100
	3x15	0,214	0,585	0,920	0,220	0,580	0,085	0,214	0,584	0,094	0,220	0,584	0,085	0,220	0,584	0,036
	3x21	0,193	0,572	0,006	0,188	0,571	0,006	0,195	0,574	0,006	0,188	0,574	0,006	0,149	0,514	0,026
	3x31	0,149	0,574	0,043	0,132	0,573	0,032	0,150	0,576	0,044	0,130	0,577	0,044	0,190	0,580	0,014
Anel	3x7	0,245	0,494	0,296	0,251	0,493	0,295	0,307	0,457	0,295	0,249	0,493	0,295	0,120	0,476	0,285
	3x15	0,249	0,494	0,243	0,337	0,567	0,244	0,294	0,567	0,244	0,338	0,567	0,244	0,193	0,524	0,234
	3x21	0,318	0,596	0,225	0,346	0,585	0,227	0,291	0,600	0,224	0,344	0,595	0,227	0,224	0,556	0,225
	3x31	0,228	0,438	0,062	0,250	0,434	0,057	0,199	0,438	0,059	0,250	0,438	0,059	0,230	0,400	0,015
Círculo	3x7	0,167	0,527	0,074	0,208	0,527	0,074	0,173	0,527	0,074	0,207	0,527	0,074	0,000	0,288	0,000
	3x15	0,161	0,592	0,049	0,174	0,576	0,045	0,160	0,592	0,065	0,412	0,598	0,048	0,117	0,307	0,035
	3x21	0,158	0,594	0,040	0,179	0,589	0,038	0,163	0,594	0,029	0,180	0,573	0,038	0,137	0,393	0,028
	3x31	0,158	0,587	0,039	0,168	0,591	0,045	0,160	0,589	0,040	0,166	0,591	0,047	0,115	0,353	0,026
Quadrados	3x7	0,185	0,643	0,311	0,190	0,630	0,323	0,187	0,630	0,310	0,190	0,630	0,323	0,222	0,457	0,139
	3x15	0,293	0,555	0,012	0,298	0,593	0,164	0,299	0,592	0,164	0,300	0,593	0,164	0,302	0,412	0,071
	3x21	0,300	0,595	0,112	0,300	0,595	0,129	0,299	0,595	0,113	0,300	0,595	0,113	0,215	0,481	0,094
	3x31	0,280	0,586	0,078	0,274	0,584	0,077	0,283	0,586	0,077	0,270	0,584	0,077	0,215	0,481	0,073

5.2 Experimentos Simulados Exclusivamente em 3x7

A simulação focalizada na mais baixa configuração é de extrema importância para a comparação do PSIRT com os demais métodos, já que é esperado um ótimo desempenho do método para essa configuração.

- **Configuração MN:** 3x7, o que gera 37 hexágonos (pixels naturais).
- **Tipo de imagem:** não binária.
- **Resoluções das imagens:** 148x144, 141x130, 146x135, 136x131.
- **Métodos comparados:** ART, MART, SIRT, SMART.
- **Fator de relaxação:** ART com 0,1 | MART com 0,01 | SIRT com 0,1 | SMART com 0,001.
- **Métrica de erro:** RMSE e GC.
- **Iterações:** definida pelo usuário, até atingir a estabilidade em cada execução.
- **Fantomas:** barra vertical direita, anel, círculo e quadrados diagonais.
- **Observação:** os parâmetros do PSIRT foram alterados entre um conjunto de iterações e outro, na tentativa de encontrar melhores resultados para o método, por ser a configuração 3x7 a situação de maior interesse na avaliação do método, dentre as configurações 3xN. Os principais parâmetros alterados foram os seguintes:
 - ✓ **Aceleração das partículas:** 0,1 | 0,01 | 0,001 | 0,0001.
 - ✓ **Quantidade de partículas:** respeitando o limite máximo de 168 ($3 \times 7 \times 8$) partículas para cada rodada.
 - ✓ **Quantidade de rodada:** na ordem de dezenas (cada rodada representa um conjunto de iterações).
 - ✓ **Quantidade de iterações pré-definida:** na ordem de centenas, que podem chegar a milhares de acordo com o parâmetro “quantidade de rodadas” escolhido.
 - ✓ **Quantidade de iterações para se atingir um melhor resultado:** um novo conjunto de iterações que é ativado caso o sistema não atinja o equilíbrio.

Fantoma BARRA VERTICAL DIREITA

Quadro 9 – Reconstrução da barra vertical direita em 3x7.

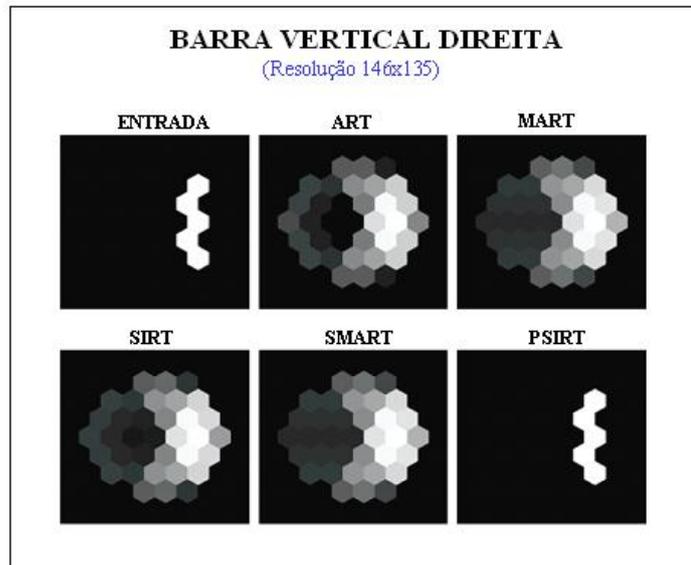
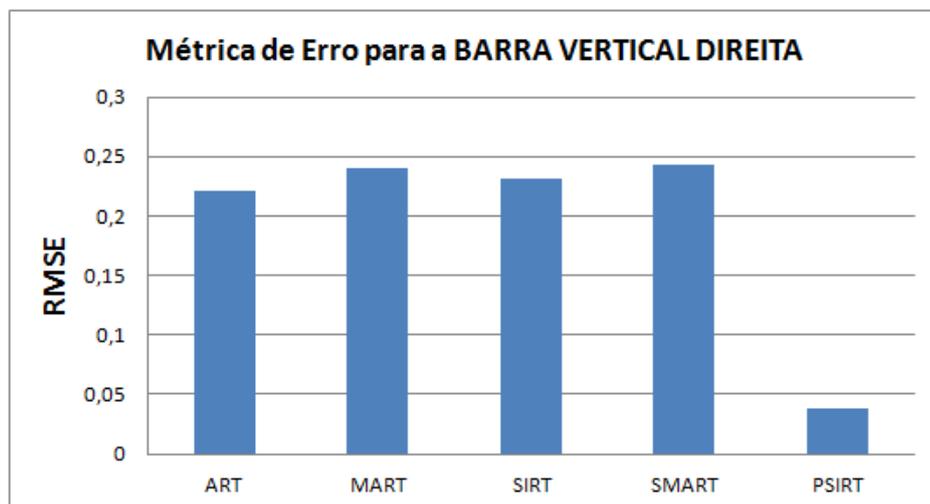


Gráfico 65: RMSE na reconstrução da barra vertical direita em 3x7.



Na barra vertical direita, a reconstrução do PSIRT foi muito mais eficiente do que nos demais métodos. O resultado do PSIRT é praticamente a representação fiel da imagem original. O MART e SMART geraram saídas muito parecidas, fato refletido no RMSE (Gráfico 65). Este gráfico também comprova uma discrepância muito grande favorável ao PSIRT nessa reconstrução.

Fantoma ANEL

Quadro 10 – Reconstrução do anel em 3x7.

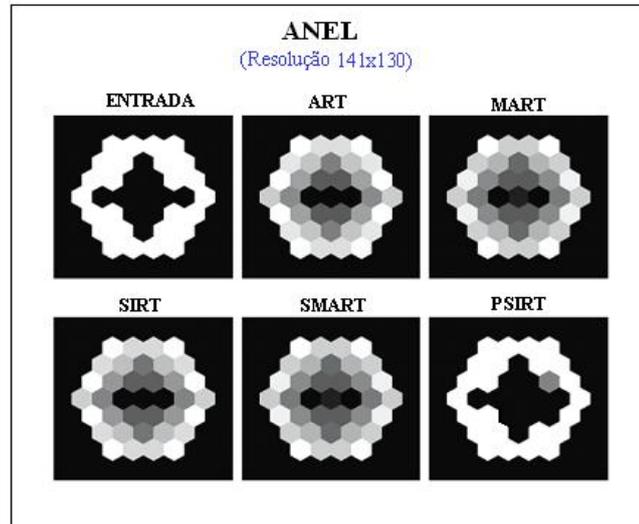
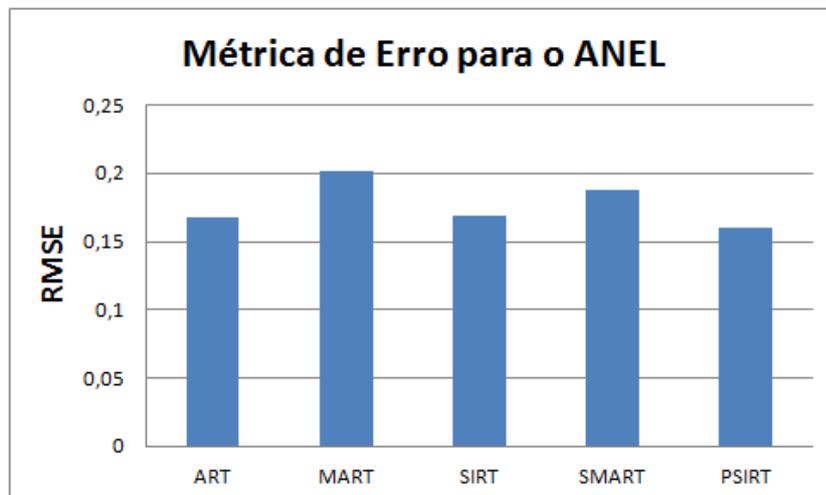


Gráfico 66 – RMSE na reconstrução do anel 3x7.



Apesar da irregularidade generalizada na reconstrução do centro do anel, o PSIRT visualmente é o que mais bem representa a reconstrução desse fantoma. É justamente na parte interna do anel que o PSIRT supera os demais métodos, os quais reconstróem o núcleo com apenas três hexágonos preto. No contorno do anel, o PSIRT também é mais eficiente, já que os outros métodos possuem alguns hexágonos sutilmente mais escuros, enquanto no PSIRT todos os hexágonos do contorno são branco. Na taxa de erro, todos os métodos foram razoáveis nessa reconstrução, com uma pequena diferença favorável ao PSIRT, conforme pode ser observado no Gráfico 66.

Fantoma CÍRCULO

Quadro 11 – Reconstrução do círculo em 3x7.

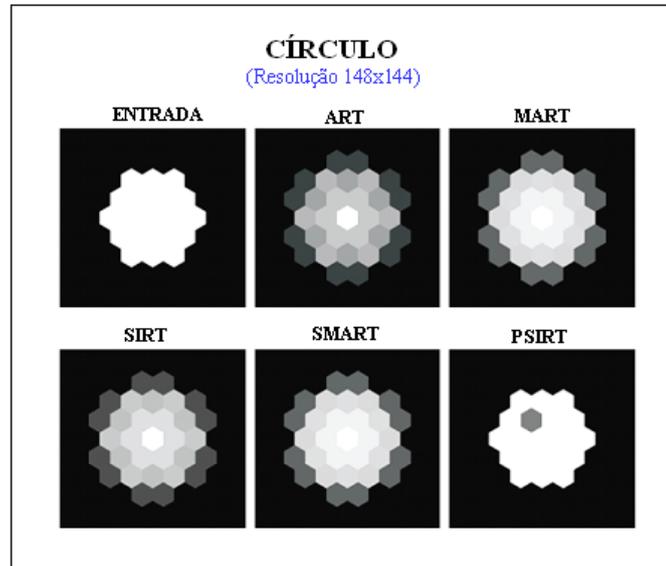
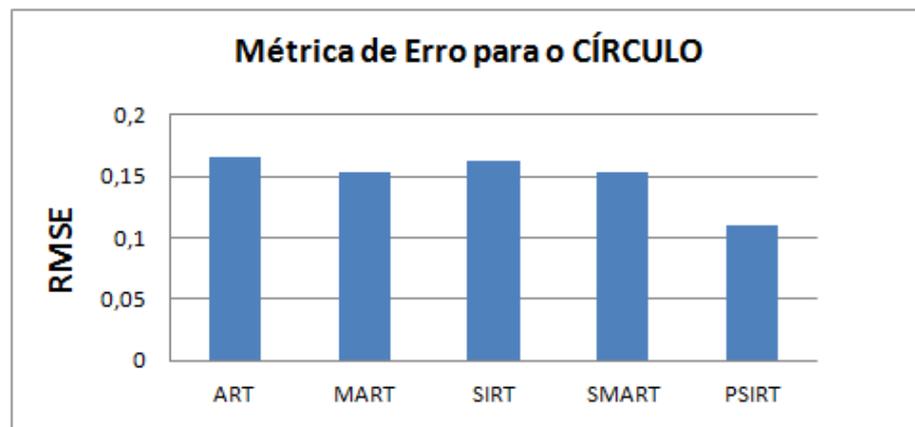


Gráfico 67 – RMSE na reconstrução do círculo em 3x7.



Na reconstrução do círculo, o PSIRT ofereceu melhor resultado visual. Aparentemente não há ruído para nenhum dos métodos, mas eles, com exceção do PSIRT, reconstróem com um círculo um pouco maior do que a figura original. Esse excesso pode sim, ser considerado ruído. Uma observação interessante é que o MART e SMART geraram praticamente a mesma imagem de saída, fato consistente com o Gráfico 67.

Fantoma QUADRADOS DIAGONAIS

Quadro 12 – Reconstrução dos quadrados diagonais em 3x7.

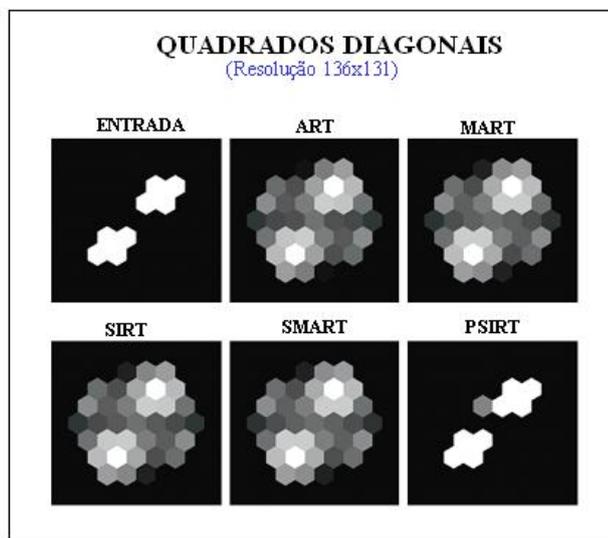
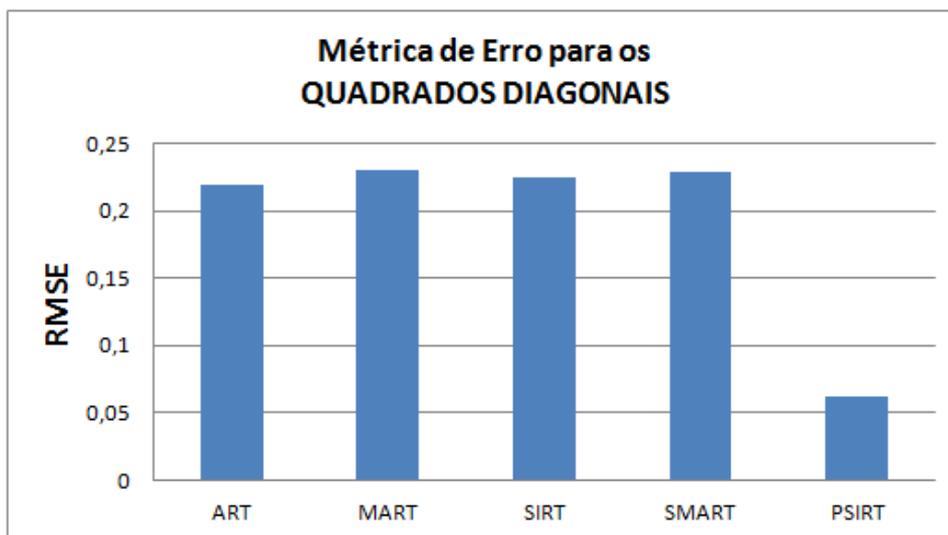


Gráfico 68 – RMSE na reconstrução dos quadrados diagonais em 3x7.



Nessa reconstrução dos quadrados diagonais, os métodos algébricos novamente geraram muito ruído global, enquanto que o PSIRT gerou apenas um ruído do tipo local, situação consistente com o RMSE (Gráfico 68).

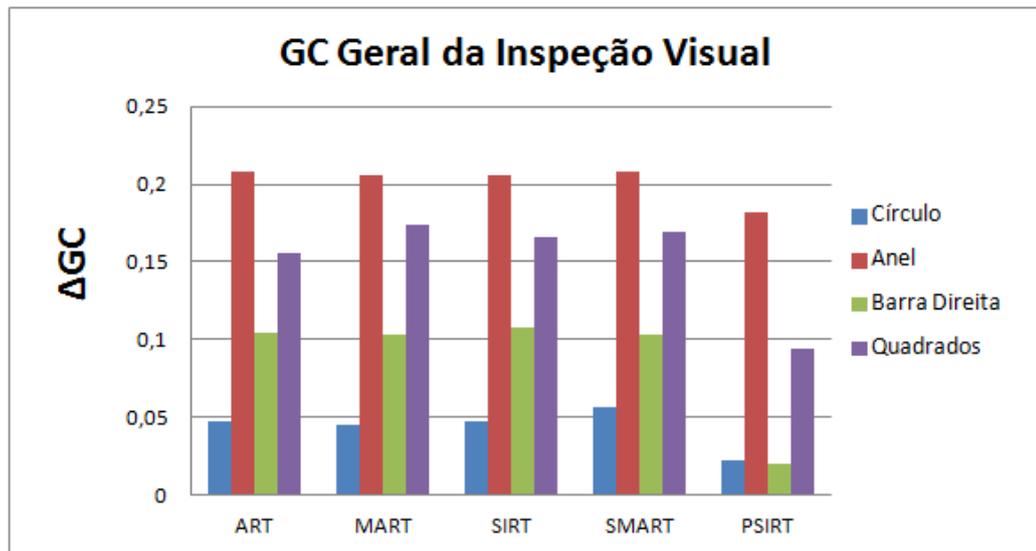
Os resultados das reconstruções da família ART foram deixados na forma não binária (preto, branco, cinza), para não enviesar a comparação, já que a escolha de uma tolerância no tom de cinza poderia adicionar desvantagens em algum aspecto. Por exemplo, a parte interna do anel do Quadro 10 na família ART, poderia ficar maior ou menor, dependendo da tolerância adotada. Enfim, não existe uma forma de homogeneizar os parâmetros nos cinco algoritmos (PSIRT e família ART), de maneira a tornar a comparação rigorosamente justa.

Porém, hexágonos na cor branca representam a melhor situação. Hexágonos intermediários (cinza) representam informações úteis à reconstrução, mas relativamente menos expressivas do que o branco. Sucessivamente, quanto mais escuro o hexágono, ou seja, quanto mais o tom de cinza se aproxima do preto, a região coberta da imagem ou é duvidosa ou é ruidosa e, hexágonos com diferentes tonalidades de cinza, são predominantes nos métodos da família ART. Já no PSIRT esses hexágonos são pouquíssimos, pois essa é uma característica do método.

Visualmente na configuração 3x7, as reconstruções do PSIRT foram mais limpas, mais enxutas, enquanto os demais métodos ofereceram diferenças de tonalidades ao redor da imagem de saída, e até muito distantes das fronteiras espaciais da referida imagem. Os cenários favoráveis ao PSIRT nessa configuração 3x7 acabaram sendo consolidados com os gráficos RMSE.

Em extensão, fazer uma avaliação também com o ΔGC é interessante, pois essa métrica tem relação direta com os pesos das cores dos hexágonos, como também com suas posições entre si. As partículas úteis estritamente vizinhas, por exemplo, definem um aglomerado de hexágonos com altíssima possibilidade de representar o material que estar sendo reconstruído. Aglomerados desse tipo, o ΔGC reconhece em seu cálculo geométrico. Assim, o Gráfico 69 apresenta o ΔGC de cada método para cada fantoma reconstruído, o qual de fato também aponta o PSIRT como o mais eficiente na comparação com os métodos da família ART nessa configuração exclusivamente em 3x7.

Gráfico 69 – ΔGC dos fantasmas círculo, anel, barra direita e quadrados na reconstrução em 3x7.

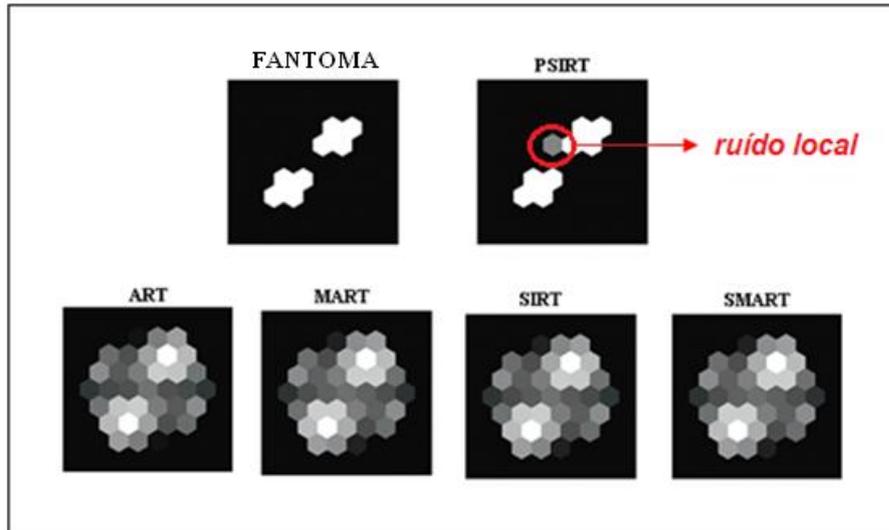


O Gráfico 69 consolida as reconstruções visuais e correspondentes RMSE dos fantasmas na configuração exclusivamente em 3x7, situações que conferem ao PSIRT melhores resultados na comparação com os métodos algébricos nesse contexto. Esse fato sinaliza o PSIRT como um método competitivo para a reconstrução tomográfica numa configuração baixa de projeções e trajetórias, requisito de extrema importância para a tomografia industrial.

5.3 Experimentos Simulados para Discutir Ruídos

As reconstruções feitas até agora sinalizaram a existência nítida de ruídos. Os tipos de ruídos são locais e globais, e eles implicam diretamente no desempenho dos métodos. O Quadro 13 mostra uma reconstrução dos métodos, destacando um ruído local no PSIRT.

Quadro 13 – Reconstruções do fantoma quadrados diagonais com ruído local.



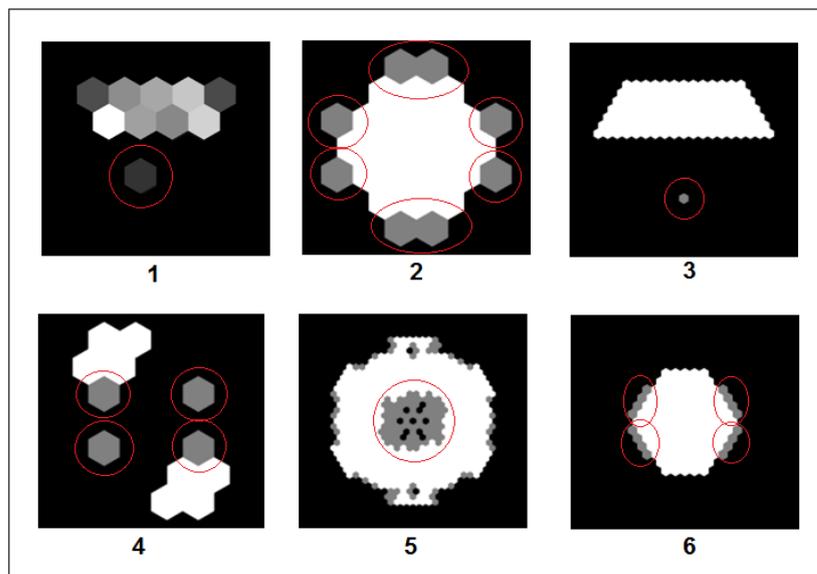
No PSIRT, uma partícula pode ficar isolada e receber um preenchimento de cor (branco a maioria das vezes, e cinza algumas vezes) como se estivesse representando a presença do material. Quando não está – como no exemplo da segunda figura do Quadro 13 –, tal partícula é considerada como sendo um ruído. Entretanto, essa situação é fácil de ser tratada e, para o PSIRT acontece assim, de forma setorizada, a qual caracteriza o conceito de ruído *local*. Enfatizando, o ruído é local quando a falha é setorizada, ou seja, quando o erro não abrange uma região muito espaçosa além das fronteiras da imagem que se pretende reconstruir, que no exemplo do quadro supracitado são dois quadrados diagonais.

Para os demais métodos, nota-se no Quadro 13 que as diferenças de tonalidades dos hexágonos são muitas e abrangem uma área considerável além dos quadrados a serem reconstruídos. Nesse caso, o ruído é considerado *global* e, devido à diferença entre a figura original e a reconstruída, os métodos precisam tratar esse problema numa relação que engloba toda região além da imagem, com os fragmentos (hexágonos) que não oferecem informações úteis para a reconstrução.

Os ruídos globais em geral, quando não são ruídos locais intensamente acumulados, se devem a vieses da forma como os algoritmos executam, similares a padrões devido ao *aliasing* espacial (padrão de Moiret, efeitos de Gibbs, dentre outros). Enfim, os ruídos globais são difíceis de serem tratados (OLIVEIRA, 2011).

Um aspecto importante é que uma partícula isolada (branca ou cinzenta) pode NÃO ser ruído, ou seja, pode estar representando material que de fato é bastante confinado, no caso, a uma região aproximadamente do tamanho de um pixel natural. Em outras palavras, os ruídos locais e os materiais confinados a regiões de tamanho de um pixel são tratados da mesma forma, por uma assunção simplificadora (que pode ser vista como uma informação a priori dos dados de entrada), de que o objeto a ser reconstruído cobre regiões contínuas de mais de um pixel natural. Esta assunção corresponde (ou dá sustentação) a uma filtragem passa-baixa. No caso particular do PSIRT, os ruídos são estritamente locais, tais como os exemplos do Quadro 14.

Quadro 14 – Exemplos de ruídos locais no PSIRT.



As partículas (hexágonos) ou aglomerados de partículas circuladas no Quadro 14 são candidatas a ruído local no PSIRT. Aquelas isoladas com certeza são ruídos, a exemplo de partículas isoladas em 1 e 3 e de duas partículas isoladas em 4. Já aquelas coladas às fronteiras da imagem, como em 2, 4 e 6, são candidatas a ruído, pois mesmo que estejam ligeiramente fora da área da imagem, suas estreitas proximidades com a imagem em questão podem impedir que as mesmas sejam ruídos quando da execução de novas iterações.

Enfim, a dinâmica de iterações do PSIRT possibilita que partículas isoladas possam ser removidas quando não são necessárias à reconstrução da imagem. Entretanto, para o caso das fronteiras da imagem, as partículas ruidosas só seriam descartadas se permanecessem assim em sucessivas iterações. Na situação esperada, elas seriam “absorvidas” pela imagem.

O caso 5 do Quadro 14 é atípico, pois não deveria ter hexágonos cinza nem branco no núcleo do Anel, além de ter poucas partículas de cor preta nessa região. Mesmo assim, para o caso 5, o ruído é local, o que dá para ser tratado, mesmo que o método necessite efetuar mais iterações, até melhorar a reconstrução da região. É muito comum na execução do PSIRT a eliminação de ruídos nas iterações sucessivas. As figuras 32 e 33 mostram isso.

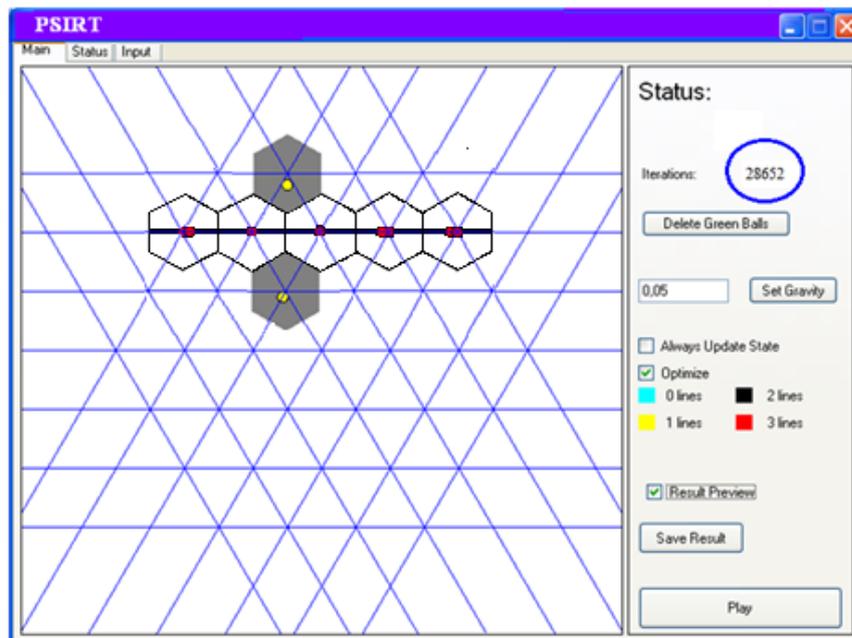


Figura 32: Reconstrução de uma barra gerando dois ruídos locais em 28.652 iterações.

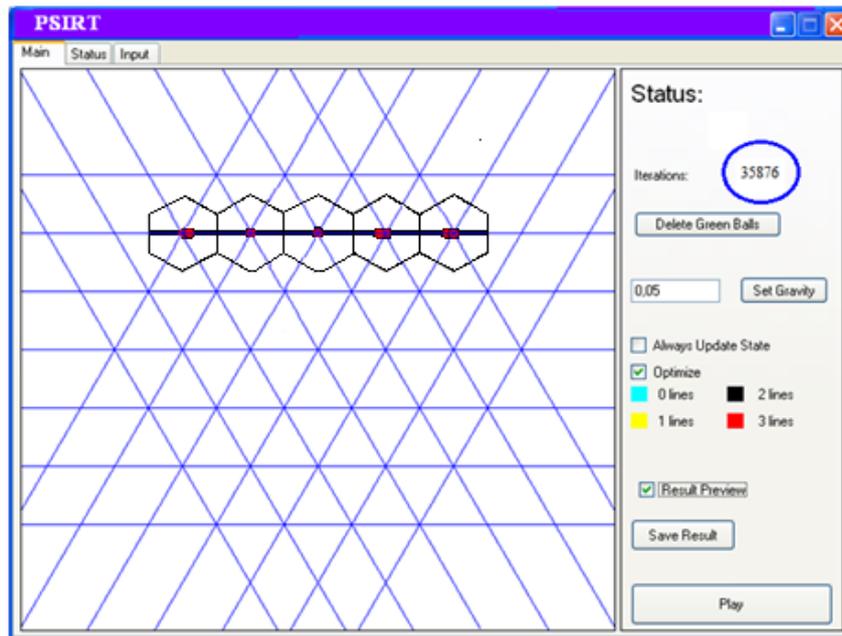


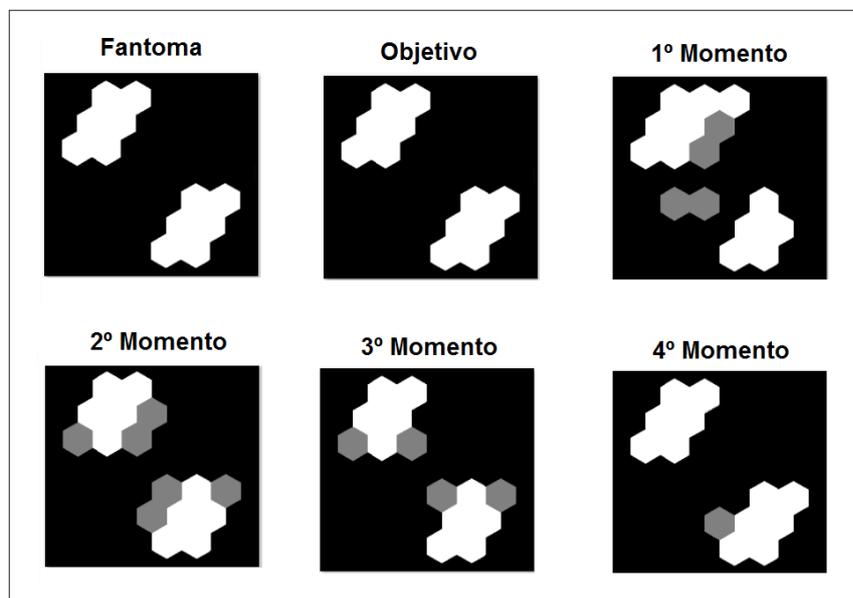
Figura 33: Eliminação dos ruídos exemplificados na Figura 34 em 35.876 iterações.

No PSIRT, os ruídos são causados por partículas atraídas por uma única trajetória, ou por partículas estacionadas em interseções duplas que ficam fora da área da imagem. A maneira padrão do PSIRT de tratar partículas que representam ruído é justamente com as iterações sucessivas do sistema, conforme exemplo das figuras 32 e 33. Ou seja, iterações seguintes podem reposicionar tais partículas ou eliminá-las naturalmente. Por exemplo, uma das trajetórias que atraiu uma partícula desse tipo pode descartá-la. Uma vez descartada e reposicionada de forma uniformemente aleatória, outra trajetória pode atrair a partícula para regiões que fazem parte da imagem do objeto, que é a situação mais desejada. Partículas em 2 e 6 do Quadro 14 são exemplos em que isso pode ocorrer.

A possibilidade de tratar ruído do tipo local é um avanço num método de reconstrução. De forma contrária, o ruído global é uma situação indesejada. Ou seja, o ruído global é difícil de ser tratado, conforme já mencionado, e é muito comum em métodos algébricos. Assim, concluí-se que o PSIRT tem o tratamento de ruído como um diferencial (também) para a reconstrução tomográfica, pelo fato do ruído no método ser do tipo local. Isso impõe flexibilidade e tolerância à falha ao método, aspectos preciosos no PSIRT que, por definição, não precisa ser rigorosamente *preciso* na reconstrução, bastando apenas reconhecer a região aproximada do objeto a ser reconstruído.

Enfim, é uma sofisticação do método o tratamento eficiente de ruído. O Quadro 15, também obtido de experimentos simulados, finaliza essa discussão com outro exemplo, o qual mostra 4 iterações (momentos) sucessivas até que o resultado do “4º Momento” se assemelhe muito com a imagem que se pretende reconstruir (Objetivo). Naturalmente a redução intrínseca de ruídos pelo PSIRT, que ocasiona o custo computacional das iterações adicionais, não isenta a reconstrução dos benefícios de uma filtragem passa-baixa.

Quadro 15 – Momentos na reconstrução do fantoma quadrados diagonais.



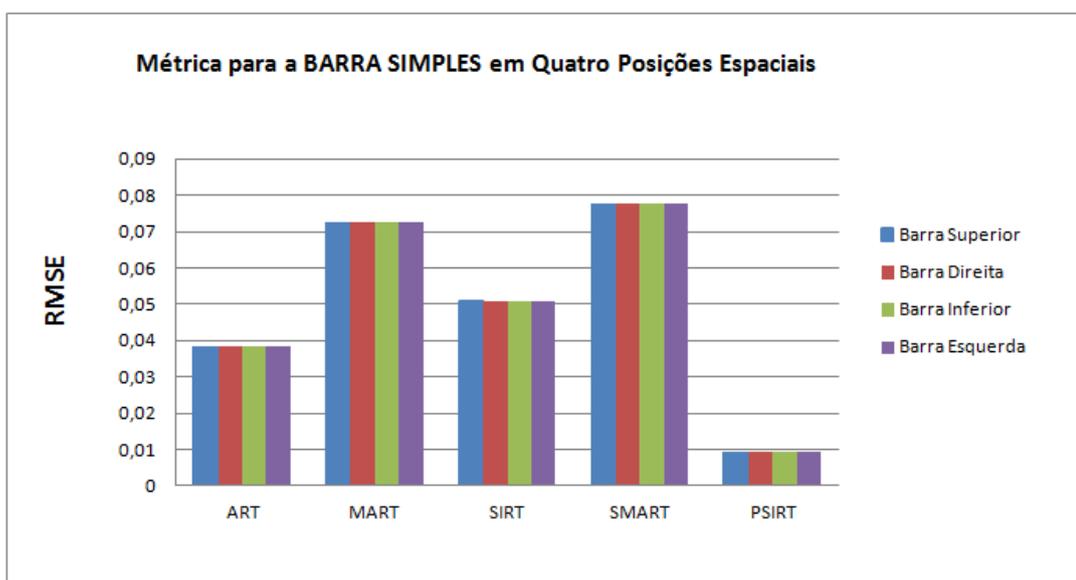
5.4 Experimento Simulado no caso Especial da Barra Simples

A barra simples foi o artefato que visualmente foi bem representado **igualmente** por todos os métodos, com destaque para o ART e PSIRT. Os experimentos foram feitos para esse tipo de barra em quatro posições espaciais (Quadro 16), para testar a robustez do PSIRT.

Quadro 16 – Reconstrução da barra simples em quatro posições espaciais.

ART	MART	SIRT	SMART	PSIRT
<i>RMSE: 0,038312</i>	<i>RMSE: 0,072625</i>	<i>RMSE: 0,050929</i>	<i>RMSE: 0,077701</i>	<i>RMSE: 0,009246</i>

Gráfico 70 – RMSE na reconstrução da barra simples.



No terceiro experimento (Gráfico 73), o PSIRT demonstrou novamente que o aumento na quantidade de partículas torna o método **extremamente** lento em relação aos métodos da família ART, pois são necessárias mais iterações para atingir o equilíbrio, que no exemplo foram 25.010 iterações.

Enfim, os métodos algébricos se mantiveram relativamente regulares nas três situações de reconstrução, enquanto que o PSIRT apresentou um comportamento desfavorável com relação ao tempo de reconstrução, conforme sintetizam os gráficos 74 e 75.

Gráfico 74 – Análise do aumento do tempo de reconstrução do PSIRT em relação ao aumento de partículas.

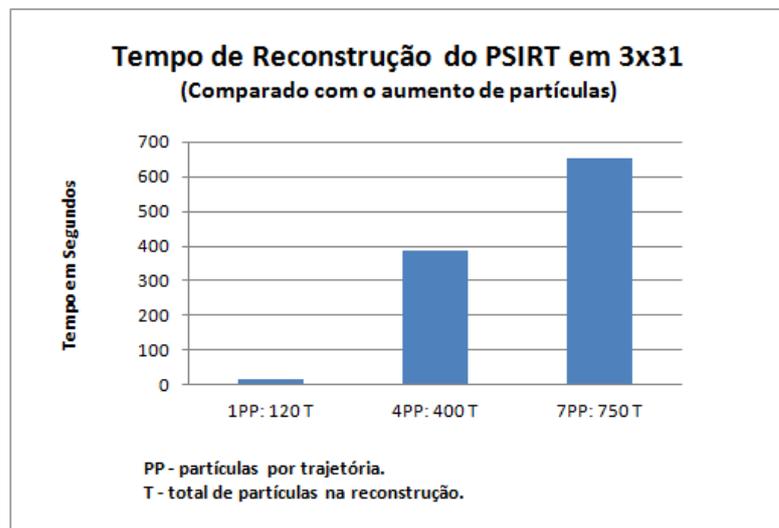
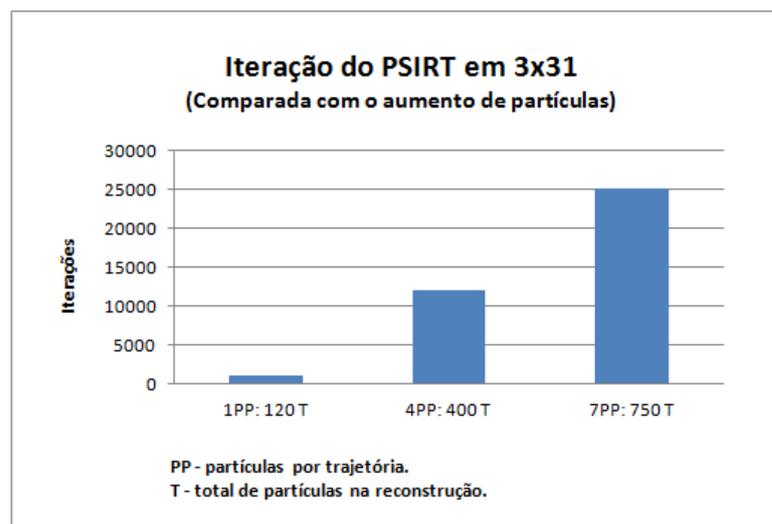


Gráfico 75 – Análise do aumento da quantidade de iterações do PSIRT em relação ao aumento de partículas.



A conclusão com esses três experimentos de fato já era esperado, pois logicamente quanto maior o número de partículas, maior será a quantidade de iterações que o PSIRT irá precisar para atingir o equilíbrio e, em consequência, maior será o tempo de reconstrução. Entretanto, os métodos algébricos apresentaram certa regularidade nesse desempenho quanto aos sinogramas analisados.

Entretanto, a forma a qual os experimentos simulados desta subseção foram realizados, favoreceu aos métodos algébricos, pois os tempos de reconstrução desses métodos foram calculados em modo *batch*²³, que é computacionalmente muito mais rápido do que o modo gráfico. O PSIRT é estritamente visual (modo gráfico). O esforço computacional para mover partículas e atualizar o *status* visual do sistema, por exemplo, aumenta significativamente o tempo de execução do método. Certamente, uma versão batch do PSIRT produziria tempos bem menores. Essa suposta versão é uma possibilidade para trabalhos futuros, já que o investimento nesta tese é na eficiência e acompanhamento visuais das reconstruções.

Uma solução para manter o PSIRT competitivo nesse aspecto *tempo* em situações da necessidade de muitas partículas é o investimento numa plataforma de hardware apropriada para o método, uma vez que hardware de alta performance é uma realidade concreta e cada vez menos onerosa no contexto tecnológico mundial. Além disso, uma versão do PSIRT rápido é uma questão natural no processo evolutivo do método, o que irá diminuir consideravelmente o tempo de reconstrução em questão, ainda que seja preservada a condição de execução gráfica no método.

²³ Execução sem a interface gráfica.

5.6 Experimentos do PSIRT com dados reais do Tomógrafo HSGT

Dados dos experimentos:

- **Configuração MN:** 5x17, o que promove interseções envolvendo 2, 3, 4 e 5 trajetórias, e não apenas interseções triplas como no modelo 3xN.
- **Fantoma:** os dados reais foram obtidos em (MAAD, 2009), que realizou experimentos a partir de um objeto cilíndrico feito de polipropileno com dois furos longitudinais, como mostra a Figura 34. O objetivo foi o de simular uma parte da dinâmica de bolhas de gás imersas em óleo dentro de um tubo. O material de polipropileno apresenta propriedades semelhantes às da densidade do óleo, que é de aproximadamente $0,92 \text{ g/cm}^3$.

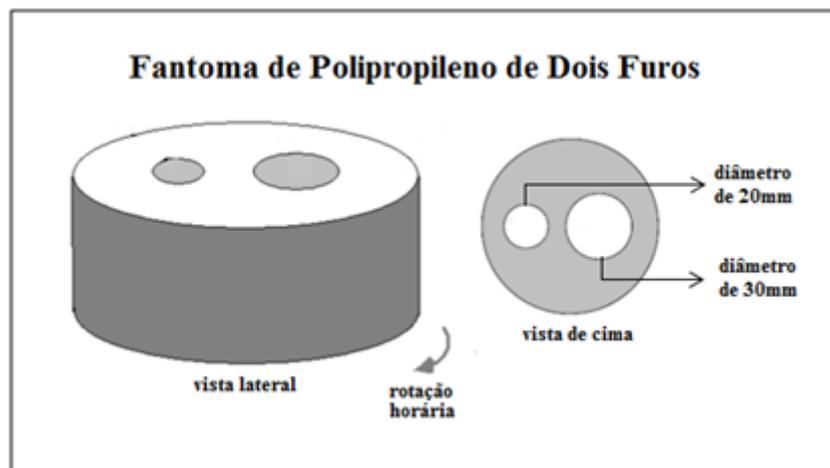


Figura 34: Polipropileno com dois furos longitudinais
Fonte da imagem: (MAAD, 2009, pág. 26).

- **Resolução do fantoma:** 32x32 pixels.
- **Quantidade de partículas:** 63 partículas por trajetória, definido tendo como referência os valores mínimo (55) e máximo (70), estimados para a simulação para esse tipo de objeto.
- **Aceleração das partículas:** 0,04 (pixel por unidade quadrada de tempo).
- **Tempo de integração²⁴ (TI):** 1, 15, 20, 35, 50 e 100ms.

²⁴ O tempo de integração é um parâmetro experimental regulado na parte de aquisição de dados, e representa o tempo que o detector fica esperando até que um quadro de imagem seja formado.

- **Velocidade de rotação (VR):** 0,6 | 0,628 | 6,283 | 18,86 | 31,416 rad/s.
- **Métrica de erro:** RMSE.
- **Iterações:** dinâmicas, mas na ordem de milhares.
- **Comparação:** a comparação foi feita justamente com os experimentos realizados em (MAAD, 2009), nas mesmas condições (configurações) definidas aqui, com relação aos quadros de imagem, TI e VR.

Quadro 17 – Reconstrução do Polipropileno de Dois Furos.

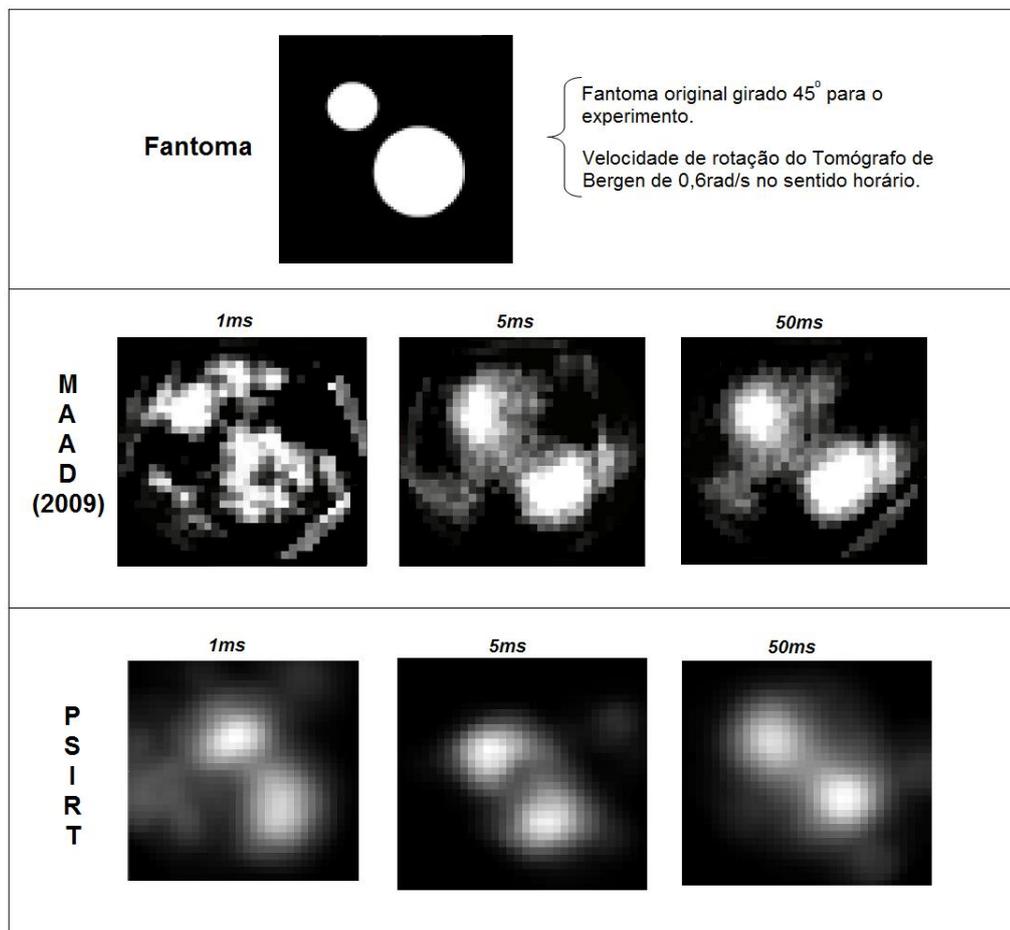
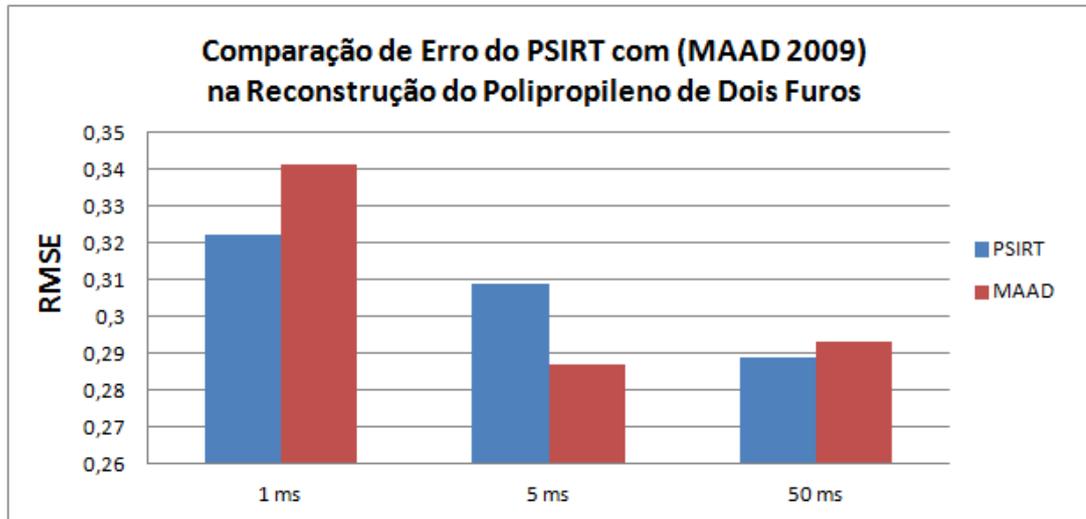


Gráfico 76 – Comparação da métrica de erro RMSE da reconstrução do PSIRT com a reconstrução feita em (MAAD, 2009), no polipropileno de dois furos.

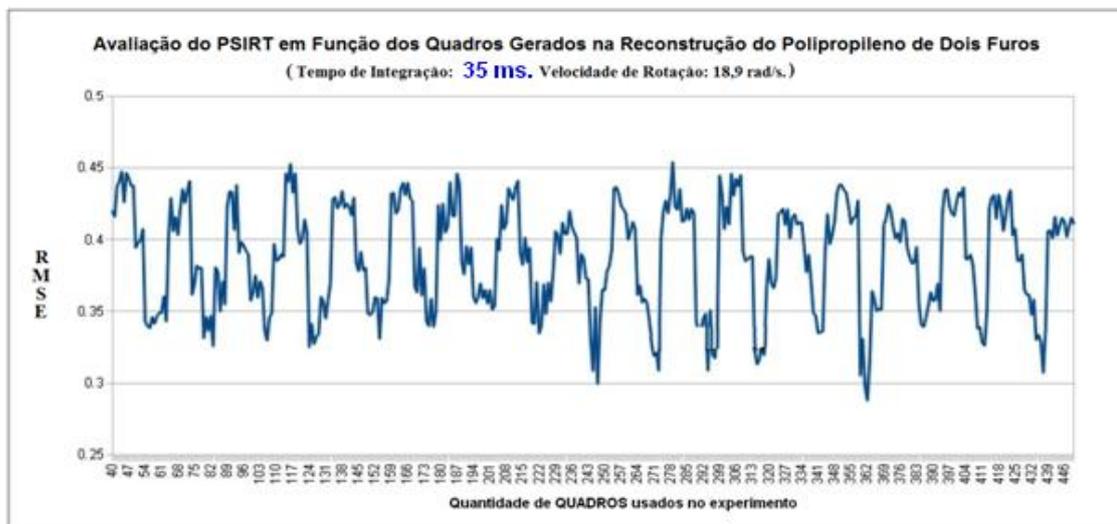


O Gráfico 76 mostra que, em duas das três saídas comparadas, o PSIRT foi mais eficiente do que a reconstrução feita em Maad (2009): no tempo de integração de 1ms e 50ms. Esse fato credencia o PSIRT para ser executado no tomógrafo HSGT. Mas é preciso explorar um conjunto maior de situações para confirmar o desempenho do método nesse tipo de tomógrafo. Uma maneira de fazer isso é variando o tempo de integração e a velocidade de rotação, para analisar a comportamento do PSIRT com relação ao RMSE. Os gráficos 77 e 78 mostram o PSIRT nos tempos de 20ms e 35ms, ambos numa rotação de 18,6rad/s, com aproximadamente 460 quadros de imagem considerados em cada experimento.

Gráfico 77 – Avaliação do RMSE do PSIRT com tempo de integração de 20ms.



Gráfico 78 – Avaliação do RMSE do PSIRT com tempo de integração de 35ms.

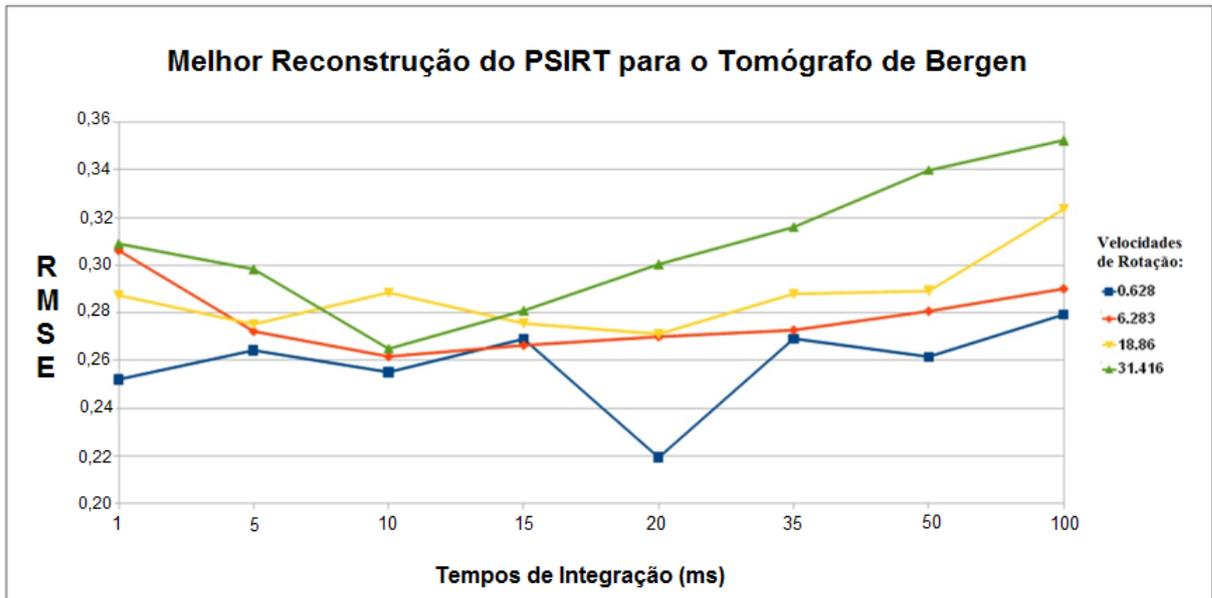


Avaliar o PSIRT com relação ao RMSE – alternando o tempo de integração e a velocidade de rotação – é um procedimento interessante para dimensionar a eficiência do método no tomógrafo HSGT, conforme dados dos gráficos 77 e 78, que mostram tal avaliação para os tempos de integração de 20 e 35ms, respectivamente, na velocidade de rotação de 18,6 rad/s. Assim, encontrar o tempo de integração e velocidade de rotação que vão apontar o melhor desempenho do PSIRT é uma informação relevante para a avaliação do método. Para isso, mais cálculos do RMSE foram realizados, nas seguintes situações: TI de 1, 5, 10, 15, 20, 35, 50 e 100ms, e VR de 0,628, 6,283, 18,860 e 31,416 rad/s. Tal avaliação é consolidada no Gráfico 79.

Ressalta-se que os experimentos planejados e executados em Maad (2008) tiveram como objetivo determinar o tempo de integração ideal para o tomógrafo HSGT, numa situação real de bolhas atravessando o riser com componentes de movimento horizontal e vertical compatíveis com o verificado na indústria. Eles continham conhecidamente tipos de ruídos devido a problemas de amostragem temporal, ou seja, a velocidade do fantoma podendo induzir na reconstrução o efeito de uma filtragem de passa-baixa temporal, a depender do tempo de integração em questão. Acontece que este estudo se tornou adequado para a validação do PSIRT, por apresentar ruídos a priori que podem ajudar na determinação da eficiência e robustez do método, e ainda podem ser abstraídos para aproximar de uma inferência dos ruídos introduzidos pelo próprio método.

Enfim, os gráficos 77 e 78 são úteis para se avaliar ou certificar-se do comportamento global do PSIRT associado ao movimento do fantoma previsto pelo estudo do RSME ao longo de cerca de 400 quadros. Os gráficos confirmam a menos de ruído da concordância dos resultados do PSIRT, incluindo as diferenças nos mínimos locais.

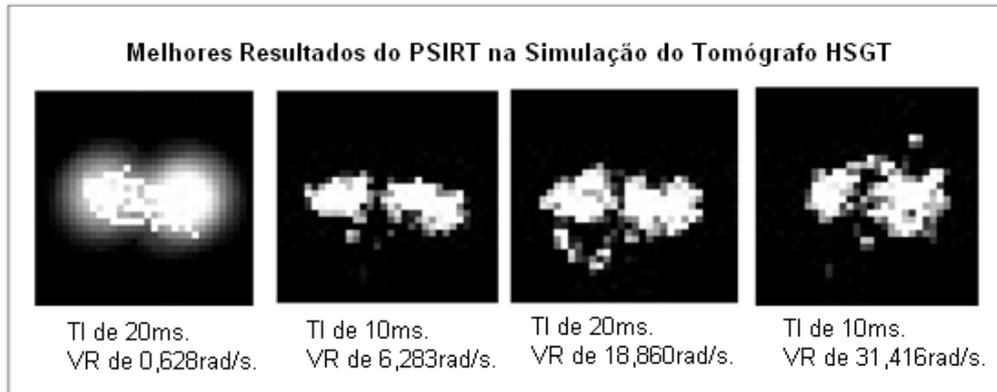
Gráfico 79 – Desempenho do PSIRT variando o TI e a VR.



Síntese do Gráfico 79:

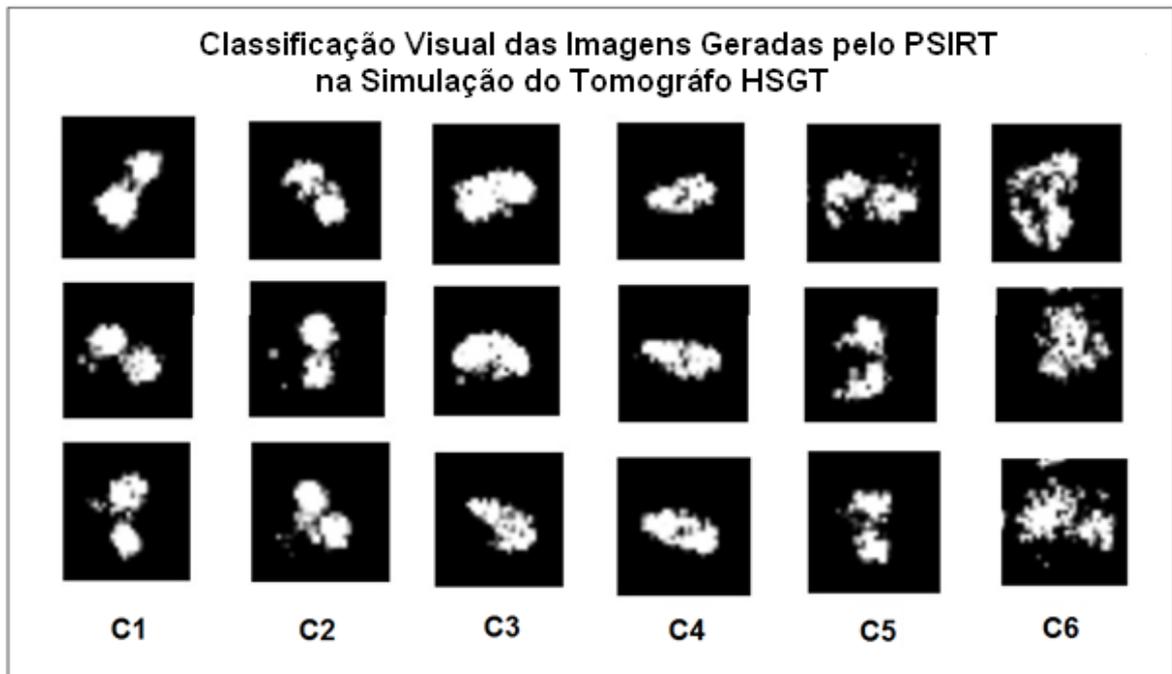
- A tendência geral é consistente com o fato de que, quanto maior a velocidade de rotação do objeto dentro do tomógrafo HSGT, pior é a reconstrução do fantoma;
- Os melhores resultados são obtidos entre os tempos de integração de 10 a 20ms;
- O PSIRT tem o melhor desempenho para o tempo de integração de 20ms na rotação de 0,628 rad/s;
- Os melhores resultados do PSIRT são os seguintes (as imagens dessas reconstruções são mostradas no Quadro 18):
 - ✓ TI de 20ms e VR de 0,628 rad/s;
 - ✓ TI de 10ms e VR de 6,283 rad/s;
 - ✓ TI de 20ms e VR 18,86 rad/s;
 - ✓ TI de 10ms e VR de 31,416 rad/s.

Quadro 18 – Melhores reconstruções do PSIRT simulando o tomógrafo HSGT com dados reais.



São os valores TI e VR que definem as formas das imagens. Diversos são esses valores e suas combinações, o que, em consequência, gera um número muito grande de imagens com distinções próximas, médias e discrepantes entre elas. Uma análise mais adequada para este caso é de se utilizar de uma estatística descritiva, criando-se a noção de classes de qualidade de reconstrução. São classes que vão ser definidas de acordo com a semelhança visual das imagens. Portanto, é uma tarefa *artesanal*, já que a classificação é feita observando a *olho nu* imagem por imagem. Foram mais de 75.000 quadros de imagem gerados nas reconstruções, dos quais foram amostradas na ordem de centenas de imagens para diferentes valores de TI e VR. Enfim, as imagens geradas pelo PSIRT foram divididas em 6 classes, conforme Quadro 19.

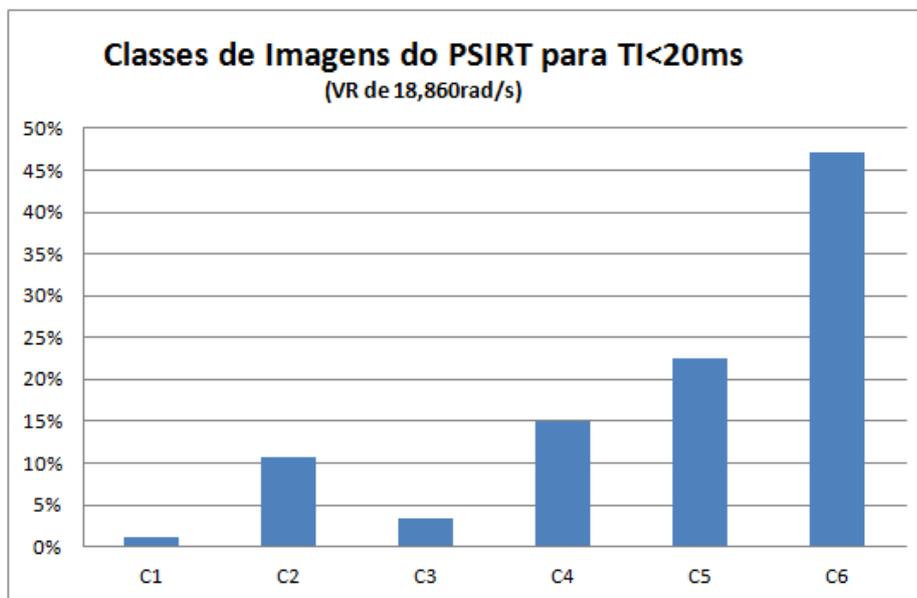
Quadro 19 – Classificação visual das imagens geradas pelo PSIRT.



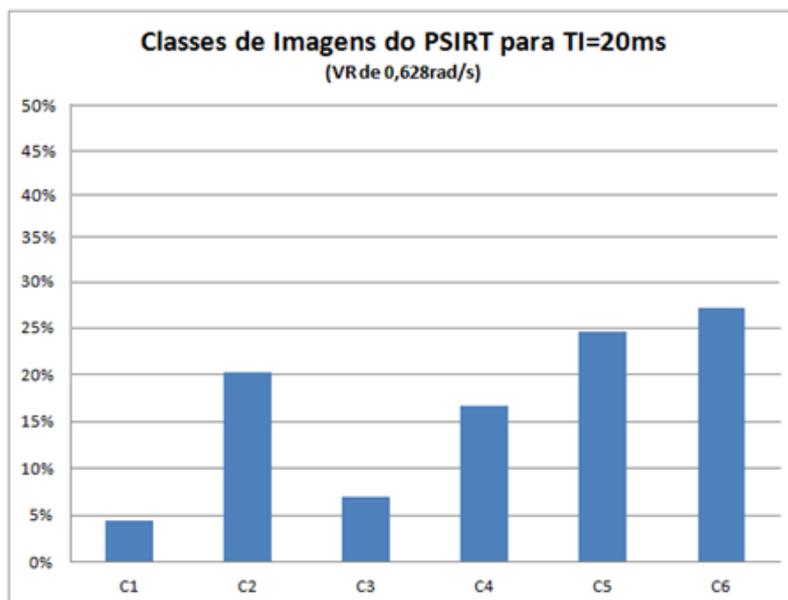
Síntese das classes do Quadro 19:

- Classe C1:** Os formatos dos furos e a distância entre eles são preservados.
- Classe C2:** Apenas o formato de um dos furos é preservado, enquanto o outro furo fica muito irregular, além da distância entre os furos não ser preservada.
- Classe C3:** A distância entre os furos é completamente anulada, mas os formatos dos furos são relativamente regulares.
- Classe C4:** A distância entre os furos é completamente anulada e os formatos dos furos são irregulares.
- Classe C5:** Os formatos dos furos são mantidos, mas a distância entre eles é distorcida.
- Classe C6:** Qualquer artefato que não preserve ao mesmo tempo o formato dos furos e a distância entre eles.

Para estimar a incidência das classes, uma amostragem com 800 imagens aleatórias foi inspecionada, inicialmente para os TIs menores que 20ms, conforme mostra o Gráfico 80.

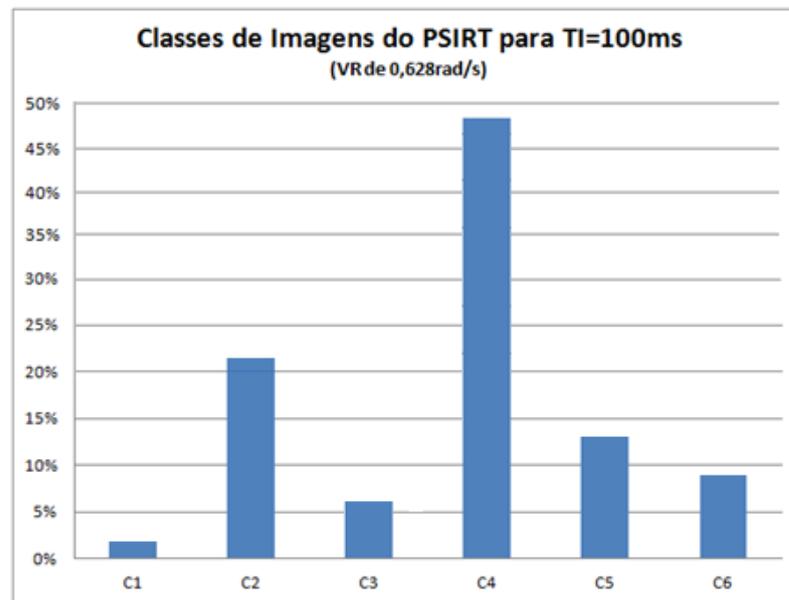
Gráfico 80 – Percentual de incidência das classes de imagem do PSIRT para $TI < 20\text{ms}$.

O Gráfico 80 mostra uma grande incidência de imagens C6, para tempos de integração mais baixos. Tal dado mostra que, para tempos de integração pequenos – independente da velocidade de rotação –, o ruído afeta seriamente o desempenho do método. Para avaliar a incidência das classes no tempo de integração ideal para o PSIRT (20ms), uma nova amostragem foi tomada, com 340 quadros de imagem, conforme mostra o Gráfico 81.

Gráfico 81 – Percentual de incidência das classes de imagem do PSIRT para $TI = 20\text{ms}$.

No Gráfico 81 dá para observar que houve uma significativa melhoria do método, devido maior incidência de C1 e C2, situação favorecida também pela expressiva redução de C6. Entretanto, é preciso também avaliar o PSIRT num valor alto do TI (100ms, por exemplo), para ver o desempenho do método nessa condição. Tal experimento foi feito com 200 quadros de imagem, e pode ser visto no Gráfico 82.

Gráfico 82 – Percentual de incidência das classes de imagem do PSIRT para o TI=100ms.



No Gráfico 82 dá para notar que os conjuntos C1 e C2 diminuíram muito pouco em relação às amostragens anteriores. Entretanto, o conjunto C6 diminuiu significativamente. Isso indica que o PSIRT responde bem aos dados de entrada que apresentam melhores taxas na relação sinal-ruído. Já o grande aumento em C4, indica que as diferenças nas distâncias entre os furos, nesses casos, podem ser desprezadas, ao passo que o ruído é diminuído. Pode-se notar também que à medida que se aumenta o tempo de integração das imagens obtidas, as mesmas se tornam visualmente mais homogêneas, ou seja, a variabilidade visual é muito maior para tempos de integração bem pequenos: 1 e 5ms.

Capítulo VI

6 Considerações Finais

6.1 Experimentos Simulados no Tomógrafo de 1ª Geração

A primeira importante conclusão sobre os experimentos em $3 \times N$ é com relação à predominância na geração de ruído global por parte dos métodos algébricos, mesmo estes métodos tendo preservado, em alguns casos, características importantes das imagens reconstruídas. Em contrapartida, o PSIRT gerou pouco ou quase nenhum ruído nas reconstruções, embora em alguns casos o método não preservou o formato do fantoma original. Essa conclusão foi uma tendência generalizada nos experimentos.

A conclusão sobre a tendência supracitada é que o excesso de ruído global é mais prejudicial à reconstrução do que, por exemplo, reconstruir a imagem sutilmente diferente da original, mas sem ruído. A predominância de ruído global nos métodos algébricos se deu tanto nas imagens binárias, quanto nas não binárias, com maior frequência nestas, por serem mais sensíveis a cores intermediárias (tons de cinza), situações consistentes com os gráficos RMSE e ΔGC gerados com os experimentos.

Outro aspecto importante é também com relação à tolerância dos tons de cinza estabelecida para os experimentos com imagens não binárias. As simulações com imagens não binárias foram feitas para se obter maior poder de avaliação dos desempenhos dos algoritmos, já que um dos parâmetros de ajustes de reconstrução nos métodos algébricos e também no PSIRT, para entradas binárias, é a tolerância utilizada para se determinar na saída o limiar de tom de cinza aceitável para se considerar pertencente ao objeto reconstruído, ou seja, para se transformar pixels com tons acima do limiar em pixels de cor branca.

O que foi observado de uma maneira ampla em relação ao conjunto de simulações com saídas não binárias é que a tolerância ideal para os métodos algébricos variou bastante de um resultado para outro, sugerindo que a escolha de um nível único acarretará em alguns casos cujos resultados não são os melhores que os algoritmos da família ART conseguem produzir para essas situações, principalmente devido a ruídos de ordem global, o que acabou beneficiando o PSIRT nesses casos.

Outra conclusão importante foi à utilização de uma métrica inovadora que é sensível ao formato final de objetos nas imagens, com relação não só a ruídos, mas também com relação aos aglomerados de pixels úteis, independente se predominantemente branco ou cinza, ou mesclados (os aglomerados): a *diferença de coerência geométrica* (ΔGC). A necessidade de se estabelecer tal métrica vem da assunção de que o objeto a ser reconstruído secciona-se numa região contínua, formada neste caso pelos aglomerados de pixels naturais. Nesse contexto, o PSIRT também teve um desempenho satisfatório e até melhor (em alguns casos) do que os métodos algébricos, principalmente em imagens binárias. É uma conclusão consistente com os valores ΔGC calculados nos experimentos, que na maioria das vezes refletiram a inspeção visual das imagens reconstruídas.

Variavelmente um ou outro método não preservou a forma e dimensões ou alguma característica importante do fantoma original, a exemplo do PSIRT nos quadrados binários na configuração 3x15, que não preservou o tamanho de ambos os quadrados; e do MART e SMART na reconstrução do núcleo do anel nas quatro configurações, que quase não reconstruíram tal região das imagens. Da mesma forma, variavelmente um ou outro método distorceu muito algum aspecto da imagem.

Em síntese, devido à diversificação em relação às configurações 3xN e tipos de fantoma, tais variações nos resultados dos métodos já eram esperadas. Entretanto, uma conclusão importante sobre isso é que o PSIRT foi mais regular nas reconstruções, inclusive, em muitos casos, reconstruindo fielmente o fantoma original. Embora os métodos algébricos também tenham produzido reconstruções precisas, o PSIRT foi mais regular nesse contexto.

Por fim, os métodos algébricos tendem a produzir muitos pixels naturais de tons intermediários, à medida que a configuração cresce, ou seja, à medida que a quantidade de trajetórias aumenta, enquanto que o PSIRT, pela sua característica de produzir mais pixels de cor branca, se mantém regular em qualquer configuração. Isso passa a ser uma desvantagem para os métodos algébricos, pois pixels cinzentos muitas vezes são ruídos, ou candidatos a ruídos, embora também possam representar a presença de material.

6.2 Experimentos Simulados Exclusivamente em 3x7

Segue extrato dos resultados dos fantasmas reconstruídos especificamente para 3x7:

- **Barra Vertical Direita:** esse fantoma foi o apogeu do PSIRT no que tange a percepção visual, pois o método produziu fielmente o fantoma original, enquanto que os demais métodos, igualmente, produziram muito ruído do tipo global, e irregularidades no formato geral da barra. As métricas RMSE e ΔGC confirmaram tal distinção, registrando extrema diferença favorável ao PSIRT. No ΔGC , por exemplo, o valor do PSIRT foi ótimo, em comparação com a média para os demais métodos: 0,03 contra 0,225.
- **Anel:** nesse fantoma o PSIRT se mostrou irregular na reconstrução da região do núcleo do anel. Mesmo assim, o método produziu melhores resultados visuais do que os outros métodos. O interessante é que é justamente no núcleo do anel que o PSIRT superou os métodos da família ART: hexágonos de cor preta são mais predominantes no PSIRT nessa região. Com relação ao ruído, só houve um hexágono candidato a ruído local na reconstrução do PSIRT, enquanto nos outros métodos houve vários e espalhados, não apenas ao redor do anel, mas também fora dele, com agravante para hexágonos colados na borda do anel, que certamente são ruídos. As métricas de erro, tanto a RMSE quanto a ΔGC , registraram extrema distinção em favor do PSIRT para esse fantoma.
- **Círculo:** nesse fantoma, o PSIRT obteve melhor resultado visual em relação aos demais métodos. A imagem produzida pelo método é praticamente a mesma do fantoma original. Além disso, o PSIRT não produziu nenhum ruído. Porém, a maior distinção favorável ao PSIRT foi o fato de ter produzido a imagem com praticamente todos os hexágonos na cor branca, com exceção apenas de um hexágono preenchido com cinza, embora localizado dentro da região da imagem, o que garante representar material. Os demais métodos produziram diversos tons de cinza. De certo, tom de cinza não é teoricamente problema algum, pois também representa a presença de material. A questão é que, sendo puramente de cor branca, a garantia de estar representando material é muito elevada, enquanto hexágonos cinzentos podem ser ruídos. De fato, isso ocorreu nos métodos da Família ART, que geraram hexágonos cinza ao redor do círculo, enfim, fora da região da imagem. O cálculo do RMSE também registrou melhor desempenho para o PSIRT, embora com diferenças muito próximas. Já o ΔGC , registrou um desempenho mais acentuado em favor do PSIRT do que o RMSE na reconstrução em questão.

- **Quadrados Diagonais:** nesse fantoma o PSIRT repetiu a eficiência da barra, ou seja, produziu os quadrados quase que fielmente, exceto por um pixel cinza que certamente representa ruído. Ainda assim, trata-se de um ruído local. Em contrapartida, os métodos da família ART produziram muito ruído global, e em regiões muito abrangentes fora das fronteiras da imagem. As métricas RMSE e ΔGC novamente confirmaram a distinção, registrando, desta vez, valores próximos a 0,06 para o PSIRT, e acima de 0,2 para os demais métodos.

Uma última conclusão importante sobre os experimentos específicos em 3×7 é que, as métricas RMSE e ΔGC , serviram para dar sentido às conclusões sobre as formas visuais das reconstruções dos fantasmas. E, com tais métricas, os resultados também foram favoráveis ao PSIRT.

6.3 Experimento Simulado no caso especial da Barra Simples

Esse experimento analisou a robustez do PSIRT. A robustez dos métodos algébricos é algo já conhecido, já esperado. É justamente essa condição de robustez dos métodos algébricos que serviu para dimensionar o quanto o PSIRT também possui essa característica tão importante num método de reconstrução. O PSIRT, então, com esse experimento, demonstrou ser robusto tanto quanto os métodos algébricos.

6.4 Tempo de Reconstrução dos Métodos

No tempo de reconstrução, o método obteve um desempenho muito inferior aos métodos algébricos. Uma solução para manter o PSIRT competitivo nesse aspecto *tempo* em situações da necessidade de muitas partículas é o investimento numa plataforma de hardware apropriada para o método, uma vez que hardware de alta performance é uma realidade concreta e cada vez menos onerosa no contexto tecnológico mundial. Além disso, uma versão do PSIRT rápido é uma questão natural no processo evolutivo do método, o que irá diminuir consideravelmente o tempo de reconstrução em questão, ainda que seja preservada a condição de execução gráfica no método.

6.5 Experimentos do PSIRT com dados reais no Tomógrafo HSGT

Para testar o PSIRT na simulação de uma geometria de tomógrafo mais complexa, foram utilizados dados experimentais produzidos no tomógrafo HSGT por Rachid Maad (MAAD, 2009). Para isso, o PSIRT sofreu uma mudança em relação a sua versão para a simulação do tomógrafo de 1ª geração, mas manteve a lógica de funcionamento de seu algoritmo, sobretudo com relação ao conceito de pixel natural, tratamento de ruído local, configurações de entrada, iterações, alcance do equilíbrio.

A mudança mais visível foi quanto à discretização, que passou a ser estabelecida por partículas atraídas por diversas interseções (2, 3, 4 e 5). Para tais partículas foram definidas cores de acordo com a interseção, e uma abrangência que representa o material, num decaimento normalizado de cores que vai do branco (material) até o preto (fora do material), passando por vários tons de cinza, considerados situações intermediárias (material ou ruído).

O fantoma reconstruído foi o polipropileno de dois furos, objeto que representa bem uma parte da dinâmica de bolhas de gás imersas em óleo dentro de um tubo, cenário semelhante ao que acontece no riser. Os dois parâmetros mais expressivos avaliados nesse contexto foram o tempo de integração (TI) e a velocidade de rotação (VR) do objeto dentro do tomógrafo HSGT. Nos TIs de 1 e 50ms, o PSIRT apresentou bons resultados, o que, de certa forma, já credenciou o método para ser executado nesse tipo de tomógrafo rápido e avançado.

Em seguida, os experimentos passaram a variar o TI e VR para avaliar o comportamento do PSIRT de forma mais abrangente. Os melhores resultados do método foram alcançados no TI de 20ms e na VR de 0,628rad/s. O PSIRT também foi bastante eficiente no TI de 10ms para o mesmo valor VR. Enfim, o PSIRT mostrou eficiência, principalmente nos tempos de 10 e 20ms para todos os valores VR (0,628 – 6,283 – 18,860 – 31,416rad/s). Isso sinaliza que o PSIRT é adequado às diversas rotações do objeto suportadas pelo tomógrafo HSGT.

No planejamento do uso dos dados experimentais decidiu-se utilizar os mais de 70.000 quadros disponíveis, tendo em vista a enorme quantidade de variáveis que contribuem para produzir ruídos. Para se avaliar os resultados da aplicação do algoritmo, devido à quantidade de quadros, decidiu-se produzir uma análise estatística descritiva, onde as imagens reconstruídas foram divididas em 6 classes que compõem uma escala de aproximação à imagem do correspondente fantoma, e cuja aderência às classes foi julgada por simples inspeção visual.

Os parâmetros de distinção das classes foram o “formato dos furos” do polipropileno e a “distância entre esses furos”. Foram várias amostras, tomadas com diferentes TIs e VRs. Da mesma forma, também foram muitas as variações do percentual das classes. Mas o importante é que o PSIRT se manteve eficiente e evolutivo nas reconstruções onde o método sinalizou ser mais eficiente: TI de 20ms e VR de 0,628rad/s.

6.6 Conclusão

O PSIRT reconstruiu satisfatoriamente fantasmas usados comumente em métodos de reconstrução tomográfica do riser tipo FCC. Trata-se de um método inovador, mais uma ramificação dos sistemas de partículas, sistemas estes que já são comprovadamente eficientes em cenários que mapeiam situações similares. A comparação para tal comprovação não foi diminuta, pois envolveu quatro métodos de reconstrução consagrados: ART, MART, SIRT e SMART, além de isso ter sido feito com métricas de erro eficientes, com destaque para a RMSE, mas combinada com a métrica GC, que tem atuação direta no formato geométrico das imagens reconstruídas, e que sinalizou uma satisfatória coerência geométrica ao PSIRT.

Um experimento simulado e específico foi realizado para a mais baixa configuração, a 3x7, por ser de interesse particular para o funcionamento do PSIRT. Em 3x7 o PSIRT superou os demais métodos, muitas das vezes com extrema distinção, a exemplo da Barra Vertical Direita e dos Quadrados Diagonais, sendo este fantoma mais complexo de ser reconstruído.

No contexto visual, o PSIRT revelou possuir apenas ruídos do tipo local, uma distinção altamente favorável, uma vez que os métodos algébricos produzem ruídos do tipo global, que são bem mais difíceis de serem tratados. De fato, as diversas situações com ruídos foram refletidas nos gráficos gerados, principalmente no RMSE.

A possibilidade de tratar ruído do tipo local é um avanço num método de reconstrução. Assim, concluí-se que o PSIRT tem o tratamento de ruído como um diferencial (também) para a reconstrução tomográfica, pelo fato do ruído no método ser do tipo local. Isso impõe flexibilidade e tolerância à falha ao método, aspectos preciosos no PSIRT que, por definição, não precisa ser rigorosamente preciso na reconstrução, bastando apenas reconhecer a região aproximada do objeto a ser reconstruído.

Em extensão à tomografia padrão, o PSIRT provou ser também eficiente quando da simulação do tomógrafo HSGT, um avançado tomógrafo de raios gama para aplicações industriais. O importante aqui é que o PSIRT produziu bons resultados para diversos tempos de integração (TI) e velocidades de rotação (VR) do objeto dentro do tomógrafo, o que credencia o método para ser executado nesse tipo de equipamento.

Finalizando, variavelmente o PSIRT mostrou ser eficiente nas reconstruções de imagens realizadas neste trabalho, no geral com mais distinções favoráveis em relação aos métodos comumente usados nesse tipo de reconstrução, com desempenho satisfatório para variados tipos de fantasmas e condições. A análise subjetiva visual (relativamente), e as métricas de erro RMSE e ΔGC comprovaram isso.

Com relação ao custo operacional da reconstrução, basicamente é o mesmo para todos os métodos, uma vez que a forma de aquisição dos dados é a mesma, em termos de arranjos tomográficos. Ou seja, os métodos funcionam a partir dos dados computados com as atenuações sofridas pelos feixes gama, com cada método tratando esses dados à sua maneira, sem interferir na estrutura física do tomógrafo. Sendo assim, o arranjo físico para a tomografia industrial não é um aspecto decisivo na relação *custo/benefício* no sentido de que irá implicar na escolha de um ou de outro método.

Enfim, o PSIRT se apresenta, pelo menos, como uma alternativa competitiva para a reconstrução tomógrafo industrial.

6.7 Sugestões para Trabalhos Futuros

Seguem algumas das sugestões que podem ser exploradas em trabalho futuros a cerca do PSIRT:

- Produzir avanços no desempenho computacional do PSIRT: uma modelagem computacional baseada puramente em apontadores (alocação dinâmica), por exemplo. Isso vai diminuir significativamente o tempo de execução do método, provavelmente colocando-o no mesmo nível de velocidade de reconstrução dos métodos algébricos;
- Da mesma forma, considerar as formas de paralelização do PSIRT;

- Considerar o uso do PSIRT na produção de sequências de imagens, como na tomografia dinâmica, podendo utilizar a saída de uma reconstrução (conjunto de partículas antes da produção da imagem) como a configuração inicial da próxima reconstrução;
- Estudar os ajustes ideais de parâmetros, com intenção de torná-los adaptativos;
- Estudar de forma mais aprofundada a incorporação de conhecimento a priori, em conexão com o estudo de ajustes de parâmetros;
- Adaptar o PSIRT para admitir sinogramas de objetos cuja área útil é formada de regiões de baixa frequência;
- Estudar a possibilidade de utilizar o resultado do PSIRT como uma máscara para outros métodos e vice-versa, possivelmente com alternância de execução para tentar aliar baixa incidência de ruído local e preservação de forma com a fidelidade nas proporções das regiões do objeto;
- Viabilizar a reconstrução de imagem no PSIRT em dimensão três (3D), para se obter maior clareza e precisão da imagem reconstruída;
- Incorporar no PSIRT algumas funcionalidades que foram feitas de forma externa ao método, a exemplo da geração dos gráficos RSME e ΔGC .
- Executar o PSIRT em modo batch, para também diminuir seu tempo de reconstrução;
- Criar uma métrica ou uma maneira mais precisa de classificar as imagens geradas pelo PSIRT com dados reais no tomógrafo HSGT.

Bibliografia

AGUIAR, W. M. “O Uso de Fontes Alternativas de Energia como Fator de Desenvolvimento Social para Segmentos Marginalizados da Sociedade”. Dissertação de Mestrado. UFRJ. Rio de Janeiro, 2004.

AKAIKE, H. “Akaike’s – An Information Criterion”, 1973.

AMRANI, M.; JAILLET F.; SHARIAT B. “Deformable Objects Modeling and Animation: Application to Organ’s Interactions Simulation”. Journal for Geometry and Graphics, Vol. 4, Nº 2, p. 181–188, 2000.

ANTON, H. “Álgebra Linear Contemporânea”. Bookman. ISBN 0-471-16362-7. Porto Alegre-RS, 2006.

AZZI, M.; TURLIER, P.; BERNAD, J. R. “Mapping Solid Concentration in a Circulating Fluid Bed Using Gammametry”. Powder Technology, 1991.

BLINN, J. F. “Light Reflection Functions for Simulation of Clouds and Dusty Surfaces”. Computer Graphics, Vol. 16, Nº 3, July, 1982.

BOZDOGAN, H. “Akaike's Information Criterion and Recent Developments in Information Complexity”. Journal of Mathematical Psychology 44, 62-91, 2000.

BREEN, D. E.; HOUSE, D. H.; WOZNY, M. J. “A Particle-Based Model for Simulating the Draping Behavior of Woven Cloth”. Textile Research Journal, Vol. 64, Nº 11, p. 663–685, November, 1994.

CALZAVARA, B. “Era do petróleo parece viver seus últimos 50 anos”. Hype Science (hypescience.com/era-do-petroleo-parece-viver-seus-ultimos-50-anos), postado em 25 de abril de 2011.

DANTAS, C. C.; MELO, S. B.; et. al. “The Tomographic Parameters Density, Spatial and Temporal Resolutions are Precisely Determined by Gamma Ray Transmission.” In:

INAC2007 International Nuclear Atlantic Conference, Vol. 1, Santos-SP. Proceedings of the INAC2007 – International Nuclear Atlantic Conference, 2007.

FREIRE, L. “Reconstrução Tomográfica em TC e em Medicina Nuclear”. IST. Curso de Medicina Biomédica. Acessado em nebm.ist.utl.pt/repositorio/download/923/8, Lisboa, Portugal, 2008.

FRIDA. “Métodos de Reconstrução Tomográfica de Imagens de SPECT”. Projeto FRIDA. PUCRS. Porto Alegre, 2006.

GAZZANI, M. H. “Reconstruções de Imagens a Partir de Projeções Paralelas”. Dissertação de Mestrado. UFU, Uberlândia-MG, 1999.

GILBERT, P. “Iterative methods for the three-dimensional reconstruction of an object from projections”. *Journal of Theoretical Biology*, Volume 36, Issue 1, July 1972, p. 105–117.

GORDON, R.; BENDER, R; HERMAN, G. T. “Algebraic Reconstruction Techniques (ART) for Three Dimensional Electron and X-Ray Photography”. *J fo Theor. Biol.* Vol.29, p. 471–481, 1970.

HOFFMANN, E. C. “Estudo de Parâmetros Físicos Envolvidos com a Qualidade da Imagem em Tomografia Computadorizada de Feixe Cônico”. Dissertação de Mestrado, publicada na *Revista da Graduação*, Vol. 4, Nº 1, 2011-22. ISBN 1983-1374. PUCRS, Faculdade de Física, Porto Alegre, 2010.

HUG, J.; BRECHDÜHLER, C.; SZÉKELY, G. “A Particle System for Robust Semi-Automatic Segmentation”. BIWI, TR-185, March, 1999.

IAEA: Industrial Process Gamma Tomography. International Atomic Energy Agency – IAEA (TEC-DOC 1598), maio de 2008.

IUSEM, A. N.; PIERRO, A. R. “Fundamentos Matemáticos da Tomografia Computadorizada: Métodos de Expansão em Série”. *Revista Matemática Universitária*, Nº 12, IMPA/UNICAMP, dezembro de 1990, p. 57–85.

KACZMARZ, S. "Angentührte Auflösung von Systemen linearer Gleichungen", Bull. Int. Acad. Pol. Sci. Lett., A, vol. 35, p. 355 -357, 1937.

KAK, A. C.; SLANEY, M. "Principles of Computerized Tomographic Imaging". Philadelphia: SIAM, 342 p., 2001.

KELLER, C. "Radioquímica". Editora UFPE. Recife, 1981, 348 páginas.

LEVY, M.; PERDEW. J. P.; SAHNI, V. "Exact Differential Equation for Density and Ionization Energy of a Many-Particle System". Physical Review, Vol. 30, Nº 5, November, 1984.

MAAD, R. "Design Optimization of High Speed Gamma-Ray Tomography". Ph.D. Thesis – University of Bergen, 2009.

MARKS, J; et al. "A General Approach to Setting Parameters for Computer Graphics and Animation". Mitsubishi Electric Research Laboratory – MERL (www.merl.com), TR 97-14, 1997.

MELO, S. B.; DANTAS, C. C.; et al. "Polynomial Reconstruction of Radial Catalyst Concentration Distribution in an Experimental Type FCC Riser for a Low Number of Transversal Projections in Gamma Ray Tomography". In: INAC2007 International Nuclear Atlantic Conference, Vol. 1, Santos-SP. Proceedings of the INAC2007 – International Nuclear Atlantic Conference, 2007.

MENDES, M. F. A; et al. "Espectrofotometria". UFRGS. Acessado em www.ufrgs.br/leo/site_espec/index.html, postado em março de 2012.

MILLER, G. "A Connected Particle System for Animating Viscous Fluids". Computer & Graphics, Vol.13, Nº 3, p. 305–309, 1989.

MUSIAL, J. E. "Técnicas de Reconstrução Algébricas Aplicadas em Tomografia Computadorizada". Dissertação de Mestrado. UFPR, Curitiba, 2010.

OLIVEIRA, E. F. “Reconstrução Tomográfica com Superfícies B-Splines”. Dissertação de Mestrado. UFPE, 2011.

PETROBRÁS. “Atuação no Pré-Sal”. Site Oficial da Petrobrás. Disponível em www.petrobras.com.br/pt/energia-e-tecnologia/fontes-de-energia/petroleo/presal, acessado em 02 de março de 2012 (PETROBRÁS, 2012-a).

PETROBRÁS. “Exploração e Produção de Petróleo e Gás”. Site Oficial da Petrobrás. Disponível em www.petrobras.com.br/pt/energia-e-tecnologia/fontes-de-energia/petroleo/presal, acessado em 02 de março de 2012 (PETROBRÁS, 2012-b).

PETROBRÁS. “Refino de Petróleo e Gás”. Site Oficial da Petrobrás. Disponível em www.petrobras.com.br/pt/quem-somos/perfil/atividades/refino, acessado em 02 de março de 2012 (PETROBRÁS, 2012-c).

RAPARIA, D.; ALESSI, J.; KPONOU, A. “The Algebraic Reconstruction Technique (ART)”. AGS Department. Brookhaven National Lab, Upton, NY 11973, Id: 0-7803-4376-X, IEEE, USA, 1998.

REEVES, W. T. “A Technique for Modeling a Class of Fuzzy Objects”. ACM Transactions on Graphics, Vol. 2, N° 2, p. 91–108, April, 1983.

RODRIGUES, T. T. T. “Estudo da Física Médica – Interação da Radiação com a Matéria no Radiodiagnóstico – Parte 4”. Acessado em www.ensinodefisicamedica.blogspot.com, postado em 22 de maio de 2011.

ROSA, B. “Pré-sal: Brasil tem potencial para explorar 55 bilhões de barris”. Globo.com (oglobo.globo.com/economia/pre-sal-brasil-tem-potencial-para-explorar-55-bilhoes-de-barris-6097763), postado em 16 de setembro de 2012.

SALINA, F. V. “Reconstrução Tomográfica de Imagens Utilizando Técnicas POCS Sequenciais e Paralelas”. Dissertação de Mestrado. UFSCar, São Carlos-SP, 2001.

SALVADOR, P. A. V. “Análise de Sistemas Multifásicos Utilizando Tomografia Computadorizada Gama Monoenergética e Polienergética”. Tese de Doutorado. Instituto de Pesquisas Energéticas e Nucleares, São Paulo, 2008.

SANCHES, I. J. “Compressão sem Perdas de Projeções de Tomografia Computadorizada Usando a Transformada Wavelet”. Dissertação de Mestrado. UFPR, Curitiba, 2011.

SANTOS, R. J. “Mudança de Coordenadas – Rotação”. Acessado em www.mat.ufmg.br/~regi/gaalt/mudco.pdf. ICEX. UFMG, 13 de dezembro de 2001.

SILVA, I. C. S. “Avaliação da Qualidade de Imagens Médicas Geradas por *Ray Casting*”. Dissertação de Mestrado”. UFRGS, Porto Alegre, 2003.

SILVA, U. R. “Influência da Formação de Agregados e de Nanopartículas de Ouro nas Propriedades Microscópicas de Corante Rodamina 6G”. Dissertação de Mestrado. UFAL, 2010.

SIMS, K. “Particle Animation and Rendering Using Data Parallel Computation”. *Computer Graphics*, Vol. 24, N° 4, August, 1990.

STEIGLEDER, M. “Integração de Sistemas de Partículas com Detecção de Colisão em Ambientes de Ray Trancing”. Dissertação de Mestrado. UFRGS, Agosto, 1997.

SUBBARAO, P. M. V.; MUNSHI, P.; MURALIDHAR, F. “Performance of Iterative Tomographic Algorithms Applied to non-Destructive Evaluation with Limited Data”. *NDT&E International*, Vol. 30, N° 6, p. 359–370, 1997.

SZELISKI, R.; TONNESEN, D. “Oriented Particle Systems Surface Modeling with Oriented Particle Systems”. Cambridge Research Lab, CRL-91/14, December, 1991.

SZKLO, A. S.; ULLER, V. C. “Fundamentos do Refino de Petróleo”, Interciência, 2ª ed. ISBN 8571932042, Rio de Janeiro, 2008.

THOMAS, J. E. “Fundamentos de Engenharia de Petróleo”. Interciência, 272 páginas, 2ª ed. ISBN 8571930996, Rio de Janeiro, 2004.

VASCONCELOS, G. V.; et al. “A Comparison of PSIRT Against Algebraic Methods in Industrial Configurations with Natural Pixels”. ISBN 978-85-99141-04-5. 2011 International Nuclear Atlantic Conference – INAC 2011. Belo Horizonte, MG, Brazil, October 24-28, 2011.

VASCONCELOS, W. E., et al. “Determinação do coeficiente de atenuação-gama de um catalisador de craqueamento catalítico fluido”. Revista Química & Tecnologia. Universidade Católica de Pernambuco. Ano 2 – nº 1, jan/jun 2003.

VERHOEVEN, D. “Limited-data Computed Tomography Algorithms for the Physical Sciences”. Applied Optics, julho de 1993.

VERÍSSIMO, A. S. “Aplicação de Métodos Tomográficos na Reconstrução Bidimensional de Propriedades de Chamas”. INPE. Dissertação de Mestrado. São José dos Campos-SP, 2006.

WITKIN, A. “Particle System Dynamics”. School of Computer Science Carnegie Mellon University. Physically Based Modeling, SIGGRAPH, 1997.

YAMAOKA, K.; NAKAGAWA, T.; UNO, T. “Application of Akaike's information criterion (AIC) in the evaluation of linear pharmacokinetic equations”. Journal of Pharmacokinetics and Biopharmaceutics, Volume 6, Issue 2, pp 165-175, April 1978.

APÊNDICE

Geovane Vitor Vasconcelos

PSIRT:

A Técnica Iterativa de Reconstrução Tomográfica Baseada em Sistemas de Partículas

IMPLEMENTAÇÃO DO PSIRT

1. Microsoft Visual Studio 2010

O PSIRT foi implementado no Microsoft Visual Studio 2010 (MVS 2010).



Símbolo do MVS 2010.

Fonte da imagem: www.microfost.com.

Trata-se de uma plataforma de desenvolvimento completa, que engloba vários recursos modernos e diversificados de programação, tais como paradigmas, ferramentas, componentes de software, *frameworks*, bibliotecas, linguagens de programação, entre outros. O MVS 2010 dá suporte para a construção de diversos tipos de sistemas, desde complexas aplicações *web*, até grandes sistemas de gerenciamento e automação. Os principais paradigmas de programação viabilizados pelo MVS 2010 são os seguintes:

- Procedimental;
- Gráfica, também chamada de programação visual;
- Orientada a objetos;
- Dirigida por eventos.
- Baseada em componentes de software;
- Baseada em *frameworks*.
- De baixo nível, também chamada de programação em linguagem de máquina;

Além de uma plataforma com diversos recursos, o MVS 2010 preserva a colaboração dinâmica entre os elementos usados no desenvolvimento dos sistemas, tanto com relação à estrutura, quanto com relação à funcionalidade. Por exemplo, um “Componente de Software” pode responder a outro “Componente” através de um “Evento”, sem haver dependências entre eles, apenas colaboração. Enfim, o ambiente fornece conceitos de tecnologia e engenharia de software avançados, tais como modularidade, flexibilidade, visibilidade, independência, colaboração, robustez, integração, portabilidade. Tudo isso confere ao MVS 2010 a possibilidade de construir, de forma eficiente e rápida, sistemas para arquiteturas tais como Linux, Windows e equipamentos de diversas naturezas.

Para sistemas científicos, como é o caso do PSIRT, o MVS 2010 possui recursos facilitadores, além de preservar linguagens mais apropriadas para esse tipo de desenvolvimento, a exemplo das linguagens C e C++. No caso do PSIRT, a linguagem usada foi a C# (*C sharp*), a versão do C orientado a objetos, incluído pela Microsoft na sua plataforma de desenvolvimento chamada de “.NET”. O C# é uma evolução do C++, e inclui recursos de linguagens antigas, a exemplo da linguagem Object Pascal, além de dá suporte a aplicações modernas, tais como aplicações Java. A versão da plataforma usada no desenvolvimento do PSIRT foi a “.NET Framework 4.5”, e a linguagem usada foi a C#. A figura que se segue mostra os principais recursos dessa versão .NET, inclusive o C#, na sua versão atual, a 5.0.

.NET Framework 4.5

Windows Presentation Foundation

- Built-in Ribbon controls
- Databinding improvements
- Ability to add breakpoints to databindings
- Data source change aware views (Live Shaping)
- Validation improvements
- Improved legacy UI integration
- Dispatcher improvements
- Speed-up of large data sets

Windows 8 support

- Support for Windows Runtime (WinRT)
- .NET Profile for Metro-style apps
- Improved support for sharing DLLs between .NET profiles

ASP.NET

- Built-in providers now natively support SQL Azure
- ASP.NET MVC 4: Async controllers, Built-in mobile templates + jQuery Mobile support, Alternate views (e.g. print version, mobile site), Support for Recipes: intelligent codegen
- ASP.NET Web Pages 2: New site templates, Versatile validation support, Support for OAuth and OpenID, Built-in map embedding tools, supports Google, Bing and others
- Web Forms: Strongly typed data binding, MVC-like support for Models, HTML encoded binding expressions, HTM5 support, Control support for new semantic elements, Validator and UpdatePanel now support new HTML5 elements
- Tooling: IntelliSense improvements, More JavaScript support, VS Express used by default, New templates and snippets
- Asynchronous pipeline support (Response, Request, Handlers)
- Performance improvements: Multicore JIT, 35% faster startup, memory optimizations, assembly sharing between sites, pre-fetch support
- Request validation improvements: AntiXSS built-in, validation usable per field

Windows Communication Foundation

- Support for UDP multicast channels
- TCP channels now work with partial trust
- Asynchronous operations
- Streaming improvements
- Simplified configuration (agent)
- Can now generate service stubs from WSDL
- WebSocket support

Windows Workflow Foundation

- C# Expressions
- State machine workflows are back
- Workflow versioning
- Code-first activity design
- Faster execution
- Designer usability improvements

Managed Extensibility Framework 2.0

- Debugging improvements
- Support for explicit bindings between objects
- Support for binding POCO: no more attribute requirements

ADO.NET

- Sparse columns support improved (SQL Server)
- Passwords are now stored encrypted
- Asynchronous operations
- SQL Express LocalDB: New light version of SQL Express for developer use. Supported in .NET 4.5, separate patch for 4.0 is coming
- SQL Server 2012 ("Denali") Support: High Availability support on connection string level, Fast failover across multiple subnets, Support for new spatial data types (polygons, arcs etc.)
- Entity Framework 4.5: Enumeration support, Migrations for schema changes, Designer improvements
- Spatial data type support
- Table-valued function support
- Multi-result spoc support
- Multiple diagrams per model
- Code-first support
- Auto-compiled LINQ queries

Base Class Library

- Usability improved for WeakReferences and Streams
- Key interfaces now have async versions
- New ArraySegment and ReadOnlyDictionary classes
- Support for CLR objects over 2 GB in size
- Resource file management performance improved
- Unicode support for console applications
- Task Parallel Library: Task thread controls improved: Task.WaitAll, WaitAny, various timeout primitives available, TPL Dataflow: Tools for parallel data flow processing

C# 5.0

- Support for async programming: async and await keywords
- Methods can access call site info as parameters (CallerInfo)

Visual Basic 11

- Iterator implementations (Yield)
- Async ja Await equal to C#
- Global keyword for namespace handling
- Call Hierarchy view available

Visual C++ 11

- C++11 standard support improved
- Auto-vectorization and parallelization of loops
- Support for controlling GPU-driven processing (C++ AMP)

Wait, there's more!

- Visual Studio "11" is also coming, including support for .NET 4.5, Windows 8 and more. There are plenty of other new goodies, too.
- Also, .NET team is working on Project Roslyn, which enables interesting scenarios for integrating the C# compiler to your own apps. Roslyn is in CTP, but will ship post-4.5.

Legend:

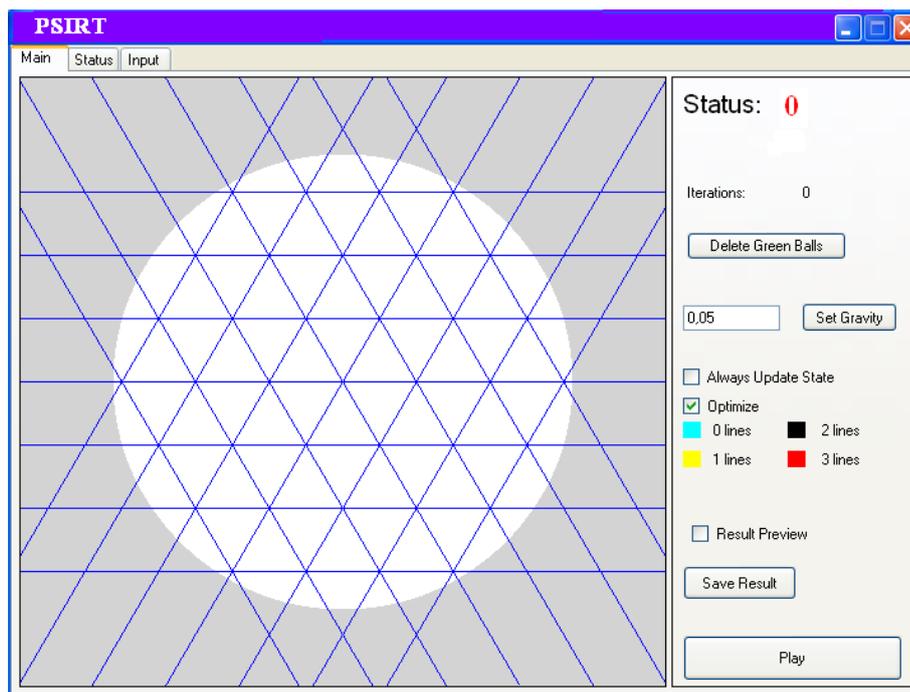
- Asynchronous support
- Performance improvement
- Significant new feature

Information based on public sources available in October 2011. Changes before RTM are likely. Composed by Avni Heilbrunn. Thanks to Sami Pihlami and Jon Aronson for the data. www.microsoft.com

Principais recursos da plataforma “.NET Framework 4.5” do MVS 2010.

Fonte da imagem: www.microsoft.com.

Uma decisão importante no desenvolvimento é quanto a qual recurso gráfico usar para a criação das interfaces gráficas do sistema. A escolha do tipo de interface GUI (*Graphical User Interface*), geralmente é feita de acordo com a característica do ambiente computacional onde o sistema irá rodar. Como o PSIRT foi implementado para ser executado no ambiente Windows, o recurso gráfico usado foi o “Windows Form”, provido pelo .NET Framework. A figura que se segue mostra uma das telas do PSIRT feita com o Windows Form.



Tela principal do PSIRT criado com a ferramenta GUI do .NET Framework “Windows Form”.

A maneira padrão de se construir sistemas com o MVS 2010 é criando um projeto. Um projeto representa uma estrutura organizada para manter os elementos do programa, tais como objetos, funções, variáveis, classes e bibliotecas de classes, arquivos, pastas, banco de dados. Tal estrutura facilita o desenvolvimento, futuras necessidades de alteração, entendimento do código. Como o PSIRT foi codificado em C#, uma linguagem orientada a objetos, o principal elemento da estrutura do programa são as classes. Uma classe representa um conjunto de objetos de características comuns. Assim, classes tais como “Projeção”, “Trajetória” e “Partícula” são elementos de presença garantida na estrutura computacional do PSIRT.

2. Código-fonte do PSIRT

Segue código fonte comentado dos principais elementos do PSIRT:

// PANEL PRINCIPAL

```
namespace Tomografia
{
    partial class Panel
    {
        /// <summary>
        /// Required designer variable.
        /// </summary>
        private System.ComponentModel.IContainer components = null;

        /// <summary>
        /// Clean up any resources being used.
        /// </summary>
        /// <param name="disposing">true if managed resources should be disposed; otherwise,
        false.</param>
        protected override void Dispose(bool disposing)
        {
            if (disposing && (components != null))
            {
                components.Dispose();
            }
            base.Dispose(disposing);
        }

        #region Windows Form Designer generated code

        /// <summary>
        /// Required method for Designer support - do not modify
        /// the contents of this method with the code editor.
        /// </summary>
        private void InitializeComponent()
        {
            this.components = new System.ComponentModel.Container();
            this.splitContainer1 = new System.Windows.Forms.SplitContainer();
        }
    }
}
```

```

this.doubleBufferPanel1 = new WindowsFormsApplication1.DoubleBufferPanel();
this.previewResult = new System.Windows.Forms.CheckBox();
this.panel4 = new System.Windows.Forms.Panel();
this.panel3 = new System.Windows.Forms.Panel();
this.panel2 = new System.Windows.Forms.Panel();
this.panel1 = new System.Windows.Forms.Panel();
this.label4 = new System.Windows.Forms.Label();
this.label3 = new System.Windows.Forms.Label();
this.label2 = new System.Windows.Forms.Label();
this.label1 = new System.Windows.Forms.Label();
this.updateState = new System.Windows.Forms.CheckBox();
this.graviyText = new System.Windows.Forms.TextBox();
this.gButton = new System.Windows.Forms.Button();
this.result = new System.Windows.Forms.Button();
this.deleteMarkedBalls = new System.Windows.Forms.Button();
this.playStopButton = new System.Windows.Forms.Button();
this.statusLabel = new System.Windows.Forms.Label();
this.statusTextLabel = new System.Windows.Forms.Label();
this.refresh = new System.Windows.Forms.Timer(this.components);
this.tabPanel = new System.Windows.Forms.TabControl();
this.mainPage = new System.Windows.Forms.TabPage();
this.statusPage = new System.Windows.Forms.TabPage();
this.statusBox = new System.Windows.Forms.ListBox();
this.input = new System.Windows.Forms.TabPage();
this.inputTextBox = new System.Windows.Forms.TextBox();
this.save = new System.Windows.Forms.Button();
this.splitContainer1.Panel1.SuspendLayout();
this.splitContainer1.Panel2.SuspendLayout();
this.splitContainer1.SuspendLayout();
this.tabPanel.SuspendLayout();
this.mainPage.SuspendLayout();
this.statusPage.SuspendLayout();
this.input.SuspendLayout();
this.SuspendLayout();
//
// splitContainer1
//
this.splitContainer1.BorderStyle = System.Windows.Forms.BorderStyle.FixedSingle;
this.splitContainer1.Dock = System.Windows.Forms.DockStyle.Fill;
this.splitContainer1.ForeColor = System.Drawing.SystemColors.ControlLightLight;
this.splitContainer1.Location = new System.Drawing.Point(3, 3);
this.splitContainer1.Name = "splitContainer1";
//
// splitContainer1.Panel1
//

```

```

        this.splitContainer1.Panel1.BackColor = System.Drawing.SystemColors.ControlLightLight;
        this.splitContainer1.Panel1.Controls.Add(this.doubleBufferPanel1);
        this.splitContainer1.Panel1.Paint += new
System.Windows.Forms.PaintEventHandler(this.splitContainer1_Panel1_Paint_1);
        //
        // splitContainer1.Panel2
        //
        this.splitContainer1.Panel2.Controls.Add(this.previewResult);
        this.splitContainer1.Panel2.Controls.Add(this.panel4);
        this.splitContainer1.Panel2.Controls.Add(this.panel3);
        this.splitContainer1.Panel2.Controls.Add(this.panel2);
        this.splitContainer1.Panel2.Controls.Add(this.panel1);
        this.splitContainer1.Panel2.Controls.Add(this.label4);
        this.splitContainer1.Panel2.Controls.Add(this.label3);
        this.splitContainer1.Panel2.Controls.Add(this.label2);
        this.splitContainer1.Panel2.Controls.Add(this.label1);
        this.splitContainer1.Panel2.Controls.Add(this.updateState);
        this.splitContainer1.Panel2.Controls.Add(this.graviyText);
        this.splitContainer1.Panel2.Controls.Add(this.gButton);
        this.splitContainer1.Panel2.Controls.Add(this.result);
        this.splitContainer1.Panel2.Controls.Add(this.deleteMarkedBalls);
        this.splitContainer1.Panel2.Controls.Add(this.playStopButton);
        this.splitContainer1.Panel2.Controls.Add(this.statusLabel);
        this.splitContainer1.Panel2.Controls.Add(this.statusTextLabel);
        this.splitContainer1.Size = new System.Drawing.Size(701, 483);
        this.splitContainer1.SplitterDistance = 507;
        this.splitContainer1.TabIndex = 0;
        //
        // doubleBufferPanel1
        //
        this.doubleBufferPanel1.BackColor = System.Drawing.Color.White;
        this.doubleBufferPanel1.Dock = System.Windows.Forms.DockStyle.Fill;
        this.doubleBufferPanel1.ForeColor = System.Drawing.SystemColors.ControlLight;
        this.doubleBufferPanel1.Location = new System.Drawing.Point(0, 0);
        this.doubleBufferPanel1.Name = "doubleBufferPanel1";
        this.doubleBufferPanel1.Size = new System.Drawing.Size(505, 481);
        this.doubleBufferPanel1.TabIndex = 0;
        this.doubleBufferPanel1.Paint += new
System.Windows.Forms.PaintEventHandler(this.doubleBufferPanel1_Paint);
        this.doubleBufferPanel1.MouseClick += new
System.Windows.Forms.MouseEventHandler(this.doubleBufferPanel1_MouseClick);
        //
        // previewResult
        //
        this.previewResult.AutoSize = true;

```

```

this.previewResult.ForeColor = System.Drawing.SystemColors.ControlText;
this.previewResult.Location = new System.Drawing.Point(8, 352);
this.previewResult.Name = "previewResult";
this.previewResult.Size = new System.Drawing.Size(96, 17);
this.previewResult.TabIndex = 17;
this.previewResult.Text = "Result preview";
this.previewResult.UseVisualStyleBackColor = true;
this.previewResult.CheckedChanged += new
System.EventHandler(this.previewResult_CheckedChanged);
//
// panel4
//
this.panel4.BackColor = System.Drawing.Color.Red;
this.panel4.Location = new System.Drawing.Point(90, 295);
this.panel4.Name = "panel4";
this.panel4.Size = new System.Drawing.Size(14, 13);
this.panel4.TabIndex = 16;
//
// panel3
//
this.panel3.BackColor = System.Drawing.Color.Orange;
this.panel3.Location = new System.Drawing.Point(90, 272);
this.panel3.Name = "panel3";
this.panel3.Size = new System.Drawing.Size(14, 13);
this.panel3.TabIndex = 15;
//
// panel2
//
this.panel2.BackColor = System.Drawing.Color.Yellow;
this.panel2.Location = new System.Drawing.Point(8, 295);
this.panel2.Name = "panel2";
this.panel2.Size = new System.Drawing.Size(14, 13);
this.panel2.TabIndex = 14;
//
// panel1
//
this.panel1.BackColor = System.Drawing.Color.Chartreuse;
this.panel1.Location = new System.Drawing.Point(8, 272);
this.panel1.Name = "panel1";
this.panel1.Size = new System.Drawing.Size(14, 13);
this.panel1.TabIndex = 13;
//
// label4
//
this.label4.AutoSize = true;

```

```
this.label4.ForeColor = System.Drawing.SystemColors.ControlText;
this.label4.Location = new System.Drawing.Point(113, 295);
this.label4.Name = "label4";
this.label4.Size = new System.Drawing.Size(37, 13);
this.label4.TabIndex = 12;
this.label4.Text = "3 lines";
//
// label3
//
this.label3.AutoSize = true;
this.label3.ForeColor = System.Drawing.SystemColors.ControlText;
this.label3.Location = new System.Drawing.Point(113, 272);
this.label3.Name = "label3";
this.label3.Size = new System.Drawing.Size(37, 13);
this.label3.TabIndex = 11;
this.label3.Text = "2 lines";
//
// label2
//
this.label2.AutoSize = true;
this.label2.ForeColor = System.Drawing.SystemColors.ControlText;
this.label2.Location = new System.Drawing.Point(28, 295);
this.label2.Name = "label2";
this.label2.Size = new System.Drawing.Size(37, 13);
this.label2.TabIndex = 10;
this.label2.Text = "1 lines";
//
// label1
//
this.label1.AutoSize = true;
this.label1.ForeColor = System.Drawing.SystemColors.ControlText;
this.label1.Location = new System.Drawing.Point(28, 272);
this.label1.Name = "label1";
this.label1.Size = new System.Drawing.Size(37, 13);
this.label1.TabIndex = 9;
this.label1.Text = "0 lines";
//
// updateState
//
this.updateState.AutoSize = true;
this.updateState.ForeColor = System.Drawing.SystemColors.ControlText;
this.updateState.Location = new System.Drawing.Point(8, 237);
this.updateState.Name = "updateState";
this.updateState.Size = new System.Drawing.Size(125, 17);
this.updateState.TabIndex = 8;
```

```

        this.updateState.Text = "Always Update State";
        this.updateState.UseVisualStyleBackColor = true;
        this.updateState.CheckedChanged += new
System.EventHandler(this.updateState_CheckedChanged);
        //
        // graviyText
        //
        this.graviyText.Location = new System.Drawing.Point(8, 182);
        this.graviyText.Name = "graviyText";
        this.graviyText.Size = new System.Drawing.Size(76, 20);
        this.graviyText.TabIndex = 7;
        //
        // gButton
        //
        this.gButton.ForeColor = System.Drawing.SystemColors.ControlText;
        this.gButton.Location = new System.Drawing.Point(90, 180);
        this.gButton.Name = "gButton";
        this.gButton.Size = new System.Drawing.Size(75, 23);
        this.gButton.TabIndex = 6;
        this.gButton.Text = "Set Gravity";
        this.gButton.UseVisualStyleBackColor = true;
        this.gButton.Click += new System.EventHandler(this.gButton_Click);
        //
        // result
        //
        this.result.ForeColor = System.Drawing.SystemColors.ControlText;
        this.result.Location = new System.Drawing.Point(8, 386);
        this.result.Name = "result";
        this.result.Size = new System.Drawing.Size(89, 27);
        this.result.TabIndex = 5;
        this.result.Text = "Save Result";
        this.result.UseVisualStyleBackColor = true;
        this.result.Click += new System.EventHandler(this.result_Click);
        //
        // deleteMarkedBalls
        //
        this.deleteMarkedBalls.ForeColor = System.Drawing.SystemColors.ControlText;
        this.deleteMarkedBalls.Location = new System.Drawing.Point(8, 122);
        this.deleteMarkedBalls.Name = "deleteMarkedBalls";
        this.deleteMarkedBalls.Size = new System.Drawing.Size(125, 22);
        this.deleteMarkedBalls.TabIndex = 4;
        this.deleteMarkedBalls.Text = "Delete green balls";
        this.deleteMarkedBalls.UseVisualStyleBackColor = true;
        this.deleteMarkedBalls.Click += new System.EventHandler(this.deleteMarkedBalls_Click);

```

```

//
// playStopButton
//
this.playStopButton.ForeColor = System.Drawing.SystemColors.ControlText;
this.playStopButton.Location = new System.Drawing.Point(8, 441);
this.playStopButton.Name = "playStopButton";
this.playStopButton.Size = new System.Drawing.Size(172, 36);
this.playStopButton.TabIndex = 2;
this.playStopButton.Text = "Play";
this.playStopButton.UseVisualStyleBackColor = true;
this.playStopButton.Click += new System.EventHandler(this.playStopButton_Click);
//
// statusLabel
//
this.statusLabel.AutoSize = true;
this.statusLabel.Font = new System.Drawing.Font("Microsoft Sans Serif", 15F,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)0));
this.statusLabel.ForeColor = System.Drawing.SystemColors.ControlText;
this.statusLabel.Location = new System.Drawing.Point(74, 37);
this.statusLabel.Name = "statusLabel";
this.statusLabel.Size = new System.Drawing.Size(23, 25);
this.statusLabel.TabIndex = 1;
this.statusLabel.Text = "0";
//
// statusTitleLabel
//
this.statusTitleLabel.AutoSize = true;
this.statusTitleLabel.Font = new System.Drawing.Font("Microsoft Sans Serif", 15F,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)0));
this.statusTitleLabel.ForeColor = System.Drawing.SystemColors.ControlText;
this.statusTitleLabel.Location = new System.Drawing.Point(3, 8);
this.statusTitleLabel.Name = "statusTitleLabel";
this.statusTitleLabel.Size = new System.Drawing.Size(74, 25);
this.statusTitleLabel.TabIndex = 0;
this.statusTitleLabel.Text = "Status:";
//
// refresh
//
this.refresh.Enabled = true;
this.refresh.Interval = 20;
this.refresh.Tick += new System.EventHandler(this.refresh_Tick);
//
// tabPage1
//
this.tabPage1.Controls.Add(this.mainPage);

```

```

this.tabPanel.Controls.Add(this.statusPage);
this.tabPanel.Controls.Add(this.input);
this.tabPanel.Dock = System.Windows.Forms.DockStyle.Fill;
this.tabPanel.Location = new System.Drawing.Point(0, 0);
this.tabPanel.Name = "tabPanel";
this.tabPanel.SelectedIndex = 0;
this.tabPanel.Size = new System.Drawing.Size(715, 515);
this.tabPanel.TabIndex = 1;
//
// mainPage
//
this.mainPage.Controls.Add(this.splitContainer1);
this.mainPage.Location = new System.Drawing.Point(4, 22);
this.mainPage.Name = "mainPage";
this.mainPage.Padding = new System.Windows.Forms.Padding(3);
this.mainPage.Size = new System.Drawing.Size(707, 489);
this.mainPage.TabIndex = 0;
this.mainPage.Text = "Main";
this.mainPage.UseVisualStyleBackColor = true;
//
// statusPage
//
this.statusPage.AutoScroll = true;
this.statusPage.Controls.Add(this.statusBox);
this.statusPage.Location = new System.Drawing.Point(4, 22);
this.statusPage.Name = "statusPage";
this.statusPage.Padding = new System.Windows.Forms.Padding(3);
this.statusPage.Size = new System.Drawing.Size(707, 489);
this.statusPage.TabIndex = 1;
this.statusPage.Text = "Status";
this.statusPage.UseVisualStyleBackColor = true;
//
// statusBox
//
this.statusBox.Dock = System.Windows.Forms.DockStyle.Fill;
this.statusBox.FormattingEnabled = true;
this.statusBox.Location = new System.Drawing.Point(3, 3);
this.statusBox.Name = "statusBox";
this.statusBox.Size = new System.Drawing.Size(701, 472);
this.statusBox.TabIndex = 0;
//
// input
//
this.input.Controls.Add(this.inputTextBox);
this.input.Controls.Add(this.save);

```

```

this.input.Location = new System.Drawing.Point(4, 22);
this.input.Name = "input";
this.input.Padding = new System.Windows.Forms.Padding(3);
this.input.Size = new System.Drawing.Size(707, 489);
this.input.TabIndex = 2;
this.input.Text = "Input";
this.input.UseVisualStyleBackColor = true;
//
// inputTextBox
//
this.inputTextBox.Font = new System.Drawing.Font("Arial", 12F,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)0));
this.inputTextBox.Location = new System.Drawing.Point(3, 3);
this.inputTextBox.Multiline = true;
this.inputTextBox.Name = "inputTextBox";
this.inputTextBox.ScrollBars = System.Windows.Forms.ScrollBars.Vertical;
this.inputTextBox.Size = new System.Drawing.Size(696, 428);
this.inputTextBox.TabIndex = 7;
//
// save
//
this.save.Location = new System.Drawing.Point(581, 437);
this.save.Name = "save";
this.save.Size = new System.Drawing.Size(118, 46);
this.save.TabIndex = 6;
this.save.Text = "Save";
this.save.UseVisualStyleBackColor = true;
this.save.Click += new System.EventHandler(this.save_Click);
//
// Panel
//
this.AutoScaleDimensions = new System.Drawing.SizeF(6F, 13F);
this.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font;
this.ClientSize = new System.Drawing.Size(715, 515);
this.Controls.Add(this.tabPanel);
this.FormBorderStyle = System.Windows.Forms.FormBorderStyle.FixedSingle;
this.MaximizeBox = false;
this.Name = "Panel";
this.Text = "Tomografia";
this.Load += new System.EventHandler(this.Panel_Load);
this.splitContainer1.Panel1.ResumeLayout(false);
this.splitContainer1.Panel2.ResumeLayout(false);
this.splitContainer1.Panel2.PerformLayout();
this.splitContainer1.ResumeLayout(false);

```

```
    this.tabPanel.ResumeLayout(false);  
    this.mainPage.ResumeLayout(false);  
    this.statusPage.ResumeLayout(false);  
    this.input.ResumeLayout(false);  
    this.input.PerformLayout();  
    this.ResumeLayout(false);  
  
}
```

```
#endregion
```

```
private System.Windows.Forms.SplitContainer splitContainer1;  
private System.Windows.Forms.Button playStopButton;  
private System.Windows.Forms.Label statusLabel;  
private System.Windows.Forms.Label statusTitleLabel;  
private WindowsFormsApplication1.DoubleBufferPanel doubleBufferPanel1;  
private System.Windows.Forms.Timer refresh;  
private System.Windows.Forms.TabControl tabPanel;  
private System.Windows.Forms.TabPage mainPage;  
private System.Windows.Forms.TabPage statusPage;  
private System.Windows.Forms.ListBox statusBox;  
private System.Windows.Forms.Button deleteMarkedBalls;  
private System.Windows.Forms.TabPage input;  
private System.Windows.Forms.TextBox inputTextBox;  
private System.Windows.Forms.Button save;  
private System.Windows.Forms.Button result;  
private System.Windows.Forms.TextBox graviyText;  
private System.Windows.Forms.Button gButton;  
private System.Windows.Forms.CheckBox updateState;  
private System.Windows.Forms.Label label4;  
private System.Windows.Forms.Label label3;  
private System.Windows.Forms.Label label2;  
private System.Windows.Forms.Label label1;  
private System.Windows.Forms.Panel panel4;  
private System.Windows.Forms.Panel panel3;  
private System.Windows.Forms.Panel panel2;  
private System.Windows.Forms.Panel panel1;  
private System.Windows.Forms.CheckBox previewResult;  
  
}  
}
```

// PARTÍCULA

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Drawing;
using WindowsFormsApplication1.Manager;

namespace WindowsFormsApplication1.Util
{
    public static class ParticleConstants
    {
        public const float DEFAULT_RADIUS = 5.0f; //TAMANHO DO RAIOS.
        public const float AVAILABLE_RANGE = 195f; // DELIMITA A PARTICULA DENTRO DO RISER.
    }

    /// <summary>
    /// CLASSE QUE REPRESENTA A PARTÍCULA (SÃO AS BOLINHAS NA ANIMAÇÃO).
    /// </summary>
    public class Particle
    {
        #region Fields
        private PointF _position;
        private float _radius; // LOCALIZAÇÃO CARTESIANA(x,y) DO PONTO (DA PARTÍCULA).
        private Color _color; // COR DO RAIOS DE ACORDO COM O Nº DE TRAJETÓRIAS QUE A PARTÍCULA
        É ATINGIDA.

        private int intersectLinesNumber; // NÚMERO DE INTERSECÇÃO DO RAIOS.

        #endregion Fields

        #region Properties
        /// <summary>
        /// POSIÇÃO DA PARTÍCULA.
        /// </summary>
        public PointF Position // Array de pontos flutuantes
        {
            get { return _position; }
            set { _position = value; }
        }
    }
}
```

```

/// <summary>
/// RAIO DA PARTÍCULA.
/// </summary>
public float Radius
{
    get { return _radius; }
    set { _radius = value; }
}
/// <summary>
/// NÚMERO DE LINHAS DE INTERSECÇÃO.
/// </summary>
public int IntersectLinesNumber
{
    get { return intersectLinesNumber; }
    set { intersectLinesNumber = value; }
}
/// <summary>
/// COR COM A QUAL A PARTÍCULA VAI SER IMPRESSA (EXIBIDA NA TELA).
/// </summary>
public Color Color
{
    get { return _color; }
    set { _color = value; }
}
#endregion Properties

#region Constructor
/// <summary>
/// MÉTODO CONSTRUTOR PARA CADA PARTÍCULA.
/// </summary>
/// <param name="position"></param>
public Particle(PointF position)
{
    _position = position;
    _radius = ParticleConstants.DEFAULT_RADIUS;
    Color = Color.Red;
}
#endregion Constructor

#region Methods
/// <summary>
/// MÉTODO QUE APLICA UMA FORÇA A PARTÍCULA.
/// </summary>

```

```

/// <param name="force"></param>
public void ApplyForce(PointF force)
{
    // ALTERA A POSIÇÃO DA PARTÍCULA CONSIDERANDO A ATRAÇÃO SOFRIDA PELA MESMA.
    this.Position = new PointF(Position.X + force.X, Position.Y + force.Y);
}

/// <summary>
/// RESETA (LIMPA) AS VARIÁVEIS.
/// </summary>
public void ResetStatus() // LIMPA AS PARTÍCULAS PARA NOVA EXECUÇÃO
{
    this.instersectLinesNumber = 0;
}

/// <summary>
/// MÉTODO PARA IMPRIMIR (EXIBIR) A PARTÍCULA NA TELA.
/// </summary>
/// <param name="graphics"></param>
public void Draw(Graphics graphics)
{
    float _shapeRadius = _radius * 0.8f;
    float _outLineRadius = _radius;
    Color[] colors = new Color[] { Color.Chartreuse, Color.Yellow, Color.Orange, Color.Red };
    SolidBrush shapeBrush = new SolidBrush(colors[instersectLinesNumber]);
    SolidBrush outLineBrush = new SolidBrush(Color.Black);
    graphics.FillEllipse(outLineBrush, _position.X - _outLineRadius, _position.Y - _outLineRadius, 2
* _outLineRadius, 2 * _outLineRadius);
    graphics.FillEllipse(shapeBrush, _position.X - _shapeRadius, _position.Y - _shapeRadius, 2 *
_shapeRadius, 2 * _shapeRadius);
}

public void CheckStatus() // VERIFICA O MODELO DE FORMA GERAL (A SEÇÃO DO RISER). VER SE
TEM PARTÍCULA FORA DO RISER.
{
    // CONSIDERA A DISTÂNCIA PARTÍCULA-CENTRO ATRAVÉS DA NORMA.
    PointF center = new PointF(ProjectionConstants.PANEL_SIZE.X / 2,
ProjectionConstants.PANEL_SIZE.Y / 2);
    PointF vectorDistance = new PointF(this.Position.X - center.X, this.Position.Y - center.Y);

    float distance = (float) Math.Sqrt(vectorDistance.X * vectorDistance.X + vectorDistance.Y *
vectorDistance.Y);

    if (distance > ParticleConstants.AVAILABLE_RANGE)
    {

```

```

        vectorDistance.X = vectorDistance.X * (ParticleConstants.AVAILABLE_RANGE / distance);
        vectorDistance.Y = vectorDistance.Y * (ParticleConstants.AVAILABLE_RANGE / distance);
        this.Position = new PointF(center.X + vectorDistance.X, center.Y + vectorDistance.Y);
    }
}
#endregion Methods
}
}

```

// PROJEÇÃO

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Drawing;
using WindowsFormsApplication1.Manager;

namespace WindowsFormsApplication1.Util
{
    /// <summary>
    /// CONSTANTES RELACIONADAS A GRAVIDADE APLICADA A PARTÍCULA.
    /// </summary>
    public class ProjectionConstants
    {
        public static float GRAVITY = 0.05f; // ATRAÇÃO QUE A TRAJETÓRIA APLICA A PARTÍCULA.
        public static Point PANEL_SIZE = new Point(505, 481); // TAMANHO DA TELA EM PIXELS.
        public const float PROJECTION_AREA = 200.0f; // DELIMITA O CONJUNTO DE TRAJETÓRIAS DE
        UMA MESMA PROJEÇÃO.
    }

    /// <summary>
    /// CLASSE QUE REPRESENTA A ENTIDADE DE PROJEÇÃO.
    /// </summary>
    public class Projection
    {
        #region Fields
        private List<Trajectory> _trajectories; // LISTA DINÂMICA DE TRAJETÓRIAS (ARRAY).
        private float _angle; // ANGULOS QUE SÃO 0, 60 E 120 GRAUS.
        private PointF _center; // REPRESENTA O CENTRO DO RISER.

```

#endregion Fields

#region Properties

/// <summary>

/// LISTA DE TRAJETÓRIAS RELACIONADAS A CERTA PROJEÇÃO.

/// </summary>

public List<Trajectory> Trajectories

{

 get { return _trajectories; }

 set { _trajectories = value; }

}

/// <summary>

/// ÂNGULO ASSOCIADO A CERTA PROJEÇÃO.

/// </summary>

public float Angle

{

 get { return _angle; }

 set { _angle = value; }

}

/// <summary>

/// CENTRO DO PROJEÇÃO, QUE NESTE SISTEMA COINCIDE COM O CENTRO DA TELA.

/// AS TRAJETÓRIAS SÃO GERADAS A PARTIR DESTA PONTO.

/// </summary>

public PointF Center

{

 get { return _center; }

 set { _center = value; }

}

#endregion Properties

#region Constructor

/// <summary>

/// MÉTODO CONSTRUTOR DE UMA PROJEÇÃO.

/// </summary>

/// <param name="angle">Ângulo associado a uma projeção</param>

/// <param name="n">Numero de trajetorias por projeção</param>

/// <param name="particlesPerTrajectory">Números de partículas por trajetória</param>

public Projection(float angle, int n, int[] particlesPerTrajectory)

{

 _trajectories = new List<Trajectory>();

 _center = new PointF(ProjectionConstants.PANEL_SIZE.X / 2,

ProjectionConstants.PANEL_SIZE.Y / 2);

 _angle = angle;

 AddTrajectories(n, particlesPerTrajectory);

```

}
#endregion Constructor

#region Methods
/// <summary>
/// MÉTODO PARA ADICIONAR TRAJETÓRIAS A CERTA PROJEÇÃO.
/// </summary>
/// <param name="n">Número de trajetórias</param>
/// <param name="particlesPerTrajectory">Número de partículas por projeção</param>
public void AddTrajectories(int n, int[] particlesPerTrajectory)
{
    //CALCULA O ÂNGULO EM RADIANOS E DESCOBRE O VETOR QUE POSSUI A RESPECTIVA
    DIREÇÃO.
    double angleRadian = (2 * Math.PI * (_angle + 90)) / 360;
    PointF ortogonalVector = new PointF((float)Math.Cos(angleRadian),
(float)Math.Sin(angleRadian));
    PointF vector = new PointF(-ortogonalVector.Y/ortogonalVector.X, 1);
    float modvector = vector.X*vector.X + vector.Y*vector.Y;
    modvector = (float)Math.Sqrt(modvector);
    vector = new PointF(vector.X / modvector, vector.Y / modvector);
    //CRIA A LINHA GRÁFICA DA PROJEÇÃO.
    PointF begin = new PointF(_center.X - (ortogonalVector.X *
ProjectionConstants.PROJECTION_AREA), _center.Y + (ortogonalVector.Y *
ProjectionConstants.PROJECTION_AREA));
    PointF end = new PointF(_center.X + (ortogonalVector.X *
ProjectionConstants.PROJECTION_AREA), _center.Y - (ortogonalVector.Y *
ProjectionConstants.PROJECTION_AREA));

    float incX = (end.X - begin.X) / (n + 1);
    float incY = (end.Y - begin.Y) / (n + 1);

    float modvector3 = incX * incX + incY * incY;
    modvector3 = (float)Math.Sqrt(modvector3);

    for (int i = 1; i <= n; i++)
    {
        PointF center = new PointF(begin.X + incX * i, begin.Y + incY * i);
        _trajectories.Add(Trajectory.makeProjection(_angle, center, particlesPerTrajectory[i - 1]));
    }
}

```

```

/// <summary>
/// ATUALIZA O ESTADO DA PROJEÇÃO, RECALCULANDO O ESTADO DE CADA PARTÍCULA E
TRAJETÓRIA.
/// </summary>
/// <param name="particles"></param>
public void UpdateState(List<Particle> particles)
{
    //RESETA (LIMPA) O ESTADO ATUAL DA TRAJETÓRIA.
    foreach (Trajectory trajectory in Trajectories)
    {
        trajectory.CurrentNumberParticles = 0;
    }

    //ATRIBUE UM NOVO ESTADO A TRAJETÓRIA.
    foreach (Particle particle in particles)
    {
        foreach (Trajectory trajectory in Trajectories)
        {
            if (trajectory.Distance(particle.Position) < particle.Radius)
            {
                trajectory.CurrentNumberParticles++;
                particle.InstersectLinesNumber++;
            }
        }
    }
}
/// <summary>
/// MÉTODO PARA ATUALIZAR O ALGORITMO.
/// </summary>
/// <param name="particles"></param>
public void UpdateGravity(List<Particle> particles)
{
    float unstableTrajectories = 0;
    foreach (Trajectory trajectories in Trajectories)
    {
        if (trajectories.NumberParticles != trajectories.CurrentNumberParticles)
        {
            foreach (Particle particle in particles)
            {
                //OBTENDO UMA DIREÇÃO.
                PointF force = trajectories.DirectionFrom(particle.Position);
                //mod
                float mod = (float)Math.Sqrt(force.X * force.X + force.Y * force.Y);
            }
        }
    }
}

```

```

        if (mod != 0)
        {
            force.X = force.X / mod;
            force.Y = force.Y / mod;

            //CALCULANDO A FORÇA.
            float alfa = 0;
            float distance = trajectories.Distance(particle.Position);
            distance = Math.Max(10, distance);
            distance = Math.Min(distance, 100);
            distance = 100 - distance;
            distance = distance / 10;

            alfa = ProjectionConstants.GRAVITY * -(trajectories.NumberParticles -
trajectories.CurrentNumberParticles);
            alfa *= distance;
            unstableTrajectories++;
            force.X = force.X * alfa;
            force.Y = force.Y * alfa;
            //APLICANDO A FORÇA.
            particle.ApplyForce(force);
            if (MainManager.AlwaysUpdateState)
                MainManager.Singleton.UpdateState();
        }
    }

}
}
}

/// <summary>
/// IMPRIMINDO (EXIBINDO) A PROJEÇÃO NA TELA.
/// </summary>
/// <param name="graphics"></param>
public void Draw(Graphics graphics)
{
    foreach (Trajectory projection in Trajectories)
    {
        projection.Draw(graphics);
    }
}

#endregion Methods
}
}

```

// TRAJETÓRIA

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Drawing;

namespace WindowsFormsApplication1.Util
{
    /// <summary>
    /// CONSTANTES RELACIONADAS A TRAJETÓRIA.
    /// </summary>
    public static class TrajectoryConstants
    {
        public const float TRAJECTORY_SIZE = 400.0f; // TAMANHO DA TRAJETÓRIA.
    }
    /// <summary>
    /// CLASSE QUE REPRESENTA A ENTIDADE DE UMA TRAJETÓRIA.
    /// </summary>
    public class Trajectory
    {
        #region Fields
        private PointF _begin; // DELIMITA O INÍCIO DA TRAJETÓRIA.
        private PointF _end; // DELIMITA O FIM DA TRAJETÓRIA.
        private int _numberParticles; // DEFINE O NÚMERO DE PARTÍCULAS EM CERTA TRAJETÓRIA.
        private int _currentNumberParticles; // CONSIDERA O NÚMERO ATUAL DE
PARTÍCULAS EM CERTA TRAJETÓRIA.
        #endregion Fields

        #region Properties
        /// <summary>
        /// NÚMERO DE PARTÍCULAS QUE ESTÃO RELACIONADAS A CERTA TRAJETÓRIA NUM DADO
MOMENTO.
        /// </summary>
        public int CurrentNumberParticles
        {
            get { return _currentNumberParticles; }
        }
    }
}
```

```

    set { _currentNumberParticles = value; }
}

/// <summary>
/// DEFINE O INÍCIO DO VETOR TRAJETÓRIA.
/// </summary>
public PointF Begin
{
    get { return _begin; }
    set { _begin = value; }
}

/// <summary>
/// DEFINE O FIM DO VETOR TRAJETÓRIA.
/// </summary>
public PointF End
{
    get { return _end; }
    set { _end = value; }
}

/// <summary>
/// DEFINE O CENTRO DA TRAJETÓRIA.
/// </summary>
public PointF Center
{
    get
    {
        return new PointF((_end.X - _begin.X) / 2 + _begin.X, (_end.Y - _begin.Y) / 2 + _begin.Y);
    }
}

/// <summary>
/// NÚMERO DE PARTÍCULAS QUE CERTA TRAJETÓRIA DEVE TER.
/// </summary>
public int NumberParticles
{
    get { return _numberParticles; }
    set { _numberParticles = value; }
}

#endregion Properties

#region Constructor

```

```

/// <summary>
/// MÉTODO CONSTRUTOR DA TRAJETÓRIA.
/// </summary>
/// <param name="begin">Ponto usado pra definir o vetor da trajetória</param>
/// <param name="end">Segundo ponto usado pra definir o vetor da trajetória</param>
/// <param name="numberParticles">Número de partículas que deixa o sistema
estável</param>
public Trajectory(PointF begin, PointF end, int numberParticles)
{
    _begin = begin;
    _end = end;
    _numberParticles = numberParticles;
}
#endregion Constructor

#region Methods
/// <summary>
/// DISTÂNCIA DE UM PONTO (PARTÍCULA) À TRAJETÓRIA.
/// </summary>
/// <param name="point"></param>
/// <returns></returns>
public float Distance(PointF point)
{
    PointF u = new PointF(point.X - Begin.X, point.Y - Begin.Y);
    PointF v = new PointF(End.X - Begin.X, End.Y - Begin.Y);

    //PROJEÇÃO u v

    float pp = ((u.X*v.X)+(u.Y*v.Y))/((v.X*v.X)+(v.Y*v.Y));
    PointF projuv = new PointF(pp * v.X, pp * v.Y);

    PointF vectorDistance = DirectionFrom(point);

    return (float)Math.Sqrt(vectorDistance.X * vectorDistance.X + vectorDistance.Y *
vectorDistance.Y);
}

/// <summary>
/// DISTÂNCIA E DIREÇÃO DE UM PONTO (PARTÍCULA) A CERTA TRAJETÓRIA.
/// </summary>
/// <param name="point"></param>
/// <returns></returns>
public PointF DirectionFrom(PointF point)
{
    PointF u = new PointF(point.X - Begin.X, point.Y - Begin.Y);

```

```

    PointF v = new PointF(End.X - Begin.X, End.Y - Begin.Y);
    float pp = ((u.X * v.X) + (u.Y * v.Y)) / ((v.X * v.X) + (v.Y * v.Y));
    PointF projuv = new PointF(pp * v.X, pp * v.Y);

    PointF vectorDistance = new PointF(u.X - projuv.X, u.Y - projuv.Y);

    return vectorDistance;
}
/// <summary>
/// MÉTODO PARA IMPRIMIR (EXIBIR) UMA TRAJETÓRIA NA TELA.
/// </summary>
/// <param name="graphics"></param>
public void Draw(Graphics graphics)
{
    Pen linePen = new Pen(Color.Blue);
    graphics.DrawLine(linePen, Begin, End);
}
#endregion Methods

#region Static Methods

/// <summary>
/// CRIA UMA TRAJETÓRIA.
/// </summary>
/// <param name="angle">Ângulo da trajetória</param>
/// <param name="center">Centro da trajetória</param>
/// <param name="numberParticles">Número de partículas associadas aquela
trajetória.</param>
/// <returns></returns>
public static Trajectory makeProjection(float angle, PointF center, int numberParticles)
{
    double angleRadian = (2 * Math.PI * angle) / 360;

    PointF angleVector = new PointF((float)Math.Cos(angleRadian), (float)Math.Sin(angleRadian));
    PointF begin = new PointF(center.X - (angleVector.X * TrajectoryConstants.TRAJECTORY_SIZE),
center.Y + (angleVector.Y * TrajectoryConstants.TRAJECTORY_SIZE));
    PointF end = new PointF(center.X + (angleVector.X * TrajectoryConstants.TRAJECTORY_SIZE),
center.Y - (angleVector.Y * TrajectoryConstants.TRAJECTORY_SIZE));

    return new Trajectory(begin, end, numberParticles);
}
#endregion Static Methods

```

```

/// <summary>
/// CALCULA A INTERSECÇÃO DE DUAS VARIÁVEIS (BASEADA NA DISTÂNCIA ENTRE DUAS RETAS).
/// </summary>
/// <param name="av2"></param>
/// <returns></returns>
public PointF Intersect(Trajectory av2)
{
    PointF p1 = this.Begin;
    PointF p2 = this.End;
    PointF p3 = av2.Begin;
    PointF p4 = av2.End;

    // a, b SÃO OS COEFICIENTES DE CADA RETA.

    float ua = ((p4.X - p3.X) * (p1.Y - p3.Y) - (p4.Y - p3.Y) * (p1.X - p3.X)) /
        ((p4.Y - p3.Y) * (p2.X - p1.X) - (p4.X - p3.X) * (p2.Y - p1.Y));

    float ub = ((p2.X - p1.X) * (p1.Y - p3.Y) - (p2.Y - p1.Y) * (p1.X - p3.X)) /
        ((p4.Y - p3.Y) * (p2.X - p1.X) - (p4.X - p3.X) * (p2.Y - p1.Y));

    return new PointF(p1.X + ua * (p2.X - p1.X), p1.Y + ua * (p2.Y - p1.Y));
}
}
}

```