UNIVERSIDADE DE SÃO PAULO
INSTITUTO DE FÍSICA DE SÃO CARLOS

MAURICIO FERNANDO LIMA PEREIRA

***Um Modelo de Reconstrução Tomográfica 3D para
Amostras Agrícolas com Filtragem de Wiener em
Processamento Paralelo***

São Carlos

2007

MAURICIO FERNANDO LIMA PEREIRA

***Um Modelo de Reconstrução Tomográfica 3D para
Amostras Agrícolas com Filtragem de Wiener em
Processamento Paralelo***

Tese apresentada ao Instituto de Física de São Carlos
da Universidade de São Paulo para obtenção do título
de Doutor em Ciências - Área: Física Aplicada.

Orientador: Prof. Dr. Paulo Estevão Cruvinel

Co-orientador: Prof. Dr. Luciano da Fontoura Costa

São Carlos

2007

Dedicatória

Dedicado este trabalho às duas mulheres mais importante de minha vida, a Dona Isabel, mãe e grande amiga, e a minha adorada e eterna companheira Ana Paula.

Agradecimentos

A Deus e Nossa Senhora, que nos momentos de dificuldade me ajudaram com sua força e sua luz para continuar o meu trabalho.

Ao Professor Paulo Estevão Cruvinel, pela oportunidade, pelas discussões, pelos conselhos, pela ajuda incondicional, pela compreensão e confiança depositada em mim.

Ao Professor Luciano da Fontoura Costa pela oportunidade e co-orientação deste trabalho.

Ao meu amado pai Josué da Silva Pereira (*in memorian*), por ser sempre uma fonte de inspiração para minha vida.

Às minhas irmãs Sônia Maria Lima Pereira e Sílvia Lima Pereira pelo apoio, pelas risadas e os ótimos momentos nos finais de semana em São Paulo.

Ao meu tio Pedro da Silva Pereira (*in memorian*) pelos momentos bons.

Ao Professor Tito José Bonagamba pela confiança e valorosa oportunidade.

Ao Professor José Hiroki Saito por contribuir para o desenvolvimento do trabalho.

À secretária Wladerez Aparecida Gounella Caiado, pela atenção, suporte e apoio para a conclusão deste trabalho.

Aos Professores Gonzalo Travieso, Nelson Delfino D'Ávila Mascarenhas, Sergio Shiguemi Furuie e ao Pesquisador Mateus José Martins pelas valorosas contribuições para o aprimoramento do trabalho.

Ao grande amigo Marcelo Aparecido Marchioli pelos conselhos, caminhadas, cachorros quentes e saudosos finais de semanas de bate papo e risadas.

Aos amigos do futebol da pós-graduação, pelos momentos de desconcentração.

Aos pesquisadores Álvaro Macedo da Silva e João de Mendonça Naime, do grupo de tomografia de Raios X da Embrapa Instrumentação Agropecuária.

Aos colegas que participaram do grupo de pesquisa em Instrumentação Agropecuária (CNPq) Luciano Vieira Koenigkan, Edson Roberto Minatel, Felipe Pazzinato, Gabriel Marcelino Alves, Luciana Betetto e Marcos Antônio de Matos Laia pelo companheirismo.

Ao CNPq e à Embrapa pelo apoio financeiro e institucional.

A todos que de alguma forma contribuíram para o desenvolvimento deste trabalho.

Certeza

De tudo, ficaram três coisas:

A certeza de que estamos sempre começando...

A certeza de que precisamos continuar...

A certeza de que seremos interrompidos antes de terminar...

Portanto devemos:

Fazer da interrupção um caminho novo...

Da queda um passo de dança...

Do medo, uma escada...

Do sonho, uma ponte...

Da procura, um encontro...

Fernando Pessoa

Resumo

Neste trabalho, é apresentado um novo modelo de reconstrução tridimensional (3D) para amostras agrícolas com filtragem de Wiener em processamento paralelo, o qual é obtido a partir de reconstruções tomográficas bidimensionais (2D). No desenvolvimento, foram modelados algoritmos paralelos de retroprojeção filtrada e reconstrução tridimensional, baseando-se na inserção de um conjunto de planos virtuais entre pares de planos reais obtidos em ensaios tomográficos de raios X na faixa de energia de 56 keV a 662 keV. No modelo, os planos virtuais gerados em algoritmo paralelo são implementados com base na técnica de interpolação por *B-Spline-Wavelet*. Para validação do modelo desenvolvido, foi utilizada uma plataforma paralela composta de 4 processadores DSP, a qual possibilitou a troca de dados entre os processadores DSP e o envio de informações para o *host*, um computador *desktop* com processador Pentium III operando em 800 MHz. A extração de medidas de eficiência, de ganho e de precisão dos algoritmos paralelos foi realizada com base em um conjunto de amostras agrícolas (solo, vidro e madeiras) e de *phantoms* de calibração. Nessa avaliação, observou-se que o algoritmo de reconstrução 2D, utilizado como base para o algoritmo de reconstrução 3D, possibilitou uma alta eficiência para imagens de maior resolução, atingindo um pico de 92% de eficiência na resolução de 181×181 pixels. O algoritmo paralelo de reconstrução 3D foi analisado para um conjunto de amostras, sob diferentes configurações de planos reais e virtuais, organizados de forma a possibilitarem a avaliação do impacto causado pelo aumento da granularidade da comunicação e da carga de trabalho. Um melhor desempenho, com ganho médio igual a 3,4, foi obtido na reconstrução de objetos que demandaram o cálculo de um maior número de planos. Também, buscou-se conhecer a adaptabilidade do modelo para uso em arquitetura convencional, sendo que neste caso o uso de MPI permitiu a comunicação entre as tarefas projetadas em cada algoritmo paralelo. Adicionalmente, foram incluídas ferramentas de visualização 2D e 3D para que usuários possam analisar as imagens e as características das amostras agrícolas em ambiente tridimensional. Os resultados obtidos indicam que o modelo de reconstrução 3D paralela trouxe contribuições originais para a área de tomografia agrícola aplicada à física de solos, bem como para a criação de ferramentas que viabilizem explorar recursos computacionais disponíveis em arquiteturas paralelas que demandem elevada capacidade de processamento.

Palavras-chave: Reconstrução Tomográfica. Processamento Paralelo. Física de solos. Filtragem de Wiener. Tomografia de solos.

Abstract

This work presents a new method for three dimensional (3D) image reconstruction dedicated to the investigation in soil physics by means of X-ray tomography which is obtained using two-dimensional (2D) tomographic image reconstructed slices. The conception of the 3D model for reconstruction and visualization was based on the filtered back projection algorithm, operating under parallel environment together the insertion of virtual planes between pairs of real planes obtained by X-Ray tomography under energies varying from 56 keV to 662 keV. In this model, the virtual planes were generated by interpolation with the use of B-Spline-Wavelets. The evaluation of the 3D reconstruction model was established by using a set of agricultural samples (i.e., soil, glass, wood and calibration phantoms) having different configuration for the planes. Such configuration was based on setting not only the sizes and the number of the real but also the virtual planes in the volume. This procedure allows the impact measurements as a function of the increasing in workload and the communication granularity. To validate the reconstruction model, a dedicated parallel architecture composed of 4 DSP processors was used. This board enables data exchange between DSP processors and communication with host computer. A measurement of efficiency with a speed up equal to 3.4 was obtained using the same set of samples and a better performance was observed with a higher number of planes. Also, to understand about its adaptability, the model was implemented in conventional architecture, using MPI library to enable communication between designed tasks. Additionally, 2D and 3D visualization tools based on Visualization ToolKit were included in order to help users to analyze images and their characteristics. Results have shown that the 3D parallel model reconstruction brought original contributions for the soil science diagnosis by X-Ray tomography, as well as to explore the available computational resources in parallel architectures, which demands great processing capacity.

Keywords: Tomographic Reconstruction. Parallel Processing. Soil Physics. Wiener Filtering. Soil Analysis.

Conteúdo

Dedicatória

Agradecimentos

Lista de Figuras

Lista de Tabelas

Lista de abreviaturas e siglas

Lista de símbolos

1	Motivações, contribuições deste trabalho e estrutura dos tópicos	21
1.1	Objetivos	21
1.2	Motivações	22
1.3	Contribuições do Trabalho	26
1.4	Estrutura dos tópicos	27
2	Métodos de Reconstrução Tomográfica de Raios X	28
2.1	Introdução	28
2.2	Fundamentos matemáticos da reconstrução tomográfica	30
2.2.1	Teorema das secções de Fourier	31

2.2.2	Retroprojeção Filtrada	35
2.3	Problemas relacionados à reconstrução de imagens tomográficas	38
2.4	Aplicação de tomografia na agricultura	40
2.5	Reconstrução tridimensional de amostras agrícolas	46
2.5.1	Método de interpolação baseada em <i>B-Wavelet</i>	48
3	Filtragem de projeções tomográficas <i>a priori</i> na reconstrução das imagens	54
3.1	Fundamentos	54
3.1.1	Estacionaridade no sentido amplo	54
3.1.2	Coeficiente de autocorrelação	54
3.1.3	Coeficiente de correlação cruzada	55
3.2	Filtragem <i>a priori</i>	55
3.2.1	Transformação de Anscombe	56
3.3	Métodos de filtragem	57
3.3.1	Filtro Mediana	57
3.3.2	Filtros preditivos	57
3.4	Filtro de Wiener	58
3.4.1	Filtro de Wiener FIR	59
3.4.1.1	Filtragem	62
3.4.1.2	Predição Linear	62
4	Processamento Paralelo	65
4.1	Introdução	65

4.2	Projeto de sistemas paralelos	69
4.2.1	Particionamento	70
4.2.2	Comunicação	72
4.2.3	Aglomeracao	73
4.2.3.1	Reducao dos custos de comunicacao através do aumento da granularidade de tarefas e comunicacao	74
4.2.3.2	Preservar Flexibilidade	76
4.2.4	Mapeamento	77
4.3	Medidas de desempenho	79
4.4	Plataforma DSP	80
4.4.1	Configuracao de tarefas em uma aplicacao	80
5	Modelo de Reconstrucao 3D e Aplicacao da Filtragem de Wiener Dedicada às Ci- ências do Solo em processamento paralelo	85
5.1	Introducao	85
5.2	Algoritmo paralelo de reconstrucao bidimensional	87
5.2.1	Modelagem do algoritmo paralelo de reconstrucao bidimensional . . .	87
5.2.2	Implementacao na plataforma DSP	91
5.3	Modelagem do algoritmo de reconstrucao tridimensional	94
5.3.1	Implementacao da reconstrucao 3D na plataforma DSP	99
5.3.1.1	Formato de gravacao dos cortes tomograficos	100
6	Resultados e Conclusões	105

6.1	Aplicação do modelo de filtragem de Wiener em amostras de <i>phantoms</i> homogêneos e heterogêneos	105
6.2	Interface com o usuário e algoritmos de reconstrução	110
6.2.1	Interface de visualização bidimensional	110
6.2.2	Avaliação da correlação entre coeficiente de atenuação linear e tons de cinza	111
6.2.3	Interface de visualização tridimensional	114
6.2.3.1	Avaliação da precisão da aglomeração de blocos	117
6.3	Estudo de caso	120
6.4	Resultados do método paralelo	124
6.4.1	Resultados obtidos na reconstrução bidimensional na plataforma DSP	124
6.4.1.1	Implementação em ambiente convencional do algoritmo paralelo	130
6.4.2	Resultados obtidos na reconstrução tridimensional	133
6.4.2.1	Organização da aplicação de reconstrução 3D na plataforma DSP	133
6.4.2.2	Medidas de desempenho do algoritmo paralelo de reconstrução 3D	136
6.5	Conclusões	139

Lista de Figuras

1.1	Diagrama com a concepção inicial do modelo de reconstrução 3D de amostras agrícolas, baseado em técnicas paralelas	25
2.1	Ilustração da tomografia de transmissão	30
2.2	Projeção paralela de $f(x,y)$ para Transformada de Radon	31
2.3	Teorema das secções de Fourier	32
2.4	Conjunto de estimativas nas linhas radiais no espaço de frequências, das projeções de um objeto	34
2.5	Retroprojeção dos pontos sobre a linha LM a partir do dado $Q_{\theta_i}(t)$ da projeção filtrada Q_{θ_i}	36
2.6	Imagens tomográficas de um torrão de terra adquirida com o minitomógrafo de resolução micrométrica (MACEDO, 1997) (a) Sem ajuste do contraste (b) Equalização do contraste	39
2.7	Minitomógrafo de resolução milimétrica de laboratório (CRUVINEL, 1987) . . .	43
2.8	Minitomógrafo portátil para estudo de solo e plantas, em campo (NAIME, 1994)	43
2.9	Novos tomógrafos da Embrapa - (a)Minitomógrafo de resolução micrométrica (MACEDO, 1997) - (b)Minitomógrafo de varredura em leque, desenvolvido por Naime (NAIME, 2001)	45
2.10	Minitomógrafo Compton desenvolvido por Cruvinel e Balogun (CRUVINEL; BALOGUN, 2006)	46
2.11	Ilustração da interpolação tridimensional a partir de fatias reconstruídas	47

2.12	Ilustração apresentando a diferença existente entre a interpolação e aproximação por <i>B-Wavelet</i>	48
2.13	Vetor com os valores originais.	51
2.14	Inserção de pontos fantasmas.	52
2.15	Função obtida com a interpolação por <i>B-Wavelets</i> - (a) Visualização dos pontos do vetor A_i juntamente com os demais pontos; (b) Função de interpolação incluindo pontos fantasmas e as retas que conectam os pontos de controle; (c) Função gerada pela interpolação; (d) Interpolação de 4 pontos entre cada par de pontos.	53
3.1	Diagrama da filtragem <i>a priori</i> das projeções	56
3.2	Ilustração de um problema geral do filtro de Wiener. Dados dois processos estacionários, $x(n)$ e $d(n)$, que são estatisticamente relacionados entre si, o filtro $W(z)$ minimiza a estimativa do erro médio quadrático, $\hat{d}(n)$, de $d(n)$ (HAYES, 1996)	58
3.3	Predição linear como um tipo de problema onde busca-se encontrar a estimativa $x(n+1)$ através da combinação linear de p valores de $x(n)$ até $x(n-p+1)$ (HAYES, 1996)	63
4.1	Execução em <i>pipeline</i> com uso de instruções sucessivas, onde o sistema passa a executar, após alguns ciclos, uma instrução por ciclo (HWANG, 1993)	67
4.2	Metodologia para desenvolvimento de sistemas paralelos PCAM	70
4.3	Ilustração da decomposição de um problema com dados em três dimensões	71
4.4	Ilustração da aglomeração de tarefas para aumentar a granularidade da comunicação e da computação	75
4.5	Ilustração de 8 processos trocando dados e realizando o somatório de valores, de forma que, ao final de 3 etapas, todos os processos possuem o valor do somatório	76

4.6	Ilustração da comunicação e envio de cargas de trabalho de gerente para trabalhadores	78
4.7	Modelo de aplicação configurada numa rede de processadores DSP com arquivo de configuração - (a)Modelagem da aplicação X, com 4 tarefas que se comunicam; (b)A aplicação X com suas tarefas mapeadas entre os processadores <i>Root</i> , P1 e P2	83
4.8	Placa HEPC2E, utilizada como plataforma paralela para avaliação do desempenho do método paralelo	84
5.1	Diagrama com o modelo de alto nível dos processos do trabalho	86
5.2	Ilustração das tarefas resultantes do particionamento do algoritmo de reconstrução bidimensional e do estabelecimento da comunicação entre as elas	88
5.3	Aglomeração de tarefas do algoritmo paralelo de reconstrução 2d	90
5.4	Funções que compõe a tarefa gerente na reconstrução bidimensional	93
5.5	Funções que compõem a tarefa trabalhador apresentadas juntamente com a tarefa gerente na reconstrução bidimensional	95
5.6	Esquematização do particionamento e comunicação das tarefas da reconstrução tridimensional baseada em interpolação por <i>B-Wavelet</i>	97
5.7	Aglomeração das tarefas de reconstrução tridimensional paralela	99
5.8	Algoritmo em alto nível do processo gerente	101
5.9	Algoritmo em alto nível do processo trabalhador	101
5.10	Diagrama de blocos dos algoritmo paralelo de reconstrução 3D, ilustrando a interação entre gerente e trabalhador	102
5.11	Formato de arquivo <i>.vtk</i> utilizado na descrição de um objeto tridimensional	104
6.1	Diagrama de blocos da filtragem de Wiener por predição	105

6.2	Projeção homogênea - (a)Filtro de Wiener por predição; (b)Filtragem por mediana	106
6.3	Conjunto de projeções do <i>phantom</i> homogêneo - (a) Originais; (b)Ruidosas; (d)Filtro por Mediana com máscara [1x5] (d)Filtro de Wiener por Predição com 6 pesos.	107
6.4	Imagens reconstruídas de um <i>phantom</i> homogêneo, a partir das projeções (a)originais; (b)ruidosas; (c)filtradas predição de Wiener com 6 pesos.	108
6.5	Comparação do maior erro após a aplicação das filtragens	109
6.6	Aplicação da filtragem em uma projeção da amostra heterogênea (a)Filtro de Wiener por Predição (b) Filtro por Mediana	109
6.7	Conjunto de projeções do <i>phantom</i> heterogêneo - (a) Originais; (b)Ruidosas; (c)Filtro por Mediana com mascara [7x1]; (d)Filtro de Wiener por Predição de 2 pesos	110
6.8	Imagens reconstruídas de um <i>phantom</i> heterogêneo, a partir das projeções (a)originais; (b)ruidosas; (c)filtradas predição de Wiener com 2 pesos.	110
6.9	Janela do sistema que permite realizar a escolha da amostra tomográfica e visu- alização dos cortes reconstruídos	112
6.10	<i>Phantom</i> de calibração com os elementos Cálcio(Ca), Alumínio(Al), Fósforo(P), Água(H ₂ O) e <i>Plexiglass</i>	113
6.11	Gráfico do ajuste da curva de calibração obtida a partir dos dados de coeficiente de atenuação(cm^{-1}) $\times$ tons de cinza da imagem	114
6.12	Diagrama de classes da aplicação Viewer3D	116
6.13	Interface do software de visualização onde é possível ao usuário interagir com o objeto reconstruído	117
6.14	Janela do sistema apresentando a visualização de voxels do objeto que estão dentro de uma região de interesse	118

6.15	Visualização do corte sagital do <i>phantom</i> heterogêneo.	118
6.16	Visualização do corte coronal do <i>phantom</i> heterogêneo.	119
6.17	Visualização do corte transversal do <i>phantom</i> heterogêneo.	119
6.18	Visualização de um corte sagital combinada com a extração de uma medida de intensidade de um <i>voxel</i>	120
6.19	Plano gerado para calibrar a aglomeração de blocos durante a reconstrução 3D .	121
6.20	Objeto tridimensional gerado na plataforma DSP a partir do plano padrão . . .	121
6.21	Imagens de dois cortes reconstruídos de um torrão de solo nos horizontes de (a)88 mm; (b) 158 mm	122
6.22	Imagens dos cortes transversal(a), coronal(b), sagital(c) e medidas de coeficiente de atenuação linear(d) do torrão de solo.	123
6.23	Amostra do torrão de solo apresentada com diferentes faixas de limiares para as tonalidades: (a) Entre 255 e 0; (b) Entre 255 e 100; (c) Entre 99 e 10	124
6.24	Exemplo de interatividade do modelo de visualização, o qual permite que se execute movimentos de rotação e aproximação das amostras	125
6.25	Grupo de imagens reconstruídas com o algoritmo paralelo - (a)Latossolo roxo ($41 \times 41 pixels$); (b)Solo Gley(51×51); (c) <i>Phantom</i> de Calibração (64×65); (d)Madeira (76×76); (e)Madeira no Horizonte B (81×81); (f)Esfera de vidro (101×101); (g)Granito (121×121); (h)Grãos de areia (145×145); (i)Bloco de solo (151×151); (j)Torrão de solo (181×181).	126
6.26	Comparação entre a eficiência da reconstrução bidimensional	127
6.27	Tempo de execução do algoritmo seqüencial e dos algoritmos paralelos, na arquitetura paralela dedicada, com 2, 3 e 4 processadores	129

6.28	Gráfico com o comportamento da eficiência da reconstrução bidimensional para 2, 3 e 4 processadores DSP	130
6.29	Dados em formato XML exportados por um trabalhador, durante o processo de reconstrução bidimensional	131
6.30	Perfil do desempenho médio dos processos trabalhadores	131
6.31	Desempenho dos processos trabalhadores na reconstrução da amostra com 121 projeções de 121 pontos	132
6.32	Grupo de imagens reconstruídas com o algoritmo paralelo - (a)Madeira (201×201) (b)Grãos de areia (221×221); (c)Compósito (251×251)	134
6.33	Comparação do desempenho dos três algoritmos implementados em plataforma convencional. (a)Tempos de execução das implementações A, B e C (b)Ganho obtido em cada implementação.	135
6.34	Medidas obtidas no estudo do desempenho do algoritmo paralelo na configuração 1 - (a)Ganho; (b)Eficiência	137
6.35	Medidas obtidas no estudo do desempenho do algoritmo paralelo na configuração 2 - (a)Ganho; (b)Eficiência	138
6.36	Medidas obtidas no estudo do desempenho do algoritmo paralelo na configuração 3 - (a)Ganho; (b)Eficiência	140

Lista de Tabelas

5.1	Tabela de carga de trabalho, utilizada no algoritmo de comunicação.	90
5.2	Arquivo de mapeamento das tarefas na criação da aplicação paralela de reconstrução 2D na rede de processadores DSP.	91
6.1	Tabela com valores de maior erro obtidos de uma projeção do <i>phantom</i> homogêneo com ruído, após a aplicação dos filtros por predição de Wiener e por mediana.	106
6.2	Tabela com valores de variância obtidos de uma região de interesse no centro das imagens reconstruídas a partir de projeções filtradas com os filtros por predição de Wiener e por mediana.	108
6.3	Tabela com valores de maior erro obtidos de um <i>phantom</i> heterogênea, após a aplicação do filtro de Wiener por predição e filtragem por mediana	108
6.4	Tabela com valores de variância obtidos de quatro ROIs, os quais foram extraídos de imagens de um <i>phantom</i> heterogêneo, reconstruídas a partir de projeções filtradas com os filtros por predição de Wiener e por mediana.	111
6.5	Tabela com os coeficientes de atenuação e os tons de cinza obtidos	113
6.6	Descrição detalhada das configurações para obtenção dos dados tomográficos das amostras utilizadas na reconstrução paralela.	125
6.7	Tabela com resultados das medidas de ganho e eficiência obtidos na reconstrução 2D nas diferentes amostras utilizadas em arquitetura paralela com 2, 3 e 4 processos trabalhadores	128

6.8	Arquivo de mapeamento das tarefas na criação da aplicação paralela de reconstrução 3D na rede de processadores DSP.	136
-----	---	-----

Lista de abreviaturas e siglas

AT	Transformada de Anscombe
CPU	Central Processing Unit
FFT	<i>Fast Fourier Transform</i> - Transformada Rápida de Fourier
IFFT	<i>Inverse Fast Fourier Transform</i> - Inversa da Transformada Rápida de Fourier
MPI	<i>Message Passing Interface</i>
PCAM	Metodologia com fases Particionamento, Comunicação, Aglomeração e Mapeamento
ROI	<i>Region of interest</i> - Região de interesse
SNR	Relação Sinal-Ruído
VTk	Visualization ToolKit
XML	eXtensible Markup Language

Lista de símbolos

$\hat{d}(n)$	Estimativa do sinal desejado
ϕ	Número de fótons
θ	Ângulo da projeção
ξ	Eficiência quântica da fotomultiplicadora
$d(n)$	Sinal desejado
$e(n)$	Erro de estimação do filtro de Wiener
$F(u, v)$	Transformada Bidimensional de Fourier de uma função $f(x, y)$
$h(t)$	Impulso resposta do filtro
K	Quantidade de projeções adquiridas
$P(\phi, t)$	Probabilidade de detecção de ϕ fótons em um tempo t de exposição
P_θ	Projeção no ângulo θ
Q_θ	Projeção filtrada, de ângulo θ
$r_{xd}(k-l)$	Função correlação cruzada entre o sinal desejado e o sinal estimado
R_x	Matriz $N \times N$ cujos elementos são as funções de autocorrelação do processo estocástico $x(n)$
$r_x(k-l)$	Função autocorrelação do sinal da entrada do filtro de Wiener
S_θ	Transformada de Fourier da projeção P_θ
$T_{filtragem}$	Quantidade de processos trabalhadores de filtragem tomográfica
$T_{retroprojektor}$	Quantidade de processos trabalhadores de retroprojeção

1 Motivações, contribuições deste trabalho e estrutura dos tópicos

1.1 Objetivos

Este trabalho apresenta um modelo de reconstrução tridimensional de amostras agrícolas que se baseie em técnicas do processamento paralelo e filtragem preditiva para eliminação de ruído das projeções. O foco do trabalho está na modelagem, implementação e validação do modelo de reconstrução e na capacidade de acelerar o processo tomográfico. Os resultados encontram aplicação em estudos sobre:

- Solos e plantas (estruturas de formação de poros e texturas);
- Movimentação de água e soluto nos solos;
- Distribuição de raízes.

Além disso, o trabalho visa desenvolver um modelo de reconstrução tridimensional que permita explorar de forma otimizada arquiteturas paralelas, buscando atender a um diferenciado grupo de arquiteturas com a paralelização de todo o processo de reconstrução, desde a filtragem dos cortes, passando pela reconstrução bidimensional, até a reconstrução tridimensional. Outro aspecto relevante é a geração de um modelo de visualização tridimensional das imagens e dos objetos reconstruídos, a partir dos tomógrafos da Empresa Brasileira de Pesquisa Agropecuária (Embrapa), o que possibilita um ambiente de análise com visualização de cortes tomográficos de forma interativa e com reconhecimento espacial, medida de coeficiente de ate-

nuação linear com a visualização de coordenada, bem como ferramenta de limiarização para auxílio ao diagnóstico.

1.2 Motivações

Avaliando a evolução que vem ocorrendo na área física de solos, percebe-se o crescente interesse da comunidade científica para o desenvolvimento e aplicação de técnicas não invasivas para o estudo de características do solo. Dentre as técnicas utilizadas, destaca-se a tomografia computadorizada de raios X, que se sobressai em relação às demais técnicas aplicadas na física de solos, como a gravimétrica e a sonda de nêutrons (TEIXEIRA *et al.*, 2005) (FERREIRA *et al.*, 1998), devido à sua precisão na extração de atributos físicos, como densidade e umidade, e pela característica de possibilitar o exame de amostras de solo de forma não destrutiva (AYLMORE; HAINSWORTH, 1983) (CRESTANA, 1985) (PEDROTTI *et al.*, 2003). Outra vantagem oferecida pela tomografia computadorizada, em relação às demais, é a possibilidade de fazer-se uso, após a reconstrução, das ferramentas do processamento de imagens (GONZALEZ; WOODS, 2000) para auxiliar a investigação dos fenômenos físicos que ocorrem solo.

No contexto da área agrícola, percebe-se que o progresso dos trabalhos da técnica tomográfica de solos ocorreu em duas vertentes bem definidas, descritas como:

- Vertente de instrumentação dos equipamentos tomográficos;
- Vertente de algoritmos de reconstrução tomográfica, desempenho de algoritmos e visualização.

A vertente de instrumentação pesquisa o desenvolvimento de novos equipamentos e o aprimoramento dos existentes. Busca aumentar a portabilidade dos equipamentos tomográficos, de forma a possibilitar seu uso em campo, fazendo com que ocorra o mínimo possível de mudanças nas condições reais em que se encontra o objeto em estudo.

Este campo de estudos também trata do desenvolvimento da geometria do conjunto fonte-detector dos equipamentos de aquisição. Nesta área, as pesquisas têm trazido, ao longo

destes últimos anos, significativos resultados, no que tange a avanços no processo de aquisição dos tomógrafos agrícolas. Essencialmente, o que se tem buscado são formas de aquisição mais rápidas e geometrias que causam uma menor destruição do ambiente de estudo.

Um importante fator levado em consideração no decorrer dos projetos de instrumentação tomográfica é o de produção de novos equipamentos com custos reduzidos em relação aos equipamentos comerciais, caros e essencialmente planejados para o uso com pessoas. Destaca-se que esta preocupação também se constitui num ponto importante para ampliação do uso da técnica tomográfica em outras áreas do conhecimento humano, além da área de aplicações médicas.

Na outra vertente de desenvolvimento da tomografia de solos, a de algoritmos de reconstrução e visualização, grande parte deste desenvolvimento se deu no aprimoramento dos algoritmos de reconstrução, com aplicação de diferentes técnicas de reconstrução e variadas formas de filtragem, as quais envolvem técnicas lineares, estatísticas e baseadas em transformada *wavelets* (GRAPS, 1995). Ainda na área de reconstrução bidimensional, realizaram-se estudos com tomografias de múltiplas energias (GRANATO, 1998) (BRAZ *et al.*, 2000), que buscaram correlacionar as informações das duas imagens obtidas e melhorar a qualidade das imagens reconstruídas.

A visualização dos objetos reconstruídos tem, no decorrer dos últimos anos, evoluído constantemente, acompanhando o desenvolvimento das áreas de computação gráfica. Em relação às ferramentas, tem-se feito desde o uso de técnicas tradicionais de computação gráfica para visualização 3D (MINATEL, 1997) até o desenvolvimento de ferramentas que utilizam bibliotecas de visualização como o *OpenGL* (PEREIRA *et al.*, 2001).

Neste panorama da aplicação da reconstrução tomográfica em física de solos, percebe-se que há pontos que têm sido pouco explorados, sendo os mesmos abertos à pesquisa. Dentre eles, destaca-se a ausência de modelos de reconstrução 3D que permitam, através da combinação de técnicas de processamento de imagens e de processamento paralelo, um maior aprofundamento dos estudos dos fenômenos dinâmicos que ocorrem no solo, com ênfase nas apli-

cações agrícolas. Sobretudo, destaca-se o desenvolvimento insuficiente de modelos de reconstrução tridimensional de amostras agrícolas que explorem paradigmas paralelos de modelagem de sistemas. Algoritmos desenvolvidos sobre esse modelo, contribuem significativamente para abreviação do tempo de execução dos algoritmos de reconstrução tomográfica bidimensional e tridimensional.

Outro aspecto relevante é o aumento do poder de processamento que pode ser obtido com o uso de arquiteturas paralelas baseadas em *clusters*, em máquinas com processadores de múltiplos núcleos, bem como com arquiteturas paralelas dedicadas baseadas em processadores DSP. É interessante ressaltar que, dado o desenvolvimento que vem ocorrendo na vertente de instrumentação tomográfica para área agrícola, a criação de um modelo que permitia explorar a capacidade destas arquiteturas paralelas é algo que está em consonância com todo o progresso que vem ocorrendo nesta área.

Um primeiro ensaio para modificar este quadro foi apresentado por Pereira (2001), obtendo-se resultados a respeito do desempenho de um algoritmo paralelo de reconstrução tridimensional em uma arquitetura paralela dedicada, baseada em 2 processadores DSP TMS320C40. Os resultados do trabalho mostraram que, mesmo não sendo um algoritmo de reconstrução tridimensional paralelo plenamente otimizado, quando este era aplicado nesta arquitetura paralela, fornecia resultados satisfatórios com relação ao tempo de reconstrução e eficiência no uso. Principalmente, percebeu-se que o algoritmo forneceu melhores resultados quando comparado com arquiteturas monoprocessadas baseadas em processadores convencionais.

Apesar dos méritos e resultados do trabalho acima mencionado, os quais incluíram inovações no uso de processamento paralelo para área de tomografia de solos, observou-se que, na questão de estabelecimento de um modelo completo de reconstrução 3D, o trabalho se encontrava incompleto. Tal trabalho não considerou aspectos da física de solos e os conceitos físicos envolvidos no processo de aquisição, como por exemplo, a energia utilizada na aquisição dos dados e a filtragem do ruído Poisson. Outro aspecto a ser considerado é que também não havia a implementação de algoritmos paralelos para reconstrução bidimensional, bem como

havia somente a opção de reconstrução 3D com o uso do DSP TMS320C40. No contexto da visualização tridimensional, o referido trabalho apenas permitiu a visualização da casca do objeto reconstruído, sem fornecer ferramentas de análise.

Desta forma, este trabalho apresenta um modelo completo de reconstrução tridimensional, que permite seu uso em diferentes modelos de arquiteturas paralelas, o qual é sistematizado e apresentado no diagrama em blocos da Figura 1.1. Incluídos neste sistema, encontram-se os modelos de visualização 2D e 3D, que viabilizam estudos na área de física de solos aplicada à agricultura.

Figura 1.1: Diagrama com a concepção inicial do modelo de reconstrução 3D de amostras agrícolas, baseado em técnicas paralelas

A filtragem *a priori* estabelece um elemento capaz de reduzir a presença de ruído Poisson, dependente do sinal, das projeções tomográficas. Essa abordagem visa garantir maior precisão na imagem reconstruída. Os conceitos envolvidos a respeito do ruído Poisson e a filtragem *a priori* utilizados neste trabalho serão abordados no Capítulo 3.

Os módulos de leitura e escrita visam estabelecer um padrão independente de arquitetura para aquisição e gravação das imagens e objetos reconstruídos. O desenvolvimento

dos algoritmos paralelos é realizado utilizando-se o paradigma de modelagem de algoritmos paralelos proposto por (FOSTER, 2005). No desenvolvimento de todos os módulos, utilizou-se a implementação em linguagem C/C++, devido à sua portabilidade em diferentes arquiteturas.

1.3 Contribuições do Trabalho

As contribuições deste trabalho são várias e dentre elas se destacam:

- a criação de um novo modelo de reconstrução 3D que permita o uso de hardware paralelo para aplicação na área de física de solo;
- a validação do método em uma arquitetura paralela dedicada baseada em processadores DSP, que pode ser acoplada ao tomógrafo ou um a computador portátil;
- a obtenção de reconstruções bidimensionais e tridimensionais de forma mais rápida, viabilizando a aplicação do modelo na Física de Solos;
- a criação de uma estrutura que permita o estudo de fenômenos dinâmicos da física de solos em 3D, a exemplo da avaliação dinâmica do movimento de água e solutos em amostras de solo e o estudo da porosidade;
- a portabilidade do modelo paralelo de reconstrução tridimensional, que foi aplicado tanto em arquitetura dedicada em DSP, quanto em arquitetura convencional de processador de duplo núcleo;
- o desenvolvimento de um modelo de visualização 2D e o 3D, baseado em biblioteca VTK (VTK, 2006), que possibilita a extração de medidas das amostras. Com relação ao modelo de visualização 3D, destaca-se que este ofereceu maior interatividade com o objeto em estudo, permitindo a visualizações de cortes e de faixas de coeficientes de interesse.

Finalmente, fica como importante contribuição a iniciativa do uso de técnicas baseadas em algoritmos paralelos na solução de problemas da área de imagens tomográficas aplicada à física de solos.

1.4 Estrutura dos tópicos

A partir dos próximos capítulos deste trabalho, será apresentada uma breve revisão bibliográfica a respeito dos tópicos estudados durante o desenvolvimento deste modelo paralelo de reconstrução tridimensional.

No Capítulo 2, serão apresentados os princípios matemáticos da reconstrução tomográfica de cortes bidimensionais, bem como os detalhes a respeito do algoritmo de retroprojeção filtrada, além de um breve histórico sobre a aplicação da tomografia na agricultura e o algoritmo de reconstrução tridimensional, que se baseia na técnica de interpolação B-Wavelets.

No Capítulo 3, aborda-se a filtragem *a priori*, com destaque para filtragem Wiener, a qual é utilizada neste trabalho.

No Capítulo 4, serão mostrados tópicos ligados a algoritmos paralelos e arquiteturas dedicadas para processamento paralelo. Neste contexto, será também apresentado um paradigma de modelagem de sistemas paralelos denominado PCAM, composto de quatro fases principais, quais sejam: particionamento, comunicação, aglomeração e mapeamento. Cada uma destas quatro fases será detalhada ao longo do capítulo, que se encerra com uma revisão das medidas de desempenho e da plataforma DSP utilizada.

No Capítulo 5, apresenta-se o desenvolvimento do modelo de reconstrução 3D paralelo. Ao longo do capítulo, demonstra-se a forma como se conduziu a modelagem dos algoritmos de paralelos e bem como a filtragem.

Finalmente, no Capítulo 6, apresentam-se os resultados e as conclusões obtidas.

2 Métodos de Reconstrução Tomográfica de Raios X

2.1 Introdução

No contexto da tomografia computadorizada, as maiores contribuições para o seu desenvolvimento foram dadas por Radon (RADON, 1917), Cormack (CORMACK, 1963) e Hounsfield (HOUNSFIELD, 1973). Em 1917, o matemático austríaco Radon foi o primeiro a apresentar uma solução matemática das equações de reconstrução de corpos a partir de projeções, isto é, a determinação da função de densidade da região estudada através de suas projeções.

Desconhecendo o trabalho de Radon, Cormack desenvolveu a técnica matemática para reconstruir imagens utilizando o método da retroprojeção. Em 1956, ele era professor de Física da *University of Cape Town* e foi solicitado para supervisionar o uso de isótopos radioativos no *Groote Schuur Hospital* devido à demissão do físico do hospital. Durante algumas semanas, Cormack trabalhou com os isótopos radioativos e acompanhou tratamentos de radioterapia. Com base em experimentos e observações, formulou uma matriz de coeficientes para cortes seccionais que poderia ser obtida pela medida da transmissão de raios X em vários ângulos através de um corpo. A partir de transmissões de raios X, aplicou-a para obter imagens de phantoms simples (CRUVINEL, 1987).

Em aplicações médicas, o primeiro tomógrafo computadorizado de raios X de caráter comercial foi apresentado em 1973, por EMI Ltda, fruto do desenvolvimento realizado por Hounsfield. Este desenvolvimento causou um grande impacto no diagnóstico radiológico. Devido a estas contribuições, em 1979, Hounsfield e Cormack dividiram o prêmio Nobel de

Medicina.

O fato de poder-se observar dados internos dos corpos, após a reconstrução das imagens tomográficas, de forma não destrutiva e não invasiva, constitui-se numa importante característica da técnica tomográfica.

A tomografia utiliza um feixe colimado de radiação, o qual define planos tão finos quanto o próprio feixe e, através de vários feixes colimados paralelos, pode-se definir vários planos. Dessa forma, ao invés de impressionar-se um filme radiográfico, como é feito em uma radiografia convencional (CRUVINEL, 1987), obtém-se, de cada reta de propagação dos feixes que partem da fonte para o detector, valores que formam uma projeção, tal qual ilustra a Figura 2.1. Com isso, pode-se dizer que os dados necessários para a reconstrução são na realidade um conjunto de integrais de linha ao longo dos raios que atravessam o objeto.

Na Figura 2.1, observa-se uma linha tracejada que representa a radiação que parte da fonte para o detector. Ela atravessa o objeto e, à medida que o conjunto caminha através dos eixos L' e L , que formam com eixo x um ângulo θ , as projeções vão sendo obtidas. As varreduras devem ser realizadas para n valores de θ dentro do intervalo $0 \leq \theta < 180^\circ$. Desta forma, o que se obtém após a varredura completa é a transformada de Radon do objeto (RADON, 1917).

Conforme será apresentado em seguida, pode-se reconstruir a imagem da fatia do objeto, através da transformada inversa de Radon ou de métodos derivados desta transformada. Para realizar tal tarefa, os algoritmos de reconstrução têm como entrada dados de projeção e produzem como saída uma imagem estimada do objeto original, baseando-se nos dados disponíveis. A estimativa da imagem varia de método para método. Contudo, a qualidade dos resultados depende de como os dados foram coletados e do objeto que está sendo estudado. Esses algoritmos, do ponto de vista matemático e computacional, determinam como reconstruir um objeto $f(x,y)$, a partir dos dados armazenados de suas projeções em diversas direções. Porém, as dificuldades matemáticas e computacionais na reconstrução são aumentadas (GORDON; HERMAN, 1975) pelo fato de as projeções serem ruidosas ou possuírem erros sistemáticos,

causados por falhas no ajuste das direções das projeções e no posicionamento dos raios.

Figura 2.1: Ilustração da tomografia de transmissão

2.2 Fundamentos matemáticos da reconstrução tomográfica

O procedimento para a reconstrução a partir da transformada de Radon está esquematizado na Figura 2.2. Nela, o raio $\overline{AB}$ no plano $z = 0$ pode ser expresso matematicamente por:

$$t = x \cos \theta + y \sin \theta \quad (2.1)$$

onde t é a distância perpendicular da origem até a linha. Com o uso desta equação do raio, a integral do raio $P_\theta(t)$ é dada por:

$$P_\theta(t) = \int_{\text{linha } \overline{AB}} f(x, y) dt = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y) \delta(x \cos \theta + y \sin \theta - t) dx dy. \quad (2.2)$$

Com $P_{\theta_i}(t)$ sendo uma função de t representando a projeção paralela com ângulo θ_i . Para θ contínuo, a função $P_\theta(t)$ é a transformada de Radon de $f(x, y)$. As projeções dadas foram obtidas paralelamente à rotação no eixo x nomeadas por t .

Figura 2.2: Projeção paralela de $f(x,y)$ para Transformada de Radon

2.2.1 Teorema das secções de Fourier

O Teorema de Fourier para a secção tomográfica é a base das técnicas de reconstrução para a maioria dos algoritmos de reconstrução (MINATEL, 1997). O teorema demonstra que a Transformada de Fourier de uma projeção paralela de uma imagem $f(x,y)$, tomada de um ângulo θ , é equivalente à fatia de uma transformada bidimensional de $f(x,y)$, definida como $F(u,v)$, subentendendo-se um ângulo θ com o eixo u de forma que a Transformada de Fourier de P_θ fornece os valores sobre a linha BB' conforme ilustra a Figura 2.3 (KAK; SLANEY, 1999). Segue desta propriedade que, a partir dos dados de projeções, é possível estimar-se a imagem $f(x,y)$ simplesmente executando a transformada inversa bidimensional de Fourier.

O Teorema de Fourier para a secção tomográfica pode ser provado de forma que

Figura 2.3: Teorema das secções de Fourier

dado $F(u, v)$ como sendo a transformada de Fourier da imagem $f(x, y)$, que é definida por

$$F(u, v) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y) e^{-j2\pi(ux+vy)} dx dy \quad (2.3)$$

e sua inversa por:

$$f(x, y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} F(u, v) e^{j2\pi(ux+vy)} du dv \quad (2.4)$$

Em seguida, utiliza-se $P_{\theta}(t)$, conforme definido anteriormente, como sendo uma projeção no ângulo θ . Com isso, sua transformada de Fourier será dada por:

$$S_{\theta}(\omega) = \int_{-\infty}^{\infty} P_{\theta}(t) e^{-j2\pi\omega t} dt \quad (2.5)$$

Partindo que $\theta = 0$, considerando-se a transformada de Fourier do objeto ao longo da linha $v = 0$, no domínio da freqüência, tem-se a Transformada de Fourier de forma simplificada:

$$\begin{aligned}
F(u, 0) &= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y) e^{-j2\pi ux} dx dy \\
&= \int_{-\infty}^{\infty} \left[\int_{-\infty}^{\infty} f(x, y) dy \right] e^{-j2\pi ux} dx \\
&= \int_{-\infty}^{\infty} P_{\theta=0}(x) e^{-j2\pi ux} dx
\end{aligned} \tag{2.6}$$

No lado direito da igualdade na Equação (2.6), tem-se representada a Transformada de Fourier unidimensional da projeção $P_{\theta=0}(t)$, com isso tem-se uma função de relacionamento entre a projeção e a transformada bidimensional do objeto dada por:

$$F(u, 0) = S_{\theta=0}(u) \tag{2.7}$$

Este resultado pode ser expandido para se obter um resultado similar para θ diferente de 0. Para tanto, faz-se a rotação dos eixos de coordenadas (x, y) por um ângulo θ para formar o eixo t e s , de acordo com a matriz de rotação dada por:

$$\begin{bmatrix} t \\ s \end{bmatrix} = \begin{bmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \tag{2.8}$$

Transcrevendo (t, s) para as coordenadas (x, y) , obtém-se (KAK; SLANEY, 1999)

$$S_{\theta}(\omega) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y) e^{-j 2\pi \omega (x \cos \theta + y \sin \theta)} dx dy \tag{2.9}$$

No lado direito da igualdade na Equação (2.9), representa a transformada de Fourier bidimensional do espaço de frequência de $(u = \omega \cos \theta, v = \omega \sin \theta)$ ou

$$S_{\theta}(\omega) = F(\omega, \theta) \tag{2.10}$$

A Equação (2.10) prova o Teorema das Secções de Fourier (KAK; SLANEY, 1999), pois relaciona a Transformada de Fourier da projeção de $P_{\theta}(t)$ a valores no domínio da frequên-

Domínio da frequência

Figura 2.4: Conjunto de estimativas nas linhas radiais no espaço de frequências, das projeções de um objeto

cia da reta BB' com ângulo θ , apresentada na Figura 2.3.

Na prática, apenas um número finito de projeções é adquirido. Neste caso, é possível observar que a função $F(u, v)$ será conhecida em apenas um número finito de pontos ao longo das linhas radiais, como ilustrado na Figura 2.4, quando a implementação da Transformada de Fourier discreta for implementada computacionalmente. Assim, para utilizar-se os valores ao longo das linhas radiais será necessário o uso de interpolações lineares ou aproximações de vizinhança. A densidade de pontos radiais se torna esparsa à medida que se afasta do centro, acarretando em aumento do erro de interpolação. Com isso, o que se pode concluir é que haverá um erro maior no cálculo dos componentes de alta frequência de uma imagem do que dos de baixa. Na imagem reconstruída, isto resultará em degradações na imagem. Com isso, ao invés de fazer-se uso direto do Teorema, utilizam-se diferentes algoritmos que garantam maior precisão e rapidez na reconstrução. Dentre estes algoritmos, destaca-se o algoritmo de retroprojeção filtrada, o qual será apresentado na seção seguinte.

2.2.2 Retroprojeção Filtrada

Um dos principais algoritmos de reconstrução é o algoritmo da retroprojeção filtrada, o qual é um dos mais utilizados em aplicações que usam fontes não difrativas. Vários fatores contribuem para a ampla divulgação desse algoritmo, dentre eles a rapidez, precisão e facilidade de implementação (CAÇÃO, 1994)(BUENO, 1995).

Algumas das idéias que fundamentam o algoritmo de retroprojeção filtrada podem ser estudadas em (KAK; SLANEY, 1999) tornando mais clara a forma como o algoritmo atua na reconstrução tomográfica do objeto. O princípio desta técnica é o mesmo que o de qualquer outra em tomografia: o coeficiente de atenuação (ou densidade) é estimado pela soma do total das densidades, isto é, a soma de todos os raios que atravessam o ponto.

O algoritmo descrito é na realidade uma derivação do teorema das secções de Fourier, com uma implementação diferente do que o teorema básico sugere. Para iniciar a derivação, faz-se necessário o uso de coordenadas polares (ω, θ) no lugar do sistema de coordenada retangulares (u, v) no domínio da frequência, para reescrever-se a Equação (2.11):

$$\begin{aligned}
 f(x, y) &= \int_0^{2\pi} \int_0^\infty F(\omega, \theta) e^{j2\pi\omega(x\cos\theta + y\sin\theta)} \omega d\omega d\theta \\
 &= \int_0^\pi \int_0^\infty F(\omega, \theta) e^{j2\pi\omega(x\cos\theta + y\sin\theta)} \omega d\omega d\theta \\
 &+ \int_0^\pi \int_0^\infty F(\omega, \theta + 180^\circ) e^{j2\pi\omega[x\cos(\theta+180^\circ) + y\sin(\theta+180^\circ)]} \omega d\omega d\theta \quad (2.11)
 \end{aligned}$$

Usando $F(\omega, \theta + 180^\circ) = F(-\omega, \theta)$ e a Equação (2.1) na Equação (2.11), pode-se escrever $f(x, y)$ com a ajuda do Teorema das Secções de Fourier e a expressão para t em termos de x e y como definido pela Transformada Inversa de Fourier, ou seja:

$$\begin{aligned}
 f(x, y) &= \int_0^\pi \left[\int_{-\infty}^\infty F(\omega, \theta) |\omega| e^{j2\pi\omega t} d\omega \right] d\theta \\
 &= \int_0^\pi \left[\int_{-\infty}^\infty S_\theta(\omega) |\omega| e^{j2\pi\omega t} d\omega \right] d\theta \quad (2.12)
 \end{aligned}$$

Para construir a Equação (2.12) em sua forma filtrada retroprojetada, é necessário separar a equação em duas operações diferentes. A primeira é a filtragem dos dados de projeção para cada ângulo θ , como se segue:

$$Q_{\theta}(t) = \int_{-\infty}^{\infty} S_{\theta}(\omega) |\omega| e^{j2\pi\omega t} d\omega \quad (2.13)$$

Depois, as projeções filtradas são retroprojetadas para obter-se a função objeto

$$f(x,y) = \int_0^{\pi} Q_{\theta}(x \cos \theta + y \sin \theta) d\theta \quad (2.14)$$

Para cada *pixel* (x,y) no plano da imagem, existirá um valor de $t = x \cos \theta + y \sin \theta$ para cada projeção filtrada, Q_{θ} , obtida no ângulo θ . Cada uma destas projeções filtradas contribuirá para reconstrução do ponto (x,y) com seu valor em t . Conforme pode ser observado na Figura 2.5, todos os pontos sobre a linha LM receberão a mesma contribuição de Q_{θ_i} para o ângulo θ_i (KAK; SLANEY, 1999).

Figura 2.5: Retroprojeção dos pontos sobre a linha LM a partir do dado $Q_{\theta_i}(t)$ da projeção filtrada Q_{θ_i}

Na atual implementação, tem-se a versão truncada da Equação (2.13) como sendo (KAK; SLANEY, 1999):

$$Q_{\theta}(n\tau) = \tau \sum_{k=0}^{N-1} h(n\tau - k\tau) P_{\theta}(k\tau) \quad n = 0, 1, 2, \dots, N-1 \quad (2.15)$$

onde τ representa o intervalo de amostragem das projeções e $P_{\theta}(k\tau) = 0$ para $k < 0$ e $k > N-1$.

A função $h(n\tau)$ é a versão amostrada da resposta ao impulso, o qual é definida por:

$$h(n\tau) = \begin{cases} \frac{1}{4}\tau^2, & n = 0 \\ 0, & n = \text{par} \\ -\frac{1}{n^2\pi^2\tau^2}, & n = \text{impar} \end{cases} \quad (2.16)$$

A implementação no domínio da frequência, sob forma de equação, pode ser expressa como:

$$Q_{\theta}(n\tau) = \tau \times \text{IFFT}\{\text{FFT}[P_{\theta}(n\tau)] \times \text{FFT}[h(n\tau)]\} \quad (2.17)$$

onde FFT e IFFT representam as Transformadas Rápidas de Fourier e sua inversa, respectivamente.

O passo seguinte no algoritmo de reconstrução é a retroprojeção das projeções filtradas que tem sua aproximação discretizada por:

$$\hat{f}(x, y) = \frac{\pi}{K} \sum_{i=1}^K Q_{\theta_i}(x \cos \theta_i + y \sin \theta_i) \quad (2.18)$$

onde K ângulos θ_i são os valores discretos de Q para cada $P_{\theta}(t)$ conhecido. Em outras palavras, a imagem da reconstrução é gerada pela soma de todos os valores t de Q_{θ_i} , para cada valor θ_i , projetados e multiplicados por $\frac{\pi}{K}$.

Quando o valor de t calculado não corresponde a algum dos valores de t na função discretizada $Q_{\theta_i}(t)$, existe a necessidade de interpolação. A utilização de uma simples interpolação linear é adequada, nestes casos, na solução do problema (STARK *et al.*, 1981).

2.3 Problemas relacionados à reconstrução de imagens tomográficas

A qualidade de uma imagem tomográfica está diretamente relacionada ao processo de aquisição, bem como às características do equipamento e dos ajustes realizados no tomógrafo antes de iniciar-se a varredura de um corpo. Antes de começar-se o processo, determinam-se parâmetros, tais como a largura do feixe; a quantidade de projeções realizadas, a qual é definida pelo passo angular utilizado; a quantidade de varreduras paralelas num determinado ângulo θ , denominada passo linear; além da energia e outros parâmetros utilizados antes do início do processo. Contudo, uma determinada seleção de parâmetros, com os quais se obtém um máximo de detalhes para uma determinada amostra, pode reduzir a visibilidade de diferenças em outros corpos, como por exemplo em corpos com tecidos mais "moles".

Em comparação com outras técnicas como a da radiografia por raios X, a tomografia computadorizada geralmente possui maior contraste. Nela, cada atributo anatômico do corpo em estudo é mostrado diretamente e não sobreposto sobre outros objetos. Isto permite que seja melhorado o contraste de áreas de interesse sem a interferência de estruturas com alto coeficiente de atenuação (MINATEL, 1997). Escalas equalizadas podem ser implementadas de forma a trabalhar o contraste, permitindo a melhor visualização de tecidos/corpos mais homogêneos (PRATT, 1991)(GRANATO, 1998). A Figura 2.6 ilustra este conceito de melhoria de contraste de uma imagem tomográfica.

No processo de aquisição tomográfica, existe a presença de fatores de borramento. Esses fatores podem ter diversas causas como a inadequação:

- da largura do raio de amostragem;
- do intervalo dos raios de amostragem;
- do tamanho dos *pixels* ou *voxels*;
- dos filtros de suavização usados na reconstrução.

Figura 2.6: Imagens tomográficas de um torrão de terra adquirida com o minitomógrafo de resolução micrométrica (MACEDO, 1997) (a) Sem ajuste do contraste (b) Equalização do contraste

A largura do raio de amostragem, conhecida como abertura de amostragem, é um dos mais significativos fatores que originam borramento em uma imagem tomográfica e que limitam a boa visualização de detalhes na mesma (MINATEL, 2003). Todos os detalhes menores que a largura do raio são borrados no seu processo de medida. A abertura do detector também é um dos fatores que influenciam na largura do raio. Um detector com pequena abertura produz um raio estreito com conseqüente baixo nível de borramento e melhores detalhes. Essa abertura do detector é ajustada com uso de colimadores.

O passo linear, ou seja, a distância entre raios adjacentes, influencia na obtenção de detalhes. Se muito distantes, os detalhes entre um raio e outro são perdidos e ocorre efeito de *aliasing* (GONZALEZ; WOODS, 2000) na imagem reconstruída.

Pode-se resumir ou classificar os ruídos de um sistema de tomografia em quatro partes (MINATEL, 2003):

- *Ruído quântico*, que ocorre devido à natureza estatística de emissão e recepção de fótons;
- *Ruído do detector*, causado pela flutuação da temperatura e de interferências externas;
- *Ruído eletrônico*, causado pelos mesmos motivos do ruído do detector;
- *Ruído de reconstrução*, relacionado diretamente com o método de reconstrução envolvido.

2.4 Aplicação de tomografia na agricultura

Em diversos assuntos referentes à sustentabilidade do planeta Terra, o solo tem encontrado papel de destaque (BOUMA; HOOSBECK, 1996). Nos últimos dez anos, uma série de esforços tem sido realizada para diminuição do impacto devido ao seu uso, com foco na minimização dos problemas decorrentes da degradação por erosão devido à poluição química, e em escala não menos relevante, devido aos processos agrícolas para produção de alimentos, produção florestal e insumos para energia de biomassa.

A forma de manejo do solo tem papel definitivo nesses processos. Por exemplo, a técnica do plantio direto minimiza perdas de solo e água pela ação do escoamento superficial. Nesta técnica, mantém-se a cobertura vegetal em índices superiores a 30%, evitando exposição direta do solo as chuvas e ao sol (SSSA, 1997). O sistema de plantio convencional, em que se têm índices inferiores a 30% de cobertura vegetal, gera uma maior exposição do solo ao impacto direto da chuva, causando o encrostamento superficial do solo, provocando desta forma o escoamento superficial e a erosão (SHIPITALO *et al.*, 2000).

Além disso, na agricultura moderna, o tráfego de máquinas com peso excessivo por eixo e que trafegam quando o solo está muito úmido contribuem para a compactação do mesmo. Esta compactação promove uma alteração estrutural e reorganização das partículas do solo, causando decréscimo da disponibilidade de água e nutrientes e da difusão de gases no solo. Também neste caso, ocorre o aumento da densidade e o decréscimo do volume de poros de maior diâmetro (BEULTER; CENTURION, 2004). Os poros grandes têm um papel importante na penetração de raízes, gases e água no volume do solo. Quanto maior a densidade de macroporos, mais as raízes podem explorar o solo. Similarmente, quanto mais contínuos são os macroporos, mais livremente os gases podem realizar trocas com a atmosfera (SHIPITALO *et al.*, 2000). Na planta, a compactação do solo reduz o crescimento radicular por impedimento mecânico, aeração deficiente e menor taxa de absorção de água e nutrientes, causando decréscimos significativos de produtividade (BEUTLER *et al.*, 2006).

Observa-se que a determinação de parâmetros físicos como a umidade e a densidade

do solo, por exemplo, são de grande importância no monitoramento hídrico de áreas agrícolas, bem como em estudos que enfoquem a relação solo-água-planta (FERREIRA *et al.*, 1998). A densidade corresponde à massa do solo seco por volume do solo. É uma propriedade variável e depende da estrutura e compactação deste. O material constituinte tem grande influência sobre seu valor, assim como os sistemas de manejo e tipo de cobertura vegetal.

Os valores de densidade podem ser extremamente variáveis. Pode-se ter, em solos de mesma textura ¹, densidades diferenciadas no perfil. Além disso, seus valores tendem a aumentar com a profundidade o que se deve a fatores como: teor reduzido de matéria orgânica, menor agregação, menor penetração de raízes, maior compactação, ocasionada pelo peso das camadas subjacentes, dentre outros.

Existem vários métodos diretos e indiretos que viabilizam a medida de parâmetros físicos de solos. Dentre eles, há diferentes vantagens e limitações (FERREIRA *et al.*, 1998). Dentre os métodos diretos, pode-se ressaltar o método gravimétrico, considerado padrão, que é demorado, destrutivo e não permite a repetição da amostragem no mesmo local. Entre os métodos indiretos, a utilização da sonda de nêutrons se destaca por permitir a aferição da umidade do solo com o mínimo de alteração no perfil. A técnica da moderação de nêutrons utiliza a relação de dependência entre o conteúdo volumétrico de água no solo e a contagem relativa (contagem solo/contagem no padrão). Uma sonda de nêutrons consiste de uma fonte radioativa que emite nêutrons rápidos, um detector de nêutrons lentos e um pré-amplificador, cujo sinal é conduzido ao sistema eletrônico de contagem. Desta forma, nêutrons rápidos (alta energia) são emitidos por esta fonte, interagindo com o meio ao redor. Através das colisões, principalmente com os átomos de hidrogênio presentes na água, nêutrons rápidos se tornam lentos (perdem energia) e retornam ao sistema de contagem, fornecendo a taxa de contagem, que, por sua vez, é relacionada com o teor de água do solo. Esta técnica tem sido utilizada há mais de cinco décadas para a determinação do conteúdo de água no solo, mas vários aspectos ainda apresentam dificuldade,

¹A textura do solo refere-se à proporção relativa em que se encontram, em determinada massa de solo, os diferentes tamanhos de partículas. Refere-se, especificamente, às proporções relativas das partículas ou frações de areia, silte (fragmento de mineral ou rocha menor com diâmetro entre 0,053 mm e 0,002 mm) e argila na terra fina seca ao ar.

tais como determinação da umidade em camadas superficiais do solo, riscos com o manuseio por tratar-se de material radioativo e a calibração do equipamento, sendo esta última talvez a mais crítica das desvantagens dessa técnica (TEIXEIRA *et al.*, 2005).

Na busca de técnicas mais apuradas para determinação e avaliação de parâmetros físicos do solo com aplicabilidade em diversos tipos de terrenos (CRESTANA *et al.*, 1996), vem se destacando há algum tempo o uso da tomografia computadorizada.

A tomografia computadorizada, como um novo método de análise e investigação na física de solos, foi introduzida por Petrovic (PETROVIC *et al.*, 1982), Aylmore (AYLMORE; HAINSWORTH, 1983) e Crestana (CRESTANA, 1985). A introdução desta técnica, até então inédita, resultou em maior precisão e vantagens com relação a métodos clássicos.

Dentre os parâmetros de interesse, pode-se destacar o uso das técnicas de tomografia para estudo:

- Compactação;
- Penetração de raízes;
- Encrostamento;
- Ciclos de umedecimento e secagem;
- Fluxos preferenciais de poluentes em solos fraturados.

Uma das vantagens do uso da técnica tomográfica para avaliar estes parâmetros é a boa resolução espacial conseguida (PEDROTTI *et al.*, 2003).

No Brasil, têm sido desenvolvidos tomógrafos de uso dedicado ao estudo de solos. Dentre os trabalhos desenvolvidos, encontra-se o minitomógrafo de raios X e γ que se constituiu num importante e pioneiro passo na aplicação das técnicas tomográficas e realização de medidas de amostras de solo em laboratório. A Figura 2.7 ilustra o minitomógrafo de raios X e γ (CRUVINEL, 1987) (CRUVINEL *et al.*, 1990).

Figura 2.7: Minitomógrafo de resolução milimétrica de laboratório (CRUVINEL, 1987)

Em 1994, Naime e colaboradores desenvolveram um minitomógrafo portátil, o que deu agilidade ao estudo de amostras de solo uma vez que permitia o estudo em campo. A figura 2.8 ilustra o minitomógrafo portátil (NAIME, 1994) (NAIME *et al.*, 1996)

Figura 2.8: Minitomógrafo portátil para estudo de solo e plantas, em campo (NAIME, 1994)

Em 1997, Silva e colaboradores desenvolveram um tomógrafo de laboratório com resolução micrométrica, mostrado na Figura 2.9(a). Com isso, tornou-se possível visualizar e estudar a geometria dos poros de amostras de solo, bem como distribuição e seu tamanho. Trabalhando nesta escala, tal equipamento permitiu também o estudo de fenômenos do selamento superficial em amostras de solo (MACEDO, 1997) (MACEDO *et al.*, 1997).

Em 2001, Naime e colaboradores (NAIME, 2001) construíram um tomógrafo de campo com esquema de varredura em leque, conforme ilustra a Figura 2.9(b). A grande diferença em relação aos tomógrafos que possuem esquema de varredura de feixe fino é que a aquisição dos dados é feita de forma mais rápida, dado que o feixe envolve toda a amostra e exige um número menor de movimentações do conjunto fonte-detecores para realizar a varredura completa da amostra.

O desenvolvimento deste equipamento deu-se devido à inexistência de instrumentação disponível e adequada para realizar uma varredura, suficientemente rápida para monitoramento e medição de forma não destrutiva, em duas e três dimensões, do movimento da água no solo na região não saturada e permitir a estimativa das propriedades hidráulicas do mesmo.

Mais recentemente, a partir de 2001, a Embrapa Instrumentação Agropecuária vem desenvolvendo um novo minitomógrafo baseado no método da tomografia Compton (CRUVINEL; BALOGUN, 2000) (CRUVINEL; BALOGUN, 2006). O efeito Compton foi apresentado pela primeira vez por Compton e Hagenow (1924). As técnicas convencionais de tomografia por transmissão e os modelos convencionais de tomógrafo são baseados no uso de fonte e detector em lados opostos. Esses modelos nem sempre podem ser utilizados em aplicações agrícolas, como por exemplo, a extração de medidas de solo diretamente no campo. A tomografia Compton possui fonte e detector situados do mesmo lado da amostra. Desta forma, não existe a necessidade, nesse modelo de tomógrafo, de se abrir trincheiras para análise de solo, como mostrado na Figura 2.8. Comparação entre esta técnica e a técnica de tomografia por transmissão foi realizada por Cruvinel e Balogun, utilizando o tomógrafo mostrado na Figura 2.10. Os resultados mostraram um adequado coeficiente de correlação linear entre as duas técnicas para

(a)

(b)

Figura 2.9: Novos tomógrafos da Embrapa - (a)Minitomógrafo de resolução micrométrica (MACEDO, 1997) - (b)Minitomógrafo de varredura em leque, desenvolvido por Naime (NAIME, 2001)

diversos estudos realizados em diferentes amostras de solo.

Figura 2.10: Minitomógrafo Compton desenvolvido por Cruvinel e Balogun (CRUVINEL; BALOGUN, 2006)

Recentemente, em 2006, parte dos tomógrafos que foram desenvolvidas pela Embrapa Instrumentação Agropecuária, foram transferidos para a iniciativa privada (ERENO, 2006). Os indícios mostram que, além das aplicações ligadas ao solo, o mercado principal deste tipo de tecnologia será a aplicação em processos industriais, direcionados principalmente à qualidade de madeira e de peças cerâmicas. Além disso, outro possível nicho da tomografia está na avaliação das condições de árvores plantadas em áreas urbanas para ornamentação, e que constantemente são expostas à poluição e a ataques de coleópteros (CRUVINEL *et al.*, 2003).

2.5 Reconstrução tridimensional de amostras agrícolas

As amostras adquiridas pelos tomógrafos da Embrapa Instrumentação e utilizadas neste trabalho possuem a característica de não deslocar a posição de análise durante o processo de aquisição. Tal característica garante que mesmo em fatias obtidas em diferentes alturas na amostra, mantenha-se a relação espacial entre os dados dos planos tomografados. Quando não ocorre movimentação na aquisição das fatias tomográficas, a reconstrução tridimensional pode

ser feita também a partir da sobreposição de fatias bidimensionais. Essa técnica consiste em montar os planos gerados pelas funções $f(x, y, z_i)$ para $i = 0, 1, 2, \dots, n$, tal qual mostrado na Figura 2.11.

Figura 2.11: Ilustração da interpolação tridimensional a partir de fatias reconstruídas

Partindo-se deste princípio, realiza-se a geração de imagens tomográficas tridimensionais utilizando-se um método de interpolação de dados que se baseia em dados realmente adquiridos para gerar os planos intermediários.

A interpolação consiste em estimar dados intermediários com base nos dados previamente conhecidos preenchendo as lacunas desconhecidas a respeito dos dados e da função que os representa. Os métodos de interpolação de dados desenvolvidos inicialmente auxiliavam na determinação do posicionamento de corpos celestes e épocas corretas de plantio (MEIJERING, 2002). Nos dias atuais, algumas soluções baseadas em interpolações de dados para a área agrícola contribuem para a melhor compreensão a respeito das características do solo, tais como porosidade, dinâmica e também na visualização das estruturas internas que o compõem (MINATEL, 1997) (MINATEL, 2003)(PEREIRA *et al.*, 2001).

Neste trabalho, utiliza-se a técnica de interpolação por *B-Spline-Wavelet*, ou simplesmente *B-Wavelet*. Esta técnica demonstrou-se de grande precisão na determinação de características sub-pixel de imagens tomográficas. Algumas informações adicionais a respeito do

uso desta técnica na área de processamento de imagens podem ser encontradas em (UNSER, 1997), (MINATEL; CRUVINEL, 1998) e (VELHO; PERLIN, 2002).

Realiza-se na próxima seção uma revisão dos conceitos matemáticos envolvidos no algoritmo de interpolação por *B-Wavelet*, bem como apresenta-se um exemplo passo a passo da aplicação do algoritmo.

2.5.1 Método de interpolação baseada em *B-Wavelet*

A interpolação por *B-Wavelet* determina um conjunto de valores intermediários entre uma sequência de pontos conhecida. Diferentemente da aproximação, a interpolação não apenas desloca a curva gerada sob a influência dos pontos conhecidos, como também faz com que essa curva passe por esses pontos. A Figura 2.12 ilustra esta diferença.

Figura 2.12: Ilustração apresentando a diferença existente entre a interpolação e aproximação por *B-Wavelet*.

Para descrever a interpolação, faz-se necessária uma discussão sobre o método de aproximação, seus problemas e suas soluções computacionais.

Seja f uma função de aproximação Spline de ordem m e com passo μ dada por:

$$f(\mu) = \sum_{i=0}^N a_i B(N\mu - i) \quad (2.19)$$

onde N é o número de pontos conhecidos.

De forma a otimizar o processo de cálculo, implementa-se a função $B(x)$, também chamada de função de *blending*, da seguinte forma:

$$B(x) = \begin{cases} \frac{1}{6}(2+x)^3 & -2 < x \leq -1 \\ \frac{1}{6}(4-6x^2-2x^3) & -1 < x \leq 0 \\ \frac{1}{6}(4-6x^2+2x^3) & 0 < x \leq 1 \\ \frac{1}{6}(2-x)^3 & 1 < x < 2 \\ 0 & 2 \leq |x| \end{cases} \quad (2.20)$$

A implementação direta desta somatória fornece os valores intermediários da aproximação. No entanto, para os pontos inicial e final para $m = 4$, por exemplo, têm-se os valores:

$$f(0) = \frac{2}{3}a_0 + \frac{1}{6}a_1 \quad (2.21)$$

e

$$f(1) = \frac{2}{3}a_N + \frac{1}{6}a_{N-1} \quad (2.22)$$

Para solucionar esse problema, nesta implementação, utilizou-se de pontos não existentes no conjunto original, os chamados "pontos fantasmas". Essa técnica consiste em considerar um ou mais valores antes do ponto inicial e depois do final para garantir a passagem pelos pontos desejados. Com isso, foram adicionados dois "pontos fantasmas" no início e dois no final da seqüência conhecida. Esses valores foram ajustados de forma dinâmica para que a curva a ser gerada pela função *B-Wavelet* passasse pelos pontos inicial e final. Os pontos iniciais têm seus valores estabelecidos pela regra:

$$a_{-1} = 2a_0 - a_1$$

$$a_{-2} = 2a_{-1} - a_0$$

e os pontos finais, por :

$$a_{N+1} = 2a_N - a_{N-1}$$

$$a_{N+2} = 2a_{N+1} - a_N$$

A função mostrada na Equação (2.19) é alterada, ficando da seguinte forma:

$$f(\mu) = \sum_{i=-2}^{N+2} a_i B(N\mu - i) \quad (2.23)$$

Para a implementação da interpolação, ao invés da simples aproximação, usa-se uma função muito próxima da função de aproximação. Essa função é dada por:

$$f(\mu) = \sum_{i=-2}^{N+2} A_i B_m(N\mu - i) \quad (2.24)$$

A idéia básica é usar a seqüência de pontos A_i , denominada pontos de controle, no lugar de a_i . Essa seqüência é dada pela multiplicação das matrizes

$$A_i = a_i M^{-1} \quad (2.25)$$

onde A é o vetor seqüência a ser encontrado, a é o conjunto de pontos conhecidos com os "pontos fantasmas" e M é uma matriz $N + 4 \times N + 4$ dada por:

$$M = \begin{bmatrix} \frac{N}{2} & -\frac{N}{2} & 0 & 0 & 0 & \dots & 0 & 0 \\ \frac{1}{6} & \frac{2}{3} & \frac{1}{6} & 0 & 0 & \dots & 0 & 0 \\ 0 & \frac{1}{6} & \frac{2}{3} & \frac{1}{6} & 0 & \dots & 0 & 0 \\ 0 & \dots & \dots & \dots & \dots & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & \frac{1}{6} & \frac{2}{3} & \frac{1}{6} & 0 \\ 0 & 0 & 0 & \dots & 0 & \frac{1}{6} & \frac{2}{3} & \frac{1}{6} \\ 0 & 0 & 0 & \dots & 0 & 0 & -\frac{N}{2} & \frac{N}{2} \end{bmatrix} \quad (2.26)$$

Para melhor compreensão da interpolação por *B-Wavelet*, apresenta-se um exemplo com valores numéricos e gráficos que demonstram a forma como ocorre o processo de interpolação de planos virtuais entre planos reais.

Exemplificando o uso da técnica, pode-se considerar que, a partir de um conjunto de cinco planos reais, obtêm-se os valores dos pixels de coordenadas (x,y) , inicialmente tendo-se o seguinte vetor:

50	60	140	105	100
----	----	-----	-----	-----

Graficamente, o vetor de pontos é representado por pontos azuis na Figura 2.13, onde o eixo de coordenadas relaciona os planos e intensidade dos pixels.

Figura 2.13: Vetor com os valores originais.

Seguindo o algoritmo, calculam-se os 4 "pontos fantasmas" que garantirão a passagem da interpolação pelos pontos a_0 e a_n , obtendo-se o vetor abaixo.

30	40	50	60	140	105	100	95	90
----	----	----	----	-----	-----	-----	----	----

e que está representado na Figura 2.14 com os pontos vermelhos representando os "pontos fantasmas".

A partir do vetor composto pelos "pontos fantasmas" e pelos pontos reais, calcula-se o vetor A_i que será utilizado para gerar a função de interpolação. Para isto, utiliza-se a Equação 2.25, a qual faz uso dos pontos reais e da matriz inversa de M , definida na Equação 2.26. Para o conjunto de pontos utilizado neste exemplo, o vetor A_i correspondente é dado por:

3.9113	74.394	48.032	33.48	178.05	94.325	74.651	207.07	12.331
--------	--------	--------	-------	--------	--------	--------	--------	--------

Na Figura 2.15(a), vê-se a apresentação destes pontos do vetor A_i como asteriscos

Figura 2.14: Inserção de pontos fantasmas.

verdes, juntamente com os "pontos fantasmas" e os pontos reais.

Utilizando-se os pontos do vetor A_i e a função de *Blending*, calcula-se a função de interpolação $f(\mu)$ para cada ponto interpolado, tal qual é definido na Equação 2.20. A Figura 2.15(b) apresenta a interpolação de pontos, incluindo os "pontos fantasmas". Na Figura 2.15(c), é mostrada a função de interpolação obtida através dos pontos reais e, na 2.15(d), apresenta-se um conjunto de 4 pontos interpolados a cada par de pontos reais com a mesma função.

Observa-se que a inserção dos 4 "pontos fantasmas" realmente garante a passagem da função de interpolação pelos pontos inicial e final.

Figura 2.15: Função obtida com a interpolação por *B-Wavelets* - (a) Visualização dos pontos do vetor A_i juntamente com os demais pontos; (b) Função de interpolação incluindo pontos fantasmas e as retas que conectam os pontos de controle; (c) Função gerada pela interpolação; (d) Interpolação de 4 pontos entre cada par de pontos.

3 *Filtragem de projeções tomográficas a priori na reconstrução das imagens*

3.1 Fundamentos

Este capítulo a respeito de filtragem *a priori* aborda durante o seu desenvolvimento um conjunto de conceitos de cunho estatístico, como estacionaridade, autocorrelação e correlação cruzada. Assim, uma sucinta apresentação destes conceitos é realizada a seguir.

3.1.1 Estacionaridade no sentido amplo

Um processo de série de tempo é considerado estacionário se sua média e variância são constantes ao longo do tempo e se a autocorrelação, considerando dois períodos de tempo, depende somente da distância entre os pontos no tempo e não do período de tempo efetivo para o qual variância é computada.

3.1.2 Coeficiente de autocorrelação

O coeficiente de autocorrelação $r_s(k)$ de uma série temporal S_t pode ser estimado através da Equação (3.1), onde $s_t = S_t - \bar{S}$, sendo $\bar{S}$ a média da série S_t

$$r_s = \frac{\sum_{t=k}^{n-1} s_t s_{t-k}}{\sum_{t=0}^{n-1} s_t^2} \quad (3.1)$$

3.1.3 Coeficiente de correlação cruzada

A função de correlação cruzada entre uma variável S_t e Y_t mede a correlação entre séries em diferentes períodos do tempo (SILVEIRA, 2004). A Equação (3.3) define a função de covariância cruzada, c_{sy} , de s_t e y_t e, na Equação (3.2), r_{sy} representa a função de correlação cruzada de S_t e Y_t . Em ambas, tem-se $k = 0, \pm 1, \pm 2, \dots, n$ sendo que μ_s e μ_y são as médias e σ_s e σ_y , os desvios-padrão das séries estacionárias S_t e Y_t

$$r_{sy}(k) = \frac{c_{sy}(k)}{\sigma_s \sigma_y} \quad (3.2)$$

$$c_{sy}(k) = E \{ [s_t - \mu_s] [y_{t+k} - \mu_y] \} \quad (3.3)$$

3.2 Filtragem *a priori*

Uma das principais limitações para a precisão da medida tomográfica computadorizada é a natureza estatística no processo de produção de fótons. A probabilidade de detecção de ϕ fótons em um intervalo de tempo de exposição t pode ser estimada pela função distribuição de probabilidade de Poisson dada pela Equação 3.4 (DERENIAK, 1984) (CRUVINEL, 1987):

$$P(\phi, t) = \frac{(\bar{\phi})^\phi}{\phi! e^{\bar{\phi}}} \quad (3.4)$$

onde ϕ é o número de fótons e $\bar{\phi}$ é a média de fotoelétrons emitidos no intervalo de tempo t , segundo a expressão

$$\bar{\phi} = \xi R t \quad (3.5)$$

onde R é a razão média de fótons (fótons/segundo) e ξ é a eficiência quântica da fotomultiplicadora (GRANATO, 1998).

Aumentar o tempo de exposição pode melhorar a relação sinal ruído, no entanto,

isto implica em maior tempo de exposição à radiação (LI *et al.*, 2001). Alternativamente, pode-se suprimir este tipo de ruído trabalhando-se a filtragem *a priori* das projeções, conforme apresenta a Figura 3.1.

Figura 3.1: Diagrama da filtragem *a priori* das projeções

A aplicação de filtros determinísticos ou preditivos reduz os efeitos do ruído Poisson nas projeções. Uma vez realizada a filtragem, o passo seguinte é aplicação do algoritmo de reconstrução para obtenção da imagem reconstruída.

3.2.1 Transformação de Anscombe

O ruído Poisson é caracterizado por ser dependente do sinal, uma vez que a sua variância depende do valor médio do sinal. Contudo, a maioria dos métodos de redução de ruído atualmente disponíveis, baseiam-se em sinais que possuem ruídos independentes do sinal com distribuição gaussiana estacionária (LI *et al.*, 2001).

Uma alternativa para contornar tal problema envolve o uso da transformada de Anscombe (AT) que transforma o ruído Poisson dependente do sinal em um que é aproximadamente gaussiano, aditivo, com média zero e variância unitária (ANSCOMBE, 1948) (HOMEM *et al.*, 2002).

Se x é uma variável aleatória de distribuição Poisson, sua AT é definida como:

$$y_i = 2\sqrt{x_i + \frac{3}{8}} \quad (3.6)$$

onde y_i representa a nova distribuição. Esta pode ser representada também por um modelo aditivo, tal qual apresentado na Equação (3.7)

$$y_i = 2\sqrt{x_i + \frac{1}{8}} + v_i = s_i + v_i \quad (3.7)$$

Na Equação (3.7), v_i representa um ruído aditivo que é aproximadamente indepen-

dente de s_i , com distribuição gaussiana, de média zero e variância unitária.

Retorna-se à variável original aplicando-se em y_i a inversa da AT, definida como:

$$b_i = \frac{1}{4}y_i^2 - \frac{1}{8} \quad (3.8)$$

3.3 Métodos de filtragem

No processamento de sinais, os filtros são responsáveis pela remoção de frequências indesejadas no sinal, como ruídos e interferências. Basicamente, no caso de filtragem linear, trabalha-se restringindo a passagem de frequências específicas, utilizando filtros clássicos, como o filtro passa-baixa, passa-alta e passa-banda (GONZALEZ; WOODS, 2000).

Por outro lado, existem paradigmas que se baseiam em técnicas não lineares, como por exemplo, o filtro por mediana. Também incluem-se na categoria dos não lineares os filtros preditores, como o filtro de Wiener e Kalman, por exemplo. Entretanto diferentemente do filtro mediana, aqueles utilizam-se de características estatísticas do sinal, para eliminação do ruído.

3.3.1 Filtro Mediana

O filtro mediana unidimensional utiliza máscaras de dimensão $1 \times p$ para trabalhar o sinal. A cada etapa, encaixam-se p valores em sua máscara, que posteriormente são ordenados de forma crescente ou decrescente. Em seguida, extrai-se o valor mediano do conjunto, que substituirá, no sinal original, o valor que está na posição central da máscara. A intensidade do efeito do filtro é controlada pelo tamanho da janela aplicada.

3.3.2 Filtros preditivos

A estimação de um sinal a partir de um outro é um dos mais importantes problemas em processamento de sinais, abrangendo um amplo conjunto de aplicações, que permeiam áreas como o tratamento de sinais em processamento de imagens (MASCARENHAS *et al.*, 1999), a bioengenharia (HOMEM *et al.*, 2002) dentre outras. Nestas aplicações, em grande parte das

situações, o sinal desejado pode estar, por várias razões, corrompido ou ruidoso (HAYES, 1996). Em ambientes simples ou idealizados, pode-se fazer uso de filtros lineares para restaurar o sinal desejado a partir das medidas realizadas do sinal. Contudo, raramente estes filtros serão ótimos no sentido de produzir a melhor estimativa do sinal desejado.

Na década de 1940, Norbert Wiener (WIENER, 1949) foi pioneiro na pesquisa para elaboração de um filtro que produziria a estimativa ótima de um sinal ruidoso. Dentre os filtros preditivos, destacam-se o filtro de Wiener para cancelamento de ruído, filtragem não linear (HAYES, 1996) e o filtro de Kalman (KALMAN, 1960)(WELCK; BISHOP, 2001).

3.4 Filtro de Wiener

A Figura 3.2 exibe o problema do filtro de Wiener, que tem como objetivo recuperar um sinal desejado, dado por $d(n)$, de uma observação com ruído $x(n)$

$$x(n) = d(n) + v(n) \quad (3.9)$$

Figura 3.2: Ilustração de um problema geral do filtro de Wiener. Dados dois processos estacionários, $x(n)$ e $d(n)$, que são estatisticamente relacionados entre si, o filtro $W(z)$ minimiza a estimativa do erro médio quadrático, $\hat{d}(n)$, de $d(n)$ (HAYES, 1996)

Assumindo que $d(n)$ e $v(n)$ são processos aleatórios estacionários, a elaboração do filtro consiste em minimizar o valor esperado do erro médio quadrático da estimativa de $d(n)$ (HAYES, 1996). Deste modo, com ξ dado por:

$$\xi = E \{ |e(n)|^2 \} \quad (3.10)$$

onde $e(n)$ é definido por

$$e(n) = d(n) - \hat{d}(n) \quad (3.11)$$

o problema na filtragem de Wiener reside em encontrar o filtro que minimize ξ

3.4.1 Filtro de Wiener FIR

A função do sistema para um filtro de ordem p é dada por

$$W(z) = \sum_{n=0}^{p-1} w(n)z^{-n} \quad (3.12)$$

sendo $w(n)$ a resposta à amostra unitária do filtro.

A saída do filtro, $\hat{d}(n)$, é dada pela convolução de $w(n)$ com $x(n)$, a entrada do filtro

$$\hat{d}(n) = \sum_{l=0}^{p-1} w(l)x(n-l) \quad (3.13)$$

Os coeficientes do filtro devem ser determinados de tal forma que minimizem o erro

$$\xi = E \{ |e(n)|^2 \} = E \{ |d(n) - \hat{d}(n)|^2 \} \quad (3.14)$$

Para tanto, é necessário e suficiente que a derivada de ξ em relação a $w^*(k)$ seja igual a zero para $k = 0, 1, \dots, p-1$ (HAYES, 1996):

$$\frac{\partial \xi}{\partial w^*(k)} = \frac{\partial}{\partial w^*(k)} E \{ e(n)e^*(n) \} = E \left\{ e(n) \frac{\partial e^*(n)}{\partial w^*(k)} \right\} = 0 \quad (3.15)$$

Com

$$e(n) = d(n) - \sum_{l=0}^{p-1} w(l)x(n-l) \quad (3.16)$$

segue que

$$\frac{\partial e^*(n)}{\partial w^*(k)} = -x^*(n-k) \quad (3.17)$$

e a Equação (3.15) torna-se

$$E \{e(n)x^*(n-k)\} = 0 \quad ; \quad k = 0, 1, 2, \dots, p-1 \quad (3.18)$$

Substituindo a Equação (3.16) na Equação (3.18), tem-se:

$$E \left\{ \left[d(n) - \sum_{l=0}^{p-1} w(l)x(n-l) \right] x^*(n-k) \right\} = 0 \quad (3.19)$$

obtendo-se em seguida

$$E \{d(n)x^*(n-k)\} - \sum_{l=0}^{p-1} w(l)E \{x(n-l)x^*(n-k)\} = 0 \quad (3.20)$$

A equação 3.19 apresenta o princípio da ortogonalidade, também conhecido como teorema da projeção.

Dado que $x(n)$ e $d(n)$ são processos estocásticos conjuntamente estacionários no sentido amplo, tem-se que na Equação (3.20), o termo $E \{x(n-l)x^*(n-k)\} = r_x(k-l)$, onde $r_x(k-l)$ denota a função de autocorrelação do sinal de entrada $x(n)$. O segundo termo da Equação (3.20) é $E \{d(n)x^*(n-k)\} = r_{dx}(k)$, onde $r_{dx}(k)$ denota a função de correlação cruzada entre o sinal desejado $d(n)$ e o sinal de entrada $x(n)$ (HAYES, 1996) (VEIGA, 2002). Desta maneira, a Equação (3.20) pode ser reescrita como sendo

$$\sum_{k=0}^{p-1} w(l)r_x(k-l) = r_{dx}(k) \quad ; \quad k = 0, 1, 2, \dots, p-1 \quad (3.21)$$

A Equação (3.21) corresponde a um conjunto de p equações lineares com p incógnitas $w(k)$, com $k = 0, 1, \dots, p-1$.

Como nesta tese serão tratados apenas sinais com valores reais, considerar-se-á a

partir deste ponto que os coeficientes do filtro de Wiener assumirão apenas valores reais. Deste modo, a Equação (3.21), na forma matricial, torna-se:

$$\begin{bmatrix} r_x(0) & r_x(1) & \dots & r_x(p-1) \\ r_x(1) & r_x(0) & \dots & r_x(p-2) \\ \dots & \dots & \dots & \dots \\ r_x(p-1) & r_x(p-2) & \dots & r_x(0) \end{bmatrix} \begin{bmatrix} w(0) \\ w(1) \\ \dots \\ w(p-1) \end{bmatrix} = \begin{bmatrix} r_{dx}(0) \\ r_{dx}(1) \\ \dots \\ r_{dx}(p-1) \end{bmatrix} \quad (3.22)$$

A Equação (3.22) corresponde à forma matricial das equações de Wiener-Hopf, as quais podem ser escritas de maneira concisa como

$$R_x w = r_{dx} \quad (3.23)$$

onde R_x é a matriz de dimensão $p \times p$ de valores de coeficientes de autocorrelações, w é o vetor de coeficientes do filtro e r_{dx} é o vetor com os coeficientes de correlações cruzadas entre o sinal desejado $d(n)$ e o sinal observado $x(n)$.

A determinação do vetor de coeficientes do filtro pode ser realizada aplicando-se a Equação (3.24)

$$w = R_x^{-1} r_{dx} \quad (3.24)$$

Utilizando-se os valores dos coeficientes, pode-se obter o erro médio quadrático mínimo, ξ_{min} , de estimativa de $d(n)$ e que pode ser avaliado através da Equação (3.25) (HAYES, 1996).

$$\xi_{min} = r_d(0) - \sum_{l=0}^{p-1} w(l) r_{dx}(l) \quad (3.25)$$

3.4.1.1 Filtragem

Em problemas de filtragem, um sinal $d(n)$ é estimado a partir de medidas ou observações que estão corrompidas por ruído, tal qual apresentado na Equação (3.9).

Utilizando-se os resultados apresentados na seção 3.4.1, o filtro FIR ótimo de Wiener pode ser facilmente derivado. Assumindo-se que o ruído tem média zero e é decorrelacionado do sinal desejado $d(n)$, é possível afirmar que $E\{d(n)v(n-k)\} = 0$. Redefinindo o coeficiente de correlação cruzada entre $d(n)$ e $x(n)$, tem-se

$$\begin{aligned} r_{dx} &= E\{d(n)x(n-k)\} \\ &= E\{d(n)d(n-k)\} + E\{d(n)v(n-k)\} \\ &= r_d(k) \end{aligned} \quad (3.26)$$

Em seguida, define-se r_x e tem-se que:

$$r_x = E\{x(n+k)x(n)\} \quad (3.27)$$

$$\begin{aligned} &= E\{[d(n+k) + v(n+k)][d(n) + v(n)]\} \\ &= r_d(k) + r_v(k) \end{aligned} \quad (3.28)$$

Desta forma, as equações de Wiener-Hopf tornam-se

$$[R_d + R_v]w = r_d \quad (3.29)$$

3.4.1.2 Predição Linear

Com observações sem ruído, a predição linear busca estimar o valor $x(n+1)$ em termos de uma combinação linear do valor corrente e dos valores anteriores a $x(n+1)$, tal qual mostra a Figura (3.3). Desta forma, um preditor linear FIR de ordem $p-1$ é formado como

$$\hat{x}(n+1) = \sum_{l=0}^{p-1} w(l)x(n-l) \quad (3.30)$$

onde $w(k)$ para $k = 0, 1, \dots, p-1$ são coeficientes do filtro preditor. Assim, a predição linear pode ser modelada com uma filtragem de Wiener onde se ajusta $d(n) = x(n+1)$.

Figura 3.3: Predição linear como um tipo de problema onde busca-se encontrar a estimativa $x(n+1)$ através da combinação linear de p valores de $x(n)$ até $x(n-p+1)$ (HAYES, 1996)

Reavaliando-se o coeficiente de correlação cruzada entre $d(n)$ e $x(n)$, obtém-se

$$r_{dx}(k) = E \{d(n)x(n-k)\} = E \{x(n+1)x(n-k)\} = r_x(k+1) \quad (3.31)$$

e as equações Wiener-Hopf para o preditor linear ótimo serão definidas como

$$\begin{bmatrix} r_x(0) & r_x(1) & \dots & r_x(p-1) \\ r_x(1) & r_x(0) & \dots & r_x(p-2) \\ \dots & \dots & \dots & \dots \\ r_x(p-1) & r_x(p-2) & \dots & r_x(0) \end{bmatrix} \begin{bmatrix} w(0) \\ w(1) \\ \dots \\ w(p-1) \end{bmatrix} = \begin{bmatrix} r_x(1) \\ r_x(2) \\ \dots \\ r_x(p) \end{bmatrix} \quad (3.32)$$

O erro médio quadrático mínimo é dado por

$$\xi_{min} = r_d(0) - \sum_{l=0}^{p-1} w(l)r_x(k+1) \quad (3.33)$$

Assumindo-se um modelo de predição mais realista, na qual utilizam-se observações obtidas na presença de ruído, tem-se a entrada do filtro dada por $y(n)$

$$y(n) = x(n) + v(n) \quad (3.34)$$

O objetivo é projetar um filtro que estime $x(n+1)$ baseando-se nas combinações lineares de p valores anteriores de $y(n)$. Desta maneira, a equação (3.30) é ajustada de modo a inserir os valores do ruído na predição, obtendo-se

$$\hat{x}(n+1) = \sum_{l=0}^{p-1} w(l)y(n-l) = \sum_{l=0}^{p-1} w(l)[x(n-l) + v(n-l)] \quad (3.35)$$

As equações de Wiener-Hopf, no caso do preditor linear com presença de ruído, então, serão dadas por

$$R_y w = r_{dy} \quad (3.36)$$

Se o ruído é descorrelacionado do sinal $x(n)$, então R_y tornar-se-á

$$r_y = E \{y(n)y(n-k)\} = r_x(k) + r_v(k) \quad (3.37)$$

e o vetor r_{dy} será dado por

$$r_{dy} = E \{d(n)y(n-k)\} = E \{x(n+1)y(n-k)\} = r_x(k+1) \quad (3.38)$$

As equações de Wiener-Hopf são

$$R_y w = r_{dy} \quad (3.39)$$

Portanto, a única diferença entre a predição linear com e sem presença de ruído reside na matriz de autocorrelação para o sinal de entrada, em que, para o caso de ruído descorrelacionado com $x(n)$, R_x é substituída por $R_y = R_x + R_v$

4 *Processamento Paralelo*

4.1 Introdução

Segundo a definição de Hwang (HWANG, 1984) o processamento paralelo é uma forma eficiente de trabalhar a informação com ênfase na exploração de eventos concorrentes no processo computacional. A idéia do processamento paralelo não é nova; em 1920, Vannevar Bush do *Massachusetts Institute of Technology* (MIT) já apresentava um computador analógico capaz de resolver equações diferenciais em paralelo. Em seus artigos no ano de 1940, Von Neumann, sugere uma grade para resolver equações diferenciais em que os pontos são atualizados em paralelo (NAVAUX, 1989).

A razão comercial do surgimento do processamento paralelo está na possibilidade de aumentar a capacidade de processamento com uma única máquina. Porém, arquiteturas sequenciais de modelo Von Neumann ainda são maioria entre as máquinas utilizadas comercialmente. Nos últimos anos, tem ocorrido uma evolução dessa arquitetura, com vista a explorar de forma mais racional a capacidade do processador, associada a uma melhor organização da estrutura de memória e dos outros componentes do sistema. Algumas alternativas têm sido implementadas com esse objetivo, dentre elas destacam-se:

- Uso de memórias *cache* em vários níveis;
- *Pipeline* de instruções

O uso dessas alternativas nos processadores tem contribuído no aumento da velocidade dos sistemas monoprocessados.

A utilização de vários níveis de memórias *cache* contribui no intuito de diminuir o número de ciclos em que o processador fica aguardando o dado vir dos níveis inferiores de memória para os seus registradores para que possam ser processados. As memórias *cache* são pequenas, mas rápidas unidades de memória colocadas entre a CPU e a memória principal que, em geral, armazenam as instruções e dados utilizados mais recentemente. A idéia na sua utilização é fazer uso do princípio da localidade, que se fundamenta na idéia de que a CPU acessa um relativamente pequeno espaço de endereçamento num determinado intervalo de tempo.

Basicamente, o princípio da localidade tem duas componentes (GENUTIS *et al.*, 2001) definidas como localidade espacial e temporal. A localidade espacial leva em consideração que são altas as chances de ocorrer um novo acesso à memória, numa região próxima a um endereço que foi acessado anteriormente. Isto ocorre principalmente devido ao uso de arranjos, matrizes e estruturas dentro dos programas, o que faz com que os dados de cada um deles fiquem próximos na memória. Com isso, trazer estes blocos da memória principal para a *cache* pode auxiliar na diminuição de falhas na busca de dados.

Já na localidade temporal, considera-se que, se um dado ou instrução foi acessado recentemente, é provável que venha a ser acessado novamente num futuro próximo. Este fato fica evidente quando se atenta ao que ocorre na memória, quando se faz uso de laços e os acessos as variáveis dentro dos programas. Com isso, de acordo com este princípio, o ideal é manter o máximo possível este dado ou instrução dentro da memória *cache* e remover aqueles que já não são acessados há algum tempo. Assim, a inserção de um nível de memória entre a memória RAM dinâmica de acesso lento e o *cache* do processador contribui para a diminuição da taxa de perda de ciclos do processador e para solução do *gap* existente entre a velocidade da memória e a capacidade de processar dados dos processadores (SCHALLER, 1997).

O *pipeline* é uma técnica de hardware que permite à CPU realizar a busca de uma ou

mais instruções, além da próxima a ser executada. Estas instruções são colocadas em uma fila de memória dentro da CPU onde aguardam o momento de serem executadas. Com isso, o *pipeline* de instruções explora cada uma das fases da execução das instruções, alocando cada subcomponente em uma unidade. A idéia é construir o hardware a partir de unidades funcionais, onde cada unidade é responsável por uma fase do ciclo de execução da instrução, e colocá-las como numa linha de montagem, (TANENBAUM, 1992) isto é, o resultado de cada unidade é passado para a outra até que o processo de execução da instrução seja finalizado. Assim, por exemplo num *pipeline* com quatro fases, composta de: busca, decodificação, execução, atualização da memória (HWANG, 1993), pode-se ter o *pipeline* totalmente utilizado, conforme ilustra a Figura 4.1, se cada fase do processo utilizar apenas um único ciclo do processador para completar sua função, fazendo com que cada instrução, após sucessivas instruções, dure um único ciclo para ser executada. Esse tipo de estrutura de hardware implementa um paralelismo funcional nos sistemas monoprocessados e conseqüentemente aumenta a velocidade de processamentos destes sistemas.

Figura 4.1: Execução em *pipeline* com uso de instruções sucessivas, onde o sistema passa a executar, após alguns ciclos, uma instrução por ciclo (HWANG, 1993)

Como se observa no gráfico apresentado na Figura 4.1, após o quarto ciclo de *clock* com instruções que tenham a etapa de execução durando apenas 1 período de *clock*, tem-se o *pipeline* totalmente preenchido e com isso as instruções que demandavam 4 ciclos passam a ter seus resultados a cada novo ciclo, reduzindo conseqüentemente o tempo demandado na resolução do problema.

Mais recentemente, no desenvolvimento de processadores, se destaca o surgimento de *chips* com múltiplos núcleos que buscam atender às demandas atuais dos usuários. Atualmente, o mercado convencional apresenta uma forte demanda pelo uso de aplicativos que executam concorrentemente, tais como processadores de texto, tocadores de MP3, aplicativos antivírus, sistemas que detectam intrusos, dentre outros, o que leva estes softwares a permanecer em grande parte do tempo, residentes na memória, fazendo uso da CPU (KNIGHT, 2005). Estas demandas têm requerido soluções dos fabricantes. Para isso, grande parte das empresas, tais como a Intel, AMD e IBM, desenvolvem processadores apostando no aumento do desempenho das arquiteturas de processadores baseando-se em pastilhas que contenham múltiplos *cores*, que distribuem o processamento entre os núcleos. Com isso, eles visam não só uma solução imediata para a demanda atual, mas um modelo de *hardware* que forneça escalabilidade para as exigências futuras do mercado. Paralelamente, existem frentes destas empresas que trabalham no desenvolvimento de ferramentas que auxiliem no desenvolvimento de software *multithread*, para melhor exploração do *hardware* (GEER, 2005).

Após esta análise do panorama, percebe-se que mesmo com esse aumento da velocidade de processamento nas máquinas sequenciais, ela ainda é limitada pela tecnologia, devido à velocidade dos circuitos que não pode crescer indefinidamente e também pela existência de um único elemento de processamento. Um fato que reforça esta afirmação é que, segundo Geer (2005), nos anos 1990, o crescimento da performance nos chips era em torno de 60% ao ano. Porém, no início da década seguinte, este fato começou a mudar. No período de 2000 a 2004, caiu para 40% ao ano, chegando aos 20% no fim deste período. Isto aconteceu principalmente devido ao crescimento do custo para aumento da performance dos processadores. Uma boa solução que vem sendo empregada é o incremento do poder de processamento com a utilização de processamento paralelo, fugindo, dessa maneira, das arquiteturas sequenciais. Entretanto, essa solução não é tão simples e diversas propostas vêm sendo apresentadas. Elas, em síntese, implicam em mudanças na construção do *hardware*, no gerenciamento e na coerência da memória e nos sistemas operacionais para controlar esse paralelismo e na construção de softwares que explorem esse tipo de estrutura paralela.

4.2 Projeto de sistemas paralelos

Apesar da aparente impressão de ser desenvolvido baseando-se apenas em criatividade e intuição, para o desenvolvimento de algoritmos paralelos existem paradigmas que permitem estruturá-los de forma organizada e com metodologia, levando em conta características do problema que se quer resolver. Utilizar uma metodologia de desenvolvimento de sistema, seja para um sistema sequencial ou paralelo, diminui as chances de haver necessidades de retrabalho e que este seja oneroso em termos de tempo e custo. Além disso, permite visualizar problemas que talvez só seriam descobertos com o sistema já implementado.

A maioria dos problemas computacionais possui um conjunto variado de soluções que podem, no caso do desenvolvimento de um algoritmo paralelo, ser bastante diferentes das soluções sequenciais. Com isso, é importante levar em conta que paradigmas para sistema paralelos auxiliam na escolha de uma melhor forma de modelagem de problemas do mundo real em um conjunto de pequenos pedaços que, reunidos, atendem às necessidades impostas pelo problema. A metodologia PCAM (FOSTER, 2005), mostra um paradigma de desenvolvimento de algoritmo paralelo baseado em quatro etapas:

1. Particionamento;
2. Comunicação;
3. Aglomeração;
4. Mapeamento

A Figura 4.2 ilustra os conceitos contidos em cada etapa. Basicamente, compreende em dividir o problema em pequenas tarefas, organizar a forma como estas tarefas irão se comunicar, aglomerar as tarefas que tenham intensa comunicação em tarefas maiores e por fim distribuir as tarefas entre os processadores disponíveis. Cada etapa será explicada com maiores detalhes nas seções seguintes.

Figura 4.2: Metodologia para desenvolvimento de sistemas paralelos PCAM

4.2.1 Particionamento

O objetivo de particionar um problema em partes menores é buscar oportunidades de paralelismo. Isso é realizado, para que se obtenha pequenas tarefas, que dão ao algoritmo e ao sistema final grande flexibilidade no que tange ao potencial de paralelização. Serão os

estágios seguintes que determinarão qual a organização entre as tarefas que melhor dará resultado; contudo, nessa fase inicial, é importante não se considerar este fato para não se perder tais oportunidades.

Uma boa divisão do problema deve separá-lo em pequenos pedaços de computação e de dados que serão utilizados no processamento. Para isto, existem duas formas pelas quais o projetista pode focar o problema e criar esta boa divisão. A primeira é baseada nos dados e denominada Decomposição de Domínio, em que inicialmente se propõe decompor os dados associados com o problema a ser resolvido, dividindo, se possível, em partes pequenas de tamanho próximo. Com isso, o particionamento produzirá tarefas e cada uma executará suas operações sobre uma porção dos dados que são utilizados no problema. Ao particionar utilizando este paradigma, Foster (2005) ressalta que, inicialmente, deve-se focar na criação de estruturas de dados grandes ou estruturas de dados que serão frequentemente utilizadas. Na Figura 4.3 são ilustradas três formas de particionar-se um problema que envolve um conjunto de dados tridimensional, através da técnica de Decomposição de Domínio

Algumas informações complementares e exemplos de uso deste paradigma podem ser encontradas em (FOSTER, 2005)

Figura 4.3: Ilustração da decomposição de um problema com dados em três dimensões

Outra forma de se particionar um problema está relacionada ao foco nas funcionalidades necessárias para resolver o problema, também conhecida como Decomposição Funcional. A idéia é separar o problema em tarefas desconectadas que necessitem de pouca comunicação ou replicação de dados. O objetivo ao final é que, desta forma, as tarefas sejam alocadas aos

processadores à medida que estes fiquem disponíveis, o que facilitará o mapeamento das tarefas e dará aos processadores mais rápidos uma maior carga de trabalho.

4.2.2 Comunicação

Finalizada a fase de particionamento, será necessário pensar na forma como se dará a troca de informações entre as tarefas que foram anteriormente definidas. Para realizar uma ação ou cálculo, qualquer tarefa sempre necessitará de dados que virão, na maioria das vezes, de outras tarefas. Assim, nesta etapa, denominada Comunicação, se define o fluxo das informações do algoritmo. Conceitualmente, define-se essa comunicação como sendo um canal capaz de conectar uma tarefa que deseja enviar mensagem, chamado de produtor, para outro que deseja receber, definido como consumidor. Com isso em mente, divide-se a comunicação em duas partes. Na primeira parte, as ligações entre as tarefas produtoras e consumidoras são definidas, sendo que estas ligações podem se dar diretamente ou indiretamente. A segunda parte especificará os tipos de mensagens que serão trocadas entre as tarefas durante a execução do algoritmo. Nesta parte, é possível pensar a respeito dos canais e tarefas que serão realmente necessários, com o intuito de proporcionar maior localidade e menor custo de comunicação.

Segundo Foster (2005), a decomposição de domínio pode acarretar em dificuldades para estabelecimento dos requisitos de comunicação, isto porque, apesar da simplicidade existente na separação dos dados e nas operações efetuadas sobre eles, ainda permanece a necessidade de troca de dados entre algumas tarefas, que recebem informação de diversas outras tarefas para execução dos passos seguintes. Estabelecer e organizar um algoritmo eficiente para contemplar essa necessidade pode se tornar um desafio ao projetista. Por outro lado, os requisitos de decomposição funcional são geralmente simples.

Os padrões utilizados para definição das tarefas podem ser divididos em quatro categorias:

- *Local ou Global*: A local ocorre quando uma tarefa estabelece comunicação com um pequeno conjunto de outras tarefas. Já a comunicação global é aquela em que um grande

número de tarefas participa;

- *Estruturada ou Desestruturada*: Na estruturada, a tarefa e suas vizinhas formam uma estrutura regular, como árvore ou grade. Já a comunicação desestruturada é formada por grafos arbitrários;
- *Síncrona ou assíncrona*: Na síncrona, as tarefas produtoras e consumidoras trocam informações de forma coordenada. Em contraste com essa forma, a assíncrona permite que um consumidor obtenha dados do produtor sem que haja uma cooperação explícita por parte dele.
- *Estática ou dinâmica*: Na estática, as identidades das tarefas parceiras não se alteram com o decorrer da execução. Já na dinâmica, as estruturas de comunicação alteram-se e são determinadas em tempo de execução.

4.2.3 Aglomeração

A fase de aglomeração tem por objetivo transportar as abstrações definidas nas fases de particionamento e comunicação para o mundo concreto. Nesta etapa do desenvolvimento de um sistema paralelo, o projetista começa a preocupar-se com o tamanho das tarefas, com os custos de comunicação, enfim, com a criação de um algoritmo eficiente para a execução nos diferentes tipos de computadores paralelos que poderão ser utilizados.

Considera-se, nesta fase, a aglomeração de tarefas pequenas definidas no particionamento em um pequeno número de tarefas de maior tamanho. Também é importante pensar se a replicação de dados e/ou computação nas tarefas não produzirá bons resultados. É possível que o número de tarefas aglomeradas seja maior que o número de processadores, inclusive isto pode ser até desejável, dependendo das características do algoritmo e do problema que se está resolvendo, porém, a forma de distribuição destas tarefas será determinada na fase seguinte.

Na busca por boa relação de aglomeração e de replicação tanto de dados quanto de computação, dois pontos principais devem ser seguidos pelo projetista do sistema.

4.2.3.1 Redução dos custos de comunicação através do aumento da granularidade de tarefas e comunicação

Os altos custos de comunicação são muito influentes na performance de sistemas paralelos. Assim, com o intuito de diminuir o impacto da comunicação, a aglomeração busca diminuir a quantidade de mensagens trocadas entre as tarefas, mesmo que a quantidade de dados enviados continue a mesma. De fato, o custo para se iniciar uma nova comunicação é fixo e percentualmente será alto para uma mensagem curta. Por outro lado, será menos oneroso enviar menos mensagens com mais *bytes* ou mais informação. Um exemplo desta técnica é ilustrado na Figura 4.4, onde se tem a aglomeração das tarefas pequenas e de seus dados representada pelos quadrados dentro dos círculos. Quando se utilizam tarefas pequenas com poucos dados, as quais estão representadas por um único quadrado dentro de um círculo, a computação será diminuta e com pequena quantidade de dados para troca com as demais tarefas. Isto elevará o custo de comunicação, principalmente devido ao custo inicial, que será alto em relação à quantidade de dados trocados entre as tarefas. Para se sanar esta situação, aglomeram-se as tarefas pequenas em maiores, onde se tem aumento na granularidade da comunicação e da computação realizada. Isto provocará uma redução do número de comunicações necessárias e também uma melhora nos custos de cada comunicação realizada.

Replicar a computação também pode ser uma forma interessante de reduzir a computação e/ou o tempo de execução de um algoritmo, principalmente em situações onde o processador fica parado aguardando o resultado de uma operação, por exemplo, no caso de um envio de resultados em N processadores para realização de um somatório e depois a redistribuição dos resultados. Na forma convencional, como ilustra a Figura 4.5(a), cada processo enviaria seus dados para um processo centralizador, que receberia os dados, faria a soma e enviaria o resultado para todos os processos. Numa forma alternativa de realizar essa mesma operação, os processos poderiam, aos pares, realizar as somas, modificando seus parceiros a cada nova etapa, tal qual ilustra a figura 4.5(b), de forma que após $\log_2 N$ etapas, para um valor de N processos que seja potência de 2, a soma estaria disponível em todas as tarefas. Assim, ocorre um uso mais racional da rede sem sobrecarga de um processo e evita-se que os outros processos fiquem

Figura 4.4: Ilustração da aglomeração de tarefas para aumentar a granularidade da comunicação e da computação

Figura 4.5: Ilustração de 8 processos trocando dados e realizando o somatório de valores, de forma que, ao final de 3 etapas, todos os processos possuem o valor do somatório

parados aguardando o resultado final.

4.2.3.2 Preservar Flexibilidade

Utilizar técnicas que permitam manter um variado número de tarefas, dando flexibilidade, e garantam a escalabilidade e a portabilidade do algoritmo desenvolvido são fundamentais. Nem sempre a aglomeração das tarefas de forma que se tenha N tarefas para N processadores pode ser um bom procedimento. Problemas que exigem um constante bloqueio das tarefas aguardando dados que estão em outros processadores podem tornar interessante o mapeamento de outras tarefas, que assumem o controle do processador no instante do bloqueio, aumentando a eficiência do algoritmo. É interessante observar que alguns destes eventos só serão detectados com estudo do modelo analítico e o estudo empírico, porém, ter formas flexíveis de mapeamento auxilia na sintonia do algoritmo para a máquina com que se está trabalhando e permite o equilíbrio da carga de trabalho nos processadores disponíveis no sistema.

4.2.4 Mapeamento

O mapeamento é a fase de definição do recurso computacional que cada tarefa utiliza. De forma geral, o mapeamento ainda não possui uma regra automática para a distribuição das tarefas em computadores paralelos, levando os projetistas a fazerem considerações sobre o endereçamento das tarefas a cada novo problema a ser paralelizado, além de pensar em formas de equilibrar a carga de trabalho dentre os processadores disponíveis.

O objetivo principal ao se mapear as tarefas é a busca da minimização do tempo total de execução e para isso duas estratégias são priorizadas. Uma delas busca colocar as tarefas que executam concorrentemente em processadores diferentes. Outra estratégia aloca tarefas que têm comunicação mais freqüente no mesmo processador com o intuito de reduzir os custos de comunicação e aumentar a localidade.

Na maior parte dos problemas, essas duas estratégias sempre apresentarão um conflito, o que exigirá do projetista, um equilíbrio entre os interesses de cada uma delas.

Nos algoritmos paralelos que são desenvolvidos utilizando decomposição de domínio complexa, com cargas de trabalho variáveis ou padrões de comunicação desestruturados, a realização de forma eficiente da fase aglomeração e o mapeamento equilibrado das tarefas entre os processadores utilizando as duas estratégias pode tornar-se algo desafiador. Assim, um boa prática a ser seguida pelo desenvolvedor é a utilização de algoritmos de balanceamento de cargas que buscam, em síntese, dar a eficiência e o equilíbrio desejado. Estes algoritmos baseiam-se em heurísticas, que determinam como redistribuir entre os processadores as tarefas ou processos. O uso deles traz um custo de tempo adicional à execução do software, o que deve ser ponderado. Levando em consideração estes custos adicionais, o balanceamento probabilístico de carga, que distribui as tarefas de forma aleatória entre os processadores e tende a equilibrar a carga de trabalho do sistema é então considerado um dos métodos que tendem a trazer o menor custo, devido principalmente à sua simplicidade e escalabilidade.

Algoritmos construídos com base na decomposição funcional produzem tarefas de

Figura 4.6: Ilustração da comunicação e envio de cargas de trabalho de gerente para trabalhadores

curto tempo de existência que são coordenadas por outras tarefas presentes durante toda a execução. Nestes casos, o uso de algoritmos de agendamento de tarefas tende a funcionar bem. Eles baseiam-se na alocação de tarefas para os processadores que estão disponíveis no sistema. Um exemplo de algoritmo de agendamento de tarefas é o Gerente/Trabalhador, ilustrado na Figura 4.6, que é eficiente para sistemas que possuem um número pequeno de processadores. A estratégia do algoritmo é de um processo central, denominado gerente, que é responsável por dividir o trabalho a ser executado com processos que requerem trabalho, denominados trabalhadores. O gerente é o único processo com acesso aos recursos externos, com isso todo contato com o mundo externo se dá através dele. Cada trabalhador fica repetidamente requisitando trabalho ao gerente e, ao receber, parte para a execução. Ocasionalmente, pode ocorrer de uma tarefa recebida pelo trabalhador gerar a criação de novas tarefas que são encaminhadas ao gerente para alocação em outro processo trabalhador. A eficiência desta estratégia depende da quantidade de trabalhadores e dos custos relacionados para a obtenção e execução dos problemas.

4.3 Medidas de desempenho

Neste trabalho são utilizados, como medidas de desempenho, o Ganho e a Eficiência. Estas medidas servem para mensurar a qualidade do paralelismo que está sendo implementado.

O Ganho é uma medida que relaciona o desempenho de um sistema baseado em n processadores em relação a um de único processador. O ganho é definido como:

$$S(n) = \frac{T_1}{T_n} \quad (4.1)$$

onde T_n corresponde ao tempo de execução do sistema com n processadores e T_1 , o tempo de execução num sistema com um único processador.

A eficiência é uma medida que fornece a fração de tempo em que os processadores estão sendo utilizados. A eficiência para um sistema com n processadores é definida por (HWANG, 1993):

$$E(n) = \frac{T_1}{nT_n} \quad (4.2)$$

ou

$$E(n) = \frac{S(n)}{n} \times 100 \quad (4.3)$$

quando a eficiência é fornecida em porcentagem. A eficiência indica o atual grau de ganho de desempenho obtido quando comparado com o valor máximo.

Existem alguns fatores que aparecem como sobrecargas de trabalho em programas paralelos e que limitam o ganho e a eficiência a valores menores que estes limites máximos, dentre os quais destacam-se: (STAROBA, 2004)

1. Períodos em que alguns dos processadores não estão trabalhando, permanecendo em estado de aguarde, o que ocorre geralmente em trechos meramente sequenciais;

2. Computação extra que aparece na versão paralela e que não existia na versão seqüencial;
3. Tempo de comunicação para envio de mensagens em programas que utilizam este paradigma;
4. Sincronização entre os processos.

4.4 Plataforma DSP

A linguagem C Paralela da 3L é uma linguagem que permite a construção de programas paralelos para plataforma DSP com um ou mais processadores. Segue o padrão ANSI C com a adição de algumas bibliotecas para implementação de códigos paralelos, gerenciamento de processos e de regiões críticas de memória.

A linguagem possui recursos para implementação de estruturas de sistemas paralelos e distribuídos, tais como múltiplas tarefas, comunicação entre processos, utilização dos dois barramentos e separação dos dados nos bancos de memórias, além da possibilidade de se carregar, no *cache*, as variáveis ou matrizes que são largamente utilizadas nos algoritmos. Também oferece suporte para implementação, gerenciamento e *threads*. A linguagem C Paralela possibilita o acesso aos recursos do sistema do *host* de forma transparente (3L, 1995).

4.4.1 Configuração de tarefas em uma aplicação

Uma aplicação em C Paralela compreende uma ou várias tarefas sendo executadas concorrentemente em uma rede de processadores. Em linguagem C Paralela, cada arquivo fonte que contenha uma função `main( )` representa uma tarefa que contém sua própria região de dados, com código na memória. Cada tarefa possui também vetores com canais de comunicação para troca de dados com outros processos que estejam sendo executados concorrentemente.

Existe uma ferramenta de configuração de aplicação que determina em qual processador da rede ficará alocada cada tarefa e qual será sua área de memória para dados e instruções.

Dentro de cada tarefa, podem ser criadas diversas *threads*, que concorrem pelo

acesso ao processador e aos recursos da tarefa. A utilização de *threads* aproveita de forma mais eficiente o processador, contudo aumenta também a complexidade na gerência do acesso à memória e aos recursos do sistema. Um dos problemas da linguagem C Paralela no tratamento de *threads* é a impossibilidade de distribuí-las entre todos os processadores da rede, acarretando uma disputa pelo acesso apenas do processador em que a tarefa está sendo executada e não aproveitando todo o poder de processamento da arquitetura.

Após a criação de todas as tarefas de uma aplicação, utiliza-se o aplicativo de configuração da 3L para determinar quais recursos do sistema serão alocados para cada uma das tarefas. Essa configuração é feita através de um arquivo texto que descreve, utilizando uma sintaxe própria, a organização das tarefas na rede de processadores. Por meio desses arquivos é possível configurar:

- Número de processadores presentes;
- A alocação de processador para as tarefas;
- Quantidades de memória local e global de cada processo;
- Canais físicos e virtuais de comunicação entre as tarefas;
- Número de entradas e saídas de cada processo;
- Utilização de blocos da memória cache.

Para configurar a aplicação X de forma a distribuir suas tarefas tal qual foi ilustrado na Figura 4.7, utiliza-se o trecho de código a seguir:

```
processor root
processor p1
processor p2
task A ins=1 outs=1
task B ins=2 outs=1
task C ins=2 outs=2
task D ins=1 outs=2
place A root
place B p1
place C p2
place D p2
connect? A[0] B[0]
connect? B[0] C[0]
connect? C[0] B[1]
connect? C[1] D[0]
connect? D[0] A[0]
connect? D[1] C[1]
```

Neste trecho de código do arquivo de configuração, destacam-se em negrito as palavras do software de configuração para determinação de processadores (*processor*) presentes no sistema, canais de conexão física entre os processadores (*wire*), a conexão virtual entre processos (*connect*), declaração de tarefas (*task*) da aplicação. À frente de cada tarefa da aplicação encontra-se o número de portas de entrada (*ins*) e saídas(*outs*).

Esse paradigma de configuração da aplicação utilizado pela 3L permite que a abstração do algoritmo realizada durante a fase de modelagem seja facilmente mapeada na arquitetura paralela disponível, além de auxiliar na sintonia do algoritmo na arquitetura e no balanceamento das cargas de trabalho entre os processadores. A Figura 4.7 ilustra a forma como a ferramenta 3L, utilizando o trecho de código de configuração apresentado acima, distribuirá as tarefas A, B, C e D que foram modeladas anteriormente .

Figura 4.7: Modelo de aplicação configurada numa rede de processadores DSP com arquivo de configuração - (a) Modelagem da aplicação X, com 4 tarefas que se comunicam; (b) A aplicação X com suas tarefas mapeadas entre os processadores *Root*, *P1* e *P2*

A avaliação dos algoritmos paralelos de reconstrução bidimensional foi realizada numa plataforma paralela de processadores DSP, a qual está acoplada a um PC. A plataforma é composta de uma placa modelo HEPC2E (HOLLIS, 1999), mostrada na Figura 4.8, onde encontram-se conectados quatro módulos TIM, com um processador TMS320C40 em cada módulo.

Figura 4.8: Placa HEPC2E, utilizada como plataforma paralela para avaliação do desempenho do método paralelo

O módulo mais à direita na Figura 4.8 é identificado como o módulo *Root*. É através dele que a rede de processadores DSP tem acesso a dados externos ao módulo. Assim, toda a informação de leitura e escrita em disco, exibição de informações no monitor e leitura de teclado, passa obrigatoriamente por ele.

5 Modelo de Reconstrução 3D e Aplicação da Filtragem de Wiener Dedicada às Ciências do Solo em processamento paralelo

5.1 Introdução

Neste capítulo realiza-se a apresentação do desenvolvimento do trabalho com ênfase na modelagem para reconstrução tridimensional, descrevendo a forma de particionamento das tarefas de cada reconstrução, a organização da comunicação entre as tarefas geradas, seu agrupamento e mapeamento na arquitetura DSP para o processamento paralelo.

De forma esquemática, a organização utilizada para o modelo de reconstrução tridimensional desenvolvido é mostrada na Figura 5.1. É interessante observar que esta organização possibilita um futuro acoplamento de novos módulos para aplicação de técnicas de reconhecimento de padrões, de visão computacional, bem como a aplicação de processamento paralelo juntamente com estas técnicas.

Um padrão para uma base de dados de cortes e de objetos reconstruídos foi organizado, em que as projeções, imagens e objetos ficam disponíveis. Para utilizá-los, os módulos fazem uso de um grupo de rotinas de leitura que entregam um conjunto de informações sobre a amostra que será utilizada. Como exemplo dessas informações, é possível obterem-se dados a respeito do processo de aquisição tomográfica, tais como passo angular, passo linear, translação linear e angular, energia, informações sobre o tipo de amostra, além, das informações das pro-

jeções. Na base de cortes, as rotinas de leitura retornam à matriz de coeficientes de atenuação do plano bem como às dimensões desta matriz. A partir destes dados, pode-se gerar objetos tridimensionais. Através de ferramentas de visualização bidimensional, é possível passar estes dados de atenuação para o formato *bitmap*. Neste trabalho, como será aprofundado nas próximas seções, utilizou-se interpolação entre planos tomográficos com *B-Wavelet*, realizando-se a sua paralelização e posterior avaliação do algoritmo em uma arquitetura paralela. O módulo de filtragem *a priori* nas projeções auxilia na redução do nível de ruídos inerentes ao processo de aquisição das projeções tomográficas, como por exemplo, o ruído Poisson.

Figura 5.1: Diagrama com o modelo de alto nível dos processos do trabalho

Os objetos gerados na interpolação são depositados na base de objetos reconstruídos. Para visualização e análise destes objetos desenvolveu-se uma interface gráfica utilizando-se o compilador *Borland Builder C++* versão 6 combinado com a biblioteca gráfica *Visualization ToolKit* (VTK, 2006).

5.2 Algoritmo paralelo de reconstrução bidimensional

5.2.1 Modelagem do algoritmo paralelo de reconstrução bidimensional

A reconstrução tomográfica utilizada neste trabalho baseia-se no algoritmo de retroprojeção filtrada, o qual foi revisado ao longo do Capítulo 2. Matematicamente, o cerne do algoritmo está na aplicação da Equação (2.17) para realização da filtragem das projeções e da Equação (2.18) que é utilizada para retroprojetar os pontos das projeções na imagem $g(x,y)$, obtendo-se, desta forma, a imagem reconstruída. Baseando-se nestas equações e em funcionalidades que devem ser fornecidas pelo algoritmo, realizou-se o particionamento do problema, obtendo-se um grupo de tarefas consideradas necessárias para implementação da reconstrução, as quais são enumeradas e apresentadas como sendo:

1. Tarefa de leitura dos dados da base de projeções;
2. Tarefa de filtragem de cada projeção obtida no ângulo θ , utilizando a Equação (2.17);
3. Tarefa de agrupamento dos resultados filtragem das projeções;
4. Tarefa de retroprojeção do *pixel* de coordenada (x,y) , utilizando a Equação (2.18);
5. Tarefa de agrupamento dos *pixels* reconstruídos;
6. Tarefa de gravação do corte reconstruído na base de dados.

A partir destas tarefas, seguindo o modelo PCAM, organizou-se a forma pela qual se deve estabelecer a comunicação entre elas. De forma ilustrada, esta comunicação em alto nível está organizada tal qual é apresentado na Figura 5.2, onde os círculos representam as tarefas obtidas durante o particionamento e as setas mostram o sentido da comunicação e, sobre elas, os dados que são trocados entre as tarefas. A tarefa 1 é responsável pela leitura dos dados de projeções na base e por enviá-los para a tarefa 2, onde cada projeção é filtrada. Os dados filtrados na tarefa 2 são enviados para tarefa 3, de aglomeração das projeções filtradas, que recebe e as reúne até que toda o procedimento de filtragem esteja terminado, no processo 2.

Quando isto acontece, o processo 3 distribui à tarefa de retroprojeção os dados necessários para reconstruir a imagem tomográfica, bem como uma carga de trabalho para reconstrução. A tarefa 5 recebe e agrupa o resultado de cada pixel retroprojetado pelo processo 4 e, ao final, encaminha estes *pixels* retroprojetados para a tarefa 6 realizar a gravação na base de cortes reconstruídos.

Figura 5.2: Ilustração das tarefas resultantes do particionamento do algoritmo de reconstrução bidimensional e do estabelecimento da comunicação entre as elas

A principal vantagem obtida nesta forma de divisão do problema é a possibilidade de se obter um alto grau de paralelismo entre as tarefas de filtragem e reconstrução. Na etapa de filtragem das projeções, pode-se criar um conjunto de tarefas de filtragem, com cada tarefa trabalhando em uma determinada projeção P_{θ_i} ou um subconjunto do total de projeções. Na retroprojeção dos *pixels*, uma matriz de processos de reconstrução poderia ser organizada para retroprojetar cada *pixel* $p(x,y)$ ou um grupo deles, em processos distintos.

A fase de aglomeração requer do analista a visão para organizar as novas tarefas já tendo em mente a arquitetura da máquina paralela onde o algoritmo será implementado. Além do mais, ele deve se ater ao fato de que sua modelagem irá equilibrar a demanda por comunicação entre as tarefas e o paralelismo entre elas. Fundamentado-se nestes princípios, na fase de aglomeração, aglutinaram-se as tarefas de modo a produzir um algoritmo que seguisse o padrão gerente trabalhador. As tarefas "Leituras de dados de projeções", "Agrupamento de projeções", "Agrupamento de pixels" e "Gravação do corte na base" foram aglutinadas num processo gerente. Esta forma de organização concentra no gerente todo o processo de entrada e saída de dados do algoritmo paralelo para as outras bases de dados tomográficos envolvidas no processo de reconstrução bidimensional.

As tarefas de filtragem e de retroprojeção foram transformadas em processos trabalhadores, possibilitando que se adicionem réplicas destes à medida que existam processadores disponíveis. A Figura 5.3 ilustra a aglomeração das tarefas, bem como o aumento da granularidade da comunicação entre os processos que foram aglutinados na tarefa gerente e os trabalhadores.

Com o processo de filtragem sendo um processo trabalhador, realizou-se um aumento na granularidade da comunicação para minimizar os seus custos. Cada processo trabalhador passou a receber um conjunto de projeções para filtrar. Este conjunto é estipulado pelo algoritmo de comunicação que foi projetado na fase aglomeração que estipula cargas de trabalho semelhantes para cada processo de filtragem. O algoritmo de comunicação está organizado em duas fases. Na fase 1, a de filtragem, a matriz de projeções, que possui K linhas e M colunas, é dividida levando-se em conta a quantidade de processos que estão realizando a filtragem, aqui representada por $T_{filtragem}$. A dimensão K da matriz representa a quantidade de passos angulares que foram varridos pelo conjunto fonte-detector e a dimensão M , a quantidade de passos lineares percorridos pelo conjunto. A partir destes dados, a carga de trabalho dos processos de filtragem, $Carga_{filtragem}$, será determinada pelo resultado inteiro da razão:

Figura 5.3: Aglomeração de tarefas do algoritmo paralelo de reconstrução 2d

$$Carga_{filtragem} = \frac{K}{T_{filtragem}} \quad (5.1)$$

O resto desta divisão de trabalho serão projeções adicionais que serão distribuídas entre alguns processos. Com isso, a tarefa gerente organiza uma tabela, apresentada na Tabela 5.1 que define os limites e a quantidade de carga de trabalho de cada processo trabalhador.

Identificação do Processo	Início	Final
0	1	$(K/T_{filtragem})$
1	$(K/T_{filtragem}) + 1$	$2 * (K/T_{filtragem})$
...	...	...

Tabela 5.1: Tabela de carga de trabalho, utilizada no algoritmo de comunicação.

Após realizar sua filtragem, cada trabalhador retorna o resultado para o processo gerente, que irá reunir as projeções filtradas deste e dos demais trabalhadores para a fase seguinte

task	master	file=Rec2dGerente	stack=150k heap=600k
task	worker	file=Rec2dTrab	data=700k
task	rworker	file=Rec2dTrab	data=700k

Tabela 5.2: Arquivo de mapeamento das tarefas na criação da aplicação paralela de reconstrução 2D na rede de processadores DSP.

de retroprojeção. A partir deste momento, tem início a fase 2 do algoritmo de comunicação que irá determinar a carga de trabalho dos processos de retroprojeção, os quais recebem as projeções filtradas e ficam encarregados de retroprojetar um conjunto de *pixels* da imagem reconstruída. Para simplificar, este conjunto de *pixels* é determinado utilizando-se a tabela de divisão da carga de trabalho da filtragem. Com isso, cada trabalhador recebe uma carga igual de trabalho, representada em linhas a reconstruir, e as linhas de sua responsabilidade são determinadas pela sua identificação e os valores presentes na tabela de filtragem. Uma vez finalizada a retroprojeção, seu resultado final é enviado para o processo gerente, que irá se encarregar de agrupar os *pixels* reconstruídos em cada processo trabalhador, organizar o resultado e gravar a matriz retroprojetada na base de cortes reconstruídos.

5.2.2 Implementação na plataforma DSP

Para implementação do algoritmo paralelo de reconstrução bidimensional, a tarefa gerente e os trabalhadores foram implementados utilizando o compilador C Paralelo da 3L. A linguagem fornece a possibilidade da criação de aplicações que tenham este paradigma, distribuindo automaticamente entre os processadores disponíveis na rede as tarefas trabalhadoras e alocando o processador *Root* para o processo gerente. No *Root* também é possível colocar um processo trabalhador executando concorrentemente com o processo gerente. Essas configurações são realizadas através do arquivo de configuração, onde determina-se o mapeamento das tarefas na rede de processadores DSP. Na Tabela 5.2 apresenta-se o trecho do arquivo de configuração que possibilita o ajuste da aplicação paralela na plataforma DSP.

Na configuração mostrada na Tabela 5.2, a tarefa definida como *task master* é mapeada no processador *Root* e será responsável por distribuir trabalho para os processos trabalhadores. Este modelo de tarefa é definido dentro do ambiente DSP como sendo uma tarefa *task worker*. A tarefa ajustada como *task rworker* é uma tarefa trabalhadora que é mapeada no processador *Root*, executando concorrentemente como o processo gerente. Os parâmetros *stack*, *heap* e *data* definem o espaço em memória que podem ser utilizados pelos processos.

Seguindo o que foi idealizado na fase de aglomeração, o processo gerente ficou encarregado da leitura/gravação e aglomerações dos dados produzidos em cada fase do algoritmo de retroprojeção. Para isso, dentro do processo gerente foram criadas funções para cada tarefa aglomerada, com algumas delas colocadas como *threads* dentro da tarefa gerente, o que possibilita a concorrência entre a execução destas funções e a recepção de dados enviados pelos trabalhadores. A partir destas definições, apresenta-se na Figura 5.4 o algoritmo em alto nível do processo gerente da reconstrução bidimensional. As funções que aparecem dentro das caixas de linhas tracejadas representam *threads* que, dentro do processo gerente são executadas concorrentemente.

O processo de identificação dos trabalhadores realizado pelo gerente implementa, no ambiente paralelo DSP, algo parecido com o que ocorre na biblioteca *Message Passing Interface* (MPI), na qual os processos são identificados, e esta identificação auxilia os processos trabalhadores, através de uma identificação única no sistema, na determinação das regiões de dados que estão sob sua responsabilidade (MPI, 2004).

Os processos trabalhadores de filtragem e retroprojeção constituem dois processos distintos, mas, na implementação, foram colocados em um único processo trabalhador para facilitar a troca de dados e a implementação no ambiente DSP. A Figura 5.5 ilustra a execução em paralelo dos processos gerente e trabalhador que reúnem, a filtragem e a retroprojeção. No início do algoritmo, cada processo trabalhador recebe uma identificação dentro da rede de pro-

Figura 5.4: Funções que compõe a tarefa gerente na reconstrução bidimensional

cessadores. Numa segunda etapa, o trabalhador recebe os parâmetros de filtragem e um bloco de projeções para que ele inicie o processo de filtragem. Em seguida, ele filtra as projeções, seguindo o algoritmo da Equação (2.17) e, finalizando esta etapa, envia os dados para o gerente. Após o encaminhamento do conjunto de todas as projeções filtradas aos trabalhadores, eles utilizam a Tabela 5.2 de comunicação para, com base na sua identificação, retroprojetar os pontos que estão sobre sua responsabilidade. Uma vez retroprojetados, os pontos são encaminhados ao gerente.

5.3 Modelagem do algoritmo de reconstrução tridimensional

Conforme foi revisado ao longo da seção 2.5, a reconstrução de objetos é realizada através da interpolação de planos virtuais entre pares de planos reais. Para isso, utiliza-se a interpolação *B-Wavelet*, aplicada em cada *pixel* de coordenada (x,y) dos planos reais adquiridos.

Com base no número real de fatias (NR) e o no número de fatias interpoladas (NI), pode-se obter o número total de fatias (NTF) do objeto interpolado pela Equação 5.2.

$$NTF = (NR - 1)NI + NR \quad (5.2)$$

Matematicamente, a interpolação de pontos ao longo do eixo Z, nas coordenadas (x,y) , em cada plano, é realizada utilizando-se a Equação (2.24), com cada ponto dos planos nesta coordenada preenchendo o vetor a_i . Com base na Equação (2.24) e nos procedimentos necessários para reconstrução tridimensional obteve-se o particionamento do algoritmo de reconstrução 3D. A partir do algoritmo sequencial, gerou-se o seguinte conjunto de tarefas:

1. Tarefa de Leitura dos dados de reconstrução tridimensional;
2. Tarefa de Envio de parâmetros dos planos;
3. Tarefa de Envio dos dados dos planos;
4. Tarefa de Cálculo da inversa da matriz M;

Figura 5.5: Funções que compõem a tarefa trabalhador apresentadas juntamente com a tarefa gerente na reconstrução bidimensional

5. Tarefa de Cálculo dos "pontos fantasmas";
6. Tarefa de Cálculo do vetor A_i da Equação 2.25;
7. Tarefa de Interpolação dos pontos virtuais nos planos;
8. Tarefa de Aglomeração de pontos interpolados;
9. Tarefa de Gravação dos planos reconstruídos.

Definidas as tarefas necessárias para paralelização da reconstrução 3D, definiu-se também a forma pela qual os processos estabeleceriam comunicação entre si e quais dados seriam trocados entre eles. A Figura 5.6 apresenta esta modelagem desenvolvida para comunicação a partir das tarefas particionadas. Nela, percebe-se que a tarefa de "Leitura dos dados de reconstrução tridimensional" é responsável pela leitura dos parâmetros da reconstrução e dados dos planos reais reconstruídos. Uma vez que tenha lido os parâmetros da reconstrução, tais como a quantidade de planos reais, suas dimensões e a quantidade de planos virtuais que serão interpolados, o processo "Envio de parâmetros dos planos" é notificado e envia estas informações à tarefa "Cálculo da matriz M^{-1} ". Desta forma, enquanto as informações dos planos reais são lidas, o processo 4 irá deixar este dado preparado e disponível para a tarefa "Cálculo dos pontos fantasmas". Com os dados de reconstrução todos lidos e a matriz inversa de M calculada, o processo "Envio dos dados dos planos" inicia a transmissão de vetores formados dos pontos de coordenada (x,y) de cada plano para o processo 5, que insere os "pontos fantasmas" no início e no fim do vetor e transmite para a tarefa "Cálculo do vetor A_i ". Esta tarefa realiza a multiplicação dos dados que recebeu pela matriz inversa de M e disponibiliza o vetor A_i para a tarefa "Interpolação dos pontos dos planos virtuais", que irá gerar, a partir destes dados, utilizando a Equação (2.23), vetores de tamanho NTF. Estes vetores contém os pontos interpolados entre os pontos reais. Cada vetor de pontos interpolados é passado para a tarefa "Aglomeração dos vetores de pontos interpolados", a qual reúne toda a informação de interpolação de forma organizada. Terminados os pontos a serem interpolados, a tarefa 8 envia os NTI planos para a tarefa "Gravação de planos reconstruídos", que armazenará em disco a informação reconstruída.

Figura 5.6: Esquematização do particionamento e comunicação das tarefas da reconstrução tridimensional baseada em interpolação por *B-Wavelet*.

Para aglomeração das tarefas, foram utilizados os mesmos moldes do que foi realizado na reconstrução bidimensional, gerando-se um algoritmo paralelo que segue o modelo gerente trabalhador. Reuniram-se em um único processo, o qual foi denominado como gerente, as tarefas 1 e 9, que necessitam acessar dados externos à plataforma paralela. Neste mesmo processo aglomerado, adicionou-se as tarefas 2 e 3, que são responsáveis por separar as informações de reconstrução, e a tarefa 8 que agrupa os dados reconstruídos. Organizando estes processos num único processo gerente se consegue obter um maior grau de paralelismo entre as tarefas que responsáveis pela separação/síntese e os processos de reconstrução tridimensional. Os processos 5, 4, 6 e 7 foram reunidos em um processo único, denominado de trabalhador, responsável pela reconstrução de blocos do objeto tridimensional. Com a interpolação tornando-se um processo trabalhador único, aumentou-se a granularidade da comunicação, buscando a redução dos custos de comunicação. Cada processo trabalhador de interpolação deixou de receber um vetor de pontos e passou a receber um bloco com parte dos planos reais para reconstruir tridimensionalmente. As dimensões de cada bloco poderiam ser determinadas de diferentes formas, uma vez que não há necessidade de troca de dados entre os processos trabalhadores durante a interpolação. Optou-se por criar blocos a partir dos dados originais para envio dos processos trabalhadores com dimensões definidas como sendo:

$$NC \times NL_t \times NR \quad (5.3)$$

onde NC é o número de colunas do plano, NL_t é a quantidade de linhas do plano dividida pelo número de processos trabalhadores e NR é o numero de planos reais utilizados para reconstrução. Após o término da interpolação, cada trabalhador retornará um bloco com dimensões $NC \times NL_t \times NTF$ ao processo gerente, que irá aglomerá-lo e salvá-lo quando todos os trabalhadores finalizarem sua carga de trabalho. A aglomeração modelada resulta em uma organização de tarefas que graficamente está representada na Figura 5.7.

A organização do algoritmo aglomerado permite que, no cálculo da interpolação dos planos, seja necessário um número menor de troca de dados entre gerente e trabalhado-

Figura 5.7: Aglomeração das tarefas de reconstrução tridimensional paralela

res, passando de $NC \times NL$ transmissões para 3 transmissões por trabalhador. Adicionalmente, o algoritmo permite que o intenso envio de pontos fantasmas e vetores A_i seja realizado dentro de um único processador. O paralelismo existente entre as tarefas aglomeradas dentro do trabalhador, desta forma, é ampliado.

5.3.1 Implementação da reconstrução 3D na plataforma DSP

Para definir os parâmetros de reconstrução, utiliza-se um arquivo de configuração onde se define:

- Número de linhas dos planos;
- Número de colunas dos planos;
- Quantidade de planos reais;
- Quantidade de planos virtuais entre cada par de planos reais;
- Lista de nomes dos arquivos de planos.

Quando se inicia a execução do algoritmo de reconstrução paralela na plataforma DSP, o gerente identifica cada trabalhador. Em seguida, como apresenta a Figura 5.8, o gerente lê o arquivo de configuração e, após a leitura, ele envia uma mensagem informando a quantidade de planos reais, planos interpolados e colunas para os processos trabalhadores para que eles gerem suas matrizes M^{-1} . Esta identificação será útil para o posicionamento dos dados interpolados quando o gerente estiver aglomerando os planos reconstruídos pelos trabalhadores. Com as informações dos parâmetros de reconstrução, os trabalhadores já se encarregarão, além do cálculo da inversa de M , de deixar alocadas as regiões de memória necessária para o processo de interpolação. Em paralelo a este processo, o gerente se encarrega de ler os dados definidos na lista de nomes de planos reais que está contida dentro do arquivo de configuração. Cada processo trabalhador, quando finaliza a sua geração da matriz M^{-1} , sinaliza ao gerente, indicando que está pronto para começar a interpolação dos planos. O algoritmo executado por cada processo trabalhador é mostrado na Figura 5.9.

Com os dados lidos, o gerente aplica um algoritmo de partição dos dados que divide os planos em blocos iguais e estes blocos são enviados a cada trabalhador que está identificado na rede, quando cada um deles começa o seu trabalho. O algoritmo de particionamento de dados divide, tal qual foi realizado na reconstrução 2D, a matriz que representa cada plano em uma mesma quantidade de linhas.

Conforme cada processo trabalhador envia seus resultados ao gerente, a função "AglomerarDados" organiza estas informações, deixando-as prontas para o processo de gravação de planos. A Figura 5.10 ilustra a dinâmica da execução e da comunicação entre os processos gerente e trabalhador.

5.3.1.1 Formato de gravação dos cortes tomográficos

Os dados resultantes da interpolação são armazenados em uma base de dados, ficando disponíveis para realização de estudos. Para gravação dos dados finais resultantes da interpolação de planos, foram utilizados três formatos, sendo um deles um formato em *rowdata*

```

Processo Gerente ( String ArquivoConfig, Int TotalTrab)
IdentificarTrabalhadores( TotalTrab );
[ Parâmetros, ListaCortes ] = LerArquivo( ArquivoConfig );
BroadCast( Parâmetros );
LerDadosCortes( ListaCortes );
RespostaTrabalhador = 0;
Enquanto ( RespostaTrabalhador < TotalTrab )
    Se ( ReceberMsg(IdTrab, OkMatrizM) )
        RespostaTrabalhador = RespostaTrabalhador + 1;
Fim Se
    Fim Enquanto
CargaTrabalho = ParticionarDados();
Para IdTrab = 1 até TotalTrab
    EnviarMensagem(IdGerente, IdTrab,
CargaTrabalho[IdTrab]);
Fim Para
RespostaTrabalhador = 0;
Enquanto ( RespostaTrabalhador < TotalTrab )
    Se( ReceberMensagem(IdTrab, DadosInterpolados) )
        AglomerarDados(DadosInterpolados, IdTrab );
        RespostaTrabalhador = RespostaTrabalhador + 1;
    Fim Se
Fim Enquanto
GravarPlanosInterpolados();
Fim Processo

```

Figura 5.8: Algoritmo em alto nível do processo gerente

```

Processo Trabalhador ( )
IdTrab = AguardarIdentificacao( );
ReceberMensagem(IdGerente, Parametros);
M = CalcularMatrizMinversa( Parametros );
AvisarGerente( );
AlocarMemoria( );
ReceberMensagem(IdGerente, CargaTrabalho );
Para i = 0 até CargaTrabalho.Linhas
Para j = 0 até CargaTrabalho.Colunas
    vetorPontos = ObterPontos(CargaTrabalho, i, j );
    vetorFantasmas = GerarPontoFantasmas( vetorPontos );
    vetorA = vetorFantasmas * M;
    vetorInterpolado = InterpolacaoBWavelet(vetorA);
    AglomerarVetor( vetorInterpolado )
Fim Para
Fim Para
EnviarMensagem(IdTrab, IdGerente, BlocoInterpolado);
Fim Processo

```

Figura 5.9: Algoritmo em alto nível do processo trabalhador

proprietário, definido como formato *.ddd* e outros dois compatíveis com os formatos definidos para o VTK (SCHRODER; AVILA, 2005).

O formato *.ddd* foi utilizado para visualização nas ferramentas desenvolvidas. Este formato compreende um arquivo em formato binário no qual tem-se um cabeçalho em cujos seis primeiros *bytes* definem-se :

- Quantidade de colunas (2 *bytes*);

Figura 5.10: Diagrama de blocos dos algoritmo paralelo de reconstrução 3D, ilustrando a interação entre gerente e trabalhador

- Quantidade de linhas (2 *bytes*);
- Quantidade de planos (2 *bytes*)

Os demais *bytes* do arquivo contêm os dados do objeto, com cada *voxel* sendo representado por um *byte*. Sendo assim, a escala de valores varia entre 0 e 255.

Em relação aos arquivos gerados em formato compreensível pelas bibliotecas do VTK, foram utilizados dois padrões.

O padrão com extensão *.vtk* é um arquivo texto que descreve, através de um conjunto de campos, os dados do objeto tridimensional. Uma das vantagens de utilizar este formato é sua portabilidade entre diversos visualizadores. Por exemplo, um arquivo gerado neste formato pode ser utilizado em uma ferramenta de código aberto como o *Paraview* (PARAVIEW, 2006). Uma característica interessante do *Paraview* é de distribuição de processamento dentre diversos computadores de uma rede, contribuindo para acelerar a visualização. Este ambiente de visualização foi desenvolvido inicialmente pela Kitware em parceria com o pesquisador Jim Ahrens do *Advanced Computing Laboratory* do *Los Alamos National Laboratory* e atualmente está sendo desenvolvido com participação de outros colaboradores. O formato do arquivo *.vtk* utilizado neste trabalho é mostrado Figura 5.11.

Outro formato compatível com a biblioteca VTK, que é exportado pela implementação, é o formato que gera imagens com dados em 16-bits. Nos arquivos, os dados de cada ponto de um plano são armazenados em um valor inteiro de 16-bits. Os arquivos de cada plano devem ser gravados seguindo um mesmo nome padrão, por exemplo "NomeCorte" e a extensão deve ser um valor inteiro $i \geq 0$. Com isso, os cortes são gravados como "NomeCorte.0", "NomeCorte.1" e assim por diante. Dentro das classes que compõem a hierarquia do VTK, existe uma classe denominada *vtkVolume16Reader*, que permite, através de seus métodos, escolher um conjunto de planos reconstruídos para visualização. Desta forma, é possível reduzir a demanda por memória e aumentar a velocidade de carregamento do objeto durante a sua visualização.

#vtk DataFile Version 3.0	→	Cabeçalho
Título	→	Nome do objeto
ASCII	→	Formato do arquivo
DATASET STRUCTURED_POINTS	→	Geometria/Topologia dos pontos)
DIMENSIONS NC NL NTI	→	Dimensões do objeto 3D)
SPACING 1 1 1	→	Espaçamento entre os pontos)
ORIGIN X_0 Y_0 Z_0	→	Origem do objeto)
POINT_DATA $NC \times NL \times NTI$	→	Quantidade de pontos)
SCALARS scalars unsigned_short	→	Formato dos dados)
LOOKUP_TABLE default	→	Tabela de cores)
$P(1,1,1)$ $P(1,2,1)$ $\cdots$ $P(1,NC,1)$		
$P(2,1,1)$ $P(2,2,1)$ $\cdots$ $P(2,NC,1)$		
$\vdots$ $\vdots$ $\vdots$ $\vdots$		
$P(NL,1,1)$ $P(NL,2,1)$ $\cdots$ $P(NL,NC,1)$	→	Conjunto de pontos
$P(1,1,2)$ $P(1,2,2)$ $\cdots$ $P(1,NC,2)$		
$\vdots$ $\vdots$ $\vdots$ $\vdots$		
$P(NL,1,1)$ $P(NL,2,1)$ $\cdots$ $P(NL,NC,NTI)$		

Figura 5.11: Formato de arquivo *.vtk* utilizado na descrição de um objeto tridimensional

Mais informações a respeito desta e de outras classe que compõem o VTK, podem ser obtidas em (KITWARE, 2006).

6 *Resultados e Conclusões*

6.1 **Aplicação do modelo de filtragem de Wiener em amostras de *phantoms* homogêneos e heterogêneos**

Dentro do modelo de reconstrução tridimensional desenvolvido, aplica-se o filtro de Wiener por predição, *a priori* para reduzir os efeitos do ruído Poisson nas projeções. Para isso, utiliza-se o modelo de filtragem apresentado na Figura 6.1.

Figura 6.1: Diagrama de blocos da filtragem de Wiener por predição

Utiliza-se a transformada Anscombe antes da entrada das projeções no filtro que tornará o ruído independente do sinal. Em seguida, realizou-se a filtragem por predição e com os dados sofrendo uma transformação inversa após esta etapa.

Para avaliar a eficiência do filtro nesta tarefa, aplicou-se a filtragem em dois *phantoms* de calibração, sendo um deles homogêneo e o outro heterogêneo. Os resultados foram comparados com os obtidos na aplicação de um filtro por mediana.

No conjunto, adicionou-se um ruído gaussiano e, em seguida, aplicou a filtragem por mediana e por predição no conjunto de projeções.

Na avaliação do filtro por mediana, foram utilizada mascaras de dimensão $[1 \times 3]$, $[1 \times 5]$ e $[1 \times 7]$. Para a filtragem por predição, utilizaram-se filtros com 2, 4 e 6 pesos. A caracterização da melhora do filtro foi avaliada através da análise do maior erro, que permite medir o

quanto o filtro aproximou o sinal ruidoso do sinal original.

No estudo do *phantom* homogêneo, normalizou-se o conjunto de projeções e aplicou-se o ruído gaussiano. Extraiu-se uma projeção com um valor de maior erro igual a 0,083. Os resultados obtidos para esta projeção após a filtragem são mostrados na Tabela 6.1.

Pesos	Wiener		Máscaras	Mediana
p = 2	0,043		1x3	0,044
p = 4	0,040		1x5	0,041
p = 6	0,039		1x7	0,071

Tabela 6.1: Tabela com valores de maior erro obtidos de uma projeção do *phantom* homogêneo com ruído, após a aplicação dos filtros por predição de Wiener e por mediana.

Como é possível observar na Tabela 6.1, no caso da filtragem do *phantom* homogêneo, o filtro de Wiener com seis pesos foi o que forneceu um melhor resultado, dado que o valor de erro foi mais atenuado nesta configuração. Os resultados obtidos com os dois modelos na filtragem, a partir projeção extraída, são mostrados na Figura 6.2.

Figura 6.2: Projeção homogênea - (a)Filtro de Wiener por predição; (b)Filtragem por mediana

A aplicação das filtrações no conjunto de projeções pode ser observado na Figura 6.3 onde apresenta-se o conjunto original de projeções, o conjunto ruidoso e os resultados obtidos com as projeções filtradas com o filtro mediana com mascara [1x5] e com o filtro por predição de Wiener com 6 pesos.

Figura 6.3: Conjunto de projeções do *phantom* homogêneo - (a) Originais; (b) Ruidosas; (c) Filtro por Mediana com máscara [1x5] (d) Filtro de Wiener por Predição com 6 pesos.

Após a filtragem das projeções, realizou-se o processo de reconstrução bidimensional e estabeleceu-se o valor de variância de uma região de interesse (ROI) no centro da imagem reconstruída com dimensão de 15x15 pixels. Como o *phantom* utilizado é homogêneo, os menores valores desta medida determinam uma melhor qualidade de filtragem. A Tabela 6.2 apresenta os valores de variância obtidos nas imagens reconstruída a partir de cada filtragem.

As Figuras 6.1(a),(b) e (c) apresentam, respectivamente, as imagens reconstruídas a partir de projeções originais, das projeções com inserção de ruído e das projeções filtradas com filtragem por predição de Wiener com 6 pesos..

Com um conjunto de projeções de um *phantom* heterogêneo, realizou-se o mesmo

Pesos	Wiener		Máscaras	Mediana
p = 2	0,010		1x3	0,016
p = 4	0,003		1x5	0,018
p = 6	0,002		1x7	0,018

Tabela 6.2: Tabela com valores de variância obtidos de uma região de interesse no centro das imagens reconstruídas a partir de projeções filtradas com os filtros por predição de Wiener e por mediana.

processo. Inseriu-se num conjunto sem ruído, um ruído gaussiano e em seguida, aplicou-se os filtros por predição e por mediana. Para verificar o desempenho dos filtros, extraíram-se medidas, apresentadas na Figura 6.5, dos valores de maior erro em todas as projeções. Desta forma estabeleceu-se o comportamento dos filtros no conjunto de projeções heterogêneas.

Observa-se que na região entre os passos 23 e 37, a filtragem por predição de Wiener demonstrou melhores resultados. Nas demais regiões, a filtragem por mediana demonstrou maior eficiência na filtragem. Na Figura 6.6 é apresentada a filtragem de uma projeção contida nessa região, cujos valores de maior erro em cada filtragem é apresentado na Tabela 6.3.

Pesos	Wiener		Máscaras	Mediana
p = 2	0,122		1x3	0,199
p = 4	0,130		1x5	0,214
p = 6	0,129		1x7	0,222

Tabela 6.3: Tabela com valores de maior erro obtidos de um *phantom* heterogênea, após a aplicação do filtro de Wiener por predição e filtragem por mediana

As Figuras 6.7 (a), (b), (c) e (d) apresentam os conjuntos com as projeções originais,

(a)

(b)

(c)

Figura 6.4: Imagens reconstruídas de um *phantom* homogêneo, a partir das projeções (a)originais; (b)ruidosas; (c)filtradas predição de Wiener com 6 pesos.

Figura 6.5: Comparação do maior erro após a aplicação das filtragens

Figura 6.6: Aplicação da filtragem em uma projeção da amostra heterogênea (a) Filtro de Wiener por Predição (b) Filtro por Mediana

ruidosas, filtradas por predição com 2 pesos e por mediana com máscara 7x1, respectivamente.

Após a filtragem das projeções, realizou-se o processo de reconstrução bidimensional e estabeleceu-se o valor de variância de quatro regiões de interesse correspondente a quatro elementos químicos presentes no *phantom* heterogêneo. A Tabela 6.4 apresenta os valores de variância obtidos nas imagens reconstruídas a partir de cada filtragem.

As Figuras 6.1(a),(b) e (c) apresentam, respectivamente, as imagens reconstruídas a partir de projeções originais, das projeções com inserção de ruído e das projeções filtradas com filtragem por predição de Wiener com 2 pesos.

Figura 6.7: Conjunto de projeções do *phantom* heterogêneo - (a) Originais; (b) Ruidosas; (c) Filtro por Mediana com máscara $[7 \times 1]$; (d) Filtro de Wiener por Predição de 2 pesos

Figura 6.8: Imagens reconstruídas de um *phantom* heterogêneo, a partir das projeções (a) originais; (b) ruidosas; (c) filtradas predição de Wiener com 2 pesos.

6.2 Interface com o usuário e algoritmos de reconstrução

6.2.1 Interface de visualização bidimensional

Para ativar a aplicação do algoritmo paralelo de reconstrução bidimensional e vi-

Filtro	ROI1	ROI2	ROI3	ROI 4
Wiener (p=2)	0,033	0,007	0,082	0,047
Wiener (p=4)	0,031	0,008	0,080	0,045
Wiener (p=6)	0,032	0,009	0,080	0,046
Mediana [1x3]	0,040	0,008	0,089	0,047
Mediana [1x5]	0,039	0,009	0,084	0,046
Mediana [1x7]	0,037	0,009	0,084	0,046

Tabela 6.4: Tabela com valores de variância obtidos de quatro ROIs, os quais foram extraídos de imagens de um *phantom* heterogêneo, reconstruídas a partir de projeções filtradas com os filtros por predição de Wiener e por mediana.

utilizando o compilador *Borland Builder C++* versão 6. Esta interface é apresentada na Figura 6.9, onde exibe-se a imagem reconstruída de um *phantom* heterogêneo.

A ferramenta atua como um *front-end*, enviando os parâmetros de reconstrução para a plataforma paralela. A interface possibilita ainda a escolha de regiões de interesse e o ajuste da escala de cores para visualização da imagem. Através dela, também se podem realizar a extração de valores de coeficiente de atenuação da amostra e a verificação dos parâmetros de varredura utilizados, tais como:

- a energia;
- a translação total ;
- a rotação total;
- o passo linear;
- o passo angular.

6.2.2 Avaliação da correlação entre coeficiente de atenuação linear e tons de cinza

Para avaliar a precisão do algoritmo paralelo de reconstrução 2D, buscou-se estabelecer a correlação existente entre as variáveis, coeficiente de atenuação e tons de cinza da imagem reconstruída. O coeficiente de correlação mede a relação existente entre as variáveis.

Figura 6.9: Janela do sistema que permite realizar a escolha da amostra tomográfica e visualização dos cortes reconstruídos

Se o seu resultado estiver próximo de +1 ou de -1, isto indica que os dados se ajustam bem à reta estimada (VANNI, 1998).

Na avaliação, utilizou-se um *phantom* heterogêneo de *Plexiglass*, com 6 cm de diâmetro, que contém os elementos cálcio, fósforo, alumínio e água, como ilustra a Figura 6.10.

Para o processo de tomografia, usaram-se os seguintes parâmetros:

Figura 6.10: *Phantom* de calibração com os elementos Cálcio(Ca), Alumínio(Al), Fósforo(P), Água(H₂O) e Plexiglass

Translação total :	6 cm;
Passo linear :	0,100 cm;
Rotação total :	180°;
Passo angular :	3,000°;
Tempo de contagem :	10s;
Fótons na região de feixe livre:	10.000
Energia:	60,0 keV.

A partir das projeções obtidas com esses parâmetros, aplicou-se o algoritmo de reconstrução paralelo. Os valores de tons de cinza obtidos, para cada um dos cinco elementos, são apresentados na Tabela 6.5, juntamente com os valores de coeficiente de atenuação linear de cada elemento.

Elemento	Coeficiente de atenuação	Tons de cinza
Cálcio	1,001	232
Fósforo	0,748	173
Alumínio	0,631	130
Água	0,220	22
Plexiglass	0,140	11

Tabela 6.5: Tabela com os coeficientes de atenuação e os tons de cinza obtidos

Calculou-se, a partir desses valores, a correlação entre os coeficientes de atenuação e os tons de cinza da imagem reconstruída. O coeficiente de correlação encontrada entre as duas

variáveis foi de 0,996, mostrando que existe uma forte relação linear entre elas, tornado viável o ajuste de uma reta.

Desta forma, a curva de calibração é obtida através do ajuste, e pode ser observada na Figura 6.11.

Figura 6.11: Gráfico do ajuste da curva de calibração obtida a partir dos dados de coeficiente de atenuação(cm^{-1}) $\times$ tons de cinza da imagem

Observa-se que a qualidade do ajuste da curva de calibração aos pontos observados, possibilita uma boa representatividade do real coeficiente de atenuação na imagem reconstruída.

A análise da variância da curva de calibração com os pontos observados apresentou uma variância igual a 31,04.

6.2.3 Interface de visualização tridimensional

Desenvolveu-se uma aplicação de visualização tridimensional que tem por objetivo, auxiliar a análise dos resultados e a precisão do algoritmo paralelo (PEREIRA; CRUVINEL, 2005). A ferramenta foi desenvolvida em ambiente *Windows*, utilizando o compilador *Borland Builder*

C++ versão 6, combinado com a biblioteca gráfica VTK, em sua versão 4.2, que foi compilada nesta mesma versão do compilador. A biblioteca VTK também fornece código-aberto para criação de um componente de renderização compatível com a biblioteca VCL do *Builder C++*, o qual pode ser inserido dentro da janela de visualização da ferramenta. A compilação deste código gera o componente de visualização *TvtkBorlandRenderWindow*.

A partir dos dados gerados pela reconstrução tridimensional, modelou-se um sistema de visualização para realização das análises que tem as seguintes funcionalidades:

- Visualização e interação com o objeto reconstruído;
- Conversão de arquivos de corte em arquivos *.vtk*;
- *Threshold* do objeto 3D, ajustando limiares mínimo e máximo;
- Visualização de cortes Sagitais, Coronais e Transversais, ao longo do eixo X, Y e Z, respectivamente;
- Criação de projetos de reconstrução;
- *Front-End* para execução dos algoritmos de reconstrução 2D e 3D.

O diagrama de classes apresentado na Figura 6.12 mostra, em alto nível, a organização de classes do cerne da aplicação. As classes de janelas do *Builder* e as classes específicas da biblioteca encontram-se representadas em alto nível.

A classe *T3DVisualizador* centraliza as ações de carregamento em memória dos objetos reconstruídos, ajuste de limiares de *threshold* e visualização de cortes. Para isto, ela interage com os diversos componentes do sistema enviando e recebendo informações. A interação com o objeto é realizada no componente *TvtkBorlandRenderWindow*, que fornece as funcionalidades para realizar os movimentos de rotação e translação do objeto 3D.

Figura 6.12: Diagrama de classes da aplicação Viewer3D

A janela principal com a visualização de objetos tridimensionais é apresentada na Figura 6.13. Nela, é exibida uma imagem 3D de um *phantom* heterogêneo que foi reconstruído utilizando o algoritmo paralelo.

Através da escolha de limiares de *threshold*, é possível estudar a distribuição dos coeficientes dentro do objeto tridimensional, realizando-se, desta maneira, um estudo mais minucioso a respeito do interior da amostra. Essa funcionalidade da ferramenta é exibida na Figura 6.14, onde exibe-se o objeto com limiares variando entre 70 e 255.

Outra característica da ferramenta é a visualização dos cortes sagitais, coronais e transversais das amostras. Essas visualizações permitem avaliar o interior do objeto nas 3 dimensões, ajudando a observar as regiões internas do objeto.

As Figuras 6.15, 6.16 e 6.17, respectivamente, apresentam um corte sagital, coronal e transversal do amostra.

Esta funcionalidade da ferramenta permite também a observação de heterogeneidades e o dimensionamento do nível de compactação do solo nas camadas superiores, tal qual pode ser observado no corte sagital e coronal apresentado nas Figuras 6.15 e 6.16.

Figura 6.13: Interface do software de visualização onde é possível ao usuário interagir com o objeto reconstruído

Utilizando-se a opção de visualização dos valores dos *voxels*¹, da ferramenta, pode-se obter o valor da atenuação relativa ao ponto de intersecção entre os três planos. Essa opção, mostrada na Figura 6.18, insere um cursor que auxilia o usuário a localizar o ponto que está sendo medido no objeto tridimensional. O valor obtido é apresentado na interface gráfica.

6.2.3.1 Avaliação da precisão da aglomeração de blocos

Durante o desenvolvimento da reconstrução tridimensional, percebeu-se a necessidade de criar um método para verificar-se a aglomeração realizada pelo algoritmo estava correta. Desta forma, criou-se um padrão, tal qual o apresentado na Figura 6.19, o qual permite perceber desalinhamentos das regiões das imagens. De fato, os primeiros resultados obtidos pela aglomeração apresentavam pequenos erros no alinhamento dos blocos gerados por falhas na implementação. A partir do uso deste padrão, foi possível detectar as falhas ocorridas na aglomeração e determinar as ações necessárias para o perfeito ajuste da aglomeração.

¹Um *voxel* é um elemento de volume que representa um valor em uma grade regular em um espaço tridimensional, de forma análoga a um pixel em uma imagem bidimensional.

Figura 6.14: Janela do sistema apresentando a visualização de voxels do objeto que estão dentro de uma região de interesse

Figura 6.15: Visualização do corte sagital do *phantom* heterogêneo.

Figura 6.16: Visualização do corte coronal do *phantom* heterogêneo. Figura 6.17: Visualização do corte transversal do *phantom* heterogêneo.

Figura 6.18: Visualização de um corte sagital combinada com a extração de uma medida de intensidade de um *voxel*.

A Figura 6.20 apresenta o resultado obtido a partir da geração de um objeto 3D, na plataforma DSP, com o uso de 4 processadores.

Na Figura 6.20, observa-se que a aglomeração realizada pelo processo gerente na plataforma DSP cria objetos que apresentam um perfeito alinhamento entre os blocos. Isso pode ser percebido pelas pequenas linhas vazadas dentro dos cubos em cinza. Este resultado garante que amostras de solo e madeira, ao serem reconstruídas, não apresentem falhas no alinhamento dos blocos.

6.3 Estudo de caso

Para avaliar o potencial do modelo de visualização 3D desenvolvido, realizou-se o estudo de uma amostra de torrão de solo, obtida a partir do tomógrafo de resolução micrométrica da Embrapa Instrumentação. O objetivo da avaliação é demonstrar os potenciais do modelo como uma ferramenta de análise para aplicação em pesquisas relacionadas à ciência do solo.

Figura 6.19: Plano gerado para calibrar a aglomeração de blocos durante a reconstrução 3D

Figura 6.20: Objeto tridimensional gerado na plataforma DSP a partir do plano padrão

No estudo, utilizaram-se dois cortes tomográficos que foram adquiridos com os seguintes parâmetros:

Translação total :	15,000 mm;
Passo linear :	0,083 mm;
Rotação total :	180°;
Passo angular :	1,000°;
Tempo de contagem :	4s;
Fótons na região de feixe livre:	20.000
Energia:	58,5keV.

As imagens reconstruídas dos dois cortes são apresentadas, em pseudocores, na Figura 6.21. Os dados de projeção destes dois cortes foram adquiridos nos horizontes 88 mm, Figura 6.21(a), e 158 mm, Figura 6.21(b), do torrão de solo. Na reconstrução tridimensional, foram inseridos 69 cortes virtuais entre o par de planos reais, fazendo com que cada plano representasse profundidades com variações de 1 mm, entre 88 mm e 158 mm.

Figura 6.21: Imagens de dois cortes reconstruídos de um torrão de solo nos horizontes de (a) 88 mm; (b) 158 mm

A Figura 6.22 apresenta os cortes transversal (plano 34), coronal (plano 89) e sagital (plano 119) do torrão de solo. O modelo disponibiliza meios para observar a variação de densidade ao longo dos planos da amostra, bem como a continuidade de poros no objeto. A Figura 6.22(d) apresenta três medidas de coeficiente de atenuação linear extraídas através do modelo de visualização.

As Figuras 6.23(a), (b) e (c) apresentam as imagens do torrão de solo, com três diferentes faixas de limiares aplicadas, as quais foram definidas, respectivamente, para apresentar as tonalidades entre 255 e 0, entre 255 e 100 e entre 99 e 10. Essa funcionalidade do modelo de visualização permite ao usuário a realização de análises, através da escolha de limiares de interesse, para a caracterização de uma determinada distribuição de densidades dentro do objeto

Figura 6.22: Imagens dos cortes transversal(a), coronal(b), sagital(c) e medidas de coeficiente de atenuação linear(d) do torrão de solo.

reconstruído.

A interatividade do modelo de visualização 3D habilita aos usuários a observação com detalhes da amostra de solo. Na Figura 6.24, é apresentado o torrão de solo onde se realizou um movimento de aproximação. Com isso, pode-se analisar e quantificar os poros do torrão de solo.

Além de aproximar o objeto, também é possível girá-lo em relação aos eixos X, Y e Z, o que permite examinar as demais faces do objeto. Esta combinação entre interatividade, escolha de faixas de limiares e visualização de cortes habilita a extração de características do solo, bem como fornece ferramentas para a análise de fenômenos dinâmicos do solo.

Figura 6.23: Amostra do torrão de solo apresentada com diferentes faixas de limiares para as tonalidades: (a) Entre 255 e 0; (b) Entre 255 e 100; (c) Entre 99 e 10

6.4 Resultados do método paralelo

6.4.1 Resultados obtidos na reconstrução bidimensional na plataforma DSP

Para avaliar o algoritmo paralelo de reconstrução bidimensional, fez-se um estudo do seu desempenho, adquirindo-se medidas com relação ao ganho e eficiência e traçando-se o perfil dos processos trabalhadores. Para este estudo, utilizou-se um conjunto de dados tomográficos adquiridos pelos minitomógrafos de resolução milimétrica, de campo e o de resolução micrométrica da Embrapa Instrumentação Agropecuária. Estes dados foram obtidos a partir de amostras de solo, madeira e *phantoms* de calibração. As amostras utilizadas possuem matrizes de projeção com resolução variando entre 41x41 até 251x251. Devido às limitações de memória, na arquitetura DSP, trabalhou-se com as resoluções variando entre 41x41 até 181 x 181. As

Figura 6.24: Exemplo de interatividade do modelo de visualização, o qual permite que se execute movimentos de rotação e aproximação das amostras

imagens obtidas através da reconstrução destas amostras são apresentadas na Figura 6.25. A tabela 6.6 apresenta as informações de cada amostra utilizada neste trabalho.

Resolução	Descrição	Dimensão	Passo Linear	Passo Angular	Energia
41 × 41	Latossolo Roxo	8,000 cm	0,200 cm	4,500°	60 keV
51 × 51	Solo Gley	5,000 cm	0,100 cm	3,600°	60 keV
64 × 64	<i>Phantom</i> de Calibração	6,400 cm	0,100 cm	2,830°	662 keV
76 × 76	Madeira (Horizonte A)	15,000 cm	0,200 cm	2,400°	60 keV
81 × 81	Madeira (Horizonte B)	8,000 cm	0,100 cm	2,250°	60 keV
101 × 101	Esfera de vidro	4,000 mm	0,400 mm	1,800°	56 keV
121 × 121	Granito	13,000 mm	0,108 mm	1,500°	58 keV
145 × 145	Grãos de areia	7,000 mm	0,048 mm	1,250°	56 keV
151 × 151	Bloco de solo	15,000 mm	0,100 mm	1,200°	58 keV
181 × 181	Torrão de solo	18,000 mm	0,100 mm	1,000°	58 keV
201 × 201	Madeira	13,000 mm	0,065 mm	0,900°	28,3 keV
221 × 221	Grãos de areia	14,000 mm	0,062 mm	0,800°	60 keV
251 × 251	Compósito	16,400 mm	0,065 mm	0,720°	28,3 keV

Tabela 6.6: Descrição detalhada das configurações para obtenção dos dados tomográficos das amostras utilizadas na reconstrução paralela.

Inicialmente, para avaliar a importância da comunicação, estudaram-se dois tipos de mapeamento dos processos gerente e trabalhador:

1. Com uso do processador *Root* exclusivamente para o processo gerente e outros 3 processadores para trabalhadores;
2. Com uso do processador *Root* para uso concorrente entre Gerente e 1 trabalhador e os outros 3 processadores com um trabalhador cada.

Figura 6.25: Grupo de imagens reconstruídas com o algoritmo paralelo - (a) Latossolo roxo (41×41 pixels); (b) Solo Gley (51×51); (c) Phantom de Calibração (64×65); (d) Madeira (76×76); (e) Madeira no Horizonte B (81×81); (f) Esfera de vidro (101×101); (g) Granito (121×121); (h) Grãos de areia (145×145); (i) Bloco de solo (151×151); (j) Torrão de solo (181×181).

Na Figura 6.26, é mostrada uma comparação da eficiência obtida nas configurações 1 e 2. Observa-se que a utilização do processador *Root* com o mapeamento apenas do processo gerente, na configuração 1, forneceu como resultado um baixo ganho para um sistema; como consequência, a eficiência do sistema foi reduzida nesta configuração. Desta forma, os resultados obtidos com a configuração 2 demonstram, como era esperado, o uso de um mapeamento que privilegie o gerente, dando a ele um processador dedicado, não traz um ganho satisfatório ao algoritmo paralelo de reconstrução.

Figura 6.26: Comparação entre a eficiência da reconstrução bidimensional

Para entender o comportamento do algoritmo de reconstrução, fez-se um estudo da eficiência e do ganho obtido nas diferentes resoluções estudadas, avaliando-se o algoritmo em diferentes quantidades de processos trabalhadores (PEREIRA *et al.*, 2006). Foram estudadas configurações com 2, 3 e 4 processadores, comparando os resultados com a execução sequencial para se extrair o ganho. A partir do ganho e da quantidade de processadores utilizados, obteve-se a medida de eficiência no uso dos processadores. A Tabela 6.7 apresenta os resultados obtidos neste estudo.

Os tempos de execução dos algoritmos de reconstrução sequencial e paralelo são

Resolução	2 Processadores		3 Processadores		4 Processadores	
	Ganho	Eficiência	Ganho	Eficiência	Ganho	Eficiência
41 × 41	1,654	82,691%	2,114	75,463%	2,798	69,958%
51 × 51	1,736	86,799%	2,121	76,712%	2,843	71,072%
65 × 64	1,786	89,323%	2,490	82,994%	3,123	78,068%
76 × 76	1,845	92,274%	2,579	85,977%	3,196	79,900%
81 × 81	1,829	91,473%	2,568	85,615%	3,232	80,805%
101 × 101	1,924	96,184%	2,693	89,777%	3,394	84,856%
121 × 121	1,822	91,090%	2,605	86,821%	3,356	83,910%
145 × 145	1,819	90,952%	2,612	87,080%	3,445	86,129%
151 × 151	1,891	94,566%	2,713	90,430%	3,523	88,075%
181 × 181	1,967	98,356%	2,825	94,172%	3,682	92,059%

Tabela 6.7: Tabela com resultados das medidas de ganho e eficiência obtidos na reconstrução 2D nas diferentes amostras utilizadas em arquitetura paralela com 2, 3 e 4 processos trabalhadores

apresentados na Figura 6.27

A Figura 6.28 apresenta os dados obtidos na análise da eficiência. Observa-se que nas altas resoluções o algoritmo atinge um alto grau eficiência .

Além da análise do ganho e da eficiência do algoritmo paralelo, avaliou-se o comportamento dos processos trabalhadores, traçando-se um perfil da execução destes, com o objetivo de melhor compreender sua execução e tentar buscar maneiras de otimizar seu desempenho através da identificação de gargalos do sistema. Para obter estes dados, incluem-se na implementação pontos de medição de desempenho e, ao término da execução, cada trabalhador envia um conjunto de dados, em formato XML, que descrevem as medidas realizadas, seu tempo de execução e a porcentagem de tempo em relação ao tempo total de execução. Um trecho de exemplo dos dados enviados por um processo trabalhador é apresentado na Figura 6.29. Estas informações XML são gravadas em disco pelo processo gerente.

Na Figura 6.29, percebe-se entre as *tags* XML, as informações enviadas, tais como, o tipo de plataforma, a dimensão da imagem reconstruída e as descrições à respeito das medidas adquiridas do trabalhador.

A partir dos dados gerados por todos os trabalhadores nas diferentes resoluções estudadas, realizou-se a análise do perfil dos trabalhadores, a qual é apresentada na Figura 6.30,

Figura 6.27: Tempo de execução do algoritmo sequencial e dos algoritmos paralelos, na arquitetura paralela dedicada, com 2, 3 e 4 processadores

onde as informações enviadas pelos trabalhadores foram resumidas em 3 tipos:

- Comunicação;
- Filtragem;
- Retroprojeção.

Na Figura 6.31, observa-se o comportamento dos trabalhadores na reconstrução de uma amostra com matriz de projeção de dimensão 121×121 . É possível observar que a tarefa trabalhadora com identificação 2 apresenta um tempo de comunicação menor do que as outras tarefas, mostrando que algumas tarefas demandam mais tempo de comunicação do que outras. Isso é explicado pelo acesso privilegiado que a tarefa 2 tem ao gerente por estar no mesmo processador. A diferença no tempo demandado para comunicação é resultado do gargalo que ocorre no instante de retorno dos dados retroprojetados para o processo gerente, dado que todas as tarefas possuem mesma carga e terminam seus trabalhos em tempos muito próximos.

Figura 6.28: Gráfico com o comportamento da eficiência da reconstrução bidimensional para 2, 3 e 4 processadores DSP

6.4.1.1 Implementação em ambiente convencional do algoritmo paralelo

Realizou-se uma comparação com os resultados obtidos na plataforma DSP, através do desenvolvimento de três diferentes implementações do método paralelo em ambiente convencional em um microcomputador PC. As implementações foram desenvolvidas em ambiente *Windows*, utilizando a biblioteca de comunicação MPI, na versão MPICH2-1.0.3-1 para *Windows*. Desta maneira, buscou-se também examinar as potencialidades do método e do algoritmo propostos para a aplicação em ambientes convencionais, compostos de processadores com múltiplos núcleos. Para fazer o estudo das implementações, utilizou-se um computador com um processador Intel Core Duo operando a 1,66 GHz, com 1GBytes de memória e 2MBytes de memória *cache*.

Foi utilizado, em duas implementações, o mesmo algoritmo proposto no modelo, alterando-se apenas a biblioteca para o cálculo da FFT em uma delas. Na implementação A,

Figura 6.29: Dados em formato XML exportados por um trabalhador, durante o processo de reconstrução bidimensional

Figura 6.30: Perfil do desempenho médio dos processos trabalhadores

Figura 6.31: Desempenho dos processos trabalhadores na reconstrução da amostra com 121 projeções de 121 pontos

utilizou-se a biblioteca desenvolvida pelo grupo de pesquisa em Instrumentação Agropecuária da Embrapa. Na implementação B, empregou-se a biblioteca FFTW (FFTW, 2007), a qual disponibiliza um conjunto de funções otimizadas para cálculo de transformada de Fourier e é de uso livre.

Uma terceira implementação buscou ainda avaliar o efeito da eliminação do envio de projeções filtradas dos trabalhadores para o processo gerente. O objetivo desta implementação é reduzir os custos de comunicação do algoritmo e avaliar o impacto no desempenho. Ressalta-se, porém, que essa eliminação implica em acréscimo de trabalho nos processos gerente e trabalhador. A explicação do novo procedimento de retroprojeção ajudará a entender as razões deste acréscimo.

A nova organização obriga cada trabalhador a retroprojetar, em todos os pontos da imagem reconstruída, a contribuição de cada projeção que foi filtrada por ele. Desta forma, ele passa a fornecer ao gerente uma matriz com contribuições do seu conjunto de projeções, e não mais a parte da imagem já reconstruída. O processo gerente recebe as matrizes de cada trabalhador e as soma, gerando a imagem reconstruída. Desta forma, ocorre um aumento de trabalho para o processo gerente, durante o processo de reconstrução.

Na análise de desempenho dessas duas implementações, utilizou-se todo o conjunto de dados tomográficos que foi disponibilizado, trabalhando-se com matrizes de projeção cuja resolução variou entre 41x41 até 251x 251. As imagens reconstruídas de resolução 201x201, 221x221 e 251x251 são mostradas na Figura 6.32

O tempo e o ganho obtido na reconstrução, em cada uma das implementações, são apresentados nas Figuras 6.33 (a) e (b). Em todas as avaliações, utilizou-se uma configuração de três tarefas, com sendo o processo 0 o gerente e, as demais, os trabalhadores. Os valores de ganho foram calculados a partir da razão do tempo de cada algoritmo paralelo pelo tempo do algoritmo sequencial em plataforma convencional numa implementação que utilizou a biblioteca desenvolvida pelo grupo de pesquisa em Instrumentação Agropecuária da Embrapa.

Na Figura 6.33 (a) e (b) observa-se que o ganho obtido nas implementações A e B atingem um ganho elevado, nas resoluções mais altas, bem como, o uso da biblioteca FFTW, que proporcionou melhor desempenho em relação a biblioteca do grupo nestas resoluções.

O desempenho da implementação C forneceu ganho em algumas resoluções, porém sua aplicação não apresentou vantagens na maior parte delas, mostrando que o aumento da carga de trabalho, gerou um impacto na performance do algoritmo minimizando os efeitos da remoção da comunicação .

6.4.2 Resultados obtidos na reconstrução tridimensional

6.4.2.1 Organização da aplicação de reconstrução 3D na plataforma DSP

Para realizar estudos da precisão, do ganho e da eficiência do algoritmo paralelo de reconstrução tridimensional, utilizou-se a configuração que mapeia o processo gerente no processador *Root* e em cada processador da rede um processo trabalhador, incluindo o processador *Root*. Para isso, utilizou-se a configuração mostrada na Tabela 6.8.

(c)

Figura 6.32: Grupo de imagens reconstruídas com o algoritmo paralelo - (a)Madeira (201×201) (b)Grãos de areia (221×221); (c)Compósito (251×251)

Comparação do tempo de execução das implementações no ambiente convencional

(a)

Comparação do ganho nas implementações no ambiente convencional

(b)

Figura 6.33: Comparação do desempenho dos três algoritmos implementados em plataforma convencional. (a) Tempos de execução das implementações A, B e C (b) Ganho obtido em cada implementação.

task	master	file=rec3dm	stack = 150k	heap= 600k
task	worker	file=rec3dw	data = 700k	
task	rworker	file=rec3dw	data = 600k	

Tabela 6.8: Arquivo de mapeamento das tarefas na criação da aplicação paralela de reconstrução 3D na rede de processadores DSP.

6.4.2.2 Medidas de desempenho do algoritmo paralelo de reconstrução 3D

Para avaliar o algoritmo de reconstrução tridimensional na plataforma DSP foram utilizadas as mesmas amostras tomográficas utilizadas para aplicadas o desempenho da reconstrução bidimensional. Na análise de desempenho, foram estudadas três diferentes configurações de interpolação:

1. Com 3 planos reais e 5 virtuais interpolados a cada par de planos, num total de 13 planos;
2. Com 3 planos reais e 15 virtuais interpolados a cada par de planos, num total de 33 planos;
3. Com 6 planos reais e 5 virtuais interpolados a cada par de planos num total de 31 planos.

O objetivo destas três diferentes configurações é avaliar o impacto do aumento gradual de carga de trabalho e da quantidade de dados trocados entre gerente e trabalhadores. O estudo foi realizado na plataforma paralela de processadores DSP, com uso de 4 processos trabalhadores e um processo gerente.

Na avaliação da configuração 1, obteve-se um ganho médio de 2,852 e a eficiência média de 71%. As medidas revelaram que este resultado foi fruto da pouca computação executada nos trabalhadores aliada ao peso que a comunicação teve no processo de reconstrução. Esse última ocorre principalmente porque o tamanho dos pacotes trocados entre gerente e trabalhador são de granularidade muito pequena.

O perfil de ganho e eficiência da execução do algoritmo paralelo na configuração 1 é apresentado na Figura 6.34.

No estudo da configuração 2, percebe-se que ocorre um aumento da eficiência que atinge valores em torno de 77%. Nesta configuração, ocorre um aumento na carga de trabalho e

(a)

(b)

Figura 6.34: Medidas obtidas no estudo do desempenho do algoritmo paralelo na configuração 1 - (a)Ganho; (b)Eficiência

na quantidade de informação transmitida. O reflexo do uso desta configuração no desempenho do algoritmo é demonstrada através do gráfico apresentado na Figura 6.35(a) que mostra um aumento de ganho em relação à configuração 1, mas ainda distante do ideal.

(a)

(b)

Figura 6.35: Medidas obtidas no estudo do desempenho do algoritmo paralelo na configuração 2 - (a)Ganho; (b)Eficiência

Como é possível observar, ocorre um impacto causado pelo aumento da carga de trabalho. O tamanho de bloco reconstruído e transmitido por cada trabalhador para o gerente depois da interpolação aumenta de 13 planos por bloco para 33 planos por bloco. No desempenho do algoritmo de reconstrução, este efeito resulta aumento de ganho e eficiência.

A configuração 3 traz consigo uma outra dimensão de avaliação do algoritmo paralelo, pois ela efetivamente não aumenta a dimensão do objeto reconstruído, porém, em cada processo trabalhador, ocorreu um aumento da quantidade de operações realizadas, uma vez que existe um conjunto maior de planos reais. Conseqüentemente nos trabalhadores, aumenta-se a dimensão da matriz M^{-1} a ser calculada, da quantidade de nós do vetor de pontos de controle e do somatório de interpolação realizado pela Equação 2.24.

Realizou-se o estudo da configuração 3, obtendo-se o perfil de ganho e eficiência apresentado na Figura 6.36 (a) e (b).

Para a configuração 3, o ganho médio e a eficiência média do algoritmo nas diferentes resoluções estudadas foi de 3,403 e 85%, respectivamente. Estes resultados demonstram que as configurações de reconstrução de objetos que contêm um número maior de planos reais irão apresentar um desempenho superior às configurações que têm um menor número de planos reais. É interessante observar que mesmo, com o aumento do tamanho das mensagens, o algoritmo pouco é influenciado por este acréscimo, mantendo ao longo das configurações estudadas um ganho e eficiência com pouca variação.

6.5 Conclusões

Inicialmente, pode-se destacar a utilização do método PCAM para o desenvolvimento de algoritmos paralelos, o qual se caracterizou como uma importante ferramenta para impulsionar o crescimento da área de processamento distribuído. Neste trabalho, comprovou-se a sua utilidade na modelagem dos algoritmos paralelos implementados independentemente da máquina paralela onde é executado. Um exemplo do potencial do método foi o desenvolvimento da reconstrução bidimensional paralela que depois de modelada, pode ser mapeada tanto para um ambiente paralelo de processadores DSP quanto para um ambiente convencional composto de um processador de duplo núcleo.

A partir do resultado na reconstrução 2D, onde o ganho foi em média de 3,259 na configuração com 4 processos trabalhadores e 1 processo gerente, conclui-se que o valor de

(a)

(b)

Figura 6.36: Medidas obtidas no estudo do desempenho do algoritmo paralelo na configuração 3 - (a)Ganho; (b)Eficiência

ganho obtido foi considerado satisfatório, à medida que o algoritmo atinge níveis próximos dos considerados ideais em resoluções mais altas. Este ganho em desempenho conseguido através do algoritmo paralelo na reconstrução 2D contribui de forma positiva para reduzir o tempo de criação dos objetos tridimensionais, a partir de cortes bidimensionais.

A análise da reconstrução com um processador alocado exclusivamente para o processo gerente demonstrou-se ineficiente, confirmando não ser adequada esta configuração para a reconstrução bidimensional.

A avaliação da escalabilidade do algoritmo paralelo de reconstrução 2D foi realizada através de sua execução em 2, 3 e 4 processadores DSP. Verificou-se que houve uma queda da eficiência média de 91% em 2 processadores para 81% em 4 processadores, nas diferentes resoluções de amostras agrícolas estudadas. O resultado demonstrou uma boa escalabilidade do algoritmo com o aumento da quantidade de processadores.

O uso da biblioteca FFTW, em arquitetura convencional, proporcionou um aumento de desempenho do algoritmo paralelo de reconstrução 2D. Na sua implementação obteve-se, nas altas resoluções, um ganho de desempenho superior ao ganho obtido com a implementação que utilizou a biblioteca de FFT do grupo de pesquisa em Instrumentação Agropecuária da Embrapa.

Na avaliação do modelo de reconstrução 3D, a análise das configurações 1 (três planos reais e cinco interpolados), 2 (três planos reais e quinze interpolados) e 3 (seis planos reais e cinco interpolados) mostrou que, apesar do aumento da quantidade de pontos calculados em cada objeto, não há um impacto significativo na eficiência do algoritmo ao longo das resoluções estudadas. A avaliação revelou ainda que a configuração 3 forneceu os melhores resultados com o algoritmo paralelo atingindo uma eficiência média de 85% e um ganho da ordem de 3,403. Também foi observado que não ocorre grande variação no desempenho do algoritmo à medida que se aumenta a dimensão do objeto reconstruído, demonstrando desta maneira a sua escalabilidade.

A aplicação desenvolvida para visualizar a reconstrução bidimensional mostrou ser uma importante ferramenta para a análise e a visualização dos resultados gerados pela reconstrução paralela, contribuindo inclusive na correção de problemas do algoritmo, como erros na aglomeração de pixels, na reconstrução e na filtragem das projeções. Adicionalmente, destaca-se o potencial da ferramenta na análise e extração de informações das amostras tomográficas de

solo e madeira, tais como densidade e umidade.

A interpolação de planos virtuais através da técnica de interpolação por *B-Wavelet* mostrou-se adequada ao modelo, principalmente quando se leva em consideração a qualidade obtida na geração dos planos interpolados e sua característica de não demandar dos algoritmos paralelos uma intensa troca de informações entre os processadores ao longo do processo de reconstrução. Conforme foi descrito, na modelagem do algoritmo paralelo de reconstrução 3D, cada processo trabalhador comunica-se três vezes com o processo gerente durante a reconstrução de seu bloco, não havendo necessidade de troca de dados com os outros processos trabalhadores ao longo do processo.

O uso da biblioteca VTK na criação da ferramenta de visualização tridimensional enriqueceu as funcionalidades da aplicação e permitiu maior interação de usuários com o objeto reconstruído. Através da modelagem projetada para a visualização, foi possível gerar objetos em formato compatível com as classes da biblioteca VTK, bem como gerar ferramentas de análise que permitiram a seleção e visualização de partes do objeto reconstruído. Seu uso também possibilitou a verificação e correção de problemas da reconstrução paralela 3D, tais como o problema de alinhamento de blocos e erros na implementação do algoritmo de interpolação por *B-Wavelet*.

Finalmente, conclui-se que os resultados apresentados contribuem com o estabelecimento de um novo modelo para reconstrução tridimensional de amostras agrícolas em ambientes que demandem alto poder de processamento, bem como indicam sua disponibilização para o estudo de fenômenos dinâmicos presentes na análise da ciência de solo.

Referências

3L. *PARALLEL C - USER GUIDE Texas Instruments TMS320C40*. [S.l.], 1995.

ANSCOMBE, F. J. The transformation of poisson, binomial and negative-binomial data. *Biometrika*, v. 35, p. 246–254, 1948.

AYLMORE, L.; HAINSWORTH, J. M. The use of the computed-assisted tomography to determine spatial distribution of soil water content. *Australian Journal Soil Res*, v. 21, n. 4, p. 435–443, 1983.

BEULTER, A. N.; CENTURION, J. F. Compactação do solo no desenvolvimento radicular e na produtividade da soja. *Pesquisa Agropecuária Brasileira*, v. 39, n. 6, p. 581–588, junho 2004.

BEUTLER, A. N. *et al.* Efeito da compactação na produtividade de cultivares de soja em latossolo vermelho. *Revista Brasileira de Ciencia do Solo*, v. 30, p. 787–794, 2006.

BOUMA, J.; HOOSBECK, M. R. The contribution and importance of soil scientists in interdisciplinary studies dealing with land. In: MADISON (Ed.). *The Role of Soil Science in Interdisciplinary Research*. [S.l.]: Soil Scientific Society American Spec., 1996. cap. 1, p. 1–15.

BRAZ, D.; LOPES, R.; MOTTA, L. Dual-energy computerized tomography in compacted soil. *Geotechnical and Geological Engineering*, Kluwer Academic Publishers, Netherlands, v. 18, p. 221–238, 2000.

BUENO, J. M. *Reconstrução e Visualização Tridimensional de Imagens Tomográficas Baseada no uso de Transformada Rápida de Fourier*. 130 p. Dissertação (Mestrado) — Universidade de São Paulo, São Carlos, 1995.

CAÇÃO, G. R. *Desenvolvimento de um algoritmo para a reconstrução tridimensional para imagens de um minitomógrafo baseado no método de reconstrução algébrica modificado e interpolação*. 122 p. Dissertação (Mestrado) — Universidade Federal de São Carlos, São Carlos, 1994.

COMPTON, A. H.; HAGENOW, C. A measurement of the polarization of secondary x-rays. *Journal of the Optical Society of America (1917-1983)*, v. 8, p. 487–+, jul 1924.

CORMACK, A. M. Representation of a foundation by its line with some radiological application. *App. Phys*, v. 34, n. 9, p. 2722–2727, 1963.

CRESTANA, S. *A Tomografia Computadorizada com um novo método para estudos da física da água no solo*. 140 p. Tese (Doutorado) — Instituto de Física de São Carlos - Universidade de São Paulo, São Carlos, 1985.

- CRESTANA, S. *et al.* Tomografia reconstrutiva. In: *Instrumentação agropecuária: contribuições no limiar do novo século*. Brasília: Crestana, S.; Cruvinel, P. E.; Mascarenhas, S.; Biscegli, C. I.; Martin-Neto, L.; Colnago, L. A., 1996. cap. 4.
- CRUVINEL, P. E. *Minitomógrafo de Raios X e Raios γ computadorizado para aplicações multidisciplinares*. 325 p. Tese (Doutorado) — Universidade de Campinas, Campinas, 1987.
- CRUVINEL, P. E.; BALOGUN, F. A. Compton scattering minitomography scanner for soil science. In: EMBRAPA AGRICULTURAL INSTRUMENTATION. *Advances in Agricultural Tomography*. São Carlos, 2000. p. 24–30.
- CRUVINEL, P. E.; BALOGUN, F. A. Compton scattering tomography for agricultural measurements. *Journal of the Brazilian Association of Agricultural Engineering*, v. 26, n. 1, p. 151–160, Jan-Abril 2006. Disponível em: <http://www.sbea.org.br/rea/v26_n1/index.html>.
- CRUVINEL, P. E. *et al.* X-and γ -rays computerized minitomograph scanner for soil science. *IEEE - Transactions on Instrumentation and Measurement*, v. 39, n. 5, p. 745–750, 1990. IEEE.
- CRUVINEL, P. E. *et al.* Beetle damage detection in forests by ct image processing. *Revista Árvore*, v. 27, n. 5, p. 747–752, 2003. In-press.
- DERENIAK, D. G. C. E. L. *Optical Radiation Detectors*. [S.l.]: Wiley, 1984. 320 p.
- ERENO, D. Texturas e sabores - da língua eletrônica ao analisador de pó de café, novos equipamentos são licenciados pela embrapa. *Revista Pesquisa Fapesp*, n. 119, p. 62–67, Janeiro 2006.
- FERREIRA, E. *et al.* Avaliação de Diferentes Tubos de Acesso para Medição da Umidade do Solo Através do Uso de Sonda de Nêutrons. Seropédica - RJ, Novembro 1998.
- FFTW. *FFTW Home Page*. 2007. Disponível em: <<http://www.fftw.org/>>. Acesso em: 20 de janeiro de 2007.
- FOSTER, I. *Designing and Building Parallel Programs*. 2005. Disponível em: <<http://www-unix.mcs.anl.gov/dbpp>>. Acesso em: 10 de setembro de 2005.
- GEER, D. Chip makers turn to multicore processors. *IEEE Computer Society*, v. 38, n. 5, p. 11–13, May 2005.
- GENUTIS, M.; KAZANAVICIUS, E.; OLSEN, O. Caches in dsp processors. *Ultragarsas journal*, v. 39, n. 2, p. 1–4, 2001.
- GONZALEZ, R. C.; WOODS, R. E. *Processamento de Imagens Digitais*. 3. ed. [S.l.]: Editora Edgard Blücher, 2000. 509 p.
- GORDON, R.; HERMAN, G. Three dimensional reconstruction from projections: a review of algorithms. In: APPLICATION OF OPTICAL INSTRUMENTATION IN MEDICINE III. *Society of Photo-Optical Instrumentation Engineers*. [S.l.], 1975. v. 47, p. 2–14.
- GRANATO, L. F. *Algoritmo Adaptativo para a Melhora em Imagens Tomográficas Obtidas em Múltiplas Energias*. 135 p. Dissertação (Mestrado) — Universidade Federal de São Carlos, São Carlos, 1998.

- GRAPS, A. An introduction to wavelets. *IEEE Computacional Science and Engineering*, v. 2, n. 2, p. 50–61, June 1995.
- HAYES, M. H. *Statistical Digital Signal Processing and Modeling*. New York, USA: John Wiley & Sons, Inc., 1996. 68-71 p.
- HOLLIS, T. *Hunt Engineering HEPC2E TIM Motherboard for AT bus - User Manual*. [S.l.], 1999.
- HOMEM, M. *et al.* Biological image restoration in optical-sectioning microscopy using prototype image constraints. *Real-Time Imaging*, v. 8, p. 475–490, 2002.
- HOUNSFIELD, G. N. Computerized transverse axial scanning (tomography) i: description of systems. *Brit J Radio*, v. 46, p. 1016–1022, 1973.
- HWANG, K. *Advanced Architecture and Parallel Processing*. [S.l.]: McGraw Hill, 1984.
- HWANG, K. *Advanced Computer Architecture - Parallelism, Scalability, Programmability*. [S.l.]: McGraw Hill, 1993. 771 p.
- KAK, A. C.; SLANEY, M. *Principles of Computerized Tomographic Imaging*. New York: IEEE Press, 1999.
- KALMAN, R. E. A new approach to linear filtering and prediction problems. *Transactions of the ASME - Journal of Basic Engineering*, 82, n. 82 (Series D), p. 35–42, 1960.
- KITWARE. *VTK 5.1.0 Documentation*. 2006. Disponível em: <<http://www.vtk.org/doc/nightly/html/>>. Acesso em: 10 de junho de 2006.
- KNIGHT, W. Two heads are better than one. *IEEE Review*, v. 51, n. 5, p. 33–35, September 2005. Disponível em: <www.ieee.org/review>.
- LI, X. *et al.* A noise reduction method for non-stationary noise model of spect sinogram based on kalman filter. *Nuclear Science Symposium Conference Record*, v. 4, p. 2134–2138, 2001.
- MACEDO, Á. *Construção e uso de um tomógrafo com resolução micrométrica para aplicações em ciências do solo e do ambiente*. Tese (Doutorado) — Escola de Engenharia de São Carlos, São Carlos, 1997.
- EMBRAPA Centro Nacional de Pesquisa e Desenvolvimento de Instrumentação Agropecuária. Álvaro Macedo, Carlos Manoel Pedro Vaz, João de Mendonça Naime, Paulo Estevão Cruvinel e Sílvia Crestana. *TOMÓGRAFO COMPUTADORIZADO DE RESOLUÇÃO MICROMÉTRICA PARA USO EM AGROPECUÁRIA*. 1997. MU7703174-1, 1997.
- MASCARENHAS, N. D. A.; SANTOS, C. A. N.; CRUVINEL, P. E. Transmission tomography under poisson noise using the anscombe transformation and wiener filtering of the projections. *Nuclear Instruments and Methods in Physics Research*, Amsterdam, v. 432, p. 265–271, 1999.
- MEIJERING, E. A chronology of interpolation: from ancient astronomy to modern signal and image processing,. *Proceedings of the IEEE*, v. 90, n. 3, p. 319–342, Março 2002.

- MINATEL, E. R. *Desenvolvimento de Algoritmo para Reconstrução e Visualização Tridimensional de Imagens Tomográficas com uso de Técnicas Freqüenciais e Wavelets*. 162 p. Dissertação (Mestrado) — Universidade Federal de São Carlos, Departamento de Computação, São Carlos - SP, 1997.
- MINATEL, E. R. *Modelo computacional baseado em técnicas Wavelets para relacionar imagens digitais obtidas em diferentes escalas e resoluções*. Tese (Doutorado) — Instituto de Física de São Carlos - Universidade de São Paulo, São Carlos, 2003.
- MINATEL, E. R.; CRUVINEL, P. E. Reconstrução e visualização de dados de tomografia aplicada ao estudo de solos usando *B-Wavelets*. In: EMBRAPA- INSTRUMENTAÇÃO AGROPECUÁRIA. *II SIAGRO - Simpósio Nacional de Instrumentação Agropecuária*. [S.l.], 1998.
- MPI. *The Message Passing Interface (MPI) standard*. 2004. Disponível em: <<http://www-unix.mcs.anl.gov/mpi/>>. Acesso em: 23 de maio de 2004.
- NAIME, J. M. *Projeto e construção de um minitomógrafo portátil para estudo de ciência de solo e plantas em campo*. Dissertação (Mestrado) — Escola de Engenharia de São Carlos - Universidade de São Paulo, São Carlos, 1994.
- NAIME, J. M. *Um novo método para estudos dinâmicos, in situ, da infiltração da água na região não-saturada do solo*. Tese (Doutorado) — Escola de Engenharia de São Carlos, Universidade de São Paulo, São Carlos, SP, Brasil, 2001.
- EMBRAPA. Centro Nacional de Pesquisa e Desenvolvimento de Instrumentação Agropecuária. João Mendonça Naime, Paulo Estevão Cruvinel, Silvio Crestana, Valentim Monzane e André Torre Neto. *TOMÓGRAFO COMPUTADORIZADO PORTÁTIL PARA ESTUDO DE SOLOS E PLANTAS, EM CAMPO*. 1996. MU7602400-8, 1996.
- NAVAUX, P. O. A. Introdução ao processamento paralelo. *Revista Brasileira de Computação*, v. 5, n. 2, p. 31–43, Out/Dez 1989.
- PARAVIEW. *Paraview - Parallel Visualization Application*. 2006. Disponível em: <www.paraview.org>. Acesso em: 20/02/2006.
- PEDROTTI, A. *et al.* Tomografia computadorizada aplicada a estudos de um planossolo. *Pesquisa Agropecuária Brasileira*, v. 38, n. 7, p. 819–826, jul 2003.
- PEREIRA, M. F. L. *Algoritmo Paralelo para Reconstrução de Imagens Tomográficas de Amostras Agrícolas em Arquitetura DSP com Técnicas Wavelets*. 178 p. Dissertação (Mestrado) — Universidade Federal de São Carlos, São Carlos, 2001.
- PEREIRA, M. F. L.; CRUVINEL, P. E. Ferramenta para reconstrução de imagens tomográficas das ciências agrícolas. In: *V Congresso Brasileiro de Agroinformática, II Simpósio Brasileiro de Tecnologia da Informação no Agronegócio Cooperativo*. Londrina, PR: [s.n.], 2005. p. 1–8.
- PEREIRA, M. F. L.; CRUVINEL, P. E.; COSTA, L. F. Um método para paralelização de algoritmos de reconstrução tomográfica, aplicado em amostras agrícolas, com uso de processadores dsp. In: *Anais do IX Encontro de Modelagem Computacional*. Belo Horizonte, MG: [s.n.], 2006. p. 1–10.

- PEREIRA, M. F. L.; KOENIGKAN, L. V.; CRUVINEL, P. E. Parallel dsp architecture for reconstruction of tomographic images using wavelets techniques. In: *Proceedings of XIV Brazilian Symposium on Computer Graphics and Image Processing (SIBGRAPI)*. Florianópolis: [s.n.], 2001. p. 384.
- PETROVIC, A. M.; SIEBERT, J.; RIEKE, P. E. Soil bulk analysis in three-dimensions by computed tomographic scanning. *Soil Science Soc. Am.*, v. 46, p. 445–450, 1982.
- PRATT, W. K. *Digital Image Processing*. [S.l.]: John Willey & Sons, 1991.
- RADON, J. In the determination of functions from their integral along certain manifold. *Berichte über die Verhandlungen*, v. 69, p. 262–277, 1917.
- SCHALLER, R. R. Moore's law: past, present and future. *IEEE SPECTRUM*, v. 34, n. 6, p. 52–59, June 1997.
- SCHRODER, I. J.; AVILA, L. S. *VTK File Formats for VTK Version 4.2*. 2005. Disponível em: <<http://www.vtk.org/pdf/file-formats.pdf>>. Acesso em: 27 de junho de 2005.
- SHIPITALO, M. J.; DICK, W. A.; EDWARDS, W. M. Conservation tillage and macropore factors that effect water movement and the fate of chemicals. *Soil & Tillage Research*, Amsterdam, v. 53, p. 167–183, 2000.
- SILVEIRA, A. M. *A relação entre preços de açúcar nos mercados domésticos e internacional*. 74 p. Dissertação (Mestrado) — Universidade de São Paulo - ESALQ, Piracicaba, Abril 2004.
- SSSA. *Glossary of soil science terms*. 1997.
- STARK, H. *et al.* Direct fourier reconstruction in computer tomography. *IEEE Trans. Acoust. Speech Signal Processing*, ASSP-29, p. 237–244, 1981.
- STAROBA, J. *Parallel Performance Modeling, Prediction and Tuning*. Tese (Doutorado) — Faculty of Information Technology - Brno University of Technology, 2004.
- TANENBAUM, A. S. *Organização Estrutura de Computadores*. [S.l.]: Prentice/Hall do Brasil, 1992. 454 p.
- TEIXEIRA, C. F. A.; MORAES, S. O.; SIMONETE, M. A. Desempenho do tensiômetro, tdr e sonda de nêutrons na determinação da umidade e condutividade hidráulica do solo. *Revista Brasileira de Ciencia do Solo*, v. 29, p. 161–168, 2005.
- UNSER, M. Ten good reason for using spline wavelets. In: *Proc. SPIE - Wavelets in Signal and Image Processing*. [S.l.: s.n.], 1997. v. 3169, p. 422–431.
- VANNI, S. M. *Modelos de Regressão: Estatística Aplicada*. São Paulo: Legmar Informática & Editora Ltda, 1998. 25-28 p.
- VEIGA, A. C. P. *Caracterização de sinais utilizando-se transformadas em algoritmos adaptativos baseados no LMS*. 244 p. Tese (Doutorado) — Universidade de Campinas, Campinas, Julho 2002.
- VELHO, L.; PERLIN, K. B-spline wavelet paint. *Revista de Informática Teórica e Aplicada (RITA)*, IX, n. 2, p. 100–119, Outubro 2002.

VTK. *Visualization ToolKit*. 2006. Disponível em: <<http://www.vtk.org/>>. Acesso em: 3 de fevereiro de 2006.

WELCK, G.; BISHOP, G. An introduction to the kalman filter. In: *SIGGRAPH 2001, Course 8*. Department of Computer Science, University of North Carolina at Chapel Hill, Chapel Hill, NC 27599-3175: ACM Inc, 2001. p. 47.

WIENER, N. *Extrapolation, Interpolation and Smoothing of Stationary Time Series, with Engineering Applications*. New York: Technology Press and Wiley, 1949.