

UNIVERSIDADE CATÓLICA DE PELOTAS
CENTRO POLITÉCNICO
PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA

***SisA³* : Sistema Automatizado de
Auditoria de Armazéns de Granéis**

por
Wagner Guimarães Al-Alam

Dissertação apresentada como
requisito parcial para a obtenção do grau de
Mestre em Ciência da Computação

Orientadora: Profa. Dra. Renata Hax Sander Reiser
Co-orientador: Prof. Dr. Maurício Lima Pilla
Colaborador: Prof. Dr. Adenauer Correa Yamin

DM-2010/1-008

Pelotas, fevereiro de 2010

AGRADECIMENTOS

Primeiramente gostaria de agradecer a todos amigos, colegas e professores com os quais convivi durante o Mestrado.

Em especial aos professores Renata Reiser e Maurício Pilla, pelo apoio, orientação e amizade, indispensáveis para a conclusão deste trabalho.

Aos familiares, em especial a minha mãe Lana, irmãos Fabiane e Leonardo que incentivaram o trabalho, motivaram o desenvolvimento e contribuíram com a conclusão deste projeto. Ao meu sobrinho Mateus, que contribuiu ativamente no projeto, através da participação na modelagem e execução dos protótipos.

Ao professor Adenauer Yamin que auxiliou, juntamente com os orientadores no desenvolvimento deste trabalho, desde o início.

À empresa Josapar por conceder o acesso aos armazéns, e auxílio nos questionamentos feitos.

À Tâmara Nascimento, Professor Maurício Tavares, Sidinei da empresa Con-tronic, e ao Afrânio Schulz que auxiliaram no desenvolvimento e montagem do protótipo *DigSisA*³.

Também aos colegas que estiveram presentes no momentos de necessidade e acompanhando a caminhada, muitas vezes fazendo grupos de estudos, em outras discutido os trabalhos, discutindo assuntos diversos ou sugerindo opções para ajudar na solução dos problemas. Em especial os colegas Luthiano, Nelsi, Christian e Marcos que sempre estiveram disponíveis para discutir assuntos pertinentes ao mestrado, e tantos outros não pertinentes.

À turma dos churrascos e jantas de confraternização, promovidos durante o curso, para a integração do grupo.

Sem o auxílio destes citados acima, e de alguém que ocasionalmente eu possa ter esquecido, não seria possível a conclusão deste trabalho.

SUMÁRIO

LISTA DE FIGURAS	6
LISTA DE ABREVIATURAS E SIGLAS	8
RESUMO	10
ABSTRACT	11
1 INTRODUÇÃO	12
1.1 Tema e Justificativa	13
1.2 Motivação	14
1.3 Proposta de Pesquisa	17
1.4 Objetivos	17
1.5 Contextualização da Proposta	18
1.6 Organização do Texto	21
2 TÉCNICAS COMPUTACIONAIS PARA CÁLCULO DO VOLUME REFERENTE À SUPERFÍCIES NÃO UNIFORMES	22
2.1 Técnicas para Cálculo de Volume Baseadas em Métodos Numéricos	22
2.1.1 Princípios do Cálculo Numérico	22
2.1.2 Etapas na Resolução Numérica de um Problema	23
2.2 Técnicas para Cálculo de Volume Baseadas no Processamento de Imagens	24
2.3 Técnicas Iterativas para Cálculo do Volume	24
2.4 Técnicas de Visualização de Dados Volumétricos	26
2.5 Método de Interpolação Numérica	27
2.5.1 Splines	28
2.5.2 Conceitualização	29
2.6 Conclusão	31
3 PROJETO D-GM: AMBIENTE DE SUPORTE AO DESENVOLVI- MENTO E EXECUÇÃO DO MÉTODO P-ICONE	32
3.1 Noção Intuitiva do Modelo GM	32
3.2 Projeto D-GM: Principais Componentes	33
3.3 VPE-GM	35
3.3.1 Editor Gráfico de Processos	36
3.3.2 Editor Gráfico de Memória	37
3.4 VirD-GM	38

3.5	Desenvolvimento e Execução de Aplicações no Ambiente D-GM	40
3.5.1	Parâmetros de Execução	41
3.6	Principais Construtores	42
3.6.1	Construtor Iterativo	42
3.6.2	Construtor Iterativo Paralelo	43
3.6.3	Construtor Produto Sequencial	43
3.7	Exploração do Paralelismo no D-GM com Processadores Multi-Core	44
3.8	Conclusão	45
4	ARQUITETURA SIS^{A3}	46
4.1	Sistema Dig-Sis^{A3}	46
4.2	Suporte ao Desenvolvimento da Aplicação p-iCone	49
4.3	Métodos iCone e p-iCone	49
4.4	Conclusão	50
5	ICONE-SIS^{A3} - CALCULO DE VOLUME DE RELEVOS NÃO-UNIFORMES	51
5.1	Protótipo iCone	51
5.1.1	Processamento da Entrada	51
5.1.2	Segmentação da Matriz	52
5.1.3	Divisão da Matriz de Pontos	52
5.1.4	Geração do Resultado Final	56
5.2	Estratégia Paralela e Distribuída	56
5.3	Conclusão	58
6	DIGSIS^{A3}: EQUIPAMENTO DIGITALIZADOR	59
6.1	Discretização do Espaço de Busca	60
6.1.1	Estrutura Tridimensional Formada pelo Relevo	60
6.2	Organização do DigSis^{A3}	63
6.2.1	Componentes Eletromecânicos	64
6.2.2	Componentes Eletrônicos	65
6.2.3	Componente de <i>software</i>	65
6.3	Descrição do Funcionamento	67
6.3.1	Parametrização	67
6.3.2	Leitura	69
6.3.3	Obtenção e Estrutura dos Dados	69
6.3.4	Saída de Dados	70
6.4	Conclusão	70
7	RESULTADOS OBTIDOS	71
7.1	Estudo de Caso para Análise de Erro no Protótipo iCone	71
7.2	Estudo de Caso para Análise de Desempenho no Protótipo p-iCone	73
7.2.1	Estratégias de Execução	73
7.3	Conclusão	77

8	CONSIDERAÇÕES FINAIS	78
8.1	Conclusões	78
8.2	Continuidade do Projeto	81
8.3	Publicações Realizadas	82
8.4	Contribuição em Projetos de Pesquisa	84
	REFERÊNCIAS	85

LISTA DE FIGURAS

Figura 1.1	Foto de um armazém	15
Figura 2.1	Visualização da formula de Smalian (FORESTS; RANGE, 2007)	25
Figura 2.2	Representação de uma manga através de elipses	25
Figura 2.3	Curva definida parametricamente	29
Figura 3.1	Organização Funcional do Projeto D-GM (FONSECA, 2008a)	34
Figura 3.2	Componentes da arquitetura VirD-GM (FONSECA, 2008a).	35
Figura 3.3	Editor Gráfico de Processos (FONSECA, 2008a).	36
Figura 3.4	Editor Gráfico de Memória (FONSECA, 2008a).	38
Figura 3.5	Diagrama de sequencia do projeto D-GM.	39
Figura 3.6	Organização do ambiente D-GM.	40
Figura 3.7	Parâmetros para execução do D-GM.	41
Figura 3.8	Construtor iterativo sequencial.	42
Figura 3.9	Construtor iterativo paralelo.	43
Figura 4.1	Organização da Arquitetura do <i>SisA</i> ³	46
Figura 4.2	Diagrama de Sequência do <i>DigSisA</i> ³	48
Figura 4.3	Diagrama de Sequência de medição do <i>DigSisA</i> ³	48
Figura 4.4	Diagrama de Sequência do Protótipo <i>iCone</i>	50
Figura 5.1	Degraus formados pela aproximação da altura	52
Figura 5.2	Método de cálculo dos polinômios de Interpolação.	53
Figura 5.3	Método de avaliação dos pontos.	53
Figura 5.4	Processo tratamento dos pontos	54
Figura 5.5	Método de cálculo das novas camadas.	54
Figura 5.6	Corte feito na matriz de pontos	55
Figura 5.7	Método de cálculo da área de uma face de fatia.	55
Figura 5.8	Area da face de uma fatia representada por 5 pontos	56
Figura 5.9	Area da face de uma fatia representada por 9 pontos	56
Figura 5.10	Modelagem do protótipo utilizando o VPE-GM	57
Figura 6.1	Visão do protótipo básico.	59
Figura 6.2	Imagem de um armazém.	61
Figura 6.3	Ampulheta	62
Figura 6.4	Orientação dos eixos na modelagem do armazém	62
Figura 6.5	Organização do protótipo <i>DigSisA</i> ³	64
Figura 6.6	Diagrama de Blocos do <i>DigSisA</i> ³	65

Figura 6.7	Imagem dos passos de medição.	66
Figura 6.8	Diagrama de atividades do protótipo DigSisA ³ (ALMEIDA; REISER, 2009).	68
Figura 6.9	Parametrização dos dados na calibração do equipamento scanner. . .	69
Figura 6.10	Matriz de pontos em XML.	70
Figura 7.1	Cone regular C(h,r).	71
Figura 7.2	Gráfico de Desempenho Distribuído	74
Figura 7.3	Gráfico de Eficiência em Arquitetura Distribuída	75
Figura 7.4	Gráfico de Desempenho em Ambiente Distribuído <i>Multi-Core</i>	75
Figura 7.5	Gráfico de Eficiência em Arquitetura Distribuída e Multiprocessada. .	76
Figura 7.6	Comparativo de Execução com 16 e 13 núcleos.	76

LISTA DE ABREVIATURAS E SIGLAS

BT	<i>Bluetooth</i>
CPU	<i>Central Processing Unit</i>
D-GM	<i>Distributed Geometric Machine</i>
DSP	<i>Digital Signal Processor</i>
EXEHDA	<i>Execution Environment for High Distributed Applications</i>
FPGA	<i>Field Programmable Gate Array</i>
G3PD	Grupo de Pesquisa em Processamento Paralelo e Distribuído
GMFC	Grupo de Matemática e Fundamentos da Computação
GPDIB	Grupo de Pesquisa e Desenvolvimento em Instrumentação Biomédica
GPRS	<i>General Packet Radio Service</i>
HSI	<i>Hue, Saturation, Intensity</i>
IP	<i>Internet Protocol</i>
JVM	<i>Java Virtual Machine</i>
MIMD	<i>Multiple Instruction Multiple Data</i>
MPI	<i>Message Passing Interface</i>
OpenMP	<i>Open Multi-Processing</i>
PDA	<i>Personal Digital Assistant</i>
PVM	<i>Parallel Virtual Machine</i>
RGB	<i>Red Green Blue</i>
SMP	<i>Symmetric Multiprocessing</i>
SIMD	<i>Single Instruction Multiple Data</i>
SisA ³	Sistema Automatizado de Auditoria em Armazéns
UCP	Unidade Central de Processamento
UCPEL	Universidade Católica de Pelotas
UFPEL	Universidade Federal de Pelotas

VirD-GM *Virtual Distributed Geometric Machine*

VPE-GM *Visual Programming Environment for the Geometric Machine Model*

XML *eXtensible Markup Language*

RESUMO

Empresas que trabalham com produtos a granel possuem locais para estocagem, durante o desenvolvimento do processo produtivo e no armazenamento do produto final, denominados armazéns ou silos. Os valores dos estoques devem ser validados periodicamente através da comparação dos estoques fiscal (controle das notas fiscais) e físico (levantamento do volume estocado na empresa). Neste contexto, o cálculo do estoque físico, ou seja, o volume de granéis presentes nos armazéns, é geralmente efetuado de forma manual e com baixa credibilidade, desta forma com propensão a erros. Os atuais processos de auditoria no conteúdo de silos, além de envolverem estimativas inexatas, estão frequentemente baseados no esvaziamento do silo.

Considerando o uso de tecnologias que viabilizam a medição eletrônica de distâncias, ângulos, e controles automáticos sobre atuadores que possibilitam movimentos mecânicos sobre estruturas de suporte, buscou-se o desenvolvimento de um sistema capaz de prover tanto soluções computacionais, quanto tecnológicas para o problema de cálculo do volume de relevos irregulares, no caso dos produtos estocados nos armazéns.

O Sistema Automatizado de Auditoria em Armazéns (*SisA*³) pretende tornar este processo automático, rápido e preciso, sem a necessidade de esvaziamento ou contato com os produtos. Para alcançar este objetivo, tem-se um sistema integrado composto de: (i) um equipamento digitalizador, consolidando o protótipo híbrido de *hardware* e *software* denominado *Dig-SisA*³, para a medição de pontos do relevo não-uniforme, formado pelos produtos estocados; e (ii) método para o cálculo do volume (*iCone*), que combina técnicas de visualização científica, interpolação numérica de pontos e cálculo iterativo de volume.

Além disto, introduz-se a paralelização do protótipo *iCone*, para diminuir o tempo da obtenção dos resultados do método *iCone* no processo de auditoria. A análise sobre as perspectivas em arquiteturas multiprocessadas, multi-core e paralela distribuída, utiliza o ambiente D-GM (*Distributed Geometric Machine*), a qual provê os formalismos para garantir criação, gerenciamento e processamento de aplicações paralelas e/ou distribuídas da computação científica, com ênfase na exploração do paralelismo de dados e nas etapas de sincronizações.

O protótipo de *software iCone* apresenta-se funcionalmente validado, incluindo análise de erro na execução do método. As análises de desempenho no protótipo *p-iCone* apresentaram resultados satisfatórios. O desenvolvimento deste trabalho consolida o sistema *SisA*³, viabilizando aferição automática e confiável de estoques, incluindo ampla aplicação no mercado.

Palavras-chave: Interpolação Numérica, Computação Científica, Spline Cúbica, Métodos Numéricos, Equipamento Digitalizador, Processamento Paralelo e Distribuído.

TITLE: “*SISA*³ : AN AUTOMATED AUDIT SYSTEM FOR GRAIN STORES”

ABSTRACT

Companies working with bulk materials have appropriate locations for storage during the development of the production and storage of the final product, known as warehouses or storehouses. The values of stocks need to be periodically validated by comparing the control of receipts the and the physical situation (removal of the volume stored in the company). In this context, the calculation of physical inventory as the volume of bulk present in the warehouses is usually done manually with low credibility and prone to errors. The current audit procedures on the contents of warehouses involve inaccurate estimates, and often require emptying the warehouse.

Considering the use of technologies which enable the electronic measurement of distances, angles, and automatic controls on actuators enabling mechanical movements on the supporting structures, we sought to develop a system capable of providing both computing solutions, and technology for the problem of calculation of irregular relief (products stocked in warehouses).

The Automated Auditing Warehouse *SisA*³ intends to make this process automatic, fast and precise, without the need for emptying warehouses or having contact the products. To achieve this goal, we developed an integrated system composed of: (i) a scanner equipment, consoling the hybrid prototype of hardware and software called *DigSisA*³, in order to the measurement of points of relief non-uniform, formed by the products in stock, and (ii) a method for calculating the volume *iCone*, which combines techniques of scientific visualization, numerical interpolation points and iterative calculation of volume.

The parallelization of the prototype *iCone* was also developed in order to satisfy the test of agility and performance of the method *iCone* in the audit process. The development for multiprocessor, *multi-core*, and distributed architectures was done over the DGM (*Geometric Distributed Machine*), which provides the formalities to ensure creation, management and application processing parallel and / or distributed scientific computing, with emphasis on the exploitation of data parallelism and synchronization steps.

The prototype of software *iCone* was functionally validated, including analysis of error in the method. The analysis of performance in the prototype *p-iCone* showed satisfactory results. The development of this work strengthens the system *SisA*³, enabling automatic and reliable measurement of inventories, including broad market application.

Keywords: Numerical Interpolation, Cubic Spline Interpolation, Scientific Computation, Volume Measurement, Scanner equipment, Distributed and Parallel Computing.

1 INTRODUÇÃO

O desenvolvimento dos computadores tem contribuído de forma significativa para a inovação tecnológica, introduzindo novos métodos e processos, incentivando o crescimento da produtividade de bens e geração de novos serviços, viabilizando a otimização de recursos, o acesso à informação e a melhoria da qualidade de vida.

A evolução dos componentes dos computadores, minimizando o tamanho e o custo, diversificando as funções e as interfaces e, simultaneamente, provendo suporte à demanda de maior poder computacional, resultam na expansão e aplicação da informática em novas e múltiplas tarefas, as quais integradas a sistemas eletromecânicos, também contribuem com a área de automação.

A disponibilidade das arquiteturas multiprocessadas e o atual contexto de utilização de recursos para aplicações concorrentes também incentivam o desenvolvimento de técnicas e novas linguagens para o paradigma da computação paralela e distribuída, visando melhoria de desempenho no processamento da informação.

A fim de agilizar a execução de programas, frequentemente são estudadas e aplicadas técnicas de exploração de paralelismo, caracterizando-se pela execução simultânea de trechos de programas em diferentes processadores, diminuindo o tempo total de execução dos programas paralelizáveis. Para que um programa possa ser caracterizado como paralelizável, as restrições mais fortes são as dependência de dados, ou seja, é necessário que haja dados que possam ser processados de forma independente.

Na Computação Científica, tem-se consolidado o estudo de modelos matemáticos integrados às técnicas de soluções numéricas para resolução de problemas reais, permitindo a modelagem do mundo real em n dimensões. Desta forma, permite-se o desenvolvimento de algoritmos para resolver os mais diversos problemas da indústria e da sociedade.

Neste sentido, o desenvolvimento de aplicações baseadas em modelos matemáticos integradas a ambientes computacionais, com suporte a exploração do paralelismo, tem contribuído para a inovação tecnológica em processos produtivos, caracterizado pela implantação de produtos com características de desempenho aprimoradas, fornecendo objetivamente ao consumidor novos serviços e ferramentas computacionais. Assim, os resultados obtidos podem levar a uma integração entre possíveis mudanças de equipamento, capacitação profissional e de recursos humanos, incluindo como é o caso deste trabalho, aplicação de métodos científicos visando melhoria do processo de auditoria na armazenagem da produção agrícola.

Uma metodologia para coleta de dados e geração de avaliação automática do volume de produtos estocados em armazéns de grãos pode contribuir para introduzir

mudanças quantitativas na produção e na distribuição da produção agrícola. Caracterizado como uma inovação de processo para a indústria de secagem e armazenamento de grãos, este trabalho introduz o *Sis.A³*, Sistema Automatizado de Auditoria em Armazéns, e busca contribuir com a gerência de estoques no ambiente industrial.

1.1 Tema e Justificativa

Esta dissertação tem como tema central a inovação de processo tecnológico a partir da introdução e implantação de métodos computacionais no processo de avaliação automática de estoques em armazéns de grãos, de maneira precisa e ágil, bem como o processo de digitalização do relevo formado pelos produtos estocados.

Dentre as diversas áreas produtivas no Brasil, destaca-se o agronegócio, o qual, segundo o Ministério da Agricultura, é responsável por 33% do Produto Interno Bruto (PIB) e 42% das exportações totais (ZAFALON; MASCHIO; ALBANO, 2004). Embora estes dados sejam referentes ao ano de 2004, tem-se evidenciado, desde lá, o grande desempenho econômico do agronegócio e seu correspondente crescimento comercial.

Neste ramo de negócio, os produtos são armazenados antes de serem distribuídos. Este armazenamento ocorre geralmente a granel, ou seja, sem o uso de embalagens. Há o controle de entrada e de saída de produtos, tanto pelo peso da matéria prima que entra, quanto pelo controle fiscal. Periodicamente se faz necessário um processo de auditoria, denominado inventário de produtos, para verificar se houve alguma divergência entre os controles fiscal e físico.

As indústrias que trabalham com grãos no ambiente de produção também necessitam deste procedimento de avaliação e enfrentam problemas análogos, tanto no processo de armazenagem quanto na quantidade de produtos circulando no processo industrial, pois armazéns intermediários são utilizados entre as diversas etapas do processo.

Os armazéns de grãos podem pertencer a uma área de estocagem estática, ou ainda, podem estar em fluxo dinâmico, onde o produto está sendo depositado na parte superior e retirado, simultaneamente, em um ponto na parte inferior do armazém.

Nos processos de inventário de armazéns, o fluxo de produtos necessita ser interrompido, o que torna o custo financeiro diretamente proporcional ao tempo de execução do processo, dado que o restante da planta industrial deixa de produzir. Na produção agrícola, a matéria prima (grãos, sementes, etc.) é colhida em safras que dependem de épocas definidas por características regionais. Além de serem colhidos em intervalos pré-definidos, estes produtos se deterioram rapidamente quando não tratados, o que torna essencial o funcionamento contínuo das indústrias nestes períodos.

Um processo de auditoria automatizado, além de proporcionar confiabilidade a este procedimento, reduz os custos decorrentes da parada. Com a execução automática, reduzem-se as chances de se efetuar retrabalho, situação que ocorre no processo manual. Assim, a automação deste processo torna-se uma estratégia atrativa para as empresas, no sentido de redução nos custos de parada e melhoria nos controles de estoque. A proposta de construção do *Sis.A³* está baseada em custos competitivos, quando comparadas com outras soluções no mercado.

O sistema *Sis.A³* contempla o cálculo automatizado do volume presente nos armazéns de grãos sem a necessidade de contato manual com o produto.

A literatura apresenta outras soluções baseadas em imagens, porém estas estão vulneráveis a interferência de poeira e luminosidade, sendo estes fatores tratados como

ruídos na correspondente modelagem (LIN; FU, 2007; LIN; JIANG, 2007).

Portanto, considerando a necessidade de estudos para melhoria no processo de avaliação da quantidade estocada (volume) de produtos nos diversos tipos de armazéns de graneis, de forma automática, precisa e ágil, em cada armazém, decorre o tema e justificativa deste trabalho.

1.2 Motivação

Esta seção descreve os principais focos motivacionais para o desenvolvimento deste trabalho.

1.2.1 Inovação do processo de auditoria do armazenamento de graneis

A idéia para o desenvolvimento deste trabalho surgiu a partir dos contatos desenvolvidos com empresas de comércio e indústrias, responsáveis pelo transporte e/ou armazenamento da produção agrícola, na região sul de nosso estado.

Verificou-se que, independente dos diferentes tipos de armazéns de grande porte, usualmente utilizados por estas empresas, existe uma grande dificuldade nos processos de auditorias quando da avaliação da quantidade de produtos estocados.

Estas auditorias, frequentemente, constam como exigência de contratos comerciais que também incluem as correspondentes avaliações para atualizações periódicas nos dados da empresa.

1.2.2 O uso de técnicas computacionais na solução de problemas reais

O desenvolvimento de aplicações que integram técnicas para representação e simulação do mundo real também motivaram este trabalho. Técnicas provenientes de áreas como a visualização científica, computação científica, automação, processamento paralelo, entre outras, tem-se apresentado como facilitadores na solução de problemas do mundo real.

- Técnicas de visualização para modelagem e digitalização de superfícies não uniformes

A representação geométrica do relevo que representa o amontoado de produtos armazenado pode ser visualizada como uma superfície gerada por uma sequência de cones, com sobreposição de bases. Entretanto, geralmente ocorrem deformações nesta representação geométrica, pois com a remoção parcial da quantidade de produto estocada, e considerando características de umidade, deterioração e impurezas referente ao produto, formam-se imperfeições que caracterizam a não uniformidade (irregularidade) da superfície.

Salienta-se, que a forma dos silos ou armazéns pode ser diferenciada e, por consequência, o relevo também pode ter como representação geométrica uma irregularidade decorrente destas diferenciações. Um dos tipos de armazéns largamente utilizado é o armazém horizontal, com o telhado no formato de semi-cilindro ou prismas regulares. Uma idéia geral do ambiente interno de um armazém pode ser obtida na Figura 1.1.

Mesmo nos armazéns de forma cilíndrica, dependendo do produto armazenado, também há uma grande dificuldade para mensurar o volume contido. Esta ocorre,



Figura 1.1: Foto de um armazém

principalmente, devido às propriedades físicas do produto e do ambiente, os quais impedem a formação do cone inscrito no cilindro, no processo de descarga. Tem-se então, uma etapa adicional no processo manual de auditoria, a qual consiste na distribuição uniforme do produto, a qual geralmente faz uso da distribuição manual, antecedendo cada avaliação do correspondente volume.

- Técnicas numéricas para cálculo do volume de superfícies não uniformes

Na computação científica, destaca-se o desenvolvimento de *softwares* numéricos baseados em técnicas, métodos e processos do cálculo numérico, com modelagem matemática, viabilizando aplicações de desenvolvimento científico e tecnológico (FOLEY, 1990; KREYSZIG, 1993; KIREHNER et al., 1989; SILVA, 2007).

Significativas contribuições são obtidas pela aplicação de métodos da Computação Científica na resolução numérica de problemas reais, permitindo aproximar a área de fundamentação teórica com a área de soluções tecnológicas. Esta aproximação deverá promover uma visão integrada, viabilizando soluções oportunas para o problema de prover um sistema para cálculo preciso e automatizado de volumes de relevos, considerando análise de erros e incluindo aplicação de novas tecnologias para coleta de dados.

A proposta de metodologia científica, baseada em métodos numéricos iterativos para cálculo de volume, pode contemplar solução para avaliação de estoques de granéis nas diferentes formas de armazéns. Para aumento e controle da precisão, aplica-se interpolação numérica para modelagem de curvas do relevo.

- Exploração de paralelismo

Devido à necessidade de agilidade no processo de auditoria, a exploração do paralelismo torna-se oportuna, pois possibilita a aceleração da execução do método de cálculo do volume dos produtos armazenados.

Com a utilização de grades de computadores, pode haver reaproveitamento de máquinas ociosas para processar os dados coletados do relevo. Entretanto, faz-se necessário um software para gerência da distribuição e da comunicação. Neste con-

texto, utilizando as ferramentas do projeto D-GM, o qual provê suporte à execução de programas da computação científica em ambientes paralelos e/ou distribuídos.

1.2.3 O desenvolvimento do projeto D-GM

O Projeto D-GM provê à proposta do *SisA³*, um ambiente para criação, gerenciamento e processamento paralelo do método *iCone* resultando no protótipo *p-iCone*. Este componente de software do *SisA³* é responsável pelo cálculo do volume de superfícies não uniformes e viabiliza a exploração do paralelismo por considerar as abstrações que fundamentam a concorrência no modelo de Máquina Geométrica.

Para disciplinar a elaboração de programas paralelos e distribuídos, o Projeto D-GM dispõe de um ambiente de programação visual denominado VPE-GM (*Visual Programming Environment for the Geometric Machine*) (PRESTES et al., 2005), cuja estrutura estimula o aprendizado de técnicas de programação paralela e concorrente, baseada em uma semântica bidimensional, inerente às linguagens visuais, provendo uma representação espaço-temporal da estrutura da memória e dos processos representados no modelo GM (REISER, 2002).

Como uma alternativa viável e realística à computação em grade, quando comparada aos tradicionais ambientes de supercomputadores (*cluster computing*), considera-se o uso de *middleware* EXEHDA (*Execution Environment for High Distributed Applications*) (YAMIN, 2004a) para suporte aos mecanismos de distribuição, comunicação e gerência das computações distribuídas, intermediando acessos entre o componente de *software* denominado VirD-GM (*Virtual Distributed Geometric Machine*) (FONSECA, 2008a), que gerencia a execução de aplicações, e os sistemas e linguagens nativas (JAVA) que integram o meio físico de execução.

O componente de *software* denominado VirD-GM (*Virtual Distributed Geometric Machine*) gerencia a execução de aplicações a partir das descrições obtidas pela modelagem gráfica concebida no VPE-GM.

Tratando-se o *p-iCone* como uma aplicação compatível com o modelo D-GM, o VirD-GM se mostra apropriado para execução do protótipo, pois apresenta todos os mecanismos de expressão para o paralelismo e sincronização, necessários para a implementação da solução para o problema de cálculo de relevo em superfícies não-uniformes.

Na perspectiva da computação multiprocessada, o componente VirD-GM contribui no gerenciamento e no controle da concorrência síncrona e as demandas de acesso à memória global compartilhada, implementando as interpretações concebidas no modelo GM e as correspondentes representações visuais obtidas no VPE-GM, além de garantir suporte à execução em SMP (Multiprocessamento Simétrico).

1.2.4 A integração dos grupos de pesquisa do PPGINF

O desenvolvimento desta aplicação consiste em uma proposta multidisciplinar e inter-institucional, motivada por dar continuidade e colaborar com a consolidação e a integração dos grupos de pesquisa GMFC (Grupo de Matemática e Fundamentos da Computação), GPDIB (Grupo de Pesquisa e Desenvolvimento em Instrumentação Biomédica) e G3PD (Grupo de Pesquisa em Processamento Paralelo e Distribuído), como também integrar esforços de pesquisa entre docentes e discentes do Mestrado em Ciência da Computação do PPGINF/UCPEL com o Departamento de Informática da UFPEL e com empresas agrícolas da região Sul do país.

O estudo e o desenvolvimento de um sistema integrado de hardware e software para o propósito de cálculo do volume de armazéns, além de abordar diversas tecnologias, ainda, apresenta-se como uma aplicação da computação científica que se enquadra no contexto do projeto D-GM, assim sendo usado como uma aplicação real executada neste ambiente de edição, geração e execução de aplicações paralelas e distribuídas.

1.3 Proposta de Pesquisa

Esta proposta de pesquisa introduz o sistema computacional *iCone*, para avaliação do volume de superfícies não uniformes, integrando técnicas numéricas para cálculo de volume e técnicas de visualização científica para modelagem de relevo tridimensional, com exploração do paralelismo para ganho de desempenho na sua execução.

Além disso, esta proposta também desenvolve uma estratégia para melhoria no processo de auditorias, no contexto do sistema de produção agrícola da região Sul do nosso Estado, visando à coleta automática de dados obtida por aplicação de tecnologia de medição a laser, viabilizando digitalização de pontos e modelagem do relevo em superfícies não-uniformes.

Mais especificamente, a proposta de Sistema Automatizado de Auditoria em Armazéns, denominado *SisA³*, considera o desenvolvimento de dois componentes:

- os protótipos de software denominados *iCone-SisA³* e *p-iCone-SisA³*, provendo soluções computacionais para cálculo do volume de relevos de formas irregulares.
- o protótipo híbrido de *hardware* e *software* denominado *Dig-SisA³*, referente ao equipamento digitalizador, viabilizando a medição de pontos do relevo não-uniforme formado pelo produto armazenado;

Neste contexto, este trabalho assume o desafio de contribuir para a resolução, com sucesso, da dificuldade de auditorias confiáveis e automáticas, com atualização da base de conhecimentos disponíveis nas empresas e instituições de produção e armazenamento de produção agrícola de grãos.

1.4 Objetivos

O objetivo geral deste trabalho consiste na concepção, modelagem e desenvolvimento do *SisA³* para simulação computacional do relevo em superfícies não uniformes e cálculo do correspondente volume. Este objetivo será alcançado com base no estudo e na aplicação de técnicas da computação científica integrando modelagem matemática, processamento paralelo e distribuído, e ainda, aplicação de técnicas da visualização científica.

Mais especificamente, busca-se:

- Desenvolver o protótipo *iCone* para execução sequencial do cálculo de volume de relevos.
- Determinar critérios para avaliação dos métodos, considerando o controle do erro presente no processo computacional.

- Estender o protótipo sequencial para a abordagem paralela em arquiteturas multiprocessadas, para ganho de desempenho e maior flexibilidade na utilização do *SisA*³.
- Avaliar a implementação dos algoritmos, considerando precisão e parâmetros do algoritmo.
- Desenvolver estudos de caso para validação dos protótipos desenvolvidos.
- Modelar e desenvolver o componente eletromecânico do *DigSisA*³, responsável pelos movimentos do suporte articulado que contém um medidor de distâncias a laser;
- Especificar e desenvolver o componente eletrônico, de interface de comunicação entre o computador e controle dos atuadores;
- Modelar e implementar o componente de *software*, responsável pelo controle da movimentação de motores de passo que viabilizam a movimentação da trena laser;
- Modelar a integração entre o sistema de digitalização e o método científico *iCone* do *SisA*³.

A estrutura definida para o texto contempla individualmente estes objetivos, porém os mesmos estão inter relacionados, sempre que necessário para caracterizar fatos e tendências na área da simulação computacional.

1.5 Contextualização da Proposta

Este trabalho aborda o desenvolvimento de uma solução computacional, multidisciplinar para solução do problema de cálculo de volume de armazéns de granéis. O contexto multidisciplinar é suportado pela integração de quatro áreas:

- A computação científica, como solução matemática para modelagem e cálculo de volume de superfícies não uniformes;
- O processamento paralelo e distribuído, no desenvolvimento paralelizado dos protótipos, incluindo a execução em dispositivos multi-core;
- A robótica, no desenvolvimento de um equipamento digitalizador de relevos;
- A visualização científica que trata da modelagem de objetos reais em computadores, assim como a manipulação de seus elementos.

A seguir, estas áreas serão brevemente descritas.

1.5.1 Computação Científica

O estudo e análise dos algoritmos numéricos utilizados na solução do cálculo do volume de relevos não-uniformes estão contextualizados na computação científica, a qual considera o desenvolvimento de técnicas e métodos para propor solução numérica a problemas científicos, com fundamentação baseada na modelagem matemática.

Frequentemente, esta modelagem requer uma grande quantidade de cálculo, os quais são geralmente executados utilizando técnicas desenvolvidas pelo cálculo numérico,

objetivando precisão e controle de erros (SPERANDIO; MENDES; SILVA, 2006; CLÁUDIO; MARINS, 1994).

O protótipo *iCone* recebe a matriz de pontos gerada pelo digitalizador e, através da aplicação de segmentação do relevo em fatias, obtidas pela interpolação de pontos e pela integração trapezoidal, obtém-se o resultado numérico (somatório dos volumes de cada fatia) que corresponde ao volume total armazenado.

Para análise e melhoria na modelagem do relevo, o trabalho considera a interpolação de pontos intermediários, por *splines* polinomiais cúbicas, técnica que permite que pontos intermediários do conjunto de pontos sejam calculados, levando em consideração a tendência da curva gerada pelos pontos.

1.5.2 Sistemas Paralelos e Distribuídos

As técnicas de execução de algoritmos de forma concorrente são estudadas e fundamentadas na área de processamento paralelo, possibilitando a execução de trechos de código em diferentes núcleos processadores. Este estudo se mostra interessante pois viabiliza um ganho considerável nas aplicações que se caracterizam por não possuir dependências de dados (PILLA; SANTOS; CAVALHEIRO, 2009).

Buscando ganho de desempenho, o método *p-iCone* implementa o algoritmo para cálculo do volume *iCone* promovendo execução concorrente a aplicação. Esta pode ser viabilizada pelo uso da computação em grade (*grid computing*) e pelo uso de sistemas multiprocessados, através da exploração do paralelismo de dados (decomposição de domínio).

Na abordagem da computação em grade, o componente *p-iCone* deverá executar o processamento paralelo, a partir das informações do equipamento digitalizador de relevo (*DigSisA³*), proporcionando a medição do volume de produto estocado, com o erro previamente conhecido e controlado.

Já, na abordagem da computação multiprocessada, considera-se o estudo da aplicação de *Symmetric Multi Processing* (SMP) com compartilhamento de memória na execução concorrente da aplicação *p-iCone*. Assim, a aplicação é particionada em fluxos de instruções, sendo que cada etapa paralelizável do algoritmo pode ser executada concorrentemente, em qualquer dos processadores disponíveis do multiprocessador.

1.5.3 Área de Automação

O estudo, análise e aplicação de tecnologias na área de automação se mostra uma etapa indispensável, a ser alcançada, para possibilitar o desenvolvimento do protótipo *Dig-SisA³*.

O desenvolvimento do protótipo de *software* gerenciador do *Dig-SisA³*, controlador e correspondente *hardware*, são projetados, desde sua modelagem, com flexibilidade e adaptabilidade, integrando-se ao protótipo *iCone*.

Salienta-se a relação da automação com a área de sensoriamento, na busca de uma solução computacional eficiente, o que inclui o estudo de tecnologias para digitalização de dados de relevos.

O protótipo *DigSisA³* tem como principal funcionalidade a digitalização de relevos, que ocorre efetuando n medições sobre o relevo do produto armazenado e aplicando um sistema de deslocamento integrado a três componentes: (i) componente eletromecânico, responsável pelos movimentos do suporte articulado que contém um medidor de distâncias a laser; (ii) componente eletrônico, para servir de interface de

comunicação entre o computador e controle dos atuadores; e *(iii)* componente de *software*, responsável pela controle da movimentação de motores de passo que viabilizam a geração da matriz de dados de entrada a partir dos pontos tridimensionais coletados, após a execução de todas as medições.

1.5.4 Visualização Científica

A área de pesquisa em visualização científica considerada neste trabalho refere-se à visualização de superfícies a partir de dados volumétricos e tem como objetivo a criação de imagens a partir de conjuntos de dados que representam fenômenos contendo informações científicas sobre as quais se deseja realizar uma análise e/ou exploração. Essas imagens auxiliam na extração de informações relevantes e permitem uma melhor compreensão dos fenômenos estudados.

Contextualizada como uma subárea da visualização científica, a visualização volumétrica visa a exibição do interior de objetos volumétricos, a fim de explorar sua estrutura e facilitar seu entendimento.

Salienta-se que a grande vantagem desta técnica é não necessitar decidir quando uma superfície atravessa uma certa região do volume. Essa é uma tarefa essencial para os algoritmos de detecção de superfícies, que tentam aproximar uma superfície através de primitivas geométricas e então usar os algoritmos e técnicas tradicionais de visualização (SEIXAS; GATTAS, 1996; SCHMIDT, 2000).

No entanto, os algoritmos para visualização volumétrica direta possuem um custo computacional muito alto, pois necessitam percorrer todo o volume de dados. Neste contexto, na maioria das vezes, os objetos volumétricos são definidos através de uma grade tridimensional, com um ou mais valores escalares e/ou vetoriais associados a cada ponto da grade. Os dados volumétricos são normalmente tratados como uma matriz de elementos de volume, denominados *voxels* (análogo 3D de *pixels*).

O *voxel* é a representação de uma vizinhança espacial, sobre a qual determinados atributos e características estão associados e podem ser tratados e inferidos.

Os dados volumétricos são obtidos, geralmente, de duas formas: *(i)* por simulação, na qual os dados são construídos a partir de algum modelo matemático, e *(ii)* por aquisição, em que os dados são resultantes da conversão de objetos existentes na natureza.

Um exemplo de dado volumétrico obtido por simulação ocorre na análise de fenômenos físicos através de elementos finitos ou por dinâmica de fluidos. Exames médicos como os de Ressonância Magnética, Tomografia Computadorizada e Ultrasonografia geram conjuntos de dados volumétricos adquiridos. Estes volumes representam as estruturas internas de regiões do corpo do paciente, permitindo que os médicos façam diagnósticos sem a necessidade de procedimentos invasivos.

O *SisA³* utiliza um sistemas de mapeamento do relevo por aquisição, onde através do protótipo *DigSisA³*, a digitalização da amostragem de pontos é efetuada. O armazenamento dos dados do relevo são feitos em uma grade tridimensional, que contém as coordenadas dos pontos adquiridos.

Esta diversidade de formas de aquisição de dados volumétricos vem impulsionando pesquisas na busca de novas técnicas de visualização, a serem empregadas em áreas como Medicina, Geociências, Sensoriamento Remoto, Meteorologia, dentre outras.

1.6 Organização do Texto

A apresentação deste texto está organizada em oito capítulos, brevemente resumidos a seguir.

No Capítulo 2, considera-se uma revisão das técnicas computacionais para cálculo do volume referente à superfícies não uniformes, com ênfase na interpolação numérica e nos métodos científicos para cálculo iterativo de volumes. Também foram brevemente analisadas as principais ferramentas e trabalhos relacionados com a modelagem e visualização do relevo de superfícies não uniformes, identificando as contribuições obtidas com esta revisão ao trabalho proposto.

No Capítulo 3, tem-se a descrição do projeto D-GM, apresentando as características do ambiente de desenvolvimento VPE-GM e o de execução distribuída VirD-GM relevantes para suporte ao protótipo p-iCone.

A proposta de arquitetura para o sistema *SisA*³ é apresentada no Capítulo 4, considerando as principais características e tecnologias envolvidas na metodologia proposta, que definem a solução computacional para o problema de auditorias em armazéns de grãos.

Os principais aspectos referentes à metodologia científica aplicada na modelagem e desenvolvimento dos algoritmos para cálculo do volume do relevo em superfícies irregulares, são apresentados no Capítulo 5. Neste caso, métodos numéricos são aplicados desenvolvimento do algoritmo que implementa o protótipo *iCone*, incluindo a correspondente extensão paralela.

A descrição da modelagem do componente *Dig-SisA*³ que define o equipamento digitalizador de acordo com a arquitetura do sistema *SisA*³ é apresentada no Capítulo 6. Esta modelagem considera desde a discretização do espaço de busca, a organização lógica referente aos componentes de *software* e *hardware*, bem como a descrição focando na estrutura funcional do componente *Dig-SisA*³.

No Capítulo 7, são destacados dois estudos de caso para validação dos protótipos *iCone* e *p-Icone*, com ênfase na avaliação sequencial e paralela, respectivamente.

Por fim, no Capítulo 8, as conclusões são apresentadas, além de, na perspectiva de continuidade, novas etapas do projeto são apontadas. São apresentados os artigos e resumos publicados durante o desenvolvimento do projeto.

2 TÉCNICAS COMPUTACIONAIS PARA CÁLCULO DO VOLUME REFERENTE À SUPERFÍCIES NÃO UNIFORMES

Este capítulo sintetiza os princípios gerais que norteiam o desenvolvimento de uma metodologia científica para obtenção de uma solução numérica, a partir da modelagem matemática de um problema físico real. Esta síntese considera técnicas sobre as quais a metodologia de modelagem e cálculo do relevo no *SisA*³ foi baseada, sendo feita pela revisão sob duas óticas: (i) técnicas para cálculo do volume baseadas no processamento de imagens; (ii) técnicas para cálculo do volume baseadas em métodos numéricos, descrevendo a interpolação por *spline* cúbica.

2.1 Técnicas para Cálculo de Volume Baseadas em Métodos Numéricos

Na aplicação da computação científica, o cálculo numérico tem disponibilizado e incentivado o estudo de métodos e técnicas para construção de algoritmos numéricos, visando à solução computacional de problemas que podem ser representados por um modelo matemático.

2.1.1 Princípios do Cálculo Numérico

Os princípios do cálculo numérico estão baseados em quatro importantes conceitos (SPERANDIO; MENDES; SILVA, 2006; KREYSZIG, 1993; KENDALL, 1978):

- *Iteração ou aproximação sucessiva*: Partindo-se de uma solução aproximada, inicial, repetem-se as mesmas ações/processos com base nas equações de recorrência, considerando também relevante a determinação da convergência do processo iterativo, as condições e/ou critérios de parada e os limites de erro no processo computacional.
- *Discretização*: Na solução de problemas contínuos, a discretização do domínio de dados de entrada viabiliza a descrição sequencial dos passos que definem um algoritmo, cada um envolvendo um número finito de operações de máquina. Assim, na obtenção de uma solução, a evolução temporal caracterizada pela discretização do tempo, pode ser descritas pelas computações, ao término de uma execução.

- *Aproximação*: Caracteriza-se pela substituição de um modelo matemático analítico por outro discretizado, que ofereça comportamento de interesse ao problema, mas que seja mais simples de manipular. Na definição de algoritmos numéricos e nas representações gráficas, consideram-se parâmetros para avaliação da aproximação, como erro relativo, dígitos significativos exatos, e outros.
- *Transformação do Problema*: Dado um problema P , desmembra-se ele em N subproblemas, P_1, P_2, \dots, P_N , mais simples de serem resolvidos. Por exemplo, o cálculo da área de trapézios quando obtido pelos correspondentes triângulos e retângulos.

2.1.2 Etapas na Resolução Numérica de um Problema

Na computação científica, várias etapas podem estar envolvidas na busca de uma solução numérica de um problema proposto. Dentre as principais etapas para resolução numérica de um problema, destacam-se:

- Definição do problema para avaliação de fenômeno físico, econômico, natural e outros;
- Refinamento e delimitação do problema;
- Resultados de ciências afins são considerados como parâmetros, constantes, heurísticas para modelagem do problema;
- Mensuração dos dados para modelagem matemática;
- Escolha do método científico;
- Definição e implementação do algoritmo, sendo que na etapa de codificação do algoritmo tem-se aplicação de uma linguagem de programação, incluindo a análise da convergência;
- Resolução numérica envolvendo a escolha dos parâmetros de entrada e de precisão;
- Análise do erro e desempenho computacional; e
- Resultado numérico.

Salienta-se que podem ocorrer outras etapas intermediárias, e na maioria das vezes, algumas das etapas descritas anteriormente podem ser omitidas, pois estas são dependentes do tamanho e da natureza do problema considerado.

Em muitos casos, a simulação computacional associada ao método numérico é extremamente importante, evitando por exemplo, problemas técnicos e éticos referentes a pesquisas em seres humanos ou animais, ou por viabilizar soluções de baixo custo para problemas de emergência, introduzindo inovação tecnológica, reduzindo riscos, incrementando testes, validando e melhorando a qualidade do projeto ou pesquisa.

2.2 Técnicas para Cálculo de Volume Baseadas no Processamento de Imagens

Esta seção apresenta alguns trabalhos relacionados, com aplicação de técnicas para cálculo do volume que contribuíram para o desenvolvimento da metodologia utilizada no trabalho, incluindo modelagem e implementação do *SisA*³.

Em (LIN; FU, 2007) o problema do cálculo do volume de armazéns é apresentado e propõe-se a sua resolução através da extração de vídeos do armazém e posterior tratamento das imagens. Primeiramente, as cores da imagem são extraídas e convertidas do espaço RGB (*Red Green Blue*) para o espaço HSI (*Hue Saturation Intensity(Brightness)*). Segue-se a extração das arestas delimitando o espaço que define a imagem do produto. Por fim, é aplicado um *benchmark* retangular para identificação da área da região. Nesta etapa se efetua o cálculo do volume através do gradiente de cores, o qual define a profundidade da área identificada.

Já em (LIN; JIANG, 2007), tem-se uma metodologia inteligente para auditoria de armazéns tridimensionais de granéis. Através de fotos é obtida a identificação de imagens dos agrupamentos de produtos. Neste trabalho, verifica-se a necessidade de inserção de marcações em cores diferenciadas da coloração do produto, afim de efetuar a correção dos erros de perspectiva. Após a identificação das arestas limitantes do armazém em cada fotografia, o relevo é modelado e calculado, seguindo-se a análise e correção de erros, inerentes a este processo. Salienta-se que este método não é totalmente automatizado, pois há a necessidade de uma etapa de pré-tratamento do dados para modelagem e cálculo do volume.

Os métodos baseados no tratamento de imagens apresentam uma relação direta com a qualidade das fotografias ou dos vídeos, onde a luminosidade é um fator de grande relevância. Desta forma, como importante desvantagem do uso destas técnicas nos armazéns reais, cita-se a baixa luminosidade, que é uma característica da maioria dos armazéns graneleiros. Além desta, tem-se ainda a possível ocorrência da dispersão de poeira no ar, inserindo assim, ruído na aquisição das imagem e, desta forma, acrescentando fontes de erros nos métodos numéricos.

Considerando as necessidades e ruídos inerentes às técnicas baseadas no processamento de imagens, tem-se, frequentemente, baixa confiabilidade nas soluções para o problema de cálculo do volume em superfícies não uniformes.

2.3 Técnicas Iterativas para Cálculo do Volume

O método de **Integração do Volume de Subfatias**, utilizado neste trabalho para desenvolvimento dos protótipos *iCone* e *p-iCone*, está baseado na idéia da fórmula de Smalian (FORESTS; RANGE, 2007). Esta fórmula também fundamenta o trabalho apresentado em (KIREHNER et al., 1989) para o cálculo do volume de troncos de árvores. Neste caso, técnicas iterativas são aplicadas para auxiliar no processo de medição e seleção relacionados à gestão e gerenciamento em áreas de reflorestamento, de acordo com as leis ambientalistas e direitos de exploração.

Na fórmula de Smalian, consideram-se duas seções no tronco, uma localizada na parte superior do tronco, representando o diâmetro inicial (d_1), e a outra na parte inferior do tronco, representando o diâmetro maior (d_2). Estes dados, d_1 e d_2 , consistem nas informações sobre os quais é calculada a área de cada seção, respectivamente indicadas

por A_1 e A_2 e apresentadas na Figura 2.1.

O volume médio V é obtido como produto entre a soma das áreas pela metade do comprimento do tronco, conforme a Equação 2.1:

$$V = \frac{A_1 + A_2}{2} \times L \quad (2.1)$$

sendo que L indica o comprimento do tronco. Outras variantes desta fórmula utilizam valores diferentes de comprimento L , para modelar os diversos tipos de troncos.

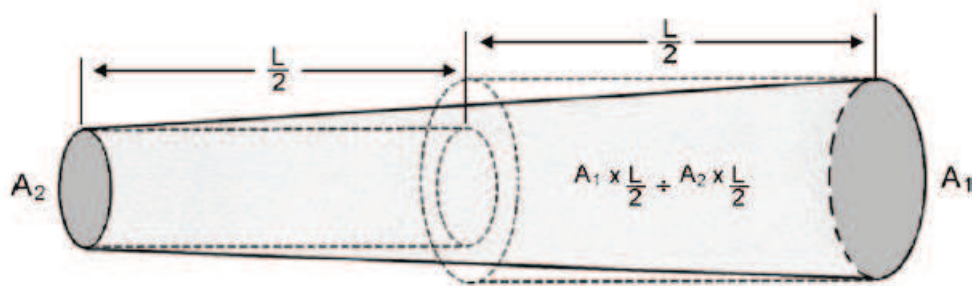


Figura 2.1: Visualização da fórmula de Smalian (FORESTS; RANGE, 2007)

Em (KIREHNER et al., 1989), tem-se um processo de medição manual na obtenção de pontos para o cálculo do volume do tronco de árvores, gerando dados para uma matriz de 6 pontos. Através da aplicação de interpolação por *splines* são introduzidos pontos intermediários, diminuindo assim a profundidade de cada fatia, e impactando em um aumento de precisão. O volume é obtido utilizando a ideia da fórmula de Smalian, através do produto da área calculada a partir da circunferência, pela profundidade de cada fatia, porém a quantidade de seções é aumentada através da aplicação da interpolação dos pontos que definem as fatias.

Outra técnica iterativa para o cálculo do volume foi desenvolvida para a fruta manga, considerando a integração de fatias virtuais obtidas a partir de uma metodologia iterativa, proposta em (CHAROENPONG et al., 2004). Neste trabalho, os dados de entrada são medidas obtidas pelo tratamento de imagens, as quais foram obtidas por fotos da fruta, considerando três distintos planos: vista superior, vista lateral e vista frontal. Nesta aplicação, o cálculo do volume considera a divisão do relevo em fatias, cada sendo geometricamente modelada por uma elipse, de acordo com a Figura 2.2. A partir do produto da área da elipse pela profundidade da fatia, o volume da fruta é determinado.

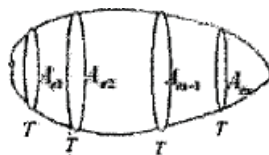


Figura 2.2: Representação de uma manga através de elipses

Após o estudo dos métodos apresentados, juntamente com a integração de trapézios para o cálculo de áreas, proporcionou-se o desenvolvimento da metodologia de Integração do volume de subfatias. Os pontos-chave para seu desenvolvimento foram a

divisão do relevo em fatias, a redivisão das fatias em subfatias pela interpolação numérica por *splines* e o cálculo da área das subfatias pela integração trapezoidal, chegando-se a um volume parcial definido pelo produto da área de cada subfatia pela profundidade da mesma. O volume final consiste no somatório das diversas subfatias que definem o objeto medido.

Pode-se também considerar a técnica iterativa de **Integração do Volume por Cubos Unitários**. Em um *cubo unitário* as arestas são pré-definidas por uma medida padrão, tomada como unidade de medida de volume. O tamanho de cada cubo unitário é definido de acordo com a precisão desejada e, deve-se avaliar o custo computacional em relação à precisão, no momento de se escolher o tamanho para o cubo (PAIVA; SEIXAS; GATTAS, 1999).

O algoritmo de integração de volume de cubos unitários tem o funcionamento semelhante ao algoritmo apresentado em (CHAROENPONG et al., 2004). A principal diferença se refere ao fato de que o volume é obtido pelo preenchimento da fatia por cubos unitários e, ao final do preenchimento, realiza-se o somatório de todos os cubos utilizados.

O cálculo do volume resulta do produto do total de cubos pelo volume associado ao cubo unitário. Salienta-se que, nesta modelagem, o comprimento de cada fatia é determinado pelo aresta do cubo.

2.4 Técnicas de Visualização de Dados Volumétricos

Podem-se classificar as técnicas de visualização de dados volumétricos em dois grandes grupos: métodos que realizam a extração de superfícies (*surface fitting*) e métodos que implementam a renderização direta de volumes (*volume rendering*).

Estas abordagens diferem, basicamente, pela utilização ou não de representações intermediárias dos dados volumétricos para a criação de imagens adequadas à aplicação. Enquanto na renderização direta de volumes a criação da imagem é realizada diretamente a partir dos dados volumétricos, na extração de superfícies os dados volumétricos são convertidos para uma representação geométrica (poligonais) a partir da qual são empregadas técnicas tradicionais de renderização de polígonos para criar a imagem final (PAIVA; SEIXAS; GATTAS, 1999).

Existem diferentes estratégias para a criação de imagens híbridas envolvendo a visualização do dado volumétrico e do modelo poligonal. Podemos classificar estas estratégias de acordo com a necessidade, ou não, de conversão dos dados para uma única representação, a fim de facilitar a etapa de combinação (LORENSEN; CLINE, 1987).

2.4.1.1 Estratégia Baseada em Superfícies

Na estratégia baseada em superfícies ocorre a conversão da representação volumétrica para a geométrica, através da criação de isosuperfícies poligonais que correspondem às estruturas presentes no dado volumétrico.

Algoritmos como o *Marching Cubes* são utilizados para realizar a extração das isosuperfícies a partir do dado volumétrico. Assim, isosuperfícies e modelo poligonal são renderizados através do *pipeline* de visualização geométrica. Empregar somente a descrição por polígonos parece uma estratégia promissora, uma vez que o *pipeline* de visualização geométrica pode ser acelerado por hardware.

No entanto, a extração de isosuperfícies tende a criar centenas de polígonos, alguns muito finos ou degenerados, exigindo que se realize uma etapa de decimação antes da visualização, o que introduz uma perda na qualidade e na precisão do modelo e um acréscimo no tempo de processamento (LEVOY, 1988; KAUFMAN, 1988; KAUFMAN; YAGEL; COHEN, 1990; KAUFMAN, 1997).

2.4.1.2 Estratégia Baseada em Elementos de Volume

A estratégia baseada em elementos de volume caracteriza a transformação do modelo poligonal para a representação volumétrica através de algoritmos de voxelização.

A representação volumétrica do dado poligonal é combinada diretamente com o dado volumétrico, empregando algoritmos como, por exemplo, o *Ray-Casting*. Também nesta abordagem pode existir perda de precisão do modelo poligonal que está sendo convertido (YAGEL, 1996).

2.4.1.3 Estratégia Baseada em Primitivas de Pontos

Na estratégia baseada em primitivas de pontos se considera a transformação do volume e do dado poligonal em primitivas do tipo ponto. Estas primitivas são compostas de informações como posição 3D, valor de cor; vetor normal, opacidade e outros dados necessários durante o processo de cálculo de cor.

A primitiva de ponto pode ser renderizada como um ponto no espaço 3D ou através de técnicas como *Splatting*, que calcula a contribuição da projeção do ponto para uma determinada vizinhança de *pixels* (WESTOVER, 1990).

2.4.1.4 Estratégia Híbrida

A estratégia híbrida emprega a renderização de cada tipo de dado separadamente, combinando as imagens do modelo poligonal e do volume em uma única imagem final. Nesta abordagem, podem-se empregar as melhores técnicas para cada tipo de representação, pois não há a necessidade de conversão de formatos, evitando perda de precisão. Trabalhos como o *Z-Merging* (KAUFMAN; YAGEL; COHEN, 1990) e o *Ray-Merging* (LEVOY, 1988) apresentam algoritmos de visualização híbrida para a combinação de volumes e modelos poligonais.

2.5 Método de Interpolação Numérica

A aproximação de funções pode ser obtida por técnicas de mínimo quadrado (regressão linear, polinomial e outras). Neste trabalho, considera-se o ajuste de dados experimentais por interpolação. Neste contexto, a interpolação numérica é um método frequentemente utilizado em processos de ajuste de curvas. Este método está definido a partir de um conjunto discreto de dados pontuais, viabilizando uma aproximação para novos dados numéricos, de acordo com uma precisão estabelecida. Estes novos dados correspondem a valores intermediários dentro do intervalo que contém o conjunto discreto de dados pontuais.

Mais detalhadamente, na interpolação, os dados pontuais coincidem com os valores de uma função de interpolação no intervalo considerado, e pela expressão analítica desta função, pode-se determinar valores intermediários. É possível ainda a inserção de

valores no conjunto de dados, de acordo com a aplicação modelada.

A procura por uma função de interpolação, dentro de uma classe específica de funções, caracteriza o método de interpolação. Por exemplo, no método denominado *spline* polinomial cúbica, tem-se que a função de interpolação é uma função polinomial de terceiro grau.

Em especial, os métodos de interpolação são imprescindíveis para a previsão e formulação de equações referentes aos mais diversos fenômenos, envolvendo medições experimentais, funções complexas, tabelas estatísticas, e outros.

Dentre os diversos métodos de interpolação (diferenças divididas, diferenças finitas, Lagrange, Newton), considera-se a interpolação polinomial cúbica por splines, para aproximação dos pontos na metodologia desenvolvida neste trabalho, dada pela aproximação a partir de um conjunto discreto de dados pontuais, que melhor se adapte a distribuição desses pontos (KIREHNER et al., 1989; CLÁUDIO; MARINS, 1994; SILVA, 2007).

Nas bibliotecas gráficas, segmentos de linhas ou polígonos são os elementos básicos, e uma curva arbitrária pode ser modelada através de uma sucessão destes elementos. Porém, diversas vezes isto não é suficiente para obter funções de aproximação, cujos gráficos garantam continuidade e diferenciabilidade, e que sejam também estáveis numericamente, apresentando suavidade satisfatória para a abordagem considerada.

Geralmente, funções de aproximação necessitam de uma grande quantidade de dados para satisfazer tais condições, e por consequência, são de difícil manipulação (BARTELS; BEATTY; BARSKY, 1987).

Técnicas mais avançadas (sombreamento contínuo, técnicas poligonais, etc.) podem resultar em imagens visualmente identificáveis pela composição de partes de objetos geométricos elementares (círculos, esferas, cilindros, etc.), mas nestes casos as funções apresentam uma regularidade na construção gráfica. Assim, a modelagem restringe-se a objetos com baixa complexidade.

Entretanto, várias classes de funções não apresentam tal regularidade em sua construção gráfica, o que torna inconveniente o uso das técnicas de modelagem citadas anteriormente. Estas funções podem apresentar curvatura irregular, ou ainda, podem ser compostas por sucessivos segmentos de curva. Neste caso, frequentemente geradas através de técnicas de interpolação ou aproximação a partir de pontos conhecidos.

Neste trabalho, considera-se o estudo, a aplicação e a implementação da interpolação cúbica por *splines* (BARTELS; BEATTY; BARSKY, 1987; KREYSZIG, 1993; SILVA, 2007; FOLEY, 1990).

2.5.1 Splines

Na modelagem de objetos reais, pode-se usar as coordenadas dos pontos reais que compõem grande parte do objeto. No entanto, é excessivamente custoso armazenar e manipular todos estes pontos na memória de um computador. Considerando tais restrições, e visando uma abordagem prática, os objetos são aproximados com segmentos de planos, curvas, ou outras formas geométricas, os quais podem ser matematicamente descritas. Deste modo, dada uma curva ou superfície cuja função f que a define é desconhecida, é possível construir, a partir de um número finito de pontos da curva ou superfície, uma boa aproximação de f (SILVA, 2007; KIREHNER et al., 1989; FOLEY, 1990; CLÁUDIO; MARINS, 1994; FORESTS; RANGE, 2007).

Visando a preservação das propriedades matemáticas e geométricas que determi-

nam a fidelidade do objeto modelado, este projeto considera a interpolação por *splines*. As *Splines* são curvas que possibilitam uma descrição das funções baseadas nestas propriedades.

O termo *spline* refere-se à longa e flexível barra de metal utilizada pelos desenhistas e engenheiros para delinear as superfícies de aviões e navios. A equivalente expressão matemática a esta barra, *spline cúbica natural*, consiste na continuidade da polinomial cúbica, a qual considera a posição, a inclinação e a curvatura como parâmetros que interpolam os pontos de controle. Justifica-se, portanto, a utilização de *splines* neste trabalho, pois consistem nas formas mais suaves de modelagem de curvas paramétricas cúbicas (FOLEY, 1990).

2.5.2 Conceitualização

Seja $Q(u) = (X(u), Y(u))$ uma curva bi-dimensional e parametrizada, tal que $X(u)$ e $Y(u)$ são funções uni-dimensionais definidas pelo parâmetro u . $X(u)$ e $Y(u)$ são definidas pelas coordenadas X e Y de cada ponto na curva $Q(u)$, para qualquer valor de u , respectivamente.

Na Figura 2.3, apresenta-se uma curva definida parametricamente, onde considera-se a interpolação linear na representação através de segmentos de retas, e a interpolação polinomial, na representação de segmentos de poligonais.

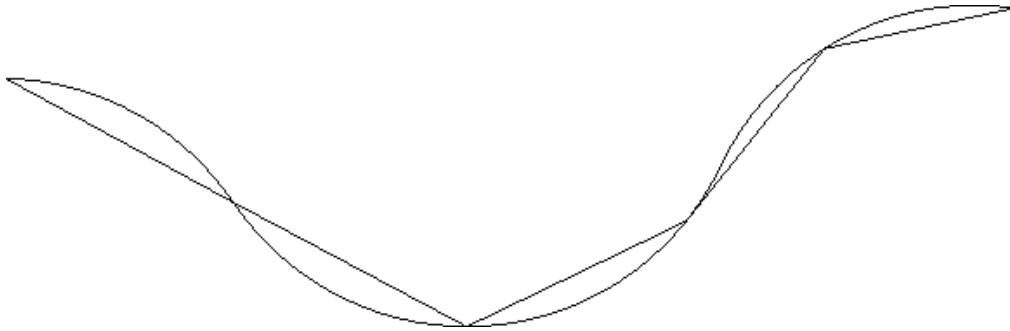


Figura 2.3: Curva definida parametricamente

As funções parametrizáveis $X(u)$, $Y(u)$ são funções polinomiais, portanto são consideradas computacionalmente eficientes e de fácil manipulação. Porém, nem sempre é possível modelar a curva $Q(u)$ aplicando somente uma função. Para modelar esta possível irregularidade, considera-se o particionamento da curva em segmentos, onde u é um parâmetro que varia de um ponto inicial u_{min} a um ponto final u_{max} .

Neste caso, cada parâmetro u , determinado pelas correspondentes funções polinomiais $X(u)$ e $Y(u)$, é denominado de nó, correspondendo à ligação entre um segmento e o próximo, com exceção dos nós extremos u_{min} e u_{max} . Salienta-se também que, a sequência de nós não pode ser decrescente, então

$$u_0 \leq \dots \leq u_j = u_{min} \leq \dots \leq u_l = u_{max} \leq \dots \leq u_n.$$

A sequência de nós $\{u_k\}_{0 \leq k \leq n} = [u_0, \dots, u_j, \dots, u_l, \dots, u_n]$ é denominada de vetor de nós. A componente $X(u)$ está segmentada por um conjunto de funções polinomiais, cobrindo todo o intervalo de definição do parâmetro u , considerando os segmentos variando de u_{min} a u_{max} . Além disto, cada componente satisfaz condições de continuidade

nas junções entre sucessivos segmentos polinomiais (garantindo existência e unicidade da função diferencial de várias ordens). O mesmo ocorre no componente $Y(u)$.

Existem duas maneiras distintas de se modelar uma curva específica: (i) modelagens baseadas em técnicas de aproximação, no caso quando a curva não passa nos pontos do domínio; ou (ii) modelagens baseadas em interpolação, quando a curva passa, obrigatoriamente, pelos pontos do domínio.

As splines polinomiais cúbicas estão inseridas no contexto das modelagens em interpolação, desta forma, os pontos inicial e final de cada curva coincidem com pontos reais.

Cálculo dos Polinômios de Interpolação

Sejam dados $m + 1$ pontos, P_0, \dots, P_m . Então a curva é definida por m polinômios, cada par de pontos $(P_i, P_{i+1})_{0 \leq i \leq m-1}$ determina o i -ésimo segmento e está conectado por um polinômio distinto.

Em geral as coordenadas $X(u)$ dos pontos da curva são determinados somente pelas coordenadas x_0, \dots, x_m , ocorrendo o mesmo com os outros eixos, ou seja, o cálculo de cada polinômio é obtido de forma independente.

Cada segmento x_0, \dots, x_m possui uma função polinomial no parâmetro u . A função polinomial cúbica

$$X_i(u) = a_i + b_i u + c_i u^2 + d_i u^3 \quad (2.2)$$

satisfaz as seguintes condições:

$$X_i(0) = x_i = a_i \quad X_i(1) = X_{i+1} = a_i + b_i + c_i + d_i$$

Considerando-se os quatro coeficientes da Eq. 2.2 que determinam completamente a função $X_i(u)$, e a continuidade das funções nos pontos dos segmentos inicial e final, aplica-se a primeira derivada D_i de $X(u)$ em cada nó u_i e tem-se:

$$X_i^{(1)}(0) = D_i = b_i \quad X_i^{(1)}(1) = D_{i+1} = b_i + 2c_i + 3d_i$$

Assim, é gerado o sistema de equações:

$$\begin{cases} a_i = x_i \\ b_i = D_i \\ c_i = 3(x_{i+1} - x_i) - 2D_i - D_{i+1} \\ d_i = 2(x_i - x_{i+1}) + D_i + D_{i+1} \end{cases} \quad (2.3)$$

A solução do sistema 2.3 consiste na determinação da coordenada $X(u)$. De forma análoga, obtém-se os coeficientes referentes aos demais eixos, neste caso $Y(u)$. Portanto, ao final, tem-se a identificação de todos os pontos intermediários do intervalo de interpolação.

A interpolação por *splines*, apresentada nesta seção, será utilizada nas metodologias que propõem a geração de uma quantidade de pontos interpolados, a qual mostra-se inúmeras vezes superior à real quantidade de pontos coletados pelo instrumento digitalizador, definindo assim a matriz de entrada.

2.6 Conclusão

Neste capítulo, foram apresentadas técnicas para modelagem computacional, tratamento dos dados, cálculos iterativos para determinação de volume e de áreas. Em especial, teve-se como foco principal a fundamentação, baseada em técnicas científicas, para viabilizar uma solução prototipada para o problema do cálculo de volume de um relevo limitado por superfície não uniforme. A solução prototipada considerou a utilização das técnicas iterativas para o cálculo de volume, no que se refere à segmentação do relevo e o cálculo de cada segmento sendo efetuado separadamente em cada iteração e a utilização da interpolação numérica por *splines* para aproximação de pontos intermediários do relevo para melhor aproximação do resultado final.

3 PROJETO D-GM: AMBIENTE DE SUPORTE AO DESENVOLVIMENTO E EXECUÇÃO DO MÉTODO P-ICONE

Este capítulo apresenta o projeto D-GM, o qual serve como base para o desenvolvimento e execução paralela e/ou distribuída do protótipo *p-iCone*, abordando as estratégias de sistemas distribuídos, processamento multinúcleo e a estratégia híbrida.

O Projeto D-GM dispõe de um ambiente para criação, gerenciamento e processamento de aplicações paralelas e/ou distribuídas da computação científica com base nas abstrações do modelo GM (*Geometric Machine Model*) (REISER, 2002; REISER; COSTA; DIMURO, 2002, 2003a). O objetivo deste capítulo é apresentar do ponto de vista do ambiente D-GM, o suporte que este oferece tanto para a modelagem e o desenvolvimento quanto para execução da aplicação *SisA*³.

3.1 Noção Intuitiva do Modelo GM

O modelo GM consiste em uma máquina abstrata, com memória compartilhada e possivelmente infinita, modelada por um espaço geométrico. Os processos e estados computacionais no modelo GM estão indexados pela discretização de pontos deste espaço geométrico. Fundamentada na Teoria dos Domínios (ABRAMSKY; JUNG, 1994), considera-se os espaços coerentes (GIRARD; LAFONT; TAYLOR, 1989), para prover semântica denotacional para processos e estados no modelo GM.

A estrutura ordenada que modela os estados no modelo GM é definida pelo espaço coerente de estados computacionais, caracterizando um estado de máquina como uma função do conjunto enumerável de posições de memória ao conjunto de valores de memória.

No contexto do projeto GM, define-se processo computacional elementar por uma sequência de funções que ao serem finalizadas alteram apenas uma posição da memória. Esta quantidade de funções necessita ser enumerável e caracteriza uma unidade de tempo computacional. O processo elementar é representado a partir da definição da ação computacional (código e parâmetros de entrada) e da posição de memória que recebe o resultado de sua execução (parâmetro de saída) (MUNHOZ, 2009). Para se alterar mais de uma posição de memória, deve-se criar vários processos elementares, estes podendo ter execução sequencial ou paralela.

Obtém-se assim a noção de computação no modelo GM, concebida como uma transição de estados associada a uma localização espacial. A estrutura ordenada dos

processos é também capaz de modelar, de forma explícita, a concorrência síncrona e o conflito de acesso a memória, modelando computações concorrentes e/ou não-determinísticas, respectivamente.

Concebido com base na versão distribuída do modelo GM (REISER; COSTA; DIMURO, 2003b, 2004a), o Projeto D-GM dispõe, de um ambiente de programação visual denominado VPE-GM (*Visual Programming Environment for the Geometric Machine*), cuja estrutura estimula o aprendizado de técnicas de programação concorrente baseada numa semântica bidimensional inerente às linguagens visuais, provendo uma representação espaço-temporal da estrutura da memória e dos processos representados no modelo GM.

Além do ambiente de programação visual, a arquitetura D-GM dispõe do ambiente de execução VirD-GM, o qual é responsável pela gerência da execução dos processos construídos através do VPE-GM.

A estrutura indutiva do modelo GM, é desenvolvida em níveis, sendo que cada nível determina um novo conjunto de processos, classificados de acordo com a quantidade de posições de memória alteradas após o final de sua execução. O primeiro nível define o conjunto enumerável de processos elementares, sobre os quais são aplicados construtores (processos sequenciais, processos paralelos, somas não-determinísticas, somas determinísticas e processos iterativos) gerando novos processos.

Salienta-se ainda que, o modelo GM provê interpretação para dois tipos especiais de paralelismo: (*i*) o espacial, com memória global compartilhada por processos modelados por estruturas matriciais, garantindo uma semântica para computações executadas em sistemas multiprocessados (SMP); e (*ii*) o temporal, na versão distribuída do modelo, considera um conjunto enumerável de modelos GM, obtendo semântica para computações distribuídas baseadas na computação em grade.

Estas estruturas de paralelismo definem construções de processos que tornam esta construção interessante, sobretudo porque mostra que a modelagem de sistemas de computação paralela e/ou distribuída pode ser obtida como uma generalização da modelagem de sistemas de computação convencionais.

O modelo GM caracteriza-se como um modelo de computação onde a exploração do paralelismo caracteriza-se como totalmente explícita, a nível de abstração, onde construções como assinalamento e composição, devem ser explicitados no programa, considerando para tal a notação posicional dos processos, testes e estados de memória. De forma análoga, o mapeamento e sincronização de tarefas, incluindo o gerenciamento do conflito de acesso à memória, no modelo GM são também formalizados pela função-posição, que define a estrutura de processo elementar e estado de memória.

O projeto D-GM está contextualizado na próxima seção, onde são apresentados os principais componentes, detalhando o funcionamento dos ambientes de desenvolvimento e execução.

3.2 Projeto D-GM: Principais Componentes

O Projeto D-GM tem como principal objetivo prover um ambiente para modelagem e execução de aplicações paralelas voltadas para computação científica, concebidas segundo as abstrações do modelo GM. Para alcançar este objetivo, o desenvolvimento do D-GM considera uma organização lógica que compreende três níveis (FONSECA, 2008b):

- Nível superior, onde estão disponíveis as abstrações para programação de aplicações distribuídas e/ou paralelas (VPE-GM), cuja natureza poderá compreender computações científicas, com ênfase nas áreas de programação estocástica (REISER; COSTA; DIMURO, 2003c, 2004b,c), intervalar (REISER; COSTA; DIMURO, 2003d,e,c, 2004d,e, 2005a) e quântica; (REISER; COSTA; DIMURO, 2004f, 2005b; REISER; COSTA; AMARAL, 2007a,b; AMARAL, 2009)
- Nível intermediário, que compreende os serviços dos mecanismos de suporte à execução da aplicação (VirD-GM). Esta fatia provê a estrutura de *software* para as funcionalidades necessárias à correta execução, viabilizando os serviços de controle do fluxo de dados e manipulação das dependências entre as computações concorrentes, comunicação e simulação da memória global;
- Nível inferior, que engloba os sistemas e linguagens nativas que integram o meio físico de execução, e a plataforma base de implementação (Máquina Virtual Java), juntamente com o sistema operacional e a rede de interconexão.

A estruturação funcional dos sistemas que integram o Projeto D-GM, apresentada na Figura 3.1, é obtida a partir de dois importantes componentes de software:

- O componente VPE-GM, responsável pela etapa de modelagem das aplicações (PRESTES et al., 2004; PRESTES; REISER, 2004; PRESTES; REISER; COSTA, 2004, 2005; PRESTES, 2005; PRESTES; REISER, 2005);
- O componente VirD-GM (FONSECA et al., 2007a,b; FONSECA, 2008a; MUNHOZ, 2009; MUNHOZ et al., 2007, 2008), definindo a arquitetura de *software* para suporte e gerenciamento da execução distribuída.



Figura 3.1: Organização Funcional do Projeto D-GM (FONSECA, 2008a)

Baseando-se nestas propostas, diversas aplicações paralelas vem sendo desenvolvidas. Todas estas aplicações necessitam de sistemas de gerência para possibilitar a exploração do paralelismo.

Os sistemas distribuídos, baseados em arranjos de computadores interligados por uma rede de interconexão, geralmente apresentam um *middleware*, composto por *software* que gerencia o paralelismo e distribui as tarefas, além de monitorar as execuções, podendo

citar o EXEHDA (YAMIN, 2004b), o qual é utilizado para gerência da distribuição de programas desenvolvidos em D-GM, MPI e Java.

Já as máquinas que possuem mais de um processador, seja na placa mãe ou no mesmo *chip* (multi-cores), tem o seu processamento paralelo gerenciado pelo sistema operacional (PILLA; SANTOS; CAVALHEIRO, 2009). Nestes sistemas multiprocessados, duas estratégias podem ser exploradas pelo VirD-GM, uma baseada na exploração de *threads*, e outra baseada na execução de diversas instâncias do VirD-GM, assim executando as diversas tarefas com a visão de nodos independentes.

Nas Seções 3.3 e 3.4, são revisados os principais conceitos de cada componente, VPE-GM e VirD-GM, respectivamente, considerando os trabalhos (PRESTES, 2005), (FONSECA, 2008b) e (MUNHOZ, 2009).

3.3 VPE-GM

Concebido segundo os fundamentos da programação visual, o VPE-GM caracteriza-se pela facilidade de uso e exploração do paralelismo explícito, sendo desenvolvido para prover interface gráfica na simulação de algoritmos paralelos e distribuídos da Computação Científica.

Implementado com base na concepção livre e multi-plataforma na linguagem de programação Python, integra duas ferramentas, o Editor Gráfico de Processos e o Editor Gráfico de Memória, as quais viabilizam o acesso do usuário ao ambiente, incluindo as correspondentes comunicações, e que se integram ao ambiente de execução VirD-GM.

Na Figura 3.2, tem-se um esquema do ambiente, com especificação dos módulos e das comunicações entre o Editor de Processos e o Editor de Memórias, incluindo módulo de controle e execução relacionados com a VirD-GM.

A comunicação entre os módulos do VPE-GM com o ambiente de execução se dá através da troca de arquivos, onde um arquivo, expresso através da linguagem XML (*eXtensible Markup Language*), é gerado para representar os dados de cada módulo, conforme visualiza-se na Figura 3.2 na parte inferior.

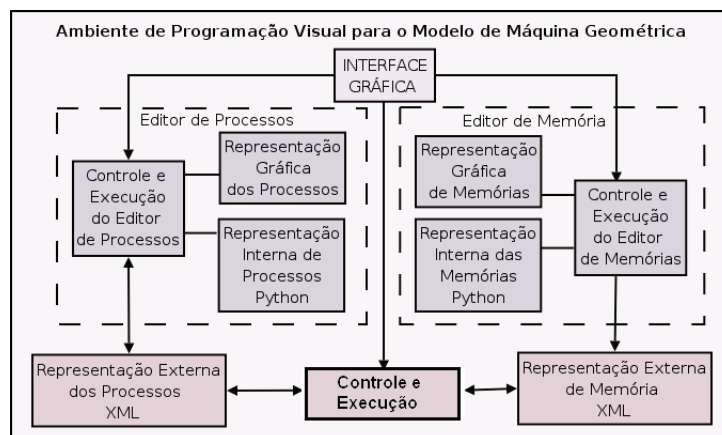


Figura 3.2: Componentes da arquitetura VirD-GM (FONSECA, 2008a).

3.3.1 Editor Gráfico de Processos

O editor de processos é a ferramenta gráfica responsável pela criação e edição dos programas a serem executados no VirD-GM. A partir de construtores básicos (componentes da interface gráfica) é permitido a definição do fluxo de programa que deverá ser executado.

Através de uma estrutura denominada envelope, permite-se o agrupamento de processos elementares, construtores (barreiras de sincronização, construtores de iteração, testes, etc.) ou ainda outros envelopes, possibilitando além da exploração de recursão, a construção de objetos com maior complexidade.

Os componentes do Editor de Processos, assim como exemplos da utilização dos mesmos, podem ser visualizados na Figura 3.3, onde tem-se representação para: (A) processos sequenciais; (B) processos paralelos; (C) somas não-determinísticas; (D) somas determinísticas; e (E) processos iterativos (abordagem sequencial e paralela).

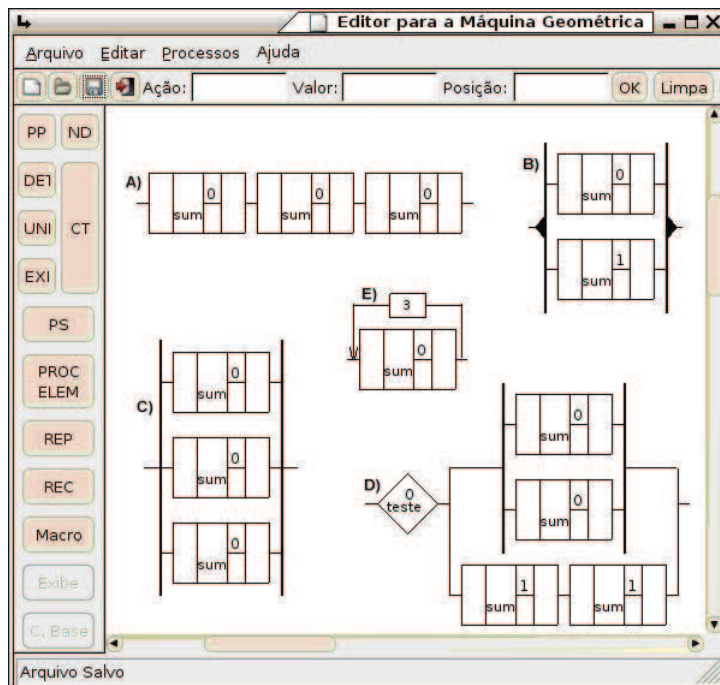


Figura 3.3: Editor Gráfico de Processos (FONSECA, 2008a).

Os objetos gráficos também são descritos por arquivos gerados na linguagem XML. A coleção destes arquivos constitui a Biblioteca de Processos, cujo acesso é realizado pelo módulo de controle e execução de processos, o qual permite a comunicação entre as diferentes representações dos objetos gráficos.

No contexto dos construtores, salienta-se que produto refere-se a uma operação entre dois ou mais processos com execução todos processos envolvidos no produto. Por outro lado, a soma refere-se a dois ou mais processos onde a execução deve considerar, ou o primeiro ou o segundo, mas não ambos.

A seguir, estão elencadas as principais construções obtidas a partir de processos elementares, associados às diversas posições de memória:

- **Produtos sequenciais:** consistem em processos gerados pelo construtor sequencial, responsável pela composição de processos no modelo GM, onde as computações

são executadas de acordo com a ordem de construção do processo no VPE-GM;

- **Produtos paralelos:** esse construtores determinam transformações aplicadas a diferentes posições de memória, mas executadas concorrentemente, caracterizando as computações síncronas e o paralelismo de dados. É utilizado quando as computações não possuem dependências de dados entre si;
- **Somas não-determinísticas:** caracterizado pela escolha aleatória do processo a ser executado, onde dois processos elementares d^k e e^k são mutuamente exclusivos pois a execução de ambos altera o valor da mesma variável associada à k -ésima posição de memória;
- **Processos iterativos:** caracterizam-se pela iteração de operações ou de posições de memória, as quais podem ocorrer concorrente ou sequencialmente, definindo duas extensões: o Construtor Iterativo Paralelo e o Construtor Iterativo Sequencial, respectivamente. No primeiro caso, contempla-se a iteração de processos paralelos sem dependências de dados entre sucessivas iterações. Já o segundo, considera a iteração de processos que possuem dependência de dados e não podem ser executados concorrentemente;
- **Somas determinísticas:** são os construtores responsáveis pela definição de estruturas de controle no modelo GM. Através dele é possível definir desvios condicionais no fluxo de execução de uma aplicação na GM (configuradas no módulo GM-App). Este construtor é definido por três parâmetros: um operador *Teste* e dois operadores *Processos* (Processo 1 e Processo 2).

Além destes processos e construtores também são definidos testes computacionais: teste determinístico, teste universal e teste existencial. A partir destes testes é possível realizar escolhas, as quais redirecionam o fluxo de dados durante a computação (FONSECA, 2008a).

3.3.2 Editor Gráfico de Memória

O editor de memória é a ferramenta gráfica responsável pela criação e edição da memória que será utilizada no momento da execução no VirD-GM. Além da escolha da estrutura adequada ao desenvolvimento da aplicação e a inicialização dos valores, permite a visualização das transformações da memória nas computações geradas, possibilitando a validação dos parâmetros que definem os estados e correspondentes processos.

Pelo módulo de Execução e Controle do Editor de Memórias, os dados são exportados para o módulo de Representação Externa de Memórias, responsável pela geração e armazenamento das correspondentes descrições em arquivos gerados utilizando a linguagem XML, assim como ocorre no editor de processos.

A interface do Editor de Memória do ambiente VPE-GM pode ser visualizada na Figura 3.4, mostrando os parâmetros de configuração e estados computacionais.

O módulo de controle e execução da VirD-GM importa os arquivos de outras bibliotecas provendo os recursos (métodos, técnicas, funções e procedimentos) necessários para posterior execução na VirD-GM. O ambiente VPE-GM também permite que os resultados parciais e finais dessa execução sejam disponibilizados para o usuário pela interface da VPE-GM.

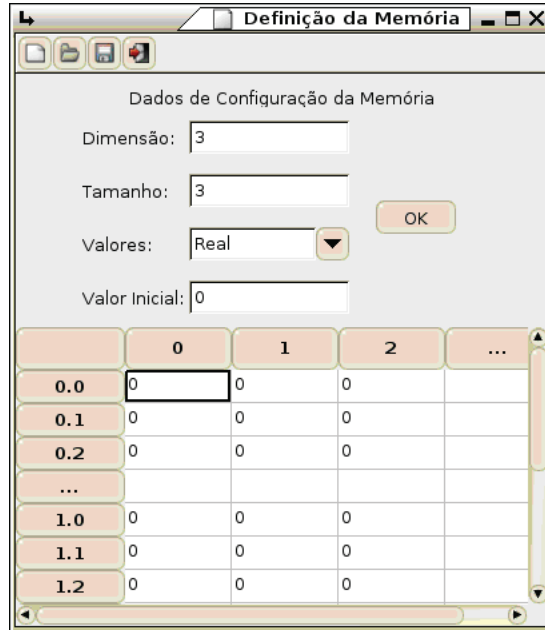


Figura 3.4: Editor Gráfico de Memória (FONSECA, 2008a).

3.4 VirD-GM

O ambiente VirD-GM executa exclusivamente as aplicações desenvolvidas no VPE-GM. Estas aplicações são, por natureza, paralelizáveis e direcionadas ao estudo de algoritmos paralelos para a Computação Científica (FONSECA, 2008a). A construção do VirD-GM integra esforços de dois grupos de pesquisa, o de Fundamentos Matemáticos da Computação (GFMC) e o de Processamento Paralelo e Distribuído (G3PD) do PPGINF/UCPEL.

O ambiente de execução VirD-GM permite que se utilize o *middleware* EXEHDA (YAMIN, 2004a) (*Execution Environment for High Distributed Applications*) para suporte aos mecanismos de distribuição, comunicação e gerência das computações distribuídas, já a gerência da execução em memória compartilhada é feita pelo próprio VirD-GM.

Na perspectiva da D-GM, os recursos de infra-estrutura física estão mapeados por três abstrações decorrentes da organização do EXEHDA:

- VirD-cel: constitui a área de atuação de uma VirD-base e dos seus VirD-nodos. As computações distribuídas e/ou paralelas acontecem no VirD-Cel concebidas seguindo o modelo D-GM;
- VirD-base: equipamento responsável pela gerência da execução da D-GM como um todo ou ainda a infra-estrutura para operação dos VirD-nodos e incluindo tanto o escalonamento das tarefas quanto pelo controle do fluxo de dados;
- VirD-nodo: são os equipamentos de processamento disponíveis, sendo responsáveis pela execução das computações da D-GM.

A Figura 3.5 apresenta o diagrama de sequencia do D-GM, mostrando a troca de mensagens entre os componentes.

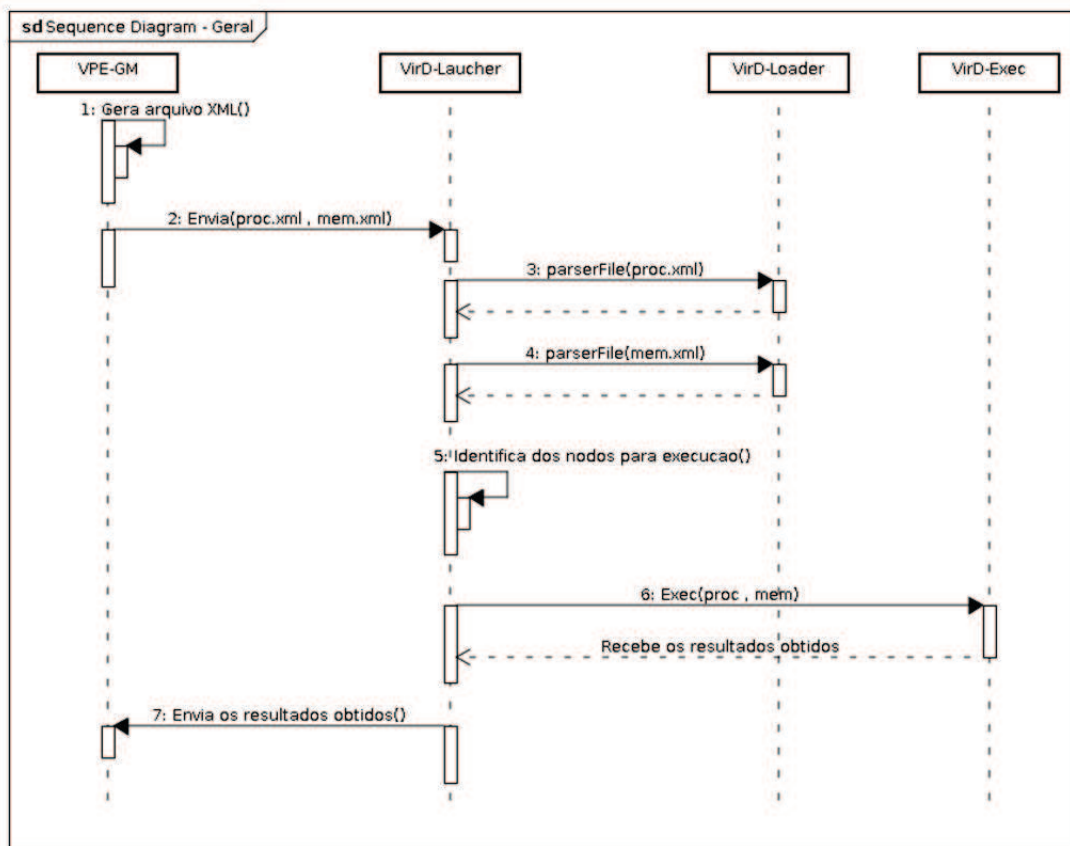


Figura 3.5: Diagrama de sequencia do projeto D-GM.

O VirD-GM se mostra apropriado para a correta execução do protótipo *p-iCone*, pois apresenta todos os mecanismos de expressão para o paralelismo e sincronização, necessários para a implementação da solução para o problema de cálculo de relevo não uniformes.

O *software p-iCone* consiste em um estudo de caso para validação do projeto D-GM, pela manipulação de estruturas matriciais tridimensionais. A paralelização pode ser considerada como um meio de validação do ambiente integrando os construtores originais (FONSECA, 2008a) e os construtores introduzidos em (MUNHOZ, 2009).

3.5 Desenvolvimento e Execução de Aplicações no Ambiente D-GM

Nesta Seção é apresentado o esquema da utilização do ambiente de desenvolvimento e execução no ambiente D-GM, com ênfase no conceito de envelope, caracterizando a estrutura que encapsula os processos elementares, construtores e outros envelopes, e ainda possibilita construções recursivas.

A seguir, descreve-se os elementos apresentados na Figura 3.6, salientando as respectivas características e as associações com os outros elementos.

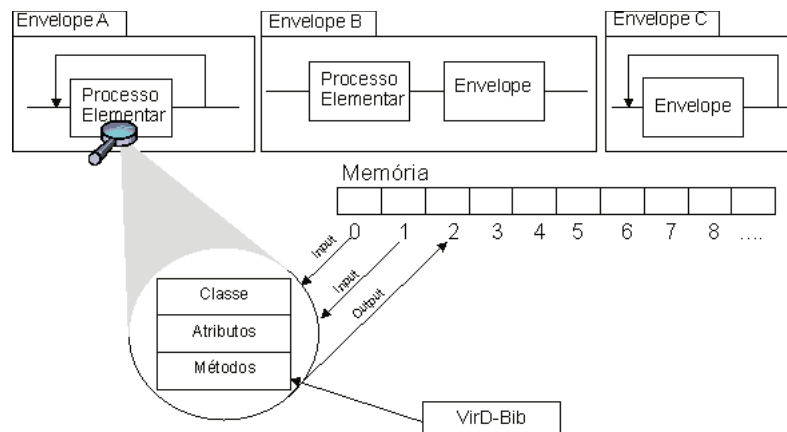


Figura 3.6: Organização do ambiente D-GM.

- Envelope A: representando uma iteração sobre um processo elementar, onde esta iteração pode ser executada sequencial ou paralelamente, de acordo com o tipo de construtor utilizado.
- Envelope B: representando a aplicação do produto sequencial, resultando na sequencialização definida por um Processo Elementar, seguida de outro envelope.
- Envelope C: representando a aplicação de um construtor de iteração, o qual pode ser sequencial (*for*) ou paralelo (*parfor*). Além dos exemplos dos envelopes A, B e C, podem ser criadas outras configurações de execução, utilizando os recursos dos construtores, associados à processos elementares e envelopes.
- Memória: modelada por uma estrutura de listas, incluindo listas multi-dimensionais que podem ser utilizadas. Cada valor da lista de dados, correlacionando sua tipagem com a lista de declaração das variáveis.

Outro recurso da memória que pode ser utilizado, é a passagem de um arquivo XML com dados para importação da memória. Neste contexto, uma posição da memória pode conter, um tipo matriz.

O acesso para leitura e gravação da memória é feito através de indexação da posição na memória, sendo que se ocorrer conflito de acesso à memória, o ambiente de execução identifica e retorna ao usuário.

- **Processo Elementar:** representado por uma classe em Java, deve conter pelo menos um método, denominado “ap” e tendo, obrigatoriamente, os parâmetros de entrada (*Integer value*, *Integer[] input*, *Integer output*, *Integer iterator*, *VirDMemory memory*, *ObjectOutputStream oos*). Além do método “app”, outros métodos podem ser implementados na classe de um processo elementar, contanto que sejam estáticos, ou seja, não necessitam que seja instanciado um objeto para possibilitar a utilização do método.
- **VirDBib:** consiste na biblioteca de classes e métodos que estão disponíveis para serem utilizados na aplicação. Além dos métodos nativos da plataforma, também métodos implementados de forma estática nas classe da aplicação podem ser utilizados.

3.5.1 Parâmetros de Execução

Na Figura 3.7 estão representados os sistemas que geram os arquivos de descrição das aplicações, juntamente com a representação da máquina virtual Java (JVM).

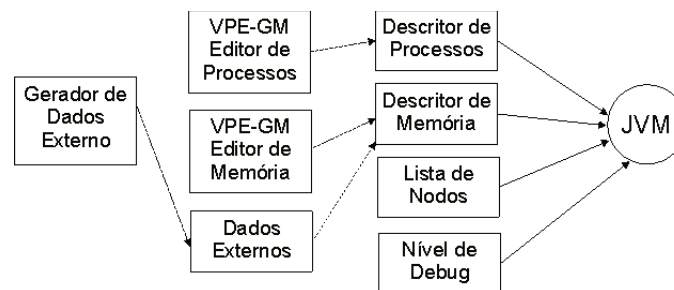


Figura 3.7: Parâmetros para execução do D-GM.

Primeiramente, tem-se o Editor de Processos do Ambiente VPE-GM, o qual é responsável pela geração de um arquivo na linguagem XML que contém as informações dos processos, envelopes e construtores, bem como a sua ordem de execução e os mecanismos de sincronização.

O Editor de Memória do Ambiente VPE-GM, é responsável pela definição da estrutura da memória, tipos de dados, e os valores iniciais das posições da memória. Assim como o Editor de Processos, ele gera um arquivo XML para representação da memória. Embora os valores das posições de memória sejam inicializados no Editor, o ambiente VirD-GM ainda possibilita a passagem de arquivos, como valores de uma posição da memória. Durante a execução de uma aplicação, o D-GM permite importar outros arquivos escritos de maneira estruturada em XML, os quais podem ser gerados por um outro sistema, externo à VPE-GM.

A lista de nodos refere-se a uma lista de endereços de *hosts* ou endereços *IP* (*Internet Protocol*) seguida da quantidade de instâncias da VirD-GM por equipamento, no formato (127.0.0.1:2) e com os nodos separados por vírgula.

Já o nível de *Debug* refere-se a um valor inteiro que indica qual dos três níveis básicos de mensagens será utilizado. As mensagens são impressas no momento da execução; (i) menor valor 0, imprime somente os resultados, (ii) o nível 1, imprime alguns dados do ambiente de execução; e (iii) o nível 2, que imprime as mensagens definidas nas classes e métodos da aplicação. Além dos níveis básicos, o ambiente suporta a definição de novos níveis para a aplicação.

Por fim, a JVM recebe todos os argumentos e promove a execução da aplicação, inclusive na execução nos nodos clientes.

3.6 Principais Construtores

Nesta seção apresentam-se os construtores utilizados no projeto D-GM juntamente com o respectivo detalhamento.

3.6.1 Construtor Iterativo

A implementação dos construtores de processos está baseada na definição recursiva da estrutura de envelope e de processo elementar. De acordo com as abstrações do modelo D-GM, o Construtor Iterativo pode ser aplicado sobre outro processo indefinido pela estrutura de dado do tipo envelope, incluindo a composição com outros construtores. Este construtor tem como principal característica a repetição de operações. O número de repetições do Construtor Iterativo é variável e definido por um parâmetro de controle no momento de sua geração no VPE-GM.

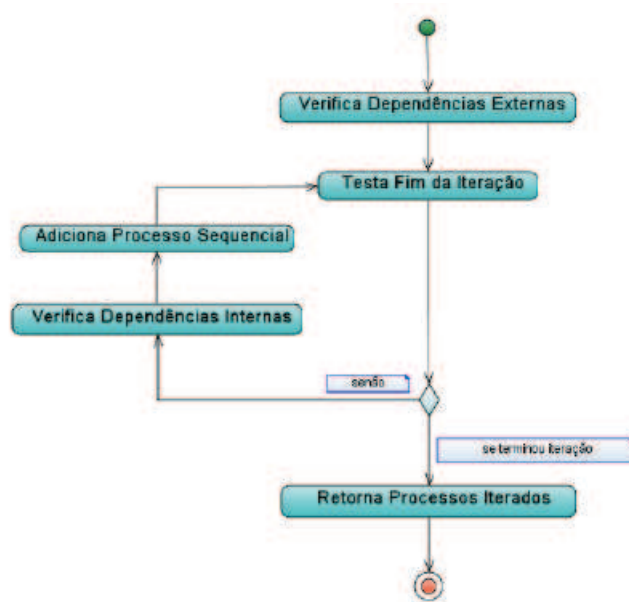


Figura 3.8: Construtor iterativo sequencial.

3.6.2 Construtor Iterativo Paralelo

A implementação do Construtor Iterativo foi realizada no módulo *VirD-Loader*. Durante a leitura do arquivo XML, caso encontrado um Construtor Iterativo, o processo descendente é replicado N vezes, onde N consiste no parâmetro definindo o número de iterações do Construtor Iterativo. Além disso, é necessário verificar se o Construtor Iterativo é sequencial ou paralelo.

No caso de Construtor Iterativo Sequencial (*for_{atgm}*), além de replicar os processo e inserir na lista, é necessário resolver as dependências de dados entre os processos gerados.

O Construtor Iterativo Paralelo (*pfor_{atgm}*) tem o funcionamento análogo ao Construtor Iterativo Sequencial, exceto pela etapa de verificação de dependências internas. Na Figura 3.9, mostra-se um diagrama que descreve o funcionamento do Construtor Iterativo Sequencial e do Construtor Iterativo Paralelo respectivamente, sendo, este último caracterizado pela não dependência de dados entre as iterações, ou seja, cada iteração pode ser executada concorrentemente com as demais iterações.

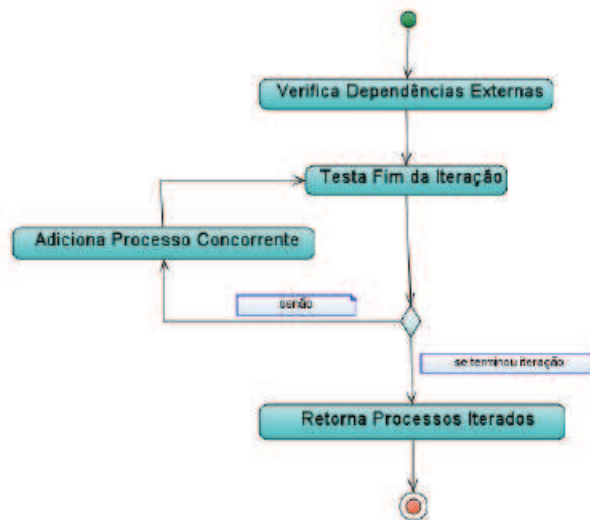


Figura 3.9: Construtor iterativo paralelo.

O Construtor Iterativo Paralelo verifica as dependências externas, identificando os processos que precisam ser finalizados para início da corrente iteração. Após, testa a finalização da iteração pela verificação do número de iterações do processo. Assim, enquanto o número de iterações não está satisfeito, adiciona o processo a iteração corrente usando uma estrutura de lista de armazenamento temporário, já com as dependências de dados resolvidas. E quando o número de iterações do processo finaliza, os processos gerados na iteração são enviados para a lista de processos de execução.

3.6.3 Construtor Produto Sequencial

O Construtor Produto Sequencial é responsável pela composição de processos no modelo GM, onde as computações são executadas de acordo com a ordem de construção do processo no VPE-GM. Estas instruções são executadas sequencialmente, assim não necessitando de validação no acesso à memória.

3.7 Exploração do Paralelismo no D-GM com Processadores Multi-Core

No contexto do projeto D-GM, além dos sistemas paralelos distribuídos, estão sendo estudadas e testadas as arquiteturas *multi-core*, as quais apresentam a capacidade de exploração de paralelismo dentro de um único *chip*, apresentando ganhos de desempenho baseado na distribuição de várias tarefas para o mesmo processador.

As unidades centrais de processamento (UCP multinúcleo) ou (*CPU multi-core*) são unidades caracterizadas por dois ou mais núcleos processadores executando de forma independente, trabalhando agrupados em um mesmo *chip* e tendo uma memória *cache* compartilhada para comunicação entre eles.

Esta organização permite que um CPU multi-core execute tarefas em paralelismo real, diferentemente dos processadores monoprocesados com suporte a *multi-thread*, os quais simulam o paralelismo através de técnicas avançadas de execução, porém continuando com a execução de uma linha de execução por vez.

Atualmente, a quantidade de núcleos por *chip* está cada vez maior. Empresas como a Intel, por exemplo, estão pesquisando e desenvolvendo processadores com capacidade semelhante ou superior de muitos dos antigos super-computadores. Um exemplo deste desenvolvimento é o *Teraflops Research Chip* desenvolvido pela Intel com 80 núcleos e capacidade de processamento de aproximadamente 1,6 *teraflops*, e ainda com baixo consumo de energia, aproximadamente 62 Watts. Tal consumo, mostra-se mais significativo se comparado ao *ASCI Red*, o primeiro computador a executar um *teraflop* em 1996. Este *cluster* utiliza perto de 10.000 processadores *Pentium® Pro*, executando a 200MHz e consumindo 500kW, além de uma grande quantidade de energia para resfriamento (RAT-TNER, 2009).

Isto demonstra que pesquisas sobre processadores *multi-core* estão sendo fortemente incentivadas e apresentam-se como tecnologia consolidada pelo mercado de processadores, tendo larga utilização nos processadores de uso geral vendidos atualmente. Com esta convergência para os processadores *multi-core*, outro paradigma se torna presente: Como suprir os n núcleos com processos para se obter o máximo desempenho do CPU?

A resposta a esta pergunta está sendo buscada através de inúmeras estratégias de execução, e sendo um tema fortemente explorado pela comunidade científica.

Os programas existentes, desenvolvidos sob a ótica de programação sequencial, dificilmente obterão ganho significativo de desempenho se submetidos a um *upgrade*, de um processador para um multi-core de mesmo *clock*.

Além de um sistema operacional que dê suporte a estes processadores, as aplicações necessitam ter sido desenvolvidas já considerando o paradigma de processamento paralelo. As aplicações concorrentes devem ser modeladas e desenvolvidas baseadas em um modelo fundamentado na teoria da concorrência.

Por outro lado, uma interessante característica associada aos processadores *multi-core* no consumo de energia, diz respeito a economia associada aos processadores ociosos, os quais são desligados automaticamente, não consumindo energia e não gerando aquecimento. No fator aquecimento, os processadores buscam distribuir os processos para núcleos afastados entre si, para que não se concentrem diversos núcleos ativos próximos, otimizando a dissipação do calor.

A linguagem Java possibilita a execução das aplicações de modo paralelo, direta-

mente nos processadores *multi-core* através do uso de *threads*. As *threads* na JVM são enviadas de forma independente para o sistema operacional, o qual geralmente efetua o escalonamento de forma a preencher os núcleos do processador. Assim qualquer programa que utilize *threads* de forma paralelizada, sempre que possível, e sem a necessidade de um *middleware* que gerencie o paralelismo. Além dessa estratégia, também pode-se considerar a simulação de vários nodos, alocando um núcleo para cada nodo, e desta forma também faz-se o uso da exploração do paralelismo nos processadores multinúcleo.

O ambiente D-GM, por ser fundamentado na teoria da concorrência, é um ambiente de desenvolvimento e execução de aplicações paralelas, capaz de explorar os recursos disponibilizados pelos processadores multinúcleos, assim como os ambientes de grade ou aglomerado de computadores.

Contudo, o atual foco de desenvolvimento de *softwares* de alta complexidade ou alto custo computacional está na exploração do paralelismo, tanto para processadores *multi-core*, quanto para sistemas paralelos distribuídos. Ambas abordagens podem ser exploradas atualmente pelo ambiente D-GM, visando o máximo desempenho em sistemas multiprocessados.

O desenvolvimento atual do projeto D-GM está considerando o estudo e a análise destas abordagens para a definição da estratégia a ser utilizada pelo VirD-GM para a exploração do paralelismo nos processadores multinúcleo.

3.8 Conclusão

Este capítulo discorreu sobre o ambiente D-GM, considerando os requisitos a serem satisfeitos quando da proposta de atender as necessidades para criação, gerenciamento e processamento de aplicações paralelas e/ou distribuídas da computação científica. Tais requisitos fundamentam-se no modelo GM, influenciando modelagem e implementação dos editores do VPE-GM, na determinação dos construtores de processos e na organização de seus serviços e no conjunto de funcionalidades a serem disponibilizadas pelo ambiente de execução VirD-GM. Apresenta-se próprio para modelagem e execução da aplicação *SisA*³, pela compatibilidade da metodologia com as abstrações do modelo GM.

4 ARQUITETURA SIS^{A3}

Neste capítulo, a modelagem dos componentes que estruturam a arquitetura do Sistema Automatizado de Auditoria em Armazéns (*SisA³*), é apresentada. São discutidas, as estruturas de *software* e de *hardware*, além dos sistemas utilizados para prover todas as abstrações necessárias para a implementação dos protótipos do *SisA³*.

Conforme apresentado na Figura 4.1, a arquitetura proposta integra três componentes, descritos nas seções que seguem:

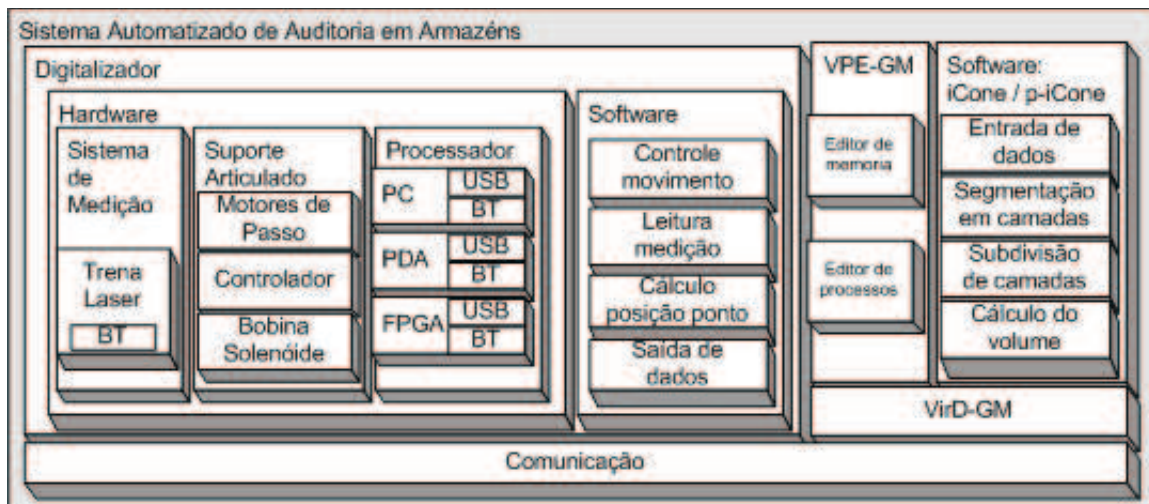


Figura 4.1: Organização da Arquitetura do *SisA³*.

4.1 Sistema Dig-Sis^{A3}

O sistema *Dig-SisA³* consiste na modelagem do equipamento digitalizador responsável por efetuar medições no relevo e geração da matriz de representação tridimensional. Este equipamento é composto por dois sistemas, um de *hardware* e outro de *software*:

O sistema de *hardware* do *DigSisA³* é responsável pela medição dos pontos, integrando 3 equipamentos, o sistema de medição, o suporte articulado que movimenta o sistema de medição e, por fim, o processador que é responsável pela execução do software de controle do *DigSisA³*.

1. Sistema de medição: composto por um sensor ótico a laser e um sistema de espe-

lhos, que articulam a movimentação e possibilitam a varredura linear sobre o relevo, enviando as medições para o processador via rede sem fio por *bluetooth*.

2. Suporte articulado: compreende o sistema eletromecânico/ótico que efetua a movimentação angular de um espelho para direcionamento do laser do sensor ótico. É composto por motores de passo e suas respectivas interfaces de controle, responsável pela movimentação precisa do espelho.
3. Processador: caracterizado por um sistema computacional de uso geral, podendo ser um *laptop*, PDA, *tabletPC*, ou ainda um *smartphone*, possuindo interface de comunicação *bluetooth* e USB (para comunicação do controlador de motor de passo).

O sistema de *software* do *DigSisA³* é responsável pelo controle das medições e exportação da matriz tridimensional. Este processo de controle é obtido pela integração de quatro módulos:

1. Módulo de controle de movimento: responsável pela comunicação com a interface de controle do motor de passo para posicionamento do laser.
2. Módulo de leitura de medição: responsável pela comunicação com o sensor ótico, solicitando as medições e recebendo os resultados.
3. Módulo de cálculo da posição do ponto: capturando os valores dos módulos de leitura de medição e de controle de movimento, além dos parâmetros pré-definidos no início da medição, calculando também a posição espacial dos pontos.
4. Módulo de saída: efetua a representação externa da matriz tridimensional capturada e a armazena em um arquivo XML.

A modelagem do processo de aquisição das coordenadas dos pontos da matriz de entrada de dados pode ser visualizada na Figura 4.2, onde apresenta-se o processo de parametrização para modelagem do armazém, e na medição dos pontos do relevo, em cada camada. O funcionamento do *DigSisA³* depende da integração entre o sistema de *software* e o sistema de hardware, a qual ocorre através de um *driver* de controle.

Mais detalhadamente, na Figura 4.3 descreve-se o relacionamento entre o drive de controle, o equipamento processador do *DigSisA³* e a trena eletrônica.

A integração dos diversos sistemas apresentados viabiliza a aquisição das coordenadas dos pontos que representam o relevo e possibilitam o cálculo computacional do volume, sem a necessidade de contato com os produtos armazenados. Assim disponibilizando uma ferramenta para auxiliar empresas e órgãos de controle do poder público ou privado na realização de auditorias de estoque.

Além de possibilitar o cálculo da quantidade de produtos, o *SisA³* tem como premissas a precisão do cálculo, tempo de resposta da auditoria e o custo de fabricação, de forma que o sistema seja acessível aos diversos perfis de produtores e inclusive à grandes indústrias.

A descrição completa do sistema *DigSisA³* esta descrita no Capítulo 6, onde cada modulo é apresentado isoladamente.

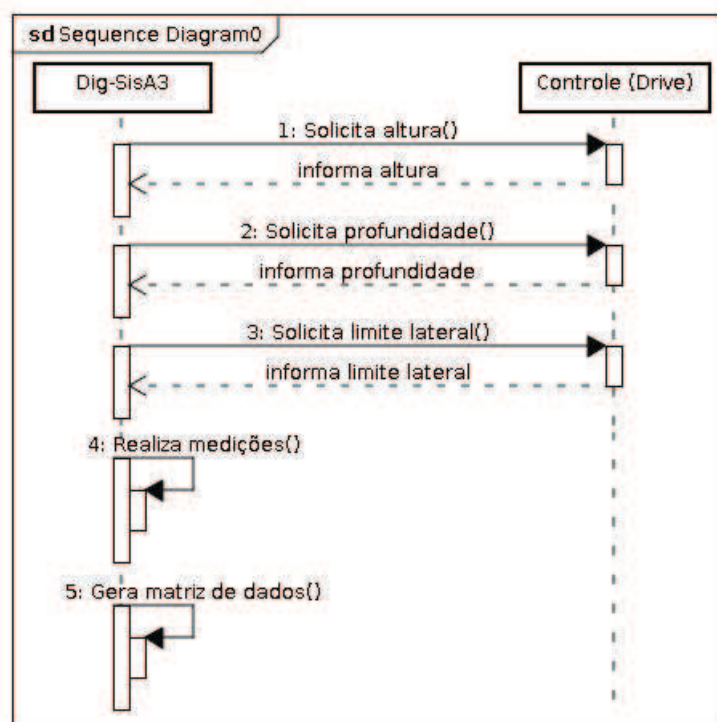


Figura 4.2: Diagrama de Sequência do *DigSisA³*.

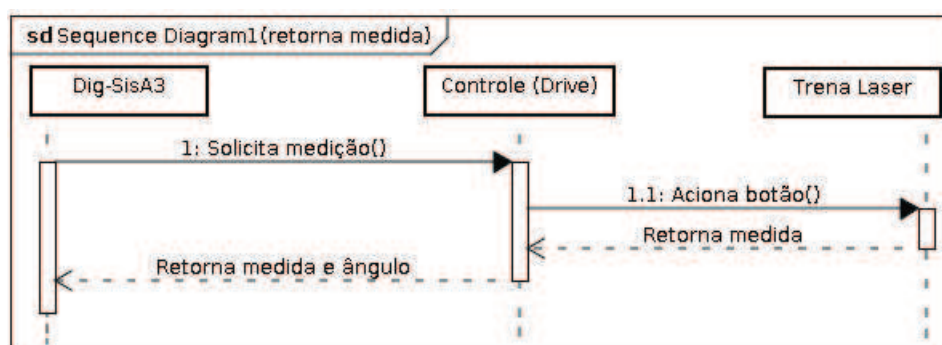


Figura 4.3: Diagrama de Sequência de medição do *DigSisA³*.

4.2 Suporte ao Desenvolvimento da Aplicação p-iCone

O Ambiente D-GM (*Distributed Geometric Machine*) foi usado para suporte ao desenvolvimento e execução de aplicações científicas, considerando a versão distribuída do modelo GM.

No contexto do projeto D-GM, o protótipo *p-iCone* é um estudo de caso, validando a arquitetura de *software* do ambiente VirD-GM, assim como seus construtores e estruturas de controle. A descrição detalhada do ambiente D-GM foi apresentada no Capítulo 3.

4.3 Métodos iCone e p-iCone

A metodologia que viabiliza o desenvolvimento do protótipo *iCone* aplica o cálculo numérico do volume identificado a partir de uma matriz de representação, onde aproximações de medições intermediárias são obtidas através da interpolação de pontos, utilizando *splines* polinomiais cúbicas. A partir do acesso aos pontos desta matriz, torna-se possível a modelagem do relevo e a aplicação de algoritmos iterativos para simulação do cálculo do volume, cuja computação considera os parâmetros relativos a distâncias, ângulos, áreas e volumes parciais para obtenção do resultado final.

O *iCone* foi desenvolvido sob a ótica de *software* para desempenhar a tarefa de cálculo do volume estocado, pela aplicação de método numérico. Neste contexto, na modelagem do protótipo, foram consideradas quatro etapas, definidas a seguir:

1. Entrada de Dados: esta etapa compreende a interpretação dos dados do arquivo de representação tridimensional do relevo, gerados pelas sucessivas leituras do *DigSisA*³;
2. Segmentação em camadas: os dados do relevo reconhecido na etapa anterior são segmentados em camadas. As camadas determinam as fatias do relevo que correspondem ao intervalo entre duas sucessivas medições. A profundidade de cada fatia é a distância entre camadas. Nesta etapa, também são calculados os coeficientes dos polinômios de interpolação, sobre os pontos da matriz.
3. Subdivisão de camadas e cálculo dos volumes parciais: As camadas geradas pela segmentação são subdivididas, definindo fatias menores, pela aplicação da interpolação numérica polinomial cúbica por *Splines*. Na sequência calcula-se o volume de cada fatia.
4. Somatório dos volumes parciais: Efetua-se o somatório dos volumes parciais, que gera o resultado final e exporta o resultado para a interface de visualização da aplicação.

Na Figura 4.4 apresenta-se o diagrama de sequência do *iCone*, onde é representada a troca de mensagens juntamente com a ordem de execução.

A descrição completa de todos os componentes do sistema *iCone* está apresentada no Capítulo 5.

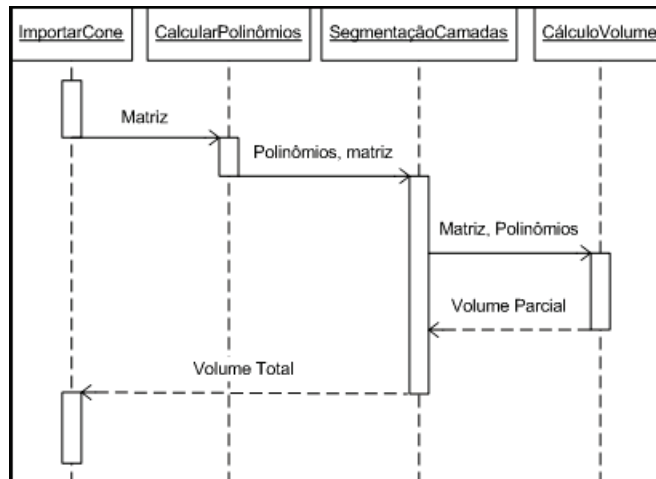


Figura 4.4: Diagrama de Sequência do Protótipo *iCone*.

4.4 Conclusão

Este capítulo apresentou a arquitetura do *SisA³* - Sistema Automatizado de Auditoria em Armazéns, o qual consiste em uma proposta para viabilizar a avaliação do volume de estoques da produção agrícola armazenada em silos. A visão geral da modelagem do sistema e da organização de seus principais componentes foi brevemente apresentada.

5 ICONE-SISA³ - CALCULO DE VOLUME DE RELEVOS NÃO-UNIFORMES

Os protótipos *iCone* e *p-iCone* apresentam o funcionamento interno semelhantes, diferenciando-se pela estratégia de acesso a memória e pela forma de iniciar os processos que podem ser sequenciais ou concorrentes.

Parte-se da apresentação da abordagem sequencial (vide Seção 5.1), onde são detalhados os quatro estágios do algoritmo, contemplando todo seu funcionamento interno. Posteriormente, é discutida a abordagem paralela e/ou distribuída, assim como suas estruturas de controle do paralelismo.

5.1 Protótipo *iCone*

O protótipo *iCone* implementa a metodologia *iCone* de integração do volume de subfatias e, nesta seção, a descrição detalhada do algoritmo é apresentada.

O algoritmo considera quatro etapas para o cálculo do volume:

- **Processamento da entrada:** Efetua a leitura da matriz de entrada, cujos dados são convertidos para o formato de uma matriz tridimensional de pontos.
- **Segmentação da matriz:** Particionando a matriz para geração de fatias, as quais são determinadas por sucessivos intervalos de medições(fatias).
- **Divisão de fatias:** Subdivide-se cada fatia em subfatias intermediárias, através da aplicação de interpolação numérica por *splines* polinomiais cúbicas. Calcula-se a área de cada face de uma subfatia e multiplica-se pela sua profundidade, afim de obter o volume.
- **Geração do Resultado Final:** Efetuando o somatório dos volumes das diversas fatias iniciais e subfatias, apresentando como retorno, o resultado final.

Segue a descrição de cada etapa do algoritmo utilizado no protótipo *iCone*.

5.1.1 Processamento da Entrada

Partindo de uma matriz tridimensional que representa o relevo de produtos, obtida pelo *DigSisA³*, a qual serve como de entrada de dados para o *iCone*, contendo as coordenadas dos pontos que definem o relevo, calculam-se os polinômios de interpolação, cujos coeficientes são armazenados em nova estrutura matricial. Desta forma, os polinômios de interpolação referentes a cada fatia, correspondem às linhas desta estrutura matricial.

5.1.2 Segmentação da Matriz

Cada fatia é definida pelo intervalo de pontos localizado entre as medições de $Z(i)$ e $Z(i + 1)$, cujos os respectivos polinômios de interpolação já foram previamente calculados na etapa de Processamento da Entrada.

Nas medições dos pontos, em cada iteração, os valores para $Z(i)$, com i indicando o índice do passo de medição, são mantidos constantes. A partir desta etapa de segmentação da matriz, cada fatia pode ser processada de forma autônoma, deixando de existir dependências de dados.

5.1.3 Divisão da Matriz de Pontos

Além do cálculo do volume das camadas, esta etapa apresenta estratégias para melhoria na precisão e controle de erros do algoritmo.

O processo da divisão de fatias em subfatias, devido a não existência da dependência de dados, pode ser executada tanto sequencialmente quanto paralelamente, dependendo somente da estratégia adotada para execução.

Pela forma como se calcula o volume, considerando a segmentação em fatias, ocorre a inconveniente formação de degraus na modelagem do relevo, conforme é apresentado na Figura 5.1.

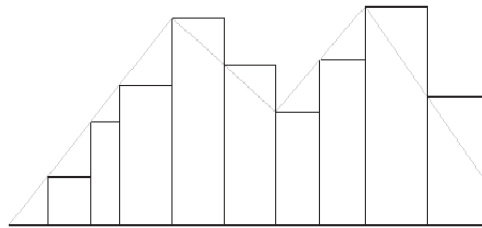


Figura 5.1: Degraus formados pela aproximação da altura

Para se reduzir o efeito computacional desta deformação, a quantidade de fatias é alterada pela aplicação da interpolação polinomial cúbica, desta forma, obtendo a construção de fatias intermediárias. Assim, tem-se a geração de nova matriz, com fatias de tamanho padronizado, as quais serão utilizadas a partir deste momento para o cálculo de volume.

As dimensões obtidas através da interpolação polinomial por *splines*, são aproximações das dimensões reais, mesmo com uma quantidade reduzida de pontos coletados, minimizando os erros no cálculo do modelo matemático.

Na etapa da divisão das fatias, faz-se a inserção de camadas na matriz através da interpolação entre os pontos reais, criando pontos aproximados na direção do eixo Z . Desta forma, obtendo uma nova matriz com uma quantidade elevada de pontos e, assim, chegando a um valor mais próximo do valor real do volume.

O método que implementa o cálculo dos polinômios de interpolação e cálculo da posição dos pontos pode ser visualizado nas figuras 5.2 e 5.3, respectivamente:

```

public static Cubic[] calcNaturalCubic(Double [] x) {
    Integer n = x.length - 1;
    Double[] gamma = new Double[n+1];
    Double[] delta = new Double[n+1];
    Double[] D = new Double[n+1];
    int i;
    /* We solve the equation
    
$$\begin{bmatrix} 2 & 1 & & & \\ 1 & 4 & 1 & & \\ & 1 & 4 & 1 & \\ & & & \dots & \\ & & & & 1 & 4 & 1 \\ & & & & & 1 & 2 \end{bmatrix} \begin{bmatrix} D[0] \\ D[1] \\ \vdots \\ \vdots \\ D[n] \end{bmatrix} = \begin{bmatrix} 3(x[1] - x[0]) \\ 3(x[2] - x[0]) \\ \vdots \\ 3(x[n] - x[n-2]) \\ 3(x[n] - x[n-1]) \end{bmatrix}$$

    by using row operations to convert the matrix to upper triangular
    and then back substitution. The D[i] are the derivatives at the knots.
    */

    gamma[0] = 1.0d/2.0d;
    for ( i = 1; i < n; i++) {
        gamma[i] = 1/(4-gamma[i-1]);
    }
    gamma[n] = 1/(2-gamma[n-1]);

    delta[0] = 3*(x[1]-x[0])*gamma[0];
    for ( i = 1; i < n; i++) {
        delta[i] = (3*(x[i+1]-x[i-1])-delta[i-1])*gamma[i];
    }
    delta[n] = (3*(x[n]-x[n-1])-delta[n-1])*gamma[n];

    D[n] = delta[n];
    for ( i = n-1; i >= 0; i--) {
        D[i] = delta[i] - gamma[i]*D[i+1];
    }

    /* now compute the coefficients of the cubics */
    Cubic[] C = new Cubic[n];
    for ( i = 0; i < n; i++) {
        C[i] = new Cubic((Double)x[i], D[i], 3*(x[i+1] - x[i]) - 2*D[i] - D[i+1],
            2*(x[i] - x[i+1]) + D[i] + D[i+1]);
    }
    return C;
}

```

Figura 5.2: Método de cálculo dos polinômios de Interpolação.

```

public class Cubic implements Serializable {
    private static final long serialVersionUID = 5238634079067377792L;
    Double a,b,c,d; /* a + b*u + c*u^2 +d*u^3 */

    public Cubic(Double a, Double b, Double c, Double d){
        this.a = a;
        this.b = b;
        this.c = c;
        this.d = d;
    }
    /** evaluate cubic */
    public Double eval(Double u) {
        return ((d*u) + c)*u + b)*u + a;
    }
}

```

Figura 5.3: Método de avaliação dos pontos.

Na Figura 5.4, apresenta-se o processo de divisão de uma seção em subseções, desde a entrada dos dados.



Figura 5.4: Processo tratamento dos pontos

Na metodologia *iCone*, a quantidade de subdivisões é definida pelo critério de tamanho máximo de uma sub-fatia após a divisão, parâmetro previamente definido na inicialização do programa. Através deste parâmetro, calcula-se em quantas partes cada intervalo de medição real será dividido, para que não seja ultrapassado o limite definido.

A estratégia de implementação é escolhida pelo usuário, pois dependendo da quantidade de pontos coletados e da precisão desejada o processamento sequencial da execução pode tornar-se, computacionalmente inviável, justificando uma execução paralela e/ou distribuída do algoritmo.

A interpolação dos pontos intermediários é efetuada por um polinômio linear nos eixos *Y* e *Z*. Salienta-se que no eixo *X* é aplicada a interpolação por *splines* cúbicas, o que aproxima as sub-fatias interpoladas do relevo original.

Na Figura 5.5 é apresentado o método que cria as novas camadas do relevo, na matriz.

```

/**
 * Método que re-divide a quantidade de pontos de uma sub-camada em n-novas
 * para aproximar da curva
 * @param beginLayer Vector que indica os pontos da face da sub-camada
 * @param numDiv Número de trapézios ao final da interpolação
 * @return Retorna um Vector com os pontos iniciais e outros que foram interpolados
 */
private Vector makeNewLayer(Vector beginLayer, Double numDiv){
    Double z = (Double) ((Point) beginLayer.get(0)).getZ();
    Vector aux = new Vector();
    //Inicio dos Polinomios para a segunda interpolação
    Cubic [] polinomials2X = new Cubic [beginLayer.size()-1];
    polinomials2X = calcPolinomialsX(divisor(beginLayer, true));
    Cubic [] polinomials2Y = new Cubic [beginLayer.size()-1];
    polinomials2Y = calcPolinomialsY(divisor(beginLayer, false));
    //Fim do cálculo dos polinomios

    //Re-divide aqui

    for (int t = 0; t < beginLayer.size()-1; t++){
        for (int i = 0; i < numDiv; i++){
            Double y = polinomials2Y[t].eval((Double)(i / numDiv));
            Double x = polinomials2X[t].eval((Double)(i / numDiv));

            if (x < 0)
                x = 0.0;
            if (y < 0)
                y = 0.0;
            aux.add(new Point( x, y, z ));
        }
    }
    Double y0 = polinomials2Y[polinomials2Y.length-1].eval(1.0);
    Double x0 = polinomials2X[polinomials2X.length-1].eval(1.0);
    if (x0 < 0)
        x0 = 0.0;
    if (y0 < 0)
        y0 = 0.0;
    aux.add(new Point( x0, y0, z ));

    return aux; //se retornar beginLayer a função de interpolação no cálculo da área é desconsiderada
}

```

Figura 5.5: Método de cálculo das novas camadas.

O cálculo da área é efetuado a partir da face (poligonal) de cada fatia, considerando os dados dos pontos sob a visão bi-dimensional, ignorando as coordenadas do eixo Z . A partir do gráfico formado pelos pontos com coordenadas X e Y , aplica-se a integração trapezoidal para o cálculo da área. A Figura 5.6 mostra uma representação pelo corte feito na matriz tridimensional de pontos, sendo cada ponto identificado com uma seta e o cálculo da área sendo demonstrado na Figura 5.7.

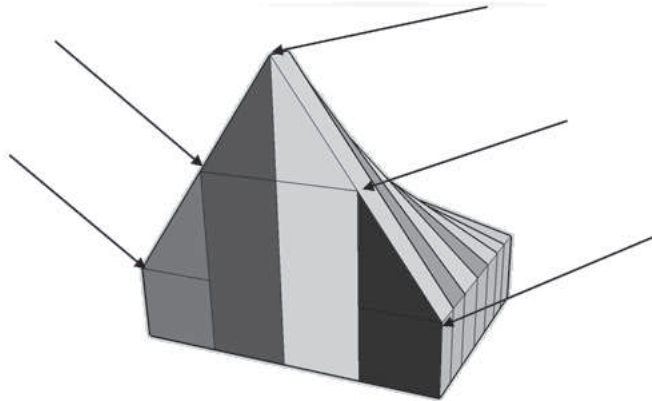


Figura 5.6: Corte feito na matriz de pontos

```

/**
 *
 * @param p0 Ponto inicial do Trapézio
 * @param p1 Ponto final do Trapézio
 * @return Retorna a Área do trapézio
 */
private Double calcArea(Point p0, Point p1){
    Double dx = p1.getX() - p0.getX();
    if(p0.getY() > 0){
        if(p1.getY() > p0.getY()){
            return((((p1.getY() - p0.getY()) * dx) / 2) + (p0.getY() * dx));
        }
        if(p1.getY() < p0.getY()){
            return((((p0.getY() - p1.getY()) * dx) / 2) + (p1.getY() * dx));
        } else{
            return(p0.getY() * dx);
        }
    } else {
        return((p1.getY() * dx) / 2);
    }
}

```

Figura 5.7: Método de cálculo da área de uma face de fatia.

Esta integração trapezoidal, associada à subdivisão das fatias já apresenta uma razoável aproximação do valor real, entretanto, para se obter um valor mais preciso, considera-se uma novo processo complementar nesta etapa do algoritmo.

Devido a uma sessão transversal em um cone apresentar a forma de uma hipérbole, figura a qual caracteriza as seções feitas no relevo de produtos, a nova etapa aplica a interpolação polinomial cúbica por *splines* nos pontos bidimensionais X e Y da face desta

subfatia. Assim, através da aplicação da interpolação por *splines*, permite-se que a quantidade de trapézios identificados na face da sub-fatia seja aumentada, através da divisão dos pontos originais em n novos pontos. Com o aumento da quantidade de trapézios, o erro no cálculo da área diminui.

Este processo pode ser melhor compreendido através da análise das Figuras 5.8 e 5.9. Na primeira, a representação bi-dimensional através de cinco pontos determina a área limitada pelos segmentos lineares, sendo visível a quantidade não considerada, ou seja o erro no método. Na outra figura, a mesma área foi re-dividida em nove pontos, reduzindo significativamente a quantidade não considerada.

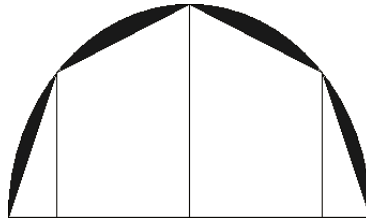


Figura 5.8: Área da face de uma fatia representada por 5 pontos

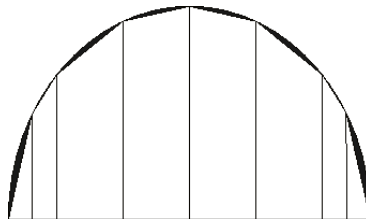


Figura 5.9: Área da face de uma fatia representada por 9 pontos

Como verificado no exemplo da Figura 5.9, foi efetuada a divisão de dois pontos no meio, resultando um novo ponto intermediário, e isto já reduz significativamente o erro desta etapa. Este processo apresenta uma eficiência maior que o processo da divisão em sub-fatias, pois para se obter uma aproximação satisfatória, não é necessário a divisão por uma quantidade excessiva de pontos.

Com o valor da área da face da fatia já conhecido, multiplica-se este valor pela profundidade, obtendo-se o volume parcial referente a uma subfatia.

5.1.4 Geração do Resultado Final

Na última etapa do algoritmo, ocorre o somatório final dos volumes parciais, considerando os resultados de cada fatia, para alcançar o valor do volume total do relevo.

Esta etapa é executada sequencialmente no algoritmo, entretanto nas abordagens paralelas e distribuída, ela serve como barreira de sincronização, de modo que somente será executada após terem sido calculadas todas as etapas anteriores.

5.2 Estratégia Paralela e Distribuída

A paralelização do protótipo *iCone*, denominada *p-iCone*, foi concebida sob duas visões distintas, uma voltada a execução em processadores *multi-core* (memória compar-

tilhada) e outra voltada ao processamento paralelo e distribuído (memória distribuída), ambas possuindo o objetivo de aumentar o desempenho da aplicação.

A primeira prototipação paralela visa a utilização de processadores *multi-core*, onde através da exploração de *threads*, possibilita-se o processamento paralelo em uma máquina de n núcleos, reduzindo o tempo de execução.

Já a segunda prototipação paralelizada do *software iCone*, foi desenvolvida sobre a arquitetura VirD-GM (*Virtual Distributed Geometric Machine Model*), a qual implementa abstrações da versão distribuída do modelo GM. Esta arquitetura apresenta todas as abstrações necessárias para a exploração do paralelismo do protótipo *p-iCone* e apresenta-se como solução para a execução do protótipo em sistemas paralelos e distribuídos.

A execução é gerenciada pelos módulos que compõem o VirD-GM e a configuração de memória e processos é modelada no VPE-GM. A memória de entrada é representada por uma matriz tridimensional com os pontos que devem ser aplicados no algoritmo, onde a posição de saída aguarda o retorno do parâmetro com o valor do volume.

Na Figura 5.10, tem-se:

1. A geração da matriz de entrada é feita por um processo elementar ($invert^1$);
2. O cálculo dos polinômios de interpolação na etapa de segmentação é efetuado pelo processo ($segm^2$);
3. O paralelismo é explorado por processos síncronos que aplicam a mesma função à intervalos distintos de dados, na figura representados por alocação estática de processos paralelos, porém esta alocação pode ser dinâmica com a utilização do construtor de iteração paralela;
4. Ao final da execução dos processos paralelos, é efetuado o somatório dos resultados parciais relativo ao processo $subdiv^k$, $k = 3(1)7$, assim retornando à posição de memória definida como saída o resultado final pelo processo ($sumC^4$). Esta etapa, precedida por uma barreira de sincronização, e executa o somatório após finalizar os processos $subdiv$. O somatório final contém os resultados de cada fatia e o valor do volume total do relevo.

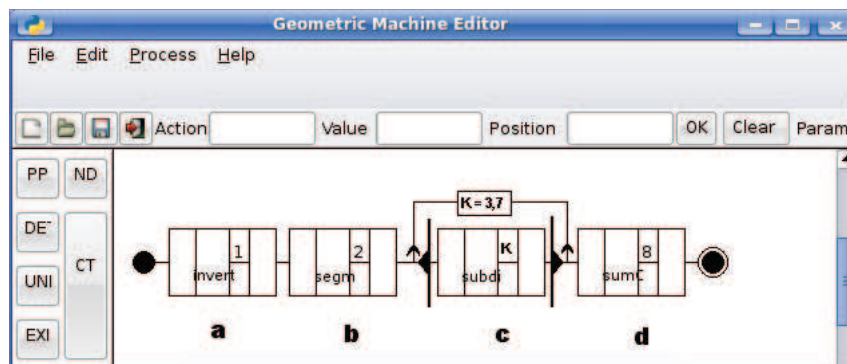


Figura 5.10: Modelagem do protótipo utilizando o VPE-GM

Inicialmente, o desenvolvimento estava focado em dois protótipos distintos, porém, considerando que o ambiente VirD-GM atualmente proporciona a execução das

aplicações em grades de computadores multiprocessados, as duas visões convergiram para uma que contempla as duas visões. Salienta-se que os protótipos foram desenvolvidos utilizando a linguagem Java, a qual se mostra adequada, portátil aos dispositivos tipicamente disponíveis em ambientes paralelos, e possui suporte às bibliotecas necessárias para a modelagem prevista.

5.3 Conclusão

Neste capítulo, apresentou-se os protótipos *iCone* e *p-iCone*, os quais implementam a metodologia *iCone*. Estes protótipos são responsáveis pelo cálculo do volume de produtos a partir de uma matriz tridimensional de pontos que define o relevo do produto armazenado no armazém.

6 *DIGSISA*³: EQUIPAMENTO DIGITALIZADOR

A etapa que antecede o cálculo do volume de um relevo não uniforme compreende a identificação dos pontos do relevo. Para que isto seja viável, encontra-se em desenvolvimento o equipamento digitalizador *DigSisA*³, descrito e modelado neste capítulo. Este equipamento é responsável por efetuar medições sobre o relevo e a partir dos resultados, gerar um conjunto finito de pontos, representado por uma matriz tridimensional. Uma visão do protótipo *DigSisA*³ pode ser obtida na Figura 6.1.

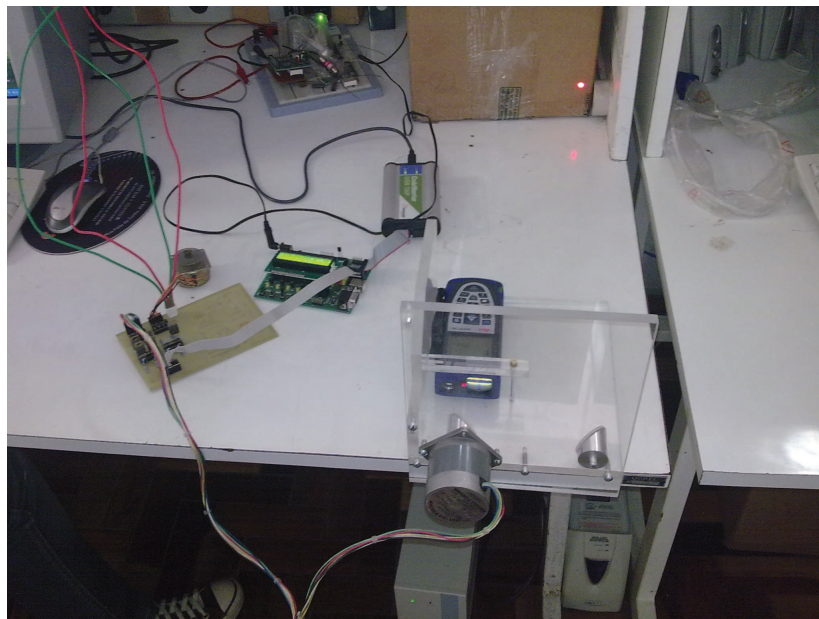


Figura 6.1: Visão do protótipo básico.

O *DigSisA*³ é um dispositivo capaz de armazenar as propriedades do relevo de produtos, a partir de um processo de medição. Tais propriedades são identificáveis através da captura de coordenadas dos pontos.

Este equipamento é caracterizado como um *laser track system*, onde são efetuadas medições de distâncias a *laser* através de uma trena eletrônica, associadas a movimentos angulares, obtendo deste modo, a caracterização tridimensional do relevo .

De forma análoga a esta proposta, o mercado de equipamentos digitalizadores baseados em laser apresenta soluções de funcionamento semelhante, porém com

restrições de alcance e/ou com custo relativamente elevado. Um exemplo é o modelo *AR4000 da Acuity*, avaliado em torno de R\$ 22.000,00, e com alcance aproximado de 16,5 metros (DISTAGE, 2003). Além deste, outros modelos de *scanners* tridimensionais estão no mercado, entretanto apresentando custos elevado e necessidade de movimentação para eliminar possíveis sombras formadas por sobreposições do relevo.

Consciente do contexto social e econômico de nossa região, o protótipo busca aliar uma proposta de baixo custo total, com alcance compatível a grande diversidade de armazéns e comercialmente viável, buscando desta forma, colaborar com o desenvolvimento de um produto acessível ao setor industrial e também a pequenos produtores rurais.

A redução dos custos é alcançada pelo uso de tecnologias exploradas na construção do equipamento de medição, incluindo os tipos de atuadores e o modelo de trena a *laser*. A combinação e integração destes componentes, permite um alcance para medições de aproximadamente 100 metros, para o digitalizador do *SisA³*. Outro aspecto que impacta nos custos é a estratégia de deslocamento sobre o armazém, o que proporciona uma maior detalhamento em áreas de avaliação e redução de custos.

Outras possibilidades também foram objeto de estudo para o *DigSisA³*, onde pode-se destacar: (i) desenvolver um sistema de articulação com mais de um eixo, assim cobrindo uma maior área e reduzindo a quantidade de deslocamentos do equipamento. (ii) a utilização da combinação de vários digitalizadores e, desta forma, não necessitando de movimentação do equipamento. Tais otimizações, apresentam-se oportunas para um cenário de alto fluxo de produtos, onde o tempo de parada impacte, profundamente, na produção, representando perdas financeiras elevadas, justificando o custo do *DigSisA³* com as otimizações descritas.

Além do *DigSisA³*, o projeto possibilita a utilização de outros tipos de digitalizadores, sendo que estes, de maneira geral, apresentam custos elevados. A utilização destes é possível através da conversão dos dados capturados, para o formato matricial e com interpretação pelo *VirD-GM*.

Além de grãos, o *DigSisA³* também possibilita a medição de outros produtos armazenados a granel como cavaco de madeira, adubos, cimento, além de uma diversidade de produtos com pequeno tamanho de grão ou no formato de farelo ou pó.

Na seção 6.1 apresenta-se uma descrição do relevo formado pelos produtos agrícolas armazenados a granel, salientando suas características específicas.

6.1 Discretização do Espaço de Busca

Nesta seção, descreve-se o ambiente tridimensional formado pelo relevo de produtos em um armazém horizontal, sua representação geométrica, incluindo as posições dos eixos coordenados e o ponto zero (origem), na intersecção dos planos formados por estes eixos.

6.1.1 Estrutura Tridimensional Formada pelo Relevo

A Figura 6.2 mostra uma imagem da visão interna da organização e da estruturação de um armazém horizontal com a base no formato de um paralelepípedo, onde o telhado tem formato de semi-cilindro.

Os armazéns horizontais, como o da Figura 6.2 apresentam limites laterais que são compostos por paredes que partem da base e altura definida de acordo com a sua capaci-

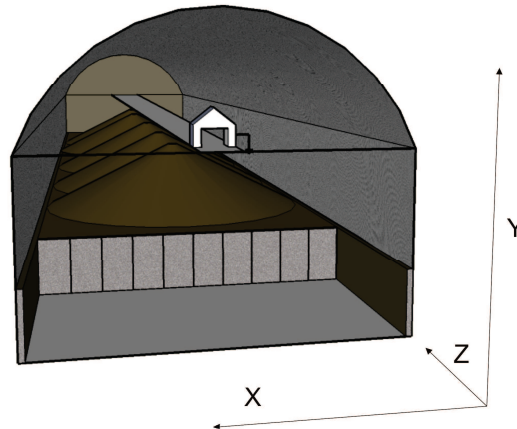


Figura 6.2: Imagem de um armazém.

dade e especificação de produto. Acima das paredes possui um espaço de escoamento, caso ocorra descarga excessiva de produtos, o limite de capacidade do armazém não é extrapolado. Este espaço de escoamento evita danos estruturais no armazém, os quais poderiam ser causados pela pressão gerada pelo excesso de produtos. Por fim, sobre pilares e acima da região de escoamento, está fixado a estrutura do telhado, geralmente modelada por um prisma triangular reto ou por um semi-cilindro.

A descarga de produtos no interior de um armazém de granéis, em geral, ocorre através de um esteira transportadora, fixada na posição central do armazém (eixo OX na Figura 6.2) e percorre toda sua extensão. Para descarregar produto adiciona-se um anteparo, no qual o produto esbarra e é redirecionado para baixo.

Frequentemente ocorre que a posição da descarga nos armazéns é variável, ou seja, o anteparo que redireciona os produtos tem a capacidade de movimentar-se sobre a esteira, possibilitando a formação de uma sequência de cones, com possível superposições nas bases. Esta descarga pode ser melhor compreendida se comparada a uma ampulheta, antigo dispositivo de marcação de tempo apresentado na Figura 6.3, cuja transferência de areia apresenta semelhanças com a descarga de produtos nos armazéns (MADEHOW, 2007).

Observa-se na Figura 6.3 que a areia cai sobre uma superfície do vidro que dá forma à ampulheta. Durante a queda forma-se um cone cuja variação da altura é proporcional a variação do tempo. Assim, quando a base do cone alcança o vidro que limita lateralmente a ampulheta, a altura do sólido geométrico gerado pela base da ampulheta, assim como a base do cone, aumentam. Este aumento decorre do preenchimento dos limites laterais, não havendo alteração no ângulo do cone. Tem-se, então, a figura tridimensional gerada pela forma limítrofe da ampulheta e representada por um cone.

Observando a parte superior da ampulheta (reservatório de areia), verifica-se a geração de um vale, formado por um cone inscrito no reservatório. Em uma ampulheta, este vale costuma ser uniformemente distribuído, devido às características da areia. Em outros produtos, isto pode não ocorrer, pois podem haver variações na umidade ou até mesmo deterioração do produto, agrupando-o de forma não uniforme e ocorrendo deformações no relevo, quando da remoção do produto do armazém.



Figura 6.3: Ampulheta

Estas variações se apresentam como fatores que adicionam complexidade ao processo de auditoria, pois se não consideradas, resultarão em um valor de volume final inexato.

6.1.1.1 Orientação dos Eixos na Representação dos Pontos do Relevo

O $SisA^3$ considera o espaço euclidiano tridimensional $3D$ para modelagem do relevo de produtos estocados em um armazém, apresentando três planos (XY , YZ , XZ), perpendiculares dois a dois, onde os três eixos coordenados são indicados pelas expressões \vec{OX} , \vec{OY} e \vec{OZ} .

Neste contexto, X refere-se à medida da distância entre o centro e os limites laterais, Y a altura e Z a profundidade do armazém, conforme indicado na Figura 6.4. Pode-se referenciar um ponto através das três coordenadas, correspondente à medida da distância entre o a origem e orientação em relação aos eixos coordenados.

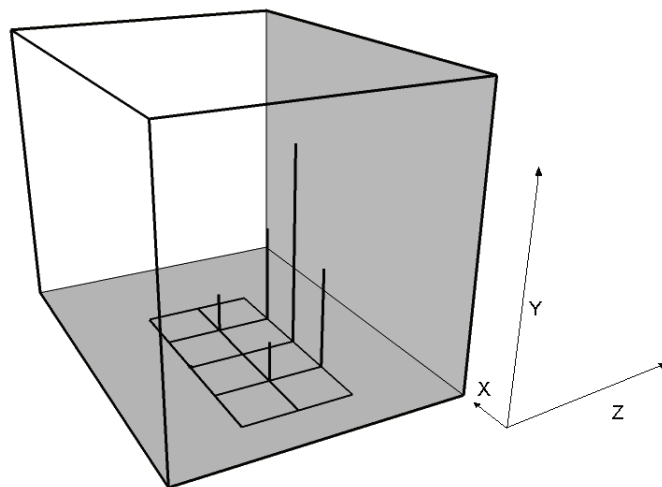


Figura 6.4: Orientação dos eixos na modelagem do armazém

6.2 Organização do DigSisA³

Na modelagem e organização do equipamento *DigSisA³*, considera-se a integração de três componentes:

- O **componente eletromecânico**, responsável pelos movimentos do suporte articulado, sendo composto por uma trena eletrônica *laser*, e um motor de passo para movimentação de um espelho. Este tipo de motor se caracteriza por apresentar um deslocamento angular preciso, e bloqueio de posição mesmo sem alimentação.
- O **componente eletrônico**, para servir de interface de comunicação entre o computador e controle dos atuadores;
- O **componente de software**, responsável pela gerência das movimentações de motores, para efetuar medições de distâncias, manipulação da trena eletrônica e o cálculo da posição dos pontos.

Salienta-se que, além destes componentes é necessária a utilização de um computador para a execução do *software*. Este equipamento não necessita possuir grande poder computacional, como por exemplo um PDA (*Personal Digital Assistant*), possuindo basicamente, as interfaces de comunicação entre equipamentos disponíveis.

O ponto inicial do *DigSisA³* é definido apontando-se o laser na direção paralela ao eixo \overline{OY} (vertical), e sentido para baixo.

O digitalizador é utilizado para determinar os pontos sobre uma linha paralela ao eixo \overline{OX} , ou seja uma varredura bidimensional, na direção centro-limites laterais. Apresenta como posição normal, o ângulo de 0 graus em relação ao eixo vertical, e limites máximos para medição de +90 ou -90 graus (90° para cada lado).

Para obter a profundidade, o equipamento é reposicionado diversas vezes no sentido do comprimento do armazém, afim de cobrir toda sua extensão. Esta movimentação, pode ser manual, ou por um processo automatizado, apresentando assim total independência do sistema.

Os motores de passo, comercialmente disponíveis com baixo custo, em geral apresentam o passo angular de 1,8 graus. Neste caso, esta medida de ângulo é definida como unidade mínima de deslocamento angular, e a leitura de uma linha pode conter no máximo 100 medições. Há a possibilidade de utilização de caixas de redução, para se aumentar a quantidade máxima de medições, viabiliza a criação de relações diferenciadas de ângulo por passo, no equipamento *DigSisA³*. Além das caixas de redução, podem ser usados outros tipos de motores de passo com ângulo de passo menor, porém com impacto no custo final de desenvolvimento do *DigSisA³*.

Salienta-se ainda que um motor de passo pode operar com o deslocamento de meio passo, ou seja, no caso de um motor com 1,8 graus de deslocamento por passo, este poderia realizar o deslocamento de 0,9 graus. Entretanto, esta operação consome uma quantidade maior de energia, necessitando que duas bobinas estejam alimentadas, alterando os requisitos da alimentação elétrica do sistema.

A geração da matriz de pontos de saída do sistema se dá a partir dos valores armazenados nas medições combinados com parâmetros medidos na etapa de calibração, incluindo os respectivos ângulos das medições para o cálculo do posicionamento de cada ponto no plano tridimensional.

6.2.1 Componentes Eletromecânicos

O digitalizador projetado neste trabalho, possui como componentes eletromecânicos: motor de passo, o medidor de distâncias *laser* (trena eletrônica) e o acionador de requisição de medições. A visualização da modelagem e organização do protótipo inicial pode ser vista na Figura 6.5, cuja descrição das principais características e funcionalidades é apresentada a seguir:

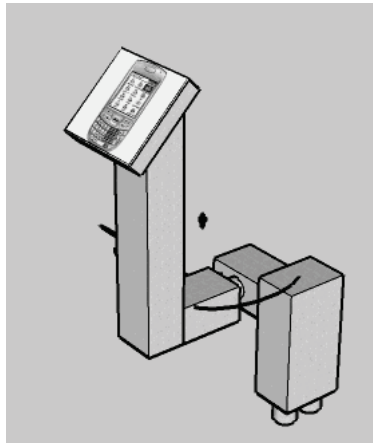


Figura 6.5: Organização do protótipo *DigSisA*³.

- O **motor de passo**: é um transdutor que converte energia elétrica em movimentos angulares precisos, sendo controlado através de pulsos. Assim, para a obtenção de uma rotação específica, referente a uma medida de ângulo, calculam-se o número de pulsos, múltiplos do ângulo referente a um passo. Estes motores apresentam bloqueio de posição, o que é uma garantia de que o motor irá permanecer na posição desejada durante o tempo necessário (JONES, 2004). No protótipo, utilizou-se um motor de passo de 12V e passo de 1,8 graus da marca Minebea;
- Um **medidor de distância a laser**: consiste em um equipamento que envia um feixe de luz contra uma determinada superfície e recebe o retorno através da reflexão ocorrida na superfície medida. Pelos cálculos sobre o tempo entre o envio e o retorno do laser, é identificada a distância entre o medidor e o alvo. Existem outros modelos de medição a laser que também efetuam cálculos precisos e que podem ser utilizados neste trabalho conforme (CHEN; BROWN; SONG, 2000; DISTAGE, 2003). Esta distância é um dos principais parâmetros para geração dos pontos da matriz de entrada. Salienta-se que pela propriedade das trenas eletrônicas a *laser* em efetuar cálculos sobre um feixe de luz para medição de distâncias, possibilita-se a utilização de espelhos planos para movimentação do laser na direção do objeto medido. Logo, obtém-se um ganho movendo um espelho, e não todo sistema. No protótipo inicial, são utilizados espelhos normais, os quais não possuem um índice de reflexão do feixe laser próximo a 100%, necessitando da utilização de espelhos especiais no caso de um protótipo para medição de grandes distâncias. No protótipo, utilizou-se uma trena eletrônica a laser da marca Leica, modelo Disto A6, com alcance máximo de 200 metros e comunicação *bluetooth*.

6.2.2 Componentes Eletrônicos

São utilizados dois equipamentos inteiramente eletrônicos no digitalizador do *SisA*³, um equipamento com processador para execução do *software* de aquisição de dados e outro equipamento controlador, para prover a interface de comunicação entre os equipamentos e o processador. Este último, sendo responsável por transformar as requisições do *software* de controle para sinais que possam ser entendidos pelos equipamentos eletromecânicos.

- **Componente Processador**, tem como funcionalidade executar o *software* de controle do digitalizador. Este componente pode ser representado por diferentes dispositivos computacionais (PC, PDA ou sistema embarcado), entretanto devendo possuir interface de comunicação para conexões (USB e Bluetooth (BT)). Um fator extremamente relevante na escolha do processador é o consumo de energia, pois o *software* de controle não apresenta processamento intensivo de dados, assim quanto menor o consumo, menor será o seu impacto no consumo das baterias. Nos testes do protótipo, foi utilizado um PC como processador.
- **Controlador**, possuindo duas funcionalidades: (i) acionar a trena e (ii) efetuar a tradução dos comandos de movimentação para controle dos motores de passo. Este equipamento de controle, recebe a quantidade de pulsos do computador, e repassa ao motor através da conversão para sinais elétricos. Assim o motor efetua a rotação correta, ocorrendo o mesmo com a solicitação de medições para a trena (JONES, 2004). No protótipo, o controlador é composto de um DSP, um driver de comunicação e uma placa de potência para alimentação das bobinas do motor de passo.

A Figura 6.6 esquematiza o sistema eletrônico que envia os comandos para o DSP (*Digital Signal Processor*) via USB. Este dispositivo processa os dados, codificando os comandos para os *drivers* dos motores de passo, a fim de realizar a movimentação mecânica. A comunicação entre o DSP e o processador será obtida por aplicação de protocolo pré-estabelecido, onde serão empregados os comandos para os movimentos do suporte eletromecânico.

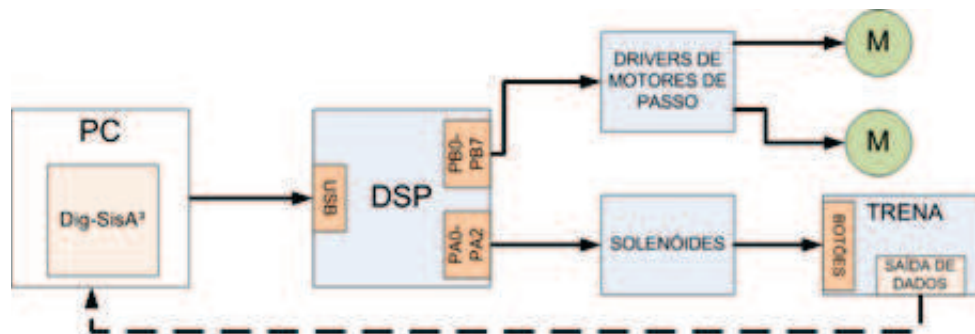


Figura 6.6: Diagrama de Blocos do DigSisA³.

6.2.3 Componente de *software*

O componente de *software* é responsável pelo posicionamento da trena eletrônica, solicitação de medição e tratamento do resultado. Combinando estes dados aos

parâmetros previamente conhecidos, tem-se a identificação da posição de um ponto e o armazenamento de suas coordenadas em um conjunto de pontos tridimensionais para modelagem do relevo. Estes pontos são armazenados em um arquivo, o qual posteriormente é utilizado no *software iCone* do *SisA³*.

O processo de aquisição das coordenadas dos pontos da matriz pode ser visualizado na Figura 6.7, onde na parte superior está representado o equipamento *DigSisA³*, logo abaixo, tem-se um corte do cone de representação dos produtos, representação da leitura bidimensional. Nas laterais estão representados os limites (paredes) e acima das paredes há uma região denominada escape de produtos.

Para controle do *DigSisA³*, foi definido um protocolo simples para comunicação entre o software de controle e o controlador. Cada comando consiste de três *bytes*, sendo o primeiro a definição da operação ou o aviso de confirmação de recebimento de algum *byte*, o segundo define o sentido do movimento e o terceiro define o parâmetro da operação, que pode ser a quantidade de passos do motor ou o tempo de pressionamento de um botão.

A região de escape para os produtos, garante que não exista agrupamento de produtos acima das paredes, limitando a capacidade do armazém. A representação gráfica, na Figura 6.7 mostra um passo de medição. Neste caso, não há variação na posição do *scanner* em relação ao eixo Z, mas podem ocorrer variações nas medições que identificam as coordenadas X e Y de cada ponto. Sendo assim, cada passo de medição consiste em n pontos com mesmo valor para a coordenada Z.

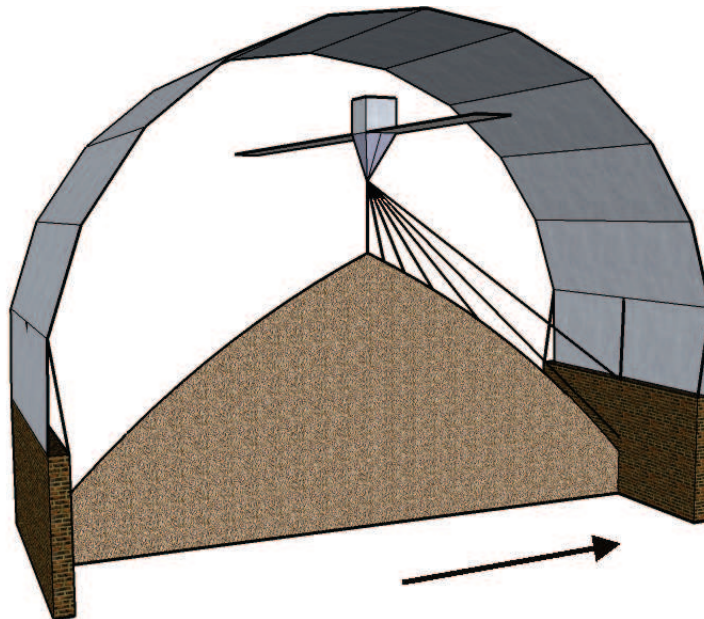


Figura 6.7: Imagem dos passos de medição.

Para o cálculo da localização de um ponto é necessário o prévio conhecimento da altura do *DigSisA³*. Utilizando como referência o piso do armazém, esta altura indicada por H_t corresponde à distância entre o aparelho e o piso.

A posição em relação ao eixo OZ é obtida posicionando o feixe laser em direção ao final do armazém, indicado por P_z . O valor da posição de um ponto P_z é calculado pela diferença da distância inicial do equipamento (origem) e a distância atual.

O ângulo Θ é definido pelo *DigSisA³*, no momento do posicionamento da trena eletrônica, consistindo no parâmetro repassado pelo sistema de controle dos motores. A

distância medida é indicada por d_{Laser} .

A posição de um ponto P_x no eixo \overrightarrow{OX} é obtida pela expressão:

$$P_x = d_{Laser} \times \sin \Theta \quad (6.1)$$

Por outro lado, a posição de um ponto P_y no eixo OY é definida pela expressão:

$$P_y = H_t - (d_{Laser} \times \cos \Theta) \quad (6.2)$$

Assim, o ponto resultante é definido por $P(P_x, P_y, P_z)$.

A aquisição de cada conjunto de pontos para cada fatia do relevo no *DigSisA³* considera a medida da altura do ponto inicial, o qual está situado no centro do armazém, logo abaixo do equipamento. Após esta medição ele efetua n medições, sendo que em todos casos, o deslocado angular efetuado pelo equipamento é constante. Este processo pode ser visualizado na Figura 6.7, através das sete linhas que partem do *scanner* que apontam ou para superfície do produto, ou para os limites laterais.

Como n é constante para todas as camadas medidas em cada feixe durante o deslocamento no armazém, pode-se aplicar a interpolação no processo de cálculo do volume, a fim de gerar pontos intermediários para melhorar a modelagem da superfície natural do relevo.

O sentido do deslocamento do *DigSisA³* segue, conforme a seta indicada abaixo da Figura 6.7, com os ângulos constantes.

6.3 Descrição do Funcionamento

Na descrição do funcionamento do *Dig-SisA³*, tem-se, primeiramente, processo de parametrização, o qual começa com a calibração do equipamento digitalizador posicionado no ponto de origem das medições, seguido da leitura das coordenadas dos pontos do relevo, obtenção da estrutura de dados e por fim a geração da matriz de pontos.

O diagrama da Figura 6.8 descreve as principais atividades, garantindo seu correto funcionamento.

6.3.1 Parametrização

A etapa de parametrização parte da medição de alguns pontos específicos, sempre que possível efetuadas pelo próprio equipamento.

Os parâmetros necessários para calibração do *DigSisA³* consistem em:

- a medição da altura do *scanner* em relação ao solo, caso não haja visibilidade do piso do armazém no ponto de origem, a altura deve ser informada manualmente, pela interface gráfica do software de gerenciamento;
- a medição da distância e ângulo entre o equipamento e as barreiras laterais;
- a medição do comprimento entre o ponto origem e o final do armazém.

Após parametrizado, o *DigSisA³* está pronto para efetuar a digitalização do relevo, através da aquisição dos pontos das diversas seções do relevo, aqui denominadas camadas.

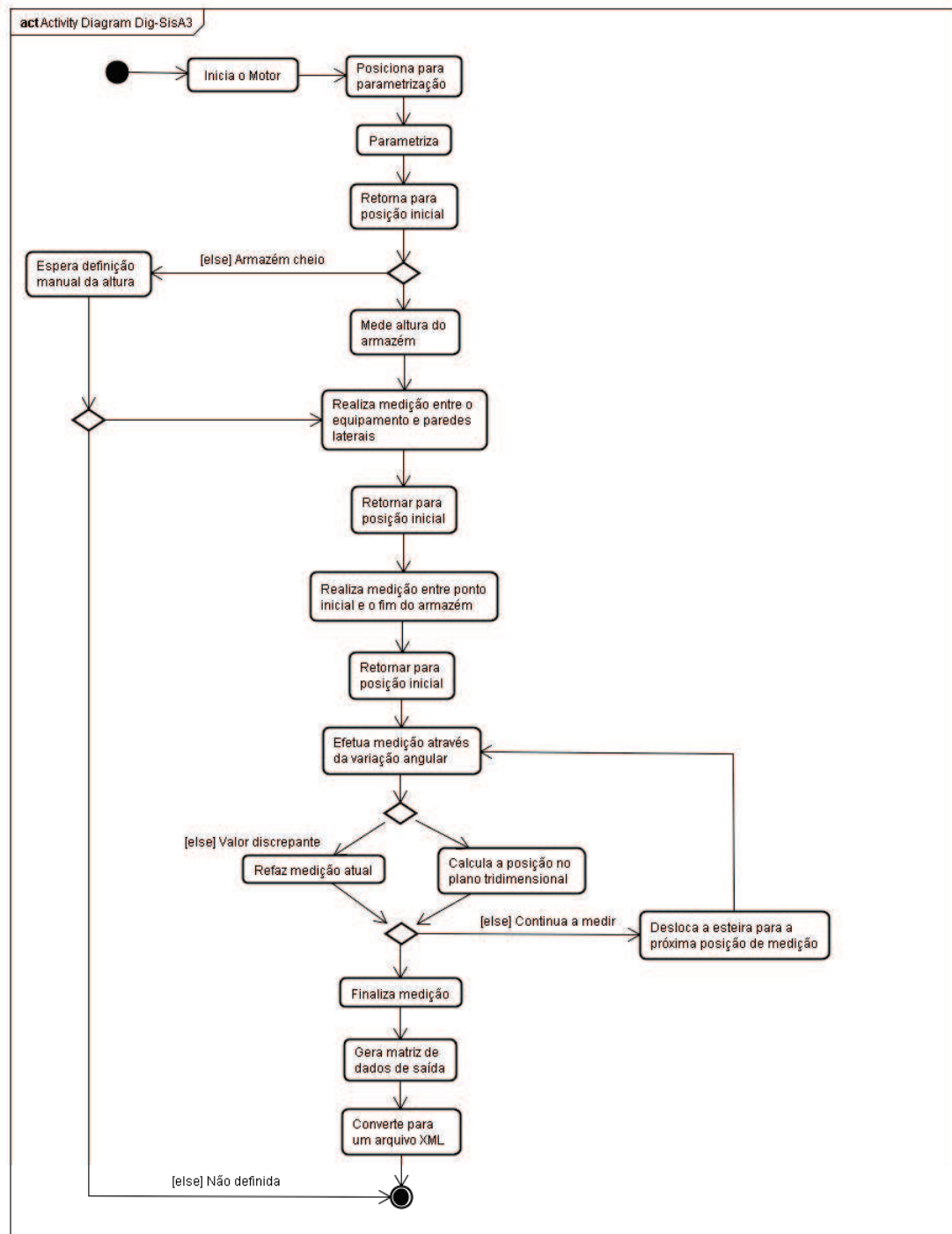


Figura 6.8: Diagrama de atividades do protótipo DigSisA³ (ALMEIDA; REISER, 2009).

6.3.2 Leitura

Após a parametrização dos dados, o equipamento é posicionado para obter a leitura dos pontos da primeira camada, sendo que cada camada é composta por pontos bidimensionais (2D), os quais representam uma seção no relevo.

Cada leitura de camada do *DigSisA*³ percorre uma quantidade pré-definida de pontos, obtida pelo deslocamento angular da trena *laser* em direção ao eixo \overline{OX} . Para medição de cada ponto, são considerados dois valores, a distância informada pela trena e o ângulo no qual ela está posicionada, conforme é apresentado na Figura 6.9.

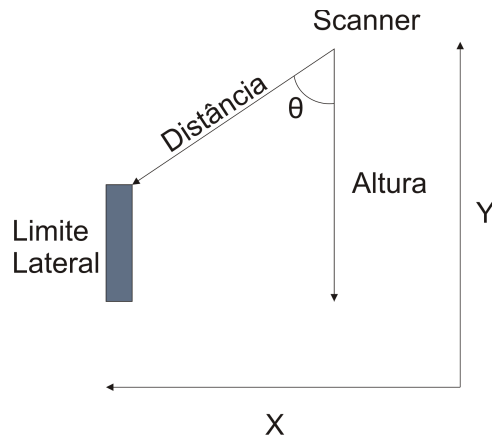


Figura 6.9: Parametrização dos dados na calibração do equipamento scanner.

O controle das medições é efetuado por uma unidade computacional, a qual pode ser executada sobre diversos dispositivos, desde um PDA até um computador. A cada medição, efetua-se o cálculo da posição do ponto.

O relevo tridimensional, modelado por uma matriz tridimensional de dados, e é exportado para um arquivo na linguagem XML, consiste na leitura de uma sequência de camadas, sendo o conjunto de camadas limitado pela distância entre o *DigSisA*³ e o final do armazém.

6.3.3 Obtenção e Estrutura dos Dados

A estrutura matricial tridimensional de dados, gerada a partir dos valores coletados pela medição de ângulo, altura e distância, inclui também a informação da localização do *DigSisA*³ dentro do armazém, de acordo com o sistema de coordenadas tridimensional descrito na Seção 6.3.2.

Para cada ponto medido, consideram-se os seguintes parâmetros:

1. Altura (variável global atribuída na calibração);
2. Distância entre o ponto de origem e a posição atual em relação ao eixo OZ ;
3. Ângulo do feixe laser no momento da medição de cada ponto;
4. Distância medida entre a trena eletrônica e o ponto.

Estes parâmetros determinam a estrutura dos dados necessária para identificação de cada ponto do relevo. Para facilitar o armazenamento dos dados, a distância do passo é associada à linha de leitura, sendo tal informação armazenada uma vez, entretanto esta definida em cada ponto na matriz de saída.

6.3.4 Saída de Dados

Após a determinação dos pontos do relevo, identificados pelo *DigSisA*³, ocorre a estruturação em um formato de transferência de dados, armazenados em um arquivo XML, utilizado como entrada para o protótipo *iCone/p-iCone*. Segue um exemplo deste arquivo, na Figura 6.10, com a definição de onze camadas possuindo cinco pontos em cada uma, porém devido a restrições do ambiente D-GM, os dados devem estar em uma matriz quadrada, assim a matriz possui tamanho onze.

```
- <memoria>
  <posicao dimensao="2" tamanho="11" />
  <valores>Point</valores>
  <dados>5:0:0, 4:0:0, 3:0:0, 2:0:0, 1:0:0, 0:0:0, 0:0:0, 0:0:0, 0:0:0, 0:0:0, 0:0:0, -</dados>
  <dados>5:0:1, 4:0:1, 3:0:1, 2:2.1115:1, 1:3.5076:1, 0:4:1, 0:0:0, 0:0:0, 0:0:0, 0:0:0, 0:0:0, -</dados>
  <dados>5:0:2, 4:0.2115:2, 3:3.0294:2, 2:5.5778:2, 1:7.3509:2, 0:8:2, 0:0:0, 0:0:0, 0:0:0, 0:0:0, 0:0:0, -</dados>
  <dados>5:0:3, 4:2.2694:3, 3:5.5778:3, 2:8.6863:3, 1:11.0557:3, 0:12:3, 0:0:0, 0:0:0, 0:0:0, 0:0:0, 0:0:0, -</dados>
  <dados>5:0:4, 4:3.5933:4, 3:7.3509:4, 2:11.0557:4, 1:14.3431:4, 0:16:4, 0:0:0, 0:0:0, 0:0:0, 0:0:0, 0:0:0, -</dados>
  <dados>5:0:5, 4:4:5, 3:8:5, 2:12:5, 1:16:5, 0:20:5, 0:0:0, 0:0:0, 0:0:0, 0:0:0, 0:0:0, -</dados>
  <dados>5:0:6, 4:3.5933:6, 3:7.3509:6, 2:11.0557:6, 1:14.3431:6, 0:16:6, 0:0:0, 0:0:0, 0:0:0, 0:0:0, 0:0:0, -</dados>
  <dados>5:0:7, 4:2.2694:7, 3:5.5778:7, 2:8.6863:7, 1:11.0557:7, 0:12:7, 0:0:0, 0:0:0, 0:0:0, 0:0:0, 0:0:0, -</dados>
  <dados>5:0:8, 4:0.2115:8, 3:3.0294:8, 2:5.5778:8, 1:7.3509:8, 0:8:8, 0:0:0, 0:0:0, 0:0:0, 0:0:0, 0:0:0, -</dados>
  <dados>5:0:9, 4:0:9, 3:0:9, 2:2.1115:9, 1:3.5076:9, 0:4:9, 0:0:0, 0:0:0, 0:0:0, 0:0:0, 0:0:0, -</dados>
  <dados>5:0:10, 4:0:10, 3:0:10, 2:0:10, 1:0:10, 0:0:10, 0:0:0, 0:0:0, 0:0:0, 0:0:0, 0:0:0, -</dados>
</memoria>
```

Figura 6.10: Matriz de pontos em XML.

Caso seja necessária a execução dos protótipos em diferentes máquinas, a transferência de dados, pode ser efetuada, tanto manualmente através de mídias removíveis, quanto através de alguma rede de comunicação de dados como WI-FI ou *bluetooth*.

6.4 Conclusão

Neste capítulo, apresentou-se o protótipo *DigSisA*³, equipamento digitalizador de relevos não-uniformes, o qual é responsável pela geração da matriz de pontos que define o relevo de produtos estocados em armazéns de granéis. Foi feita a modelagem e implementação do protótipo assim como o software de integração dos componentes, tendo o auxílio do Laboratório de Biomédica, na presença de uma bolsista para a modelagem e montagem e programação do DPS e módulo de controle do motor de passo e atuadores dos botões. Ainda teve auxílio da empresa Tornearia Schulz e da empresa Contronic na modelagem e construção do sistema mecânico do *DigSisA*³. Este equipamento engloba componentes de *hardware* e *software* que permitem o processo de captura dos pontos. O *DigSisA*³ não foi validado no armazém pois ainda encontra-se em desenvolvimento, porém as ferramentas foram validadas individualmente.

7 RESULTADOS OBTIDOS

Este capítulo considera os resultados obtidos com a implementação do método *iCone* e apresenta dois estudos de caso para validação dos protótipos *iCone* e *p-iCone*, considerando análise de erro e análise de desempenho, respectivamente. Estes estudos consideram como categorias de testes com conjuntos de dados específicos e em diferentes arquiteturas multiprocessadas, ou seja, execução em grade de computadores e multi-core.

7.1 Estudo de Caso para Análise de Erro no Protótipo *iCone*

O trabalho utiliza um ambiente simulado para avaliação do protótipo *iCone* e os dados aplicados para avaliação dos erros gerados na execução do método de cálculo do volume se referem a objetos regulares (figuras geométricas), com volume previamente conhecido e que se aproximam das principais formas geométricas do relevo formado pelos produtos armazenados.

Considerando este critério de seleção, optou-se pela figura de cones regulares para extração de pontos e validação do protótipo. Mais especificamente, tem-se na Figura 7.1, um cone reto ($C(h, r)$), onde r e h correspondem a medida do raio da base e da altura do cone, respectivamente.

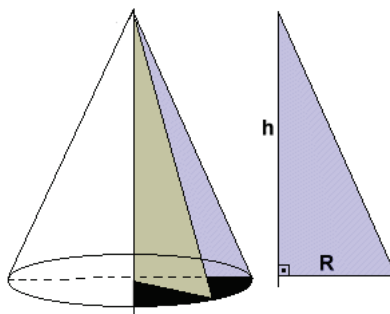


Figura 7.1: Cone regular $C(h,r)$.

O volume é obtido pela aplicação da fórmula analítica dada pela expressão 7.1:

$$V = (\pi/3 \times r^2 \times h), \quad (7.1)$$

Assim, substituindo na Eq.(7.1) os seguintes parâmetros: 20 metros de altura e 5

Tabela 7.1: Resultados da análise de erros do estudo de caso *iCone* (sequencial)

n	Volume sem dividir area sub-camada	Volume com 10 divisões da area	Profundidade sub-camada em milímetros	Erro Absoluto	Erro Relativo
1	211,4418	212,2856	100	49,5138	23,32%
2	236,4418	237,1644	50	24,6350	10,39%
3	244,7751	245,4581	33,3333	16,3413	6,66%
4	248,9418	249,6063	25	12,1931	4,88%
5	251,4418	252,0962	20	9,7032	3,85%
6	253,1085	253,7566	16,6667	8,0428	3,17%
7	254,2489	254,9431	14,2857	6,8563	2,69%
8	255,1918	255,8333	12,5	5,9661	2,33%
9	255,8862	256,5261	11,1111	5,2733	2,06%
10	256,4418	257,0808	10	4,7186	1,84%
100	260,9418	261,577	1	0,2224	0,09%

metros de raio ($C(20, 5)$) se obtém como volume:

$$V = 261,7993878 \times 2 = 523,5987756. \quad (7.2)$$

Na análise de erros, consideram-se:

- Para o cálculo do erro absoluto, a Equação 7.3:

$$Erro_{absoluto} = (Valor_{Exato} - Valor_{Aproximando}); \quad (7.3)$$

- Para o cálculo do erro relativo, a Equação 7.4:

$$Erro_{relativo} = (Erro_{absoluto}/Valor_{Exato}), \quad (7.4)$$

Com base na simetria associada à Figura 7.1, o método *iCone* para cálculo do volume e o sistema de medição podem ser reconfiguráveis, reduzindo o intervalo de deslocamento da trena e número de pontos coletados para geração dos parâmetros (distância e ângulo) que definem a matriz de entrada de dados.

A Tabela 7.1 mostra a evolução dos resultados obtidos nas sucessivas iterações do algoritmo e resume a análise dos correspondentes erros relativa ao semicone $SC(20, 5)$, considerando Eq. (7.3) e Eq. (7.4). O número n de iterações determina o número de fatias do relevo. Pela análise dos resultados, identificou-se que a partir de $n = 10$, os resultados gerados, tanto pelo cálculo direto quanto pelo o processo de subdivisão das fatias, mostram-se próximos do valor obtido na Eq. (7.2), pois há uma pequena variação.

Com o aumento da quantidade de subdivisões por fatia, o resultado converge para o valor real, conforme pode ser visualizado na última linha da tabela, quando se divide uma fatia em $n = 100$ fatias menores. Neste último teste, tem-se cada subdivisão de uma fatia apresentando 1.0 cm de profundidade e um erro relativo no cálculo do volume, pela Eq. (7.4), de aproximadamente $Erro_{relativo} = 0,09\%$.

No processo manual, além do relevo presente nos armazéns não ser regular, podem haver diferenças causadas por cálculo tendencioso, pois neste tipo de medição já se estima um valor para o produto armazenado e, tal situação, pode influenciar os auditores nos arredondamentos de medidas. Com base na metodologia utilizada atualmente nas empresas, onde as medições e os cálculos dos parâmetros como altura e raio dos cones que modelam o relevo do produto no armazém são, muito frequentemente efetuados de forma manual, e diversas vezes necessitando que o processo seja refeito, o erro relativo obtido mostra a viabilidade do método desenvolvido.

Salienta-se que, com a aplicação do processo de subdivisão dos trapézios no cálculo da área das subfatias do relevo, há melhoria na aproximação do volume real. Este processo embora não apresente uma diferença substancial neste estudo de caso, é indispensável, quando se trata de um sistema de avaliação do volume em armazéns, cuja modelagem do relevo gera superfícies não uniformes e se tem como premissa a precisão do resultado.

7.2 Estudo de Caso para Análise de Desempenho no Protótipo *p-iCone*

O protótipo *p-iCone* utiliza o mesmo algoritmo que implementa o protótipo *iCone*, diferenciando-se somente na estratégia (distribuída e/ou paralela) de execução. E assim, verificou-se, em ambas estratégias, os mesmos valores de erros para os testes efetuados.

A análise de desempenho no *p-iCone* também considera a simulação das matrizes de dados, neste caso o mapeamento e geração das coordenadas que definem as alturas do relevo está baseada na técnica denominada *heightmap* (VISION RAYTRACER PTY. LTD., 2003).

A partir de uma figura bidimensional, a aplicação de *heightmap* possibilita a geração de um relevo tridimensional, baseado na alturas dos pontos associadas ao gradiente de cores. Cada cor representa um valor de altura e, por exemplo, um círculo com cores variando de branco nas bordas a preto no centro e possuindo valores de cinza no interior, pode representar um cone ou uma esfera, dependendo da variação da função que define o gradiente de cor.

Neste contexto, fez-se uso desta técnica para a geração de uma matriz de 100×100 , modelando pontos do relevo. Além deste tamanho de matriz, utilizam-se 10.000 subdivisões de cada fatia e 100 divisões nos pontos da área de cada camada correspondente, estes valores foram definidos para gerar um caso de estresse no computador, entretanto sendo uma quantidade de pontos que facilmente se utilize na prática, dependendo do comprimento do armazém a ser medido.

Esta matriz utilizada, associada aos parâmetros de execução utilizados, promoveu a execução sequencial com tempo de execução em torno de 5 horas, tempo obtido pela média de 8 execuções e que sugere a exploração de técnicas para melhoria de desempenho.

7.2.1 Estratégias de Execução

A execução paralela no ambiente VirD-GM foi efetuada em duas arquiteturas de sistemas paralelos: (i) arquitetura multinúcleos; e (ii) grade de computadores multinúcleos, considerando um conjunto homogêneo de nodos selecionados.

De acordo com a definição da arquitetura da VirD-GM, um dos nodos distingue-se dos demais, caracterizando o nodo VirD-base, responsável tanto pelo escalonamento das tarefas quanto pelo controle do fluxo de dados.

Para a análise de desempenho da versão sequencial, a fim de avaliar o *speedup*, utilizou-se computadores multi-core, porém com restrições no sistema operacional, definidas parametricamente, para que utilizassem um núcleo somente.

A seguir, tem-se a seleção dos teste, onde, foram efetuados 7 amostras de cada configuração, tendo como valores apresentados a média dos resultados e a média dos desvios padrões, os quais resultaram em valores em torno de 0,75%, e em ambos os casos não excedendo 1,31%.

7.2.1.1 Execução em Grade de Computadores Monoprocessados

A grade de computadores utilizada é homogênea: para cada computador possui um chip (*dual core*) intel 2140 (1.6 GHz), com 994 MiB de memória RAM e comunicação através de rede *ethernet* de 100 Mbits. O escalonamento foi feito de tal forma que apenas um processador por nodo fosse utilizado.

Para o cálculo do *speedup*, considerou-se a Equação 7.5 sobre os tempos médios de execução para versões sequencial e paralela (SAHNI; THANVANTRI, 1996):

$$Speedup = (Tempo_{Sequencial} / Tempo_{Paralelo}). \quad (7.5)$$

A Figura 7.2 apresenta os testes efetuados, utilizando os mesmos parâmetros para execução. Verificam-se variação na quantidade de núcleos ativos para execução e na quantidade de computadores.

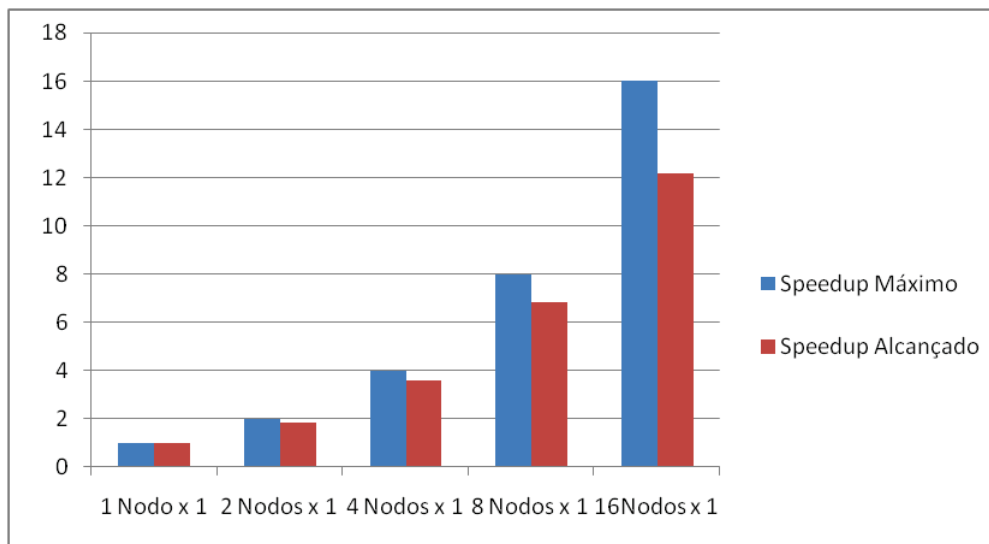


Figura 7.2: Gráfico de Desempenho Distribuído

O ganho de desempenho apresentado pela distribuição com grade de computadores com arquitetura monoprocessada, apresenta-se na Figura 7.3, onde tem-se a eficiência da paralelização, representada no eixo vertical, associada à variação da quantidade de nodos, representada no eixo horizontal.

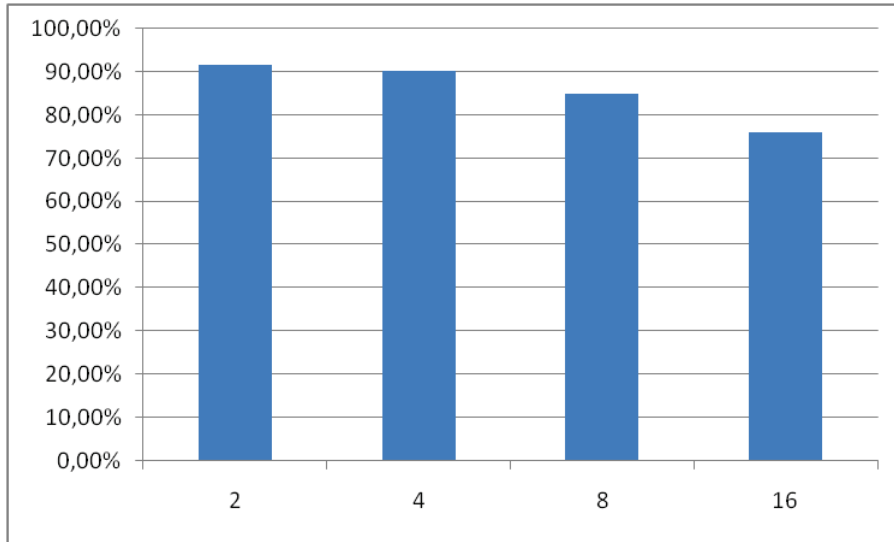


Figura 7.3: Gráfico de Eficiência em Arquitetura Distribuída

7.2.1.2 Execução em Grade de Computadores *Multi-core*

Para avaliação de desempenho do algoritmo em grades de computadores multiprocessados, buscou-se a variação tanto na quantidade de nodos, quanto na quantidade de núcleos ativos por nodo. Desta forma, a avaliação considerou tanto o desempenho pela distribuição quanto pelo compartilhamento de recursos dos computadores multiprocessados.

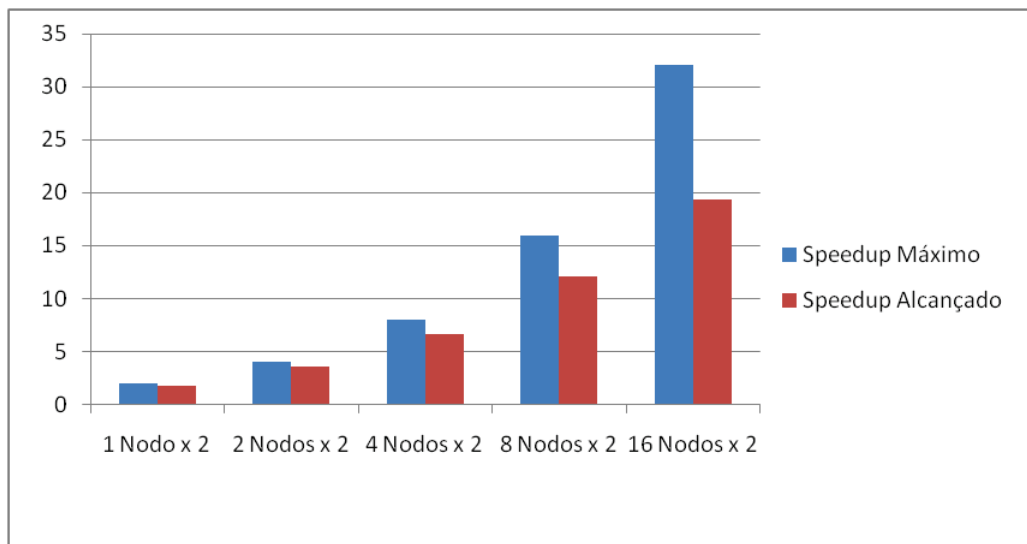


Figura 7.4: Gráfico de Desempenho em Ambiente Distribuído *Multi-Core*

De acordo com o gráfico apresentado na Figura 7.4, o *speedup* obtido com a paralelização foi satisfatório.

Para melhor visualização, apresenta-se o gráfico da Figura 7.5, onde tem-se a eficiência representada no eixo vertical e a variação na quantidade de núcleos no eixo horizontal. Por estar utilizando a arquitetura distribuída associado com a arquitetura *dual-core*, cada nodo utiliza dois núcleos.

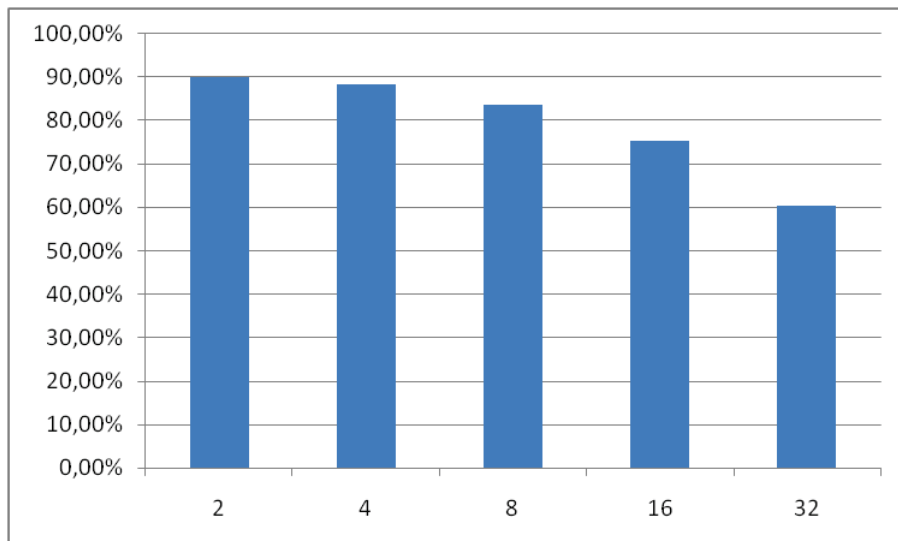


Figura 7.5: Gráfico de Eficiência em Arquitetura Distribuída e Multiprocessada.

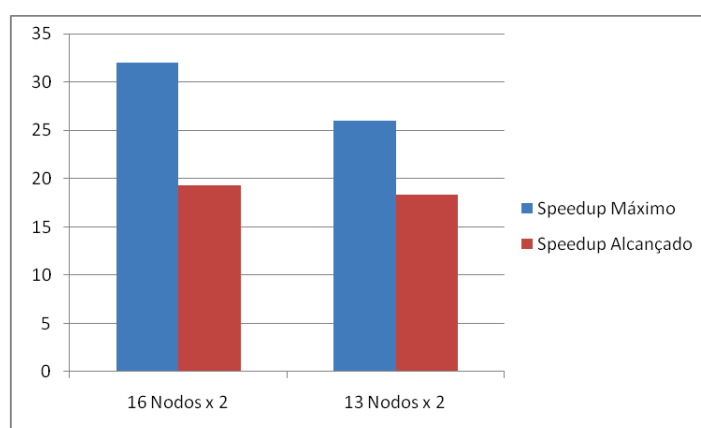


Figura 7.6: Comparativo de Execução com 16 e 13 núcleos.

Os resultados da execução com 32 núcleos processadores instigaram a análise detalhada da execução. Esta análise possibilitou que se identificasse uma característica interessante na distribuição das tarefas. O aumento na quantidade de nodos, nem sempre impacta diretamente no tempo de execução, isto pode ser visualizado no gráfico da Figura 7.6, onde as execuções com 13 Nodos e 16 nodos, apresentaram tempos de execução muito próximos, sendo ambos utilizam 2 núcleos de processadores.

Os valores de eficiência na execução sobre grades de 13 e 16 nodos resultaram em 71% e 60% de eficiência, respectivamente, valores que possibilitaram a idealização de uma nova funcionalidade para o construtor *ParFor* (Iteração paralela) do projeto D-GM. A relação estabelecida entre o número de processos disponíveis para execução e o número de processadores disponíveis determina um parâmetro que permite se identificar a melhor quantidade de nodos para execução da tarefa. Através deste controle, otimiza-se a distribuição e quantidade de nodos para execução de tarefas, deixando os demais nodos disponíveis para a grade, e ainda não impactando fortemente no tempo de execução da tarefa.

7.3 Conclusão

Este capítulo apresentou a análise sobre os resultados das avaliações dos protótipos *iCone* e *p-iCone*, onde foram descritos os erros inerentes com a execução da metodologia e o ganho de desempenho da exploração de grades de computadores e grades de computadores multi-core, respectivamente.

Os resultados da análise de erros foram satisfatórios, se comparados com a metodologia manual utilizada atualmente pelas indústrias. No que se refere ao ganho de desempenho com o paralelismo, verificou-se que o ganho foi proporcional ao aumento na quantidade de nodos processadores.

8 CONSIDERAÇÕES FINAIS

Neste último capítulo, apresentam-se as conclusões na Seção 8.1 que ressaltam as principais contribuições e caracterizam as oportunidades de trabalho na frente de pesquisa pertinente ao *SisA*³. Na sequência, nas Seções 8.2 e 8.3, tem-se as perspectivas de continuidade do trabalho e as orientações de novas etapas projetos, assim como as publicações geradas contendo os principais resultados alcançados, respectivamente.

8.1 Conclusões

Esta seção descreve as conclusões baseadas na análise, experiência e conhecimento adquiridos com o desenvolvimento deste trabalho. Para facilitar a descrição foram considerados cinco itens na descrição das principais observações conclusivas, as quais estão descritas logo a seguir.

Viabilidade do Sistema Automatizado para Melhorias de Auditorias no Armazenamento de Produção Agrícola de Grãos

Este trabalho introduziu a modelagem e implementação do Sistema Automatizado de Auditoria em Armazéns – *SisA*³, consistindo numa ferramenta integrada de *hardware* e *software*, organizada em dois subsistemas: e o protótipo para cálculo de volumes *iCone* e o sistema digitalizador *DigSisA*³.

O trabalho promoveu melhorias na automação do processo de inventário de produtos estocados a granel, baseada na digitalização de pontos obtidos pelo processo de medição eletrônica a laser e, também, apresenta-se como uma solução computacional para o problema de cálculo do volume em superfícies não uniformes.

O *SisA*³ foi concebido com o objetivo de tornar o processo de auditoria confiável, pois com a automação, varias fontes de erro são modeladas e a partir do método numérico isoladas, a fim de não influenciar no resultado. Devido ao grau de precisão, e ao controle dos erros no processo computacional, os protótipos do *SisA*³, obtidos no desenvolvimento deste trabalho, podem ser disponibilizados para testes em organizações oficiais de fiscalização.

O processo de auditoria neste tipo de armazém geralmente é complexo, pois o produto é descarregado de maneira centralizada, formando cones, e após sucessivas alterações da descarga de produtos no sentido da profundidade de armazém, formam-se intersecções entre os diversos cones. Além disso, nos momentos em que há reti-

rada de produtos, geralmente ocorre a formação de depressões caracterizando a não-uniformidade da superfície que limita superiormente o relevo, causada pelo agrupamento por deterioração ou umidade, por exemplo.

Consolidação do Método Científico para Cálculo de Volume Referente a Relevos Limitados por Superfícies Não Uniformes

Tem-se como principal contribuição deste projeto a concepção, modelagem e implementação da metodologia científica para determinação do volume em superfícies irregulares, as quais neste contexto, estão caracterizadas por produtos estocados em armazéns graneleiros.

A utilização de *splines* polinomiais cúbicas no *iCone* possibilitou uma maior aproximação na modelagem do relevo real, a partir dos pontos da matriz de entrada. Os processos de aproximação do relevo real ocorreram em duas etapas:

- Aumentando a quantidade de camadas, que definem a segmentação em fatias do relevo, e detalhando a correspondente superfície, de forma a considerar a continuidade e curvatura gerada pela sequência de pontos obtidas pelas sucessivas medições;
- Aproximando os pontos de cada seção sobre o relevo, sob a ótica bidimensional, para cálculo da área de cada camada pela integração numérica, e considerando fixada a profundidade, no cálculo do volume de cada fatia do relevo.

A análise de erros efetuada, independentemente a cada etapa, mostra o ganho nas aproximações, validando ambas estratégias de aproximação.

O erro global foi considerado satisfatório. Salienta-se que o processo manual, atualmente utilizado em grande parte das empresas, frequentemente, necessita que seja refeita a auditoria pela ocorrência de divergências entre os estoques físico e fiscal e a imprecisão do cálculo.

Ao se comparar os resultados do sistema *SisA*³ com os resultados obtidos com a metodologia manual, que geralmente ocorre nas empresas, verificou-se possível a substituição das estimativas de previsão do volume com razoável ganho. Este comparativo não está disponível, neste texto, devido a ausência de dados referentes ao erro da metodologia manual.

Desenvolvimento de Melhorias no Método *iCone* pela Exploração do Paralelismo de Dados

A metodologia utilizada no protótipo *iCone* consiste na divisão do problema em sub-problemas mais simples e de mesmo tamanho. Esta estratégia além de apresentar fácil implementação pelo uso de métodos numéricos e técnicas de visualização consolidadas, possibilitou o desenvolvimento de um protótipo para execução paralelizada, caracterizando-se pela não dependência de dados.

O desenvolvimento do protótipo paralelizado *p-iCone* visou ganho de desempenho através da exploração do paralelismo de dados, inerente a forma da modelagem do *iCone*.

Esta abordagem consiste numa alternativa viável para se utilizar os diversos núcleos dos modernos processadores *multi-cores* e/ou sistemas distribuídos de computadores.

Integração com o Ambiente D-GM para Suporte ao Desenvolvimento do *p-iCone* no *SisA*³

A integração com o Projeto D-GM, através da utilização das ferramentas de desenvolvimento (VPE-GM) e de execução (VirD-GM), possibilitou o desenvolvimento e gerência de execução na exploração do paralelismo. Ou seja, pela interface gráfica do editor de processos, obteve-se a modelagem dos processos correspondentes a cada etapa da metodologia *p-iCone*; e a configuração dos estados de memória assim como a inicialização dos valores foi obtida com a utilização do editor de memória. Ambos editores geraram arquivos descritores que foram enviados ao ambiente de execução VirD-GM, que gerenciou o controle e execução da aplicação.

Paralelamente, a aplicação *p-iCone* apresentou-se como um estudo de caso, validando os construtores (produto paralelo, iteração sequencial, iteração paralela) e os operadores no VirD-GM, além de incentivar a sua utilização em outras aplicações científicas.

Especificamente, o projeto *SisA*³ contribuiu com funcionalidades e novos requisitos a serem desenvolvidas no projeto D-GM, tais como:

- Validação dos construtores utilizados no desenvolvimento do protótipo, onde através da correta execução do algoritmo e análise das funcionalidades e comportamento de tais estruturas;
- Incentivo ao desenvolvimento do suporte a computação multiprocessada pela VirD-GM visando aumento de desempenho, não somente pela exploração da distribuição de diversos dispositivos computacionais presentes no ambiente de grade, mas também pela exploração do paralelismo presente nos nodos multiprocessados. Esta funcionalidade já está disponível, porém de maneira isolada. O foco atual desta alteração está na criação de uma versão híbrida, ou seja, distribuída e multi-core;
- Exploração e aplicação dos construtores de processos: produto sequencial e produto paralelo, iteração e iteração paralela;
- Introdução do conceito de multiplicidade na escolha dos núcleos que executarão as tarefas definidas no construtor de iteração paralela, de forma a obter o máximo desempenho com a utilização do mínimo de nodos. A especificação da multiplicidade consiste num dos módulos em desenvolvimento no VirD-GM, com base na análise detalhada sobre os resultados das execuções do protótipo *p-iCone*.
- Melhoria na distribuição de tarefas em processadores multi-core. A simples inclusão de uma maior quantidade de núcleos não garante melhoria de desempenho, incentivando análise de características dos computadores e políticas de escalonamento.

Caracterização Multidisciplinar do Projeto e Integração com Diferentes Áreas de Pesquisa e Desenvolvimento Tecnológico

Os estudos multidisciplinares nas diversas áreas englobadas no projeto foram uma etapa indispensável para a definição da melhor estratégia para modelagem e desenvolvimento dos protótipos. Neste aspecto, a integração dos grupos de pesquisa foi essencial, onde salientam-se:

- Os estudos e sistemas do grupo GMFC possibilitaram o desenvolvimento do método iterativos de cálculo, interpolação polinomial e cálculo numérico;
- O trabalho nos laboratórios de Biomédica da UCPEL junto ao grupo GPDIB auxiliaram na concepção, modelagem e montagem do protótipo *DigSisA*³, juntamente com as empresas Contronic e Tornearia Schulz;
- As técnicas e equipamentos disponibilizados junto ao grupo G3PD (Grupo de Pesquisa em Processamento Paralelo e Distribuído) contribuindo com as ferramentas de exploração do paralelismo utilizadas no projeto D-GM.

Também foi relevante a integração de esforços de pesquisa entre docentes e discentes do Mestrado em Ciência da Computação do PPGINF/UCPEL com o Departamento de Informática da UFPEL, além das empresas agrícolas da região Sul do país que contribuíram com a liberação de acesso aos silos e armazéns, levantamento de requisitos, e seus conhecimentos nos processos de auditorias e armazenagem de grãos.

8.2 Continuidade do Projeto

O sistema *SisA*³ apresenta ainda diversas oportunidades a serem exploradas, e algumas destas já foram traduzidas para trabalhos. Neste sentido, os principais temas de pesquisas em curso são

Consideram-se, logo a seguir, a relação das funcionalidades que deverão ser desenvolvidas e testadas, compreendendo desde a obtenção da coleta de pontos automática até a melhoria do desempenho do algoritmo que implementa o protótipo *p-iCone*, incluindo sua completa disposição funcional:

- Estudo e desenvolvimento de uma política de escalonamento que viabilize a exploração do máximo desempenho na distribuição de tarefas em ambientes de grades de computadores multiprocessados, integrada ao mínimo consumo de energia, no VirD-GM. Para tal, este trabalho considera:
 - Avaliação da capacidade dos nodos com base em critérios como a quantidade de núcleos, consumo energético e o tamanho de memórias, bem como a relação de desempenho para cada perfil de aplicação em cada nodo.
 - Criação de regra de multiplicidade, correspondendo a relação entre o número de processos paralelizáveis no VirD-GM e o número de processadores disponíveis, visando economia de energia, e assim, também disponibilizar nodos ociosos para a grade.
 - Busca da melhor relação de desempenho de processamento com o menor consumo de energia.
 - Avaliação dos nodos associada aos diferentes perfis de aplicação (*CPU intensive*, *Memory intensive*, Construção Híbrida, etc.) e ao correspondente consumo energético.
- Envio da nuvem de pontos via comunicação sem fio, além do retorno da computação, a ser obtido pelo estudo e aplicação de técnicas de computação móvel.

- Complementação da integração e realização de testes de *softwares* e *hardwares* para validação completa do sistema *SisA³*.
- Criação de Fatores de correção, pois na construção de armazéns, podem existir deformações na esteira onde o digitalizador será acoplado para medição, desta forma, fazendo-se necessário a correção dos ângulos de medição, a fim de que o *DigSisA³* não capture pontos fora do escopo da medição.
- Análise na deformação referente às diferenças angulares no sentido X e no sentido do eixo Y. Os fatores de correção, denominados *fatX(Z)* e *fatY(Z)*, são funções aplicáveis às medidas e assim corrigindo um possível erro estrutural do armazém.

Seguem-se outras atividades que dão sequência ao trabalho:

- Extensão do *SisA³* incluindo outras metodologias (AL-ALAM; REISER; PILLA, 2008).
- Teste com outras formas geométricas, inclusive com um relevo real, de volume já conhecido, com participação de empresas locais.
- Análise detalhada dos erros envolvidos no processo de digitalização com a utilização do *DigSisA³*.
- Desenvolvimento de um ambiente de visualização e navegação sobre o relevo digital.

A continuidade do trabalho tem como principal meta a geração de um produto que engloba *software* e *hardware*, com viabilidade comercial, e que satisfaça as necessidades de empresas da área de distribuição, armazenamento e estocagem de produtos a granel.

8.3 Publicações Realizadas

Esta seção considera a produção técnica e organização da documentação para publicação tanto do *software* do Projeto *SisA³*, como dos protótipos desenvolvidos com o trabalho:

- Publicação em Livros e Revistas
 - Wagner G. Al-Alam, Renata H. S. Reiser, Adenauer C. Yamin, Maurício L. Pilla. *SisA3: Sistema Integrado de Auditoria em Armazéns Graneleiros*. In: Revista Diálogo. UNILASALLE, 2009. Canoas-RS.
 - Wagner G. Al-Alam, Renata H. S. Reiser, Adenauer C. Yamin, Maurício L. Pilla. *SisA3: Uma Arquitetura HWSW para Auditoria em Armazéns de Granéis* In: Desafios e Avanços em Computação, inovações e tecnologia. UFPEL, 2009. Pelotas-RS.
- Publicações em Eventos
 - Eventos Latino-Americanos

- * Wagner G. Al-Alam, Renata H. S. Reiser, Adenauer C. Yamin, Maurício L. Pilla. *SisA3: Uma arquitetura HW/SW para Auditoria em Armazéns de Granéis*. In: XXXV LATIN-AMERICAN CONFERENCE ON INFORMATICS, 2009, Pelotas - RS.

– Eventos Nacionais

- * Wagner G. Al-Alam, Renata H. S. Reiser, Adenauer C. Yamin, Maurício L. Pilla. *p-iCone: Uma contribuição para o mecanismo de distribuição da produção agrícola*. In: WCAMA - I Workshop de Computação Aplicada à Gestão do Meio Ambiente e Recursos Naturais - Bento Gonçalves - RS.

– Eventos Regionais

- * Wagner G. Al-Alam, Renata H. S. Reiser, Maurício L. Pilla. *Estudo de Metodologias para Cálculo Numérico do Volume de Produtos em Armazéns*, Proceedings of Ermac 2008: VIII Encontro Regional de Matemática Aplicada e Computacional, 2008, UFPEL - Pelotas-RS.
- * Wagner G. Al-Alam, Renata H. S. Reiser, Adenauer C. Yamin, Maurício L. Pilla. *p-iCone: Paralelização de uma Metodologia para Cálculo do Volume de Produtos em Armazéns*. In: IX Escola Regional de Alto Desempenho, 2009. Anais IX Escola Regional de Alto Desempenho. Caxias do Sul - RS: UCS, 2009.
- * Wagner G. Al-Alam, Renata H. S. Reiser, Adenauer C. Yamin, Maurício L. Pilla. *SisA³: Sistema Automático de Auditoria de Armazéns* In: X Escola Regional de Alto Desempenho, 2010. Aceito para publicação. Passo Fundo - RS: UPF, 2010.
- * Tâmara Nascimento, Renata H. S. Reiser, Maurício Tavares, Wagner G. Al-Alam. *Arquitetura de SW e HW do Dig-SisA* In: X Escola Regional de Alto Desempenho, 2010. Aceito para publicação. Passo Fundo - RS: UPF, 2010.
- * Marcelo Würdig, Felipe Munhoz, Wagner G. Al-Alam, Renata Reiser, Adenauer Yamin. *VirD-GM: Modelagem e Funcionalidades do Construtor Iterativo Paralelo* In: X Escola Regional de Alto Desempenho, 2010. Aceito para publicação. Passo Fundo - RS: UPF, 2010.

– Resumos publicados em Eventos Nacionais

- * Wagner G. Al-Alam, Renata H. S. Reiser, Adenauer C. Yamin, Maurício L. Pilla. *Sistema Automatizado de Auditoria em Armazéns*. In: WCAMA - I Workshop de Computação Aplicada à Gestão do Meio Ambiente e Recursos Naturais, 2009. Bento Gonçalves - RS.

– Resumos publicados em Eventos Locais

- * Wagner G. Al-Alam, Renata H. S. Reiser, Adenauer C. Yamin, Maurício L. Pilla. *Aplicação do Processamento Paralelo no Cálculo de Volume de Armazéns Horizontais de Granéis*. In: VII Mostra de Pós Graduação - UCPEL - , 2008. Anais do XVII CIC 2008. Pelotas - RS.
- * Wagner G. Al-Alam, Renata H. S. Reiser, Adenauer C. Yamin, Maurício L. Pilla. *SISTEMA AUTOMATIZADO DE AUDITORIA EM ARMAZÉNS*

In: VIII Mostra de Pós Graduação - UCPEL - , 2009. Anais do XVIII CIC 2009. Pelotas - RS.

8.4 Contribuição em Projetos de Pesquisa

Com base nos resultados alcançados e já publicados quando do desenvolvido desta dissertação, outras etapas foram concebidas como continuidade ao desenvolvimento do Projeto D-GM e poderão contar com apoio e suporte conquistado junto ao Edital MCT/CNPq 15/2007 – Universal Faixa A, processo de número 476933/2007-2, e ao Edital CNPq 04/2008 – Apoio Técnico, processo 502999/2008-0.

Neste contexto, o trabalho apoiou também a orientação de duas bolsas de iniciação científica, as quais referem-se a: *(i)* aluna da Engenharia Elétrica da UCPEL, a qual desenvolve seus trabalhos junto aos laboratórios de Biomédica da UCPEL com apoio do grupo GPDIB; e *(ii)* aluno da Ciência da Computação que desenvolve suas atividades no NAPI com o apoio do grupo G3PD.

REFERÊNCIAS

ABRAMSKY, S.; JUNG, A. Domain Theory. In: **Handbook of Logic in Computer Science**. [S.l.]: Clarendon Press, 1994.

AL-ALAM, W. G.; REISER, R. S.; PILLA, M. L. Estudo de Metodologias para Cálculo Numérico do Volume de Produtos em Armazéns. In: VIII ENCONTRO REGIONAL DE MATEMÁTICA APLICADA E COMPUTACIONAL, 2008, Pelotas / RS. **Anais...** [S.l.: s.n.], 2008.

ALMEIDA, M.; REISER, R. Especificação baseada em UML para o Ambiente Computacional SisA3. In: CIC-UCPEL 2009, 2009. **Anais...** [S.l.: s.n.], 2009.

AMARAL, R. B. **Aplicações do Modelo qGM na Interpretação de Algoritmos Quânticos**. 2009. Dissertação para a obtenção do grau de Mestre em Ciência da Computação — , Pelotas-RS.

BARTELS, R. H.; BEATTY, J. C.; BARSKY, B. A. **Introduction to Splines in Computer Graphics and Geometric Modeling**. [S.l.]: Morgan Kaufmann Publishers, INC., 1987.

CHAROENPONG, T.; CHAMNONGTHAI, K.; KAMHOM, P.; KRAIRIKSH, M. **Volume Measurement of Mango by Using 2D Ellipse Model**. Thailand: King Mongkut's University of Technology Thonburi, 2004.

CHEN, F.; BROWN, G. M.; SONG, M. Overview of three-dimensional shape measurement using optical methods. **Optical Engineering**, [S.l.], v.39, n.1, p.10–22, 2000.

CLÁUDIO, D. M.; MARINS, J. M. **Calculo numérico computacional : teoria e pratica**. São Paulo: atlas, 1994.

COMPUTER AND COMPUTING TECHNOLOGIES IN AGRICULTURE, VOLUME I, FIRST IFIP TC 12 INTERNATIONAL CONFERENCE ON COMPUTER AND COMPUTING TECHNOLOGIES IN AGRICULTURE (CCTA 2007), WUYISHAN, CHINA, AUGUST 18-20, 2007, 2008. **Anais...** Springer, 2008. (IFIP, v.258).

DISTAGE. **General - Laser Distance Meter**. <http://www.distagage.com/gequ.html>: [s.n.], 2003.

FOLEY, J. D. **Computer Graphics: Principles and Practice**. New York: Addison Wesley, 1990.

FONSECA, V. S. **VirD-GM: Uma Contribuição Para o Modelo de Distribuição e Paralelismo do Projeto D-GM.** 2008. Dissertação para a obtenção do grau de Mestre em Ciência da Computação — , Pelotas-RS.

FONSECA, V. S. **VirD-GM: Uma Contribuição Para o Modelo de Distribuição e Paralelismo do Projeto D-GM.** 2008. Dissertação de Mestrado — Mestrado em Ciência da Computação/UCPel, Pelotas/RS.

FONSECA, V. S.; REISER, R. H. S.; YAMIN, A. C.; PILLA, M. L. VirD-GM: Towards a Grid Computing Environment. In: CCGRID 2007, 2007. **Proceedings...** [S.l.: s.n.], 2007. p.1–6.

FONSECA, V. S.; REISER, R. H. S.; YAMIN, A. C.; PILLA, M. L. O Modelo de Máquina Geométrica: Revisão das Potencialidades do Paralelismo. In: ERAD 2007, 2007. **Proceedings...** [S.l.: s.n.], 2007. p.1–3.

FORESTS, M. of; RANGE. **Scaling Manual.** British Columbia: British Columbia, 2007.

GIRARD, J.; LAFONT, Y.; TAYLOR, P. M. **Visual Object-Oriented Programming.** Cambridge: Cambridge University Press, 1989. 176p.

JONES, D. W. **Control of Stepping Motors.** Disponível em <<http://www.cs.uiowa.edu/jones/step/>> Acesso em: outubro 2008.

KAUFMAN, A. Memory and Processing Architecture for 3D Voxel-based Imagery. In: IEEE COMPUTER GRAPHICS AND APPLICATIONS, VOL. 8, NO. 6, 1988. **Anais...** [S.l.: s.n.], 1988. p.33–40.

KAUFMAN, A. Volume Visualization: Principles and Advances. In: SIGGRAPH 97 COURSE NOTES, 1997. **Anais...** [S.l.: s.n.], 1997.

KAUFMAN, A.; YAGEL, R.; COHEN, D. Intermixing Surface and Volume Rendering. In: D IMAGING IN MEDICINE, ALGORITHMS, SYSTEMS, APPLICATIONS, 3., 1990, Berlin. **Anais...** Eds.: H. H. Hohne: H. Fuchs and S. M. Pizer: Springer Verlag, 1990. p.217–227.

KENDALL, E. A. **Na Introduction to Numerical Analysis.** New York: John Wiley and Sons, 1978.

KIREHNER, F. F.; FILHO, A. F.; SCOLFORO, J. R. S.; MACHADO, S.; MITISHITA, E. A. **O uso de funções splines no cálculo de volume de árvores.** Curitiba/PR: Revista Floresta, 1989.

KREYSZIG, E. **Advanced Engeneering Mathematics.** New York: John Wiley and Sons, 1993.

LEVOY, M. Display of Surfaces from Volume Data. In: IEEE COMPUTER GRAPHICS AND APPLICATIONS, 1988. **Anais...** [S.l.: s.n.], 1988. p.29–37.

LIN, Y.; FU, Y. The Key of Bulk Warehouse Grain Quantity Recognition. In: CCTA, 2007. **Anais...** Springer, 2007. p.543–551. (IFIP, v.258).

LIN, Y.; JIANG, X. The Research on Grain Reserve Intelligent Audit Method and Implementation in Three-dimensional Stores. In: CCTA, 2007. **Anais...** Springer, 2007. p.525–532. (IFIP, v.258).

LORENSEN, W. E.; CLINE, H. E. Marching Cubes: A High Resolution 3D Surface Construction Algorithm. In: COMPUTER GRAPHICS (PROC. SIGGRAPH), VOL. 21, NO. 4, 1987. **Anais...** [S.l.: s.n.], 1987. p.163–169.

MADEHOW. **How Products Are Made,Hourglass.** Disponível em: <<http://www.madehow.com/Volume-5/Hourglass.html>> Acesso em: outubro 2008.

MUNHOZ, F. N. **Expandindo o VirD-GM para Suporte a Novos Construtores do Modelo D-GM.** 2009. Monografia de Graduação — , Pelotas-RS.

MUNHOZ, F. N.; FONSECA, V. S.; REISER, R. H. S.; PILLA, M. L.; YAMIN, A. C. Paralelismo no modelo D-GM: Emprego de Grafos de Dependências. In: WSCAD, 2007. **Anais...** [S.l.: s.n.], 2007. p.1–4.

MUNHOZ, F. N.; VIVAN, G.; FONSECA, V. S.; REISER, R. H. S.; PILLA, M. L.; YAMIN, A. C. Arquitetura de Software da VirD-GM: Modelagem e Funcionalidades. In: ERAD, 2008. **Anais...** [S.l.: s.n.], 2008. p.1–4.

PAIVA, A. C.; SEIXAS, R. B.; GATTAS, M. **Introdução à Visualização Volumétrica.** [S.l.]: PUC-RioInf.MCC03/99, UFMA/LNCC/PUC-Rio, 1999.

PILLA, M. L.; SANTOS, R. R.; CAVALHEIRO, G. G. H. Introdução à programação para arquiteturas multi-core. In: ESCOLA REGIONAL DE ALTO DESEMPENHO, 9., 2009, Caxias do Sul / RS. **Anais...** [S.l.: s.n.], 2009.

PRESTES, D. **Ambiente de Programação Visual para o Modelo de Máquina Geométrica.** Ciência da Computação - UCPel, 106p. (Monografia de Projeto de Graduação).

PRESTES, D.; CARDOSO, M.; REISER, R.; COSTA, A. Implementando o Ambiente de Programação Visual para a Máquina Geométrica. **I Workcomp Sul - Workshop de Ciências da Computação e Sistemas de Informação da Região Sul**, [S.l.], p.1–11, 2004.

PRESTES, D.; REISER, R. **Modelagem e Implementação do Editor de Processos para o Modelo de Máquina Geométrica.** Ciência da Computação - UCPel, 66p. (Relatório de bolsista PIBIC/CNPq, 2004).

PRESTES, D.; REISER, R. **Implementando o Simulador para o Modelo de Máquina Geométrica.** Ciência da Computação - UCPel, 96p. (Relatório de bolsista PIBIC/CNPq, 2005).

PRESTES, D.; REISER, R.; COSTA, A. Apresentando o Editor para o Ambiente de Programação Visual para a Máquina Geométrica. In: SIRC 2004 - SIMPÓSIO DE INFORMÁTICA DA REGIÃO CENTRO DO RS, 2004, Santo Maria, RS. **Anais...** UNIFRA-RS, 2004. p.1–5.

PRESTES, D.; REISER, R.; COSTA, A. O Simulador para o Ambiente de Programação Visual para o Modelo de Máquina Geométrica. In: CNMAC 2005 - XXVIII CONGRESSO NACIONAL DE MATEMÁTICA APLICADA E COMPUTACIONAL, 2005, Santo Amaro, SP. **Anais...** SBMAC-RJ, 2005. p.175–176. meio digital.

PRESTES, D.; REISER, R.; COSTA, A.; CARDOSO, C. Estendendo o Modelo de Máquina Geométrica a um Ambiente de Programação Visual. In: CLEI2005 XXXI CONFERÊNCIA LATINOAMERICANA DE INFORMÁTICA, 2005, Cali. **Anais...** Universidad Javeriana, 2005. p.1–10.

RATTNER, J. **Teraflops Research Chip**. Disponível em: <<http://techresearch.intel.com/articles/Tera-Scale/1449.htm>> Acesso em: junho 2009.

REISER, R. **A máquina geométrica - um modelo computacional para concorrência e não-determinismo usando como estrutura os Espaços Coerentes**. 2002. 254p. Tese (Doutorado em Ciência da Computação) — CPGCC/UFRGS - Porto Alegre.

REISER, R.; COSTA, A.; AMARAL, R. Leading to Quantum Semantic Interpretations based on Coherence Spaces. In: NANOBIO 2007 - II WORKSHOP EM NANOTECNOLOGIA E COMPUTAÇÃO INSPIRADA EM BIOLOGIA, 2007. **Anais...** PUC/RJ, 2007. p.1–6. meio digital.

REISER, R.; COSTA, A.; AMARAL, R. Quantum Computing: Computation in Coherence Spaces. In: PROCEEDINGS OF WECIQ WORKSHOP AND SCHOOL OF QUANTUM INFORMATION AND COMMUNICATION, 2., 2007. **Anais...** UFCG - PR, 2007. p.1–10.

REISER, R.; COSTA, A. C. R.; DIMURO, G. Distributed approach for the Geometric Machine Model. In: PARA 2004 - WORKSHOP ON STATE-OF-THE-ART IN SCIENTIFIC COMPUTING VALIDATED, 2004, Ligby. **Anais...** [S.l.: s.n.], 2004. n.1, p.106–114.

REISER, R.; COSTA, A.; DIMURO, G. First Steps in the Construction of the Geometric Machine Model. In: **TEMA - Tendências em Matemática Aplicada e Computacional**. [S.l.: s.n.], 2002. v.3, n.1, p.183–192.

REISER, R.; COSTA, A.; DIMURO, G. Programming in the Geometric Machine. In: **Frontiers of Artificial Intelligence**. Amsterdam: IOS Press, 2003. p.95–102.

REISER, R.; COSTA, A.; DIMURO, G. The distributed version of the geometric machine model. In: XXVI CONGRESSO NACIONAL DE MATEMÁTICA APLICADA E COMPUTACIONAL, 2003. **Anais...** SBMAC-RJ, 2003. p.163–163. meio digital.

REISER, R.; COSTA, A.; DIMURO, G. The Stochastic Geometric Machine Model. In: XXVI CONGRESSO NACIONAL DE MATEMÁTICA APLICADA E COMPUTACIONAL, 2003, São José do Rio Preto, Brazil. **Anais...** SBMAC-RJ, 2003. p.175–176. meio digital.

REISER, R.; COSTA, A.; DIMURO, G. A programming language for the Interval Geometric Machine Model. In: **Electronic Notes in Theoretical Computer Science**. [S.l.: s.n.], 2003. v.84, p.1–12.

REISER, R.; COSTA, A.; DIMURO, G. O modelo de Máquina Geométrica Intervalar. In: **TEMA - Tendências em Matemática Aplicada e Computacional**. [S.l.: s.n.], 2003. v.4, n.1, p.109–118.

REISER, R.; COSTA, A.; DIMURO, G. The Stochastic Geometric Machine Model. In: **TEMA**. [S.l.: s.n.], 2004. v.1, n.3, p.1–10.

REISER, R.; COSTA, A.; DIMURO, G. The Stochastic Geometric Machine Model. In: **TEMA TENDÊNCIAS EM MATEMÁTICA APLICADA E COMPUTACIONAL**, 2004, Rio de Janeiro, Brazil. **Anais...** SBMAC/RJ, 2004. v.5, n.2, p.305–314.

REISER, R.; COSTA, A.; DIMURO, G. The Interval Geometric Machine. In: **Numerical Algorithms**. Dordrecht: Kluwer, 2004. v.37, p.357–366.

REISER, R.; COSTA, A.; DIMURO, G. The Interval Geometric Machine. In: **Numerical Algorithms**. [S.l.]: Dordrecht, 2004. p.357–366.

REISER, R.; COSTA, A.; DIMURO, G. The Quantum Version of The Geometric Machine Model. In: **XXVII CONGRESSO NACIONAL DE MATEMÁTICA APLICADA E COMPUTACIONAL**, 2004. **Anais...** SBMAC-RJ, 2004. p.175–176. meio digital.

REISER, R.; COSTA, A.; DIMURO, G. The Distributed Interval Geometric Machine Model. In: **LECTURE NOTES IN COMPUTER SCIENCE**, 2005. **Anais...** Springer, 2005. n.3732, p.179–188.

REISER, R.; COSTA, A.; DIMURO, G. Programming in the Quantum Geometric Machine Model. In: **XXVIII CONGRESSO NACIONAL DE MATEMÁTICA APLICADA E COMPUTACIONAL**, 2005. **Anais...** SBMAC/SENAC-RJ, 2005. p.1–6. meio digital.

SAHNI, S.; THANVANTRI, V. **Performance Metrics: Keeping the Focus on Runtime**. (disponível via WWW em <http://cs.nju.edu.cn/gchen/teaching/hpca/reference/sahni96.pdf>).

SCHMIDT, A. E. F. **Visualização Tridimensional combinada de dados volumétricos e modelos poligonais usando o algoritmo shear-warp**. 2000. Tese de Doutorado em Ciência da Computação — PUC-Rio, Rio de Janeiro.

SEIXAS, R.; GATTAS, M. Visualização Volumétrica em um Ambiente de Computação Distribuída. In: **SIBGRAPI 1996**, 1996, Caxambu/MG. **Anais...** [S.l.: s.n.], 1996. p.15–22.

SILVA, L. K. M. **UM SISTEMA DE MODELAGEM GEOMÉTRICA USANDO SPLINES**. Recife/PE: Universidade Católica de Pernambuco, 2007.

SPERANDIO, D.; MENDES, T. J.; SILVA, M. L. H. **Cálculo Numérico, Características Matemáticas e Computacionais dos Métodos Numéricos**. São Paulo/SP: Pearson Prentice Hall, 2006.

VISION RAYTRACER PTY. LTD., P. of. **POV-Ray 3.6 Documentation Online View**. (disponível via WWW em <http://www.povray.org/documentation/view/3.6.1/279/>).

WESTOVER, L. Footprint Evaluation for Volume Rendering. In: COMPUTER GRAPHICS, VOL. 24, NO. 4, 1990. **Anais...** [S.l.: s.n.], 1990. p.367–376.

YAGEL, R. Volume Viewing: State of the Art Survey. In: IN COURSE NOTES, SIGGRAPH., 1996. **Anais...** [S.l.: s.n.], 1996.

YAMIN, A. **Arquitetura para um Ambiente de Grade Computacional Direcionado às Aplicações Distribuídas, Móveis e Conscientes do Contexto da Computação Pervasiva**. 2004. 195p. Tese (Doutorado em Ciência da Computação) — Instituto de Informática/UFRGS - Porto Alegre.

YAMIN, A. **Arquitetura para um Ambiente de Grade Computacional Direcionado às Aplicações Distribuídas, Móveis e Conscientes do Contexto da Computação Pervasiva**. 2004. 195p. Tese de Doutorado em Ciência da Computação — UFRGS, Porto Alegre/RS.

ZAFALON, M.; MASCHIO, J.; ALBANO, M. **Sobe Participação do Agronegócio no PIB**. (disponível via WWW em <http://www.agr.feis.unesp.br/fsp14032004.php>).