

**UNIVERSIDADE DE BRASÍLIA
FACULDADE DE TECNOLOGIA
DEPARTAMENTO DE ENGENHARIA MECÂNICA**

**SISTEMA COMPUTACIONAL PARA TRATAMENTO DE
NUVENS DE PONTOS E RECONSTRUÇÃO
TRIDIMENSIONAL DE SUPERFÍCIES BASEADA EM
MODELOS DEFORMÁVEIS**

GERARDO ANTONIO IDROBO PIZO

**ORIENTADOR: JOSÉ MAURÍCIO SANTOS TORRES DA MOTTA
DISSERTAÇÃO DE MESTRADO EM SISTEMAS MECATRÔNICOS**

**PUBLICAÇÃO: ENM.DM-027A /09
BRASÍLIA/DF: AGOSTO – 2009**

**UNIVERSIDADE DE BRASÍLIA
FACULDADE DE TECNOLOGIA
DEPARTAMENTO DE ENGENHARIA MECÂNICA**

**SISTEMA COMPUTACIONAL PARA TRATAMENTO DE NUVENS
DE PONTOS E RECONSTRUÇÃO TRIDIMENSIONAL DE
SUPERFÍCIES BASEADA EM MODELOS DEFORMÁVEIS**

GERARDO ANTONIO IDROBO PIZO

**DISSERTAÇÃO SUBMETIDA AO DEPARTAMENTO DE
ENGENHARIA MECÂNICA DA FACULDADE DE TECNOLOGIA
DA UNIVERSIDADE DE BRASÍLIA COMO PARTE DOS
REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE
MESTRE EM SISTEMAS MECATRÔNICOS.**

APROVADA POR:

**Prof. José Maurício Santos Torres Da Motta - PhD (ENM-UnB)
(Orientador)**

**Prof. Carlos Humberto Llanos Quintero PhD (ENM-UnB)
(Examinador Interno)**

**Prof. Carla Maria Chagas e Cavalcante Koike - PhD (CIC-UnB)
(Examinador Externo)**

BRASÍLIA/DF, 18 DE AGOSTO DE 2009

FICHA CATALOGRÁFICA

IDROBO PIZO, GERARDO ANTONIO

Sistema Computacional para Tratamento de Nuvens de Pontos e Reconstrução Tridimensional de Superfícies Baseada em Modelos Deformáveis: [Distrito Federal] 2009.

xvii, 126p. 210 x 297 mm (ENM/FT/UnB, Mestre, Sistemas Mecatrônicos, 2009).

Dissertação de Mestrado – Universidade de Brasília. Faculdade de Tecnologia.

Departamento de Engenharia Mecânica.

1. Nuvem de Pontos

2. Modelos Deformáveis

3. Visão Computacional

4. Construção de superfícies 3D

I. ENM/FT/UnB

II. Título (série)

REFERÊNCIA BIBLIOGRÁFICA

IDROBO PIZO, G. A. (2009). Sistema Computacional para Tratamento de Nuvens de Pontos e Reconstrução Tridimensional de Superfícies Baseada em Modelos Deformáveis. Dissertação de Mestrado em Sistemas Mecatrônicos, Publicação Agosto/2009, Departamento de Engenharia Mecânica, Universidade de Brasília, Brasília, DF, 126 p.

CESSÃO DE DIREITOS

AUTOR: Gerardo Antonio Idrobo Pizo

TÍTULO: Sistema Computacional para Tratamento de Nuvens de Pontos e Reconstrução Tridimensional de Superfícies Baseada em Modelos Deformáveis.

GRAU: Mestre

ANO: 2009

É concedida à Universidade de Brasília permissão para reproduzir cópias desta dissertação de mestrado e para emprestar ou vender tais cópias somente para propósitos acadêmicos e científicos. O autor reserva outros direitos de publicação e nenhuma parte dessa dissertação de mestrado pode ser reproduzida sem autorização por escrito do autor.

Gerardo Antonio Idrobo Pizo

Departamento de Engenharia Mecânica,
Campus Universitário Darcy Ribeiro
70.910-900 - Brasília - DF - BRASIL

DEDICATÓRIA E AGRADECIMENTOS

Primeiramente quero dedicar este trabalho como símbolo de vitória a Deus e agradecer de todo alma e coração pela vida, pela saúde e pela capacidade intelectual a mim concedida. Também quero dedicar esta triunfo a minha mãe a senhora Cristobalina Pizo minhas irmãs Amparo, Teresa e Viviana, bem como minhas sobrinhas e sobrinho Miyareth, Marcela e Miller, fontes da minha motivação e inspiração.

Novamente, quero agradecer a Deus pela força e perspicácia concedida a este humilde trabalhador incansável, que luta todos os dias na busca e domínio do conhecimento. Quero agradecer a oportunidade de ter trabalhado com meu amigo e orientador prof. PhD José Maurício Santos Torres da Motta, pela confiança, apoio e incentivo em mim depositados e pelo compartilhamento de sua sabedoria.

Também quero agradecer a Universidade de Brasília, bem como ao programa de Pós-graduação em Sistemas Mecatrônicos, por me oferecerem as condições necessárias para realização deste trabalho.

Agradeço a toda equipe do laboratório Graco (professores, técnicos e alunos) pelos conhecimentos transmitidos e pela amizade oferecida. A Capes, pela concessão da bolsa e ao Brasil como um todo pela oportunidade que me foi dada. Assim mesmo aos professores Sonia Gaona e Jauri Leon Tellez por terem me incentivado e apoiado na realização dos meus estudos de pós-graduação no Brasil.

Novamente agradeço a Deus por estas conquistas e pelas bênçãos recebidas, olhando ao céu e gritando em voz alta, com uma felicidade infinita em meu coração.

Muitíssimo obrigado Deus.

RESUMO

A reconstrução tridimensional é uma tarefa não trivial que em geral envolve cinco etapas: aquisição, registro, integração, segmentação e ajuste, sendo esta última etapa a que proporciona o modelo computacional do objeto representado (Yvart *et al.*, 2005).

Definir a qualidade de uma representação tridimensional do objeto é uma tarefa complexa quando não são conhecidas as relações de conectividade entre os pontos que compõem a amostra e, principalmente, quando a nuvem de pontos está contaminada com interferências desconhecidas que estão sujeitas de modo inerente no processo de aquisição. Essas interferências podem ser eliminadas ou atenuadas através do processo de filtragem. O principal objetivo da filtragem de sinais é melhorar a qualidade do sinal de acordo com um determinado critério de desempenho (Muller, 1992).

O presente trabalho aborda o tratamento de ruídos através de filtros adaptativos. Muitas dessas interferências que são encontradas em nuvens de pontos mudam em cada processo de aquisição e não podem atenuar-se com sucesso através de filtros digitais convencionais, passa-baixas, passa-altas ou passa-faixas (Manolakis *et al.*, 2000).

Esta dissertação aborda a combinação de duas técnicas para modelar superfícies de objetos 3D: a reconstrução de superfícies por malhas triangulares e por funções paramétricas - NURBS (*Non-Uniform Rational B-Splines*) (Farin, 2001; Anderson & Crawford-Hines, 2000; Roth & Boulanger, 1998).

Do mesmo modo são também descritos o ajuste dos entalhes paramétricos através do método iteração cíclica de Newton a fim de minimizar a distância entre a malha estruturada e os entalhes paramétricos (Elsäesser, 1998). Igualmente se descreve a avaliação de condições de fronteira para a suavização de funções paramétricas sobre toda a superfície (Hoppe *et al.*, 1992).

Essas técnicas transformam-se conjuntamente em uma ferramenta computacional de propósito específico para a construção de superfícies através da nuvem de pontos.

O desempenho do método aplicado foi avaliado em função da precisão de ajuste obtida e comparado metrologicamente com um sistema padrão, o que permitiu mostrar através do experimento que o software final serve como um instrumento de medição.

RESUMEN

La reconstrucción tridimensional es una tarea no trivial que generalmente envuelve cinco etapas: adquisición, registro, integración, segmentación e ajuste, siendo esta última etapa a que proporciona o modelo computacional do objeto representado (Yvart *et al.*, 2005).

Definir a calidad de una representación tridimensional de un objeto es una tarea compleja cuando no son conocidas las relaciones de conectividad entre los puntos que componen la nube de puntos y, principalmente, cuando está se encuentra contaminada con interferencias desconocidas que están sujetas de modo inherente en el proceso de adquisición. Estas interferencias pueden ser eliminadas o atenuadas a través de procesos de filtraje. O principal objetivo da filtraje de señales es mejorar a calidad de la señal de acuerdo con un determinado criterio de desempeño (Muller, 1992).

El presente trabajo aborda el tratamiento de ruidos a través de filtros adaptativos. Muchas de esas interferencias, que son encontradas en la nube de puntos, cambian en cada proceso de adquisición y no pueden atenuarse con suceso a través de filtros digitales convencionales: Pasa-bajas, pasa-altas o pasa-banda (Manolakis *et al.*, 2000).

Esta trabajo aborda a combinación de dos técnicas para modelar superficies de objetos 3D: a reconstrucción de superficies por mallas Triangulares e por funciones para-métricas - NURBS (*Non-Uniform Rational B-Splines*) (Farin, 2001; Anderson & Crawford-Hines, 2000; Roth & Boulanger, 1998).

Del mismo modo son también descritos el ajuste de los parches para-métricos a través da iteración cíclica de Newton con el fin de minimizar la distancia entre a malla estructurada y los parches para-métricos (Elsäesser, 1998). Igualmente se describe a la evaluación de condiciones de frontera para la suavización de funciones para-métricas sobre toda la superficie (Hoppe *et al.*, 1992).

Estas técnicas se transforman conjuntamente en una herramienta computacional de propósito específico para a construcción de superficies a través de nube de puntos.

El desempeño del método aplicado es evaluado en función de la precisión de ajuste obtenida y es comparada metrologicamente con un sistema estándar, permitiendo mostrar a través de un experimento que o software final sirva como un instrumento de medición.

ABSTRACT

The three-dimensional reconstruction is a nontrivial task that usually involves five steps: acquisition, registration, integration, segmentation and adjustment, the latter stage providing a steps computer model of the objects represented (Yvart *et al.*, 2005).

Defining the quality of a three-dimensional representation of an object is a complex task when the connectivity relations between the points that form the cloud of points are unknown. The cloud of points is subject to pollution with unknown interference that are inherent to the acquisition process. These interferences can be eliminated or mitigated through filtering processes. The main objective of these processes is to filter the signals to improve signal quality according to performance criteria (Muller, 1992).

This dissertation addresses the treatment of noise through adaptive filters. Many of these interferences change in the process of acquisition and can not be successfully mitigated through fixed digital filters: low-pass, high pass, band-pass (Manolakis *et al.*, 2000).

This dissertation focuses on a combination of two techniques for modeling surfaces of 3D objects: surface reconstruction by triangular meshes and parametric functions NURBS (Non-Uniform Rational B-Splines) (Farin, 2001; Anderson & Crawford-Hines, 2000, Roth & Boulanger, 1998).

The adjustment of the parametric patches through cyclical Newton iteration is described. This iteration minimizes the distance between a mesh structure and patches-metric (Elsäesser, 1998). The evaluation of boundary conditions for the relaxation of parametric functions over the entire surface is also demonstrated (Hoppe *et al.*, 1992).

These techniques are transformed into a joint special purpose computational tool for creating surfaces through a cloud of points.

The performance of the applied method was a function of the precision of the adjustment and was metrologically compared with a standard system. The experiment showed that the software can be applied as a valid measuring instrument.

SUMÁRIO

1. INTRODUÇÃO.....	1
1.1. MOTIVAÇÃO	2
1.2. OBJETIVOS	3
1.3. ORGANIZAÇÃO DA DISSERTAÇÃO.....	3
2. REVISÃO BIBLIOGRÁFICA	6
2.1. INTRODUÇÃO	6
2.2. TÉCNICAS DE RECONSTRUÇÃO DE SUPERFÍCIES ATRAVÉS DE NUVEM DE PONTOS	6
2.2.1 Caracterizações do problema	6
2.2.2 Classificações dos métodos.....	7
2.2.3. Métodos de decomposição espacial	7
2.2.4. Métodos incrementais	10
2.2.5. Métodos da família crust.....	11
2.2.6. Métodos implícitos.....	12
2.2.7. Métodos baseados em modelos deformáveis	17
2.3. CONSIDERAÇÕES FINAIS	22
3. FILTRAGEM DA NUVEM DE PONTOS.....	24
3.1 INTRODUÇÃO	24
3.2. PROCESSOS ESTOCÁSTICOS E DETECÇÃO DE MUDANÇA.....	24
3.2.1. Processos estocásticos	24
3.2.2. Detecção de mudança.....	25
3.3. CLASSIFICAÇÃO DE FILTROS.....	26
3.4.1. Filtragem seletiva de frequência	26
3.4.2. Filtros adaptativos	27
3.4. CONSIDERAÇÕES FINAIS	33

4. CURVAS E SUPERFÍCIES PARAMÉTRICAS	34
4.1. INTRODUÇÃO	34
4.2. CURVAS E SUPERFÍCIES DE BEZIER.....	35
4.3. FUNÇÕES-BASE <i>B-SPLINES</i>	36
4.3.1. Vetor Paramétrico de Nós	37
4.3.2. Propriedades das funções-base Splines	38
4.4. B-SPLINE.....	39
4.4.1. Curvas <i>B-Splines</i>	39
4.4.2. Superfícies <i>B-Splines</i>	39
4.5. <i>B-SPLINES</i> RACIONAIS NÃO UNIFORMES (<i>NURBS</i>).....	40
4.5.1. Propriedades da representação racional com <i>B-Spline</i>	41
4.5.2. Curvas e Superfícies <i>NURBS</i>	41
4.5.3. Graus de Liberdade	42
4.6. TÉCNICAS PROPOSTAS DE RECONSTRUÇÃO DE SUPERFÍCIES ATRAVÉS DE FUNÇÕES PARAMÉTRICAS.....	42
4.7. AJUSTE ESPACIAL DE SUPERFÍCIES PARAMÉTRICAS	44
4.7.1. Formulação do problema.....	44
4.7.2. Projeção de um ponto a uma curva	45
4.7.3. Implementação	46
4.8. CONSIDERAÇÕES FINAIS	46
5. PROCESSO DE RECONSTRUÇÃO DE SUPERFÍCIES	48
5.1. INTRODUÇÃO	48
5.2. SENSOR ÓPTICO LASER	48
5.3. TRATAMENTO DE RUÍDOS ATRAVÉS FILTROS ADAPTATIVOS.....	50
5.3.1. Formato de arquivo da nuvem de pontos	51
5.3.2. Critérios de desempenho em um algoritmo adaptativo.....	54
5.4. CONSTRUÇÃO DE UMA MALHA ESTRUTURADA.....	67

5.5. TRATAMENTO DA INFORMAÇÃO EM NUVEM DE PONTOS COM REGIOES DESCONHECIDAS	70
5.6. CONSTRUÇÃO DA SUPERFÍCIE MEDIANTE UMA REDE DE ENTALHES PARAMÉTRICOS BICÚBICOS.....	71
5.7 ANÁLISES DE CONTINUIDADE DE ENTALHES PARAMÉTRICOS	72
5.7.1 Formulação matemática	73
5.8. ADAPTAÇÃO DOS ENTALHES PARAMÉTRICOS SOBRE MALHA ESTRUTURADA	76
5.8.1 Seleção do tamanho e valores de um vetor nó	76
5.9. QUALIDADE DA APROXIMAÇÃO	81
6. MÉTODO DE VERIFICAÇÃO DA EXATIDÃO DO SISTEMA <i>LASER SCANNER</i> UTILIZANDO ALVOS ESFÉRICOS	82
6.1. INTRODUÇÃO	82
6.2. DESCRIÇÃO DO EXPERIMENTO	82
6.3. CÁLCULO DO RAIOS DA ESFERA	84
6.4. TRIANGULAÇÃO DE PONTOS	87
6.4.1 Distância entre ponto a uma reta	88
6.5. OBTENÇÃO DE DADOS.....	88
6.6. ANÁLISE E DISCUSSÃO DOS RESULTADOS.....	90
7. CONCLUSÕES	92
8. TRABALHOS FUTUROS	94
9. REFERÊNCIAS BIBLIOGRÁFICAS	95
APÊNDICES	101
APÊNDICE A - VISUALIZAÇÃO DA MALHA TRIANGULAR ATRAVÉS DE RAMPAS DE COR	102
A.1. ESCALA DE CINZA	102
A.2. ESCALA DE COR	103
APÊNDICE B - SUPERFÍCIES NO ESPAÇO ATRAVÉS DE OPENGL	106

B.1. SUPERFÍCIES EM OPENGL	106
APÊNDICE C - SOFTWARE DESENVOLVIDO.....	109
C.1. ESTRUTURA DO SOFTWARE.....	111
C.2. ORGANIZAÇÃO DO <i>SOFTWARE</i>	112
C.3. INTERFACE.....	112
C.4. MODELAGEM UML DO SISTEMA DE PROCESSAMENTO DE NUVEM DE PONTOS	116
APÊNDICE D – OUTROS RESULTADOS	118
D.1 AVALIAÇÃO GEOMÉTRICA DE UMA SUPERFÍCIE DANIFICADA POR CAVITAÇÃO	118
D.2. CONSTRUÇÃO PASSO A PASSO DE UMA CORDA DE SOLDA	119
D.3. RESOLUÇÃO DA SUPERFÍCIE	120
D.4. CONSTRUÇÃO DE UMA SUPERFÍCIE DE FORMA LIVRE.....	121
APÊNDICE E – DEFINIÇÕES.....	122

LISTA DE FIGURAS

Figura 1.1. Processo de geração de superfícies a partir de uma nuvem de pontos.....	4
Figura 2.1. Triangulação de Delaunay.....	8
Figura 2.2. Método incremental.....	10
Figura 2.3. Algoritmo Ball Pivoting 2D.....	11
Figura 2.4. Método da família crust.....	11
Figura 2.5. Modelo de <i>Blobby</i> <i>l</i>	13
Figura 2.6. Método de reconstrução da superfície (Hoppe <i>et al.</i> , 1992).....	15
Figura 2.7. Vizinhança natural de <i>p</i>	16
Figura 2.8. Organização da nuvem de pontos dentro de uma malha 2D.....	19
Figura 2.9. Seleção de pontos.....	19
Figura 2.10. Seleção de pontos utilizando o critério A.....	20
Figura 2.11. Seleção de pontos utilizando o critério C.....	20
Figura 2.12. Desenho de uma malha triangular 2D.....	21
Figura 2.13. Desenho de uma Malha triangular 3D.....	21
Figura 2.14. Processo de reconstrução de superfícies modelos deformáveis.....	22
Figura 3.1. Distribuição uniforme.....	25
Figura 3.2. Distribuição de Cauchy	25
Figura 3.3. Distribuição de Gaussiana	25
Figura 3.4. Filtros Seletivos de frequência.....	26
figura 3.5. Elementos que conformam um filtro adaptativo.....	27
Figura 3.6. Diagrama do processo de cancelamento de ruído.....	28
Figura 4.1. Funções Base-Splines uniforme	37
Figura 4.2. Funções Base com Nó não uniforme.....	38
Figura 4.3. Projeção do Plano Racional <i>w</i>	40
Figura 4.4. Superfícies NURBS com diferentes pesos (<i>w</i>).....	41
Figura 5.1. Esquemático do sistema VISSCAN-3D.....	49
Figura 5.2. Versão de campo do hardware do sistema VISSCAN-3D.....	49
Figura 5.3. Processo de escaneamento.....	49
Figura 5.4. Imagem de profundidade (nuvem de pontos).....	50
Figura 5.5. Identificação visual dos ruídos sobre a nuvem de pontos.....	50
Figura 5.6. Diagrama do processo de cancelamento de ruídos para [X,Y,Z].....	53
Figura 5.7. Plotagem dos pontos observados no plano [X,Y,Z].....	54
Figura 5.8. Plotagem dos pontos observados no plano [X,Y].....	54
Figura 5.9. Plotagem dos pontos observados no plano [X,Z].....	55

Figura 5.10. Plotagem dos pontos observados no plano [Y,Z].....	55
Figura 5.11 Plotagem dos pontos na componente X.....	56
Figura 5.12 Plotagem dos pontos na componente Y.....	57
Figura 5.13 Plotagem dos pontos na componente Z.....	57
Figura 5.14. Plotagem dos pontos observados no plano [X,Y,Z].....	58
Figura 5.15. Plotagem dos pontos observados no plano [X,Y].....	58
Figura 5.16. Plotagem dos pontos observados no plano [X,Z].....	59
Figura 5.17. Plotagem dos pontos observados no plano [Y,Z].....	59
Figura 5.18 Plotagem dos pontos na componente X.....	60
Figura 5.19 Plotagem dos pontos na componente Y.....	60
Figura 5.20 Plotagem dos pontos na componente Z.....	61
Figura 5.21. Plotagem dos pontos observados no plano [X,Y,Z]	61
Figura 5.22. Plotagem dos pontos observados no plano [X,Y].....	62
Figura 5.23. Plotagem dos pontos observados no plano [X,Z].....	62
Figura 5.24. Plotagem dos pontos observados no plano [Y,Z].....	63
Figura 5.25. Apresentação da Nuvem de Pontos Aplicando Diferentes Valores de μ_{NLMS}	64
Figura 5.26. Erro associado versus critério desempenho no NLMS.....	65
Figura 5.27. Erro associado versus critério desempenho no RLS.....	67
Figura 5.28. Projeção da nuvem de pontos de R^3 para R^2 sobre plano σ	67
Figura 5.29. Projeção da nuvem de pontos sobre uma malha padrão.....	68
Figura 5.30. Reposicionamento dos pontos nodais influenciada pela nuvem de pontos.....	69
Figura 5.31. Levantamento tridimensional da superfície (cor cinza- Apêndice C).....	69
Figura 5.32. Levantamento tridimensional da superfície (cor RGB- Apêndice C).....	69
Figura 5.33. Curva de C-Splines no plano [X,Y].....	70
Figura 5.34 - Procedimento gráfico de eliminação de regiões desconhecidas.....	71
Figura 5.35. Reconstrução de superfícies através dos entalhes paramétricos.....	72
Figura 5.36. Detalhamento visual da superfície paramétrica.....	72
Figura 5.37. G-Splines de Peters.....	73
Figura 5.38. Suavização da superfície por meio da análise de continuidades dos entalhes paramétricos.....	74
Figura 5.39. Suavização da superfície.....	74
Figura 5.40. Suavização da superfície por meio da análise de continuidades dos entalhes paramétricos.....	75
Figura 5.41. Ajuste dos vetores paramétricos através de curvas geradas pelos pontos médios.....	77
Figura 5.42. Curva de ajuste na direção u - vetor nó = {0,0,0,0,1,1,1,1}.....	78
Figura 5.43. Curva de ajuste na direção v - vetor nó = {0,0,0,0,1,1,1,1}.....	78
Figura 5.44. Curva de ajuste na direção u - vetor nó = {0.00, 0.18, 0.32, 0.46, 0.59, 0.73, 0.87, 1.0}.....	78
Figura 5.45. Curva de ajuste na direção v - vetor nó = { 0.00, 0.21, 0.27, 0.34, 0.73, 0.80, 0.87, 1.0}.....	79

Figura 5.46. Acoplamento da superfície (malha estruturada e entalhes paramétricos).....	80
Figura 5.47. Tendência do valor do erro de ajuste com a variação de t	81
Figura 6.1. Montagem esquemática da captura da nuvem de pontos.....	82
Figura 6.2. Montagem experimental.....	83
Figura 6.3. Registro da nuvem de pontos em R^2	84
Figura 6.4. Construção da malha estruturada da esfera.....	85
Figura 6.5. Cálculo do vetor normal de um triângulo.....	85
Figura 6.6. Projeção de retas desde o incentro de cada triângulo ao centro da esfera.....	85
Figura 6.7. Projeção de duas retas contíguas e opostas a um ponto nodal.....	86
Figura 6.8. Distância do ponto nodal até o ponto médio de interseção (r_i).....	86
Figura 6.9. Projeção de retas (r_i) desde todos os pontos nodais.....	87
Figura 6.10. Triangulação de pontos.....	87
Figura 6.11. Triangulação de pontos (ponto médio).....	88
Figura 6.12. Variação do raio médio(r_i) versus deslocamento da esfera (d).....	89
Figura 6.13. Precisão da versão de campo do SistemaViscan-3D.....	90
Figura 6.14. Precisão teórica dos sistemas de digitalização VISSCAN -3D.....	90
Figura A.1. Cubo de Cor RGB – Escala cinza.....	102
Figura A.2. Cubo de Cor RGB – Escala termográfica.....	103
Figura A.3 . Faixa de Cor termográfica.....	103
Figura A.4. Representação da figura através da escala termográfica (A).....	104
Figura A.5. Representação da figura através da escala termográfica (B).....	104
Figura B.1. Representação da superfície Nurbs em diferentes perspectivas.....	106
Figura B.2. Entalhe NURBS.....	108
Figura C.1. Entorno visual em que é apresentada a superfícies virtuais.....	110
Figura C.2. Diagrama básico do núcleo software.....	112
Figura C.3. Plataforma do <i>software</i> ambiente 1.....	113
Figura C.4. Plataforma do <i>software</i> ambiente 2.....	113
Figura C.5. Plataforma do <i>software</i> – ferramentas computacionais.....	113
Figura C.6. Plataforma do <i>software</i> (PDI).....	113
Figura C.7. Interface de usuário.....	114
Figura C.8. Plataforma de Visual C++ na qual se desenvolveu o sistema 3D.....	115
Figura C.9 Diagrama de caso uso do sistema de reconstrução de superfícies.....	116
Figura C.10 Diagrama de classes do sistema de reconstrução de superfícies.....	117
Figura D.1. Avaliação geométrica de uma superfície danificada por cavitação.....	118
Figura D.2. Procedimento geral para reconstrução de superfícies através de nuvens de pontos.....	119
Figura D.3. Construção de uma superfície vista em diferentes perspectivas.....	120

Figura D.4. Construção de uma superfície de forma livre.....	121
Figura E.1. Diagrama de Voronoi (dois pontos sem preenchimento) em linhas contínuas.....	122
Figura E.2.Exemplo bidimensional do fecho convexo de um conjunto de pontos.....	123
Figura E.3. Uma malha estruturada	124
Figura E.4. Uma malha não estruturada.....	124
Figura E.5. Exemplo de uma triangulação de Delaunay	124
Figura E.6. Simplexos.....	125
Figura E.7. Superquadrica.....	125
Figura E.8. Ruído dentro de uma imagem	125
Figura E.9. Exemplo de uma variedade de Dimensão 2 Imersa em R^3	126

LISTA DE TABELAS

Tabela 5.1. Acumulador de erro médio em função do critério desempenho de um algoritmo adaptativo.....	65
Tabela 5.2. Resumo dos valores obtidos do erro de ajuste da superfície.....	79
Tabela 6.2. Dados estatísticos obtidos do experimento do alvo esférico	88

LISTAS DE SÍMBOLOS, NOMENCLATURA E ABREVIACÕES

$\alpha - shape$	Parâmetro que indica o nível de detalhes da representação volumétrica
$B_{i,n}(u)$	Funções-base Bezier
$C(u)$	Função de uma Curva paramétrica
d_n	Sinal lido proveniente da nuvem de pontos
e_i	Sinal de erro
$\bar{\epsilon}_{ajuste}$	Erro de ajuste da superfície paramétrica com a malha padrão
GRACO	Grupo de Automação e Controle
h, w	Resposta impulsional
LMS	<i>Least Mean Square</i>
$N_{i,p}(u)$	Funções-base B-Spline
Nurbs	<i>Non Uniform Rational B-Spline</i>
NLMS	<i>Normal Least Mean Square</i>
OpenGL	Biblioteca Gráfica Aberta(<i>Open Graphics Library</i>)
p_i	Ponto de uma amostra
$(P_{i,j})$	Ponto nodal
Q_{RLS}	Coefficiente RLS do filtro adaptativo
R^2	Espaço em duas dimensões
R^3	Espaço em três dimensões
RLS	<i>Recursive Least Square</i>
S	Superfície do Objeto
$S(u, v)$	Função de uma Superfície paramétrica
S_{real}	Superfície do objeto real
t	Distância de separação entre linhas de uma malha padrão
$T. Entrada$	Tamanho de arquivo de entrada
$T. Saída$	Tamanho de arquivo de saída
U	Vetor de nós
μ_{LMS}	Coefficiente LMS do filtro adaptativo
μ_{NLMS}	Coefficiente NLMS do filtro adaptativo
V	Vetor de nós
VISCAN 3D	Escâner 3D

1 - INTRODUÇÃO

As turbinas hidráulicas de grande porte das usinas de geração de energia elétrica apresentam erosão por cavitação e precisam ser restauradas pela deposição de material por soldagem. Os reparos são realizados no local e, na maioria das vezes, manualmente por uma equipe de soldadores. Esse tipo de operação apresenta uma série de problemas técnicos e econômicos, tais como: dificuldade de acesso aos locais, posições de soldagem desfavoráveis, trabalho intenso e demorado, trabalho de esmerilhamento, consumo excessivo de material de deposição e deterioração progressiva da forma geométrica das pás.

Os ganhos obtidos com uma operação robotizada por soldagem são muitos. No entanto, os robôs manipuladores convencionais não podem ser utilizados nesta tarefa em função do limitado espaço de trabalho existente no interior do rotor de uma turbina hidráulica.

Este trabalho está inserido dentro de um projeto de maior envergadura, intitulado “Desenvolvimento e Construção de Sistema Robotizado para Reparos de Falhas de Cavitação em Turbinas Hidráulicas”, que tem como objetivo o projeto de um robô que detecte e repare superfícies danificadas por efeitos de cavitação nas pás de turbinas hidráulicas. O desenvolvimento do robô tem três frentes de trabalho, nas áreas de visão, robótica e soldagem, que estão sendo conduzidos de forma integrada pelos vários grupos pertencentes ao laboratório do GRACO¹ da Universidade de Brasília.

Esta pesquisa está localizada na área de visão computacional, especificamente no desenvolvimento de uma metodologia que permita o tratamento de nuvem de pontos provenientes de um sensor especificamente construído para aquisição de nuvens de pontos utilizando emissores de luz laser, câmera e sistema de processamento de informação 3-D por varredura do laser. Este sensor foi desenvolvido em trabalho anterior a este (Ginani, L. S.,2008). A construção da superfície virtual através desta metodologia permite avaliar matematicamente a zona em estudo e com a integração dos sistemas robóticos e de soldagem será possível enviar comandos ao robô para que inicialize a reparação da zona afetada pela deposição do material de soldagem.

¹ GRACO Grupo de Automação e Controle

1.1 - MOTIVAÇÃO

Inicialmente, a pesquisa sobre a reconstrução de superfícies a partir de nuvem de pontos surgiu pela necessidade de melhorar e aperfeiçoar os dados provenientes de um sensor baseado em visão computacional, o qual gera modelos digitalizados tridimensionais de superfícies.

O equipamento denominado VISSCAN-3D constituído por um hardware e um software foi projetado e construído no Grupo de Automação e Controle da Universidade de Brasília-GRACO/UnB (Ginani & Motta, 2007). O sistema captura a imagem dos pontos visíveis e calcula sua posição tridimensional como uma nuvem de pontos gerando a imagem de profundidade.

A representação exata da superfície em relação à superfície original nem sempre é possível, o que exige uma aproximação da mesma. Essa aproximação deve manter não somente a informação geométrica do objeto real, mas também requer uma visualização realista da superfície.

De acordo com Anderson & Crawford-Hines (2000), a reconstrução de superfícies suaves pode ser planejada, em termos matemáticos, a partir da nuvem de pontos: dado um conjunto de pontos $p_i = \{(x_i, y_i, z_i)_{i=0}^n\} \in R^3$ da superfície do objeto S_{real} , deseja-se encontrar uma superfície S que aproxime ou interpole os pontos amostrados da superfície S_{real} .

A reconstrução pode ser realizada pelo ponto de vista da interpolação ou pelo ponto de vista da aproximação. A interpolação exige que a superfície gerada passe exatamente por todos os pontos, mas se os dados contem algum tipo de ruído adquirido durante os processos de aquisição, registro ou integração, uma superfície aproximada é mais adequada. Neste caso, se requer somente que a superfície passe próxima aos dados de controle: dada uma seqüência de amostras p_i para $i = 0, \dots, m$, pretende-se encontrar a superfície $S(u, v)$ que adapte amostras p_i , minimizando o erro quadrático médio:

$$\text{Min} \sum_{i=0}^m \|S(u, v) - p_i\|^2 \tag{1.1}$$

1.2 - OBJETIVOS

A partir da motivação o objetivo da pesquisa foi explorar as principais técnicas de reconstrução de superfícies formalizando as etapas necessárias para a criação de um modelo abstrato de reconstrução.

O sistema deve ser capaz de criar um ambiente amigável que permita ao usuário preparar todos os dados necessários para modelar a superfície em um pequeno intervalo de tempo.

O plano de trabalho foi subdividido nas seguintes etapas:

- Identificação e eliminação do ruído adquirido durante o processo de aquisição da nuvem de pontos;
- Adaptação e aperfeiçoamento das técnicas desenvolvidas para geração de malhas de triangulares;
- Reconstrução, por meio da malha triangular, de um conjunto de superfícies paramétricas;
- Acoplamento das funções paramétricas, por meio da avaliação de condições de fronteira, procurando o ajuste e a suavização entre os entalhes adjacentes que conformam a superfície;
- Minimização da distância entre as superfícies dos entalhes paramétricos e a malha triangular utilizando como método a Iteração de Newton;
- Estudo teórico e desenvolvimento prático de técnicas para verificar a qualidade e a precisão do sensor VISSCAN 3D.
- Desenvolvimento de uma plataforma de *software* amigável, que seja de fácil manipulação no processamento dos dados.

1.3 – ORGANIZAÇÃO DA DISSERTAÇÃO

Os capítulos do presente trabalho foram organizados de acordo com os objetivos acima citados.

A figura seguinte apresenta um resumo gráfico dos passos a serem desenvolvidos durante esta dissertação:

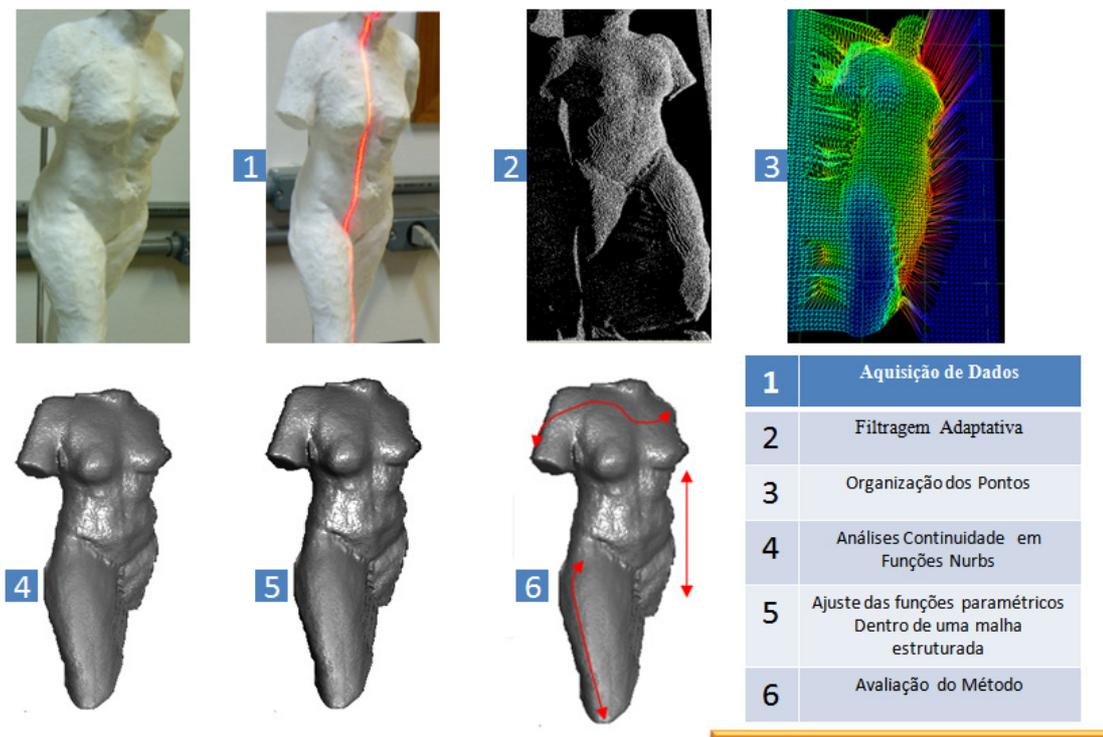


Figura 1.1. Processo de geração de superfícies a partir de uma nuvem de pontos.

Capítulo 2. Nesse capítulo é apresentada uma introdução a respeito da reconstrução da superfície a partir de nuvens de pontos. Inicialmente é fornecida uma definição formal do problema e uma classificação das abordagens. Em seguida é apresentada uma revisão bibliográfica dos métodos empregados no processo de reconstrução em particular ênfases para os métodos baseados em modelos deformáveis.

Capítulo 3. Esse capítulo aborda as bases teóricas e a aplicação prática dos filtros adaptativos no tratamento do ruído sobre nuvens de pontos. Esses filtros permitem a eliminação ou atenuação de interferências que são recolhidas no processo de aquisição e que prejudicam notavelmente a reconstrução da superfície quando é apresentado em R^3 .

Capítulo 4. Nesse capítulo é apresentada uma revisão bibliográfica para o tratamento de desenho de curvas e superfícies paramétricas, pois estas possibilitam a geração de representações de qualidade quando se trabalha com reconstrução de superfícies de forma livre.

Assim mesmo é descrito o processo matemático para gerar aproximação de ajuste espacial entre os nós da malha estruturada 3D e um conjunto e os entalhes paramétricos. A descrição do processo de ajuste primeiramente trata de curvas NURBS e logo sua formulação se estende às superfícies.

Capítulo 5. Nesse capítulo é descrito passo a passo o processo geral que foi adotado para a construção de superfícies através de uma nuvem de pontos gerada pelo sistema VISSCAN 3D.

Capítulo 6. Nesse capítulo se aborda a descrição de um experimento que permitiu determinar a variação da resolução do equipamento VISSCAN 3D *laser* em função da distância de operação.

Para fornecer confiança nos dados do experimento, foram empregados elementos padronizados como: uma mesa óptica de isolamento de vibração e uma esfera de rolamento de um gerador elétrico de diâmetro $36,50 \pm 0,005$ mm.

Capítulo 7. Conclusões; **Capítulo 8;** Trabalhos futuros **Capítulo 9;** Referências Bibliográficas

2 - REVISÃO BIBLIOGRÁFICA

2.1 - INTRODUÇÃO

O presente capítulo introduz o tema da reconstrução de superfícies a partir da nuvem de pontos. Na seção 2.2.1 é feita a caracterização do problema através da definição formal e de uma visão geral dos trabalhos desenvolvidos. No restante do capítulo são detalhadas as diversas técnicas de reconstrução de acordo com a taxonomia proposta no trabalho de Gois (2004). A divisão adotada é a seguinte: Métodos de Decomposição Espacial (Seção 2.2.3), Métodos Incrementais (Seção 2.2.4), Métodos da Família *Crust* (Seção 2.2.5), Métodos Implícitos (Seção 2.2.6) e Métodos baseados em Modelos Deformáveis (Seção 2.2.7).

2.2 - TÉCNICAS DE RECONSTRUÇÃO DE SUPERFÍCIES ATRAVÉS DE NUVEM DE PONTOS

2.2.1 - Caracterizações do Problema

Nos últimos anos as técnicas de digitalização aplicadas à engenharia reversa têm apresentado um grande desenvolvimento. Tais avanços tem permitido medir perfis com precisão na relação de 1 por 1.000 e taxas acima de 20.000 amostragens por segundo. Dependendo do objeto, milhares ou até milhões de amostragens são adquiridas e se requerem algoritmos eficientes e confiáveis que possam gerar modelos computacionais próximos de objetos reais (Mederos *et al.*, 2005).

O objetivo da construção de uma superfície pode ser definido como: “Dado um conjunto de pontos P assumindo definir uma superfície S_{real} , criar uma superfície modelo S o mais próximo possível de S_{real} ”. O processo de reconstrução da superfície não garante que ela será exatamente igual a S_{real} , visto que é conhecida uma quantidade finita de pontos. Um aumento na quantidade de pontos também não garante a exatidão da superfície. Neste caso, o algoritmo pode não conseguir reconstruir a superfície por “identificar” certos pontos como sendo ruídos. Em alguns casos é necessário definir alguns parâmetros para que o algoritmo possa solucionar o problema de reconstrução de superfície. Por esta razão, o método de reconstrução depende da aplicação. Cada aplicação requer um algoritmo adequado.

As principais dificuldades apresentadas pelo problema geral de reconstrução de superfícies são:

- *Amostras com ruídos*: Os equipamentos de aquisição de dados apresentam erros intrínsecos ao processo de obtenção de amostras.
- *Amostras Incompletas*: nem sempre a amostragem é suficiente para reconstruir a superfície.

Alguns termos a seguir apresentados estão descritos no apêndice (E).

2.2.2 - Classificações dos métodos

Diversas técnicas para resolver o problema da reconstrução de superfícies a partir de nuvem de pontos tem sido propostas e empregaram desde conceitos de geometria computacional até modelos baseados em equações diferenciais parciais (E.D.P.). Como guia da exploração dessas técnicas é necessário algum tipo de classificação.

Grande parte dos trabalhos realiza a classificação apenas em métodos paramétricos e implícitos (Hoppe *et al.*,1992; Zhao *et al.*, 2001). Recentemente, Gois *et al.* (2004) classificou os métodos em cinco grupos:

- Métodos de Decomposição Espacial;
- Métodos Incrementais;
- Métodos da Família *Crust*;
- Métodos Implícitos;
- Métodos Baseados em Modelos Deformáveis;

Todas as classificações apresentadas anteriormente baseiam-se nas características da forma em que se pode organizar nuvem de pontos. Porém outros tipos de classificação são possíveis, tais como: divisões baseadas em complexidade computacional ou qualidade de amostragem exigida pelos algoritmos.

2.2.3 - Métodos de decomposição espacial

O princípio básico da abordagem de decomposição espacial consiste em obter primeiramente a Triangulação de Delaunay do conjunto de pontos de entrada e em seguida, por meio de heurísticas denominadas de *esculpimento*, remover um conjunto de simplexes² com a finalidade de obter uma aproximação da superfície original.

² Um **n-simplexo** $\sigma \subset R^n$ é o fecho convexo de $n + 1$ pontos em posição geral (0-simplexo **ponto**, 1-simplexo **linha**, 2-simplexo **triângulo**, 3-simplexo **tetraedro**)

Os passos importantes da Triangulação de Delaunay são:

1. Definir um sistema local de eixos cartesianos, centrado em um nó, para identificar aqueles pontos mais próximos à origem para cada octante (figura 2.1.a).

2. Formar com o nó selecionado os tetraedros iniciais (figura 2.1.b).

3. Para cada tetraedro é gerada uma esfera circunscrita (figura 2.1.c).

4. Verifica-se a NÃO existência de pontos interiores adicionais dentro da esfera (figura 2.1.c). Em caso de existir um ponto interior, elimina-se o tetraedro inicial e geram-se dois novos tetraedros com o ponto mais próximo ao nó selecionado (figura 2.1.d).

5. Repete-se este processo para todos os pontos pertencentes ao domínio local (figura 2.1.e).

6. Finalmente, são escolhidos os pontos mais pertos ao nó selecionado (figura 2.1.f).

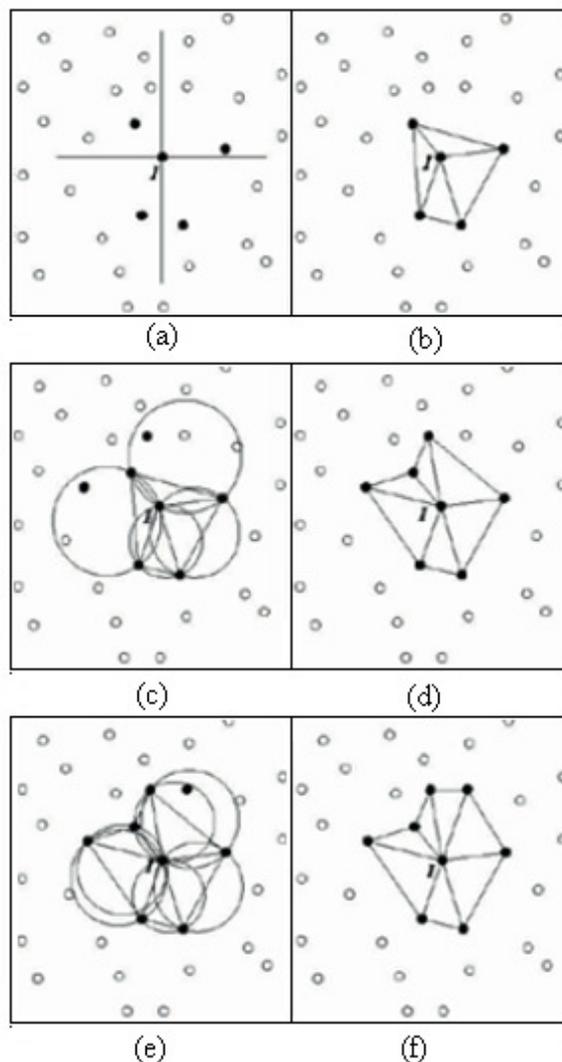


Figura 2.1 Triangulação de Delaunay

O primeiro trabalho realizado no âmbito da reconstrução de superfícies foi o de Boissonnat (1984), que introduziu dois algoritmos, sendo que um deles pode ser considerado o primeiro método de esculpimento. O algoritmo é baseado no fato de que, sob algumas restrições de amostragem, a Triangulação de Delaunay 3D contém um poliedro³ que fornece uma aproximação coerente da superfície a ser reconstruída. A heurística de esculpimento prevê a remoção de 3-simplexos enquanto a fronteira dos simplexos formar um poliedro.

³ **Poliedro** é um sólido geométrico cuja superfície é composta por um número finito de faces, em que cada uma das faces é um polígono

Outra heurística⁴ de remoção de simplexos, baseada no perímetro das faces de tetraedros, foi proposta por Rodríguez *et al.* (2000). O algoritmo de reconstrução funciona similarmente ao algoritmo descrito anteriormente, pois mantém a fronteira definida pelas faces externas dos tetraedros. Ele remove, a cada passo, todos os simplexos que possuem o perímetro da face externa menor que um limiar dado pela média dos perímetros de todas as faces externas. Tendo removido todos os possíveis simplexos, um novo limiar é calculado e comparado ao valor anterior. Se esse valor for maior que 1%, o processo é repetido.

Edelsbrunner & Mücke (1994) apresentaram um algoritmo chamado α -shape. Essa técnica foi assim chamada devido ao parâmetro α que indica o nível de detalhes da representação volumétrica obtida através da Triangulação de Delaunay 3D, ou seja, a heurística de esculpimento é definida por esse algoritmo.

Uma visão intuitiva de α -shape para complexo simplicial foi obtida ao imaginar uma esfera com raio α cujo objetivo é remover todo espaço que consegue atravessar. α -shape é adequado para pontos bem distribuídos, mas, frequentemente, é problemático para reconstrução de superfícies porque o usuário deve definir o parâmetro α que é utilizado para todo o conjunto. É mais interessante utilizar diversos valores de α para regiões específicas do domínio.

Existem outros trabalhos que se baseiam nesse autor, como o de Bajaj *et al* (1995), que apresentou uma técnica para reconstrução usando α -shape obtidos de forma automática combinando-os com funções implícitas. A fim de selecionar um valor de α são definidos critérios a respeito das propriedades da superfície desejada e, baseada nisso, é realizada uma busca binária pelo melhor parâmetro α que satisfaça as condições impostas (Bernardini *et al.*, 1999a).

Bernardini & Bajaj (1997) descreveram de maneira formal as condições necessárias para que uma superfície reconstruída através de α -shape seja homeomorfa⁵ à superfície original. A superfície somente pode ser reconstruída se for possível caminhar pelos pontos da amostra usando uma esfera de raio ε . Isso implica em duas propriedades, que correspondem às propriedades de uma amostragem uniforme:

- Quaisquer dois pontos p e q adjacentes na superfície original devem estar a uma distância menor que ε na amostra;

⁴ **Heurística** Identifica a arte ou a ciência do descobrimento, uma disciplina suscetível de ser investigada formalmente.

⁵ **Homeomorfa** Sejam os conjuntos $U \subset \mathbb{R}^n$ e $V \subset \mathbb{R}^m$. Diz-se que uma função $f: U \rightarrow V$ é um homeomorfismo, se f e f^{-1} são contínuas. Nesse caso U e V são ditos homeomorfos.

- Estando em um ponto p , somente pontos adjacentes na superfície original devem ser alcançáveis a partir de p .

2.2.4 - Métodos incrementais

Os métodos incrementais são baseados no avanço incremental de uma fronteira que compreende os pontos já processados. Em outras palavras: no início do algoritmo é definida uma fronteira de avanço composta por um conjunto de simplexes e, conforme o algoritmo processa os dados de entrada, a fronteira é expandida até que todos os pontos sejam compreendidos. Os métodos incrementais podem ser vistos como o processo oposto ao esculpimento. Ou seja, ao invés de se obter um conjunto completo de simplexes e depois esculpi-lo, a abordagem incremental gera somente os simplexes de interesse para aproximação da superfície.

Uma etapa de pré-processamento normalmente realizada pelos algoritmos incrementais é a construção de uma árvore espacial (*kd-tree* ou *quad-tree*) para que, dado um conjunto de arestas da fronteira de avanço, seja possível encontrar os pontos que são candidatos a serem processados em seguida.

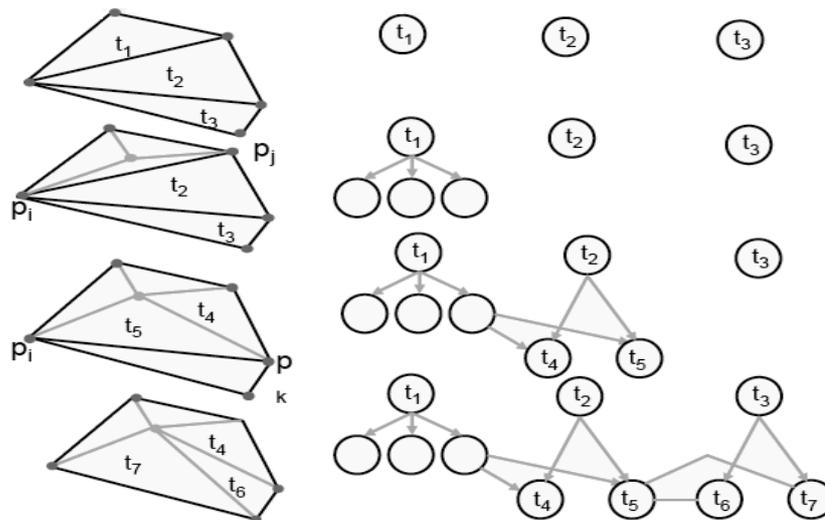


Figura 2.2. Método incremental

Como já havia sido apontado na seção (2.2.3), Bernardini & Bajaj (1997) provaram algumas propriedades sobre os α -shape utilizando algumas observações que posteriormente originaram um algoritmo incremental conhecido como *Ball Pivoting Algorithm* (Bernardini *et al.*, 1999). Esse algoritmo funciona da seguinte forma:

Seja uma bola de raio ρ em contato com três pontos (triangulação inicial). Mantendo a bola com dois desses pontos, pivote a bola até outro ponto. Cada trio de pontos tocados pela bola pivotada define um novo triângulo.

De acordo com esses autores, o Algoritmo *Ball Pivoting* gera uma aproximação simplicial que é um subconjunto dos ρ -shape e, portanto, possui as mesmas garantias de reconstrução dadas as condições de amostragem uniforme. A vantagem desse algoritmo é eliminar a necessidade de cálculo da Triangulação de Delanuy, que corresponde ao maior custo associado aos algoritmos de esculpimento. Um dos problemas apresentados pelo Algoritmo *Ball Pivoting* é a definição manual do parâmetro ρ .

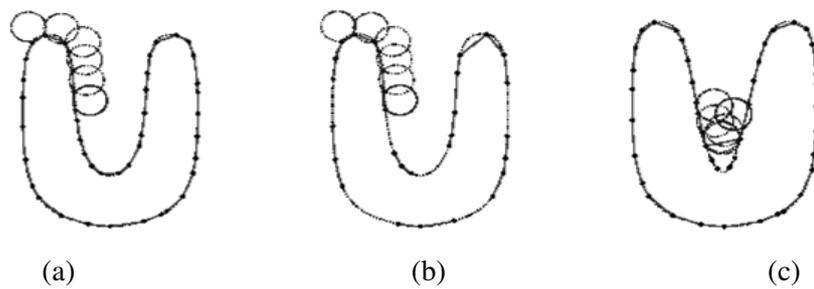


Figura 2.3. Algoritmo *Ball Pivoting 2D*

(a) Círculos de raios ρ pivotam de ponto a ponto, conectando-os pelas extremidades. (b) Quando a densidade da amostra é baixa, algumas das extremidades podem não ser criadas, restando buracos. (c) Quando a curvatura do tubo é superior a $1/\rho$, alguns pontos não serão alcançados pela bola pivotante e características serão perdidas.

2.2.5 - Métodos da família crust

Essa classe de algoritmos foi inicialmente proposta por Amenta *et al* (1998a) e tem como principal característica o emprego das propriedades e relacionamentos entre o Diagrama de Voronoi e a Triangulação de Delaunay.

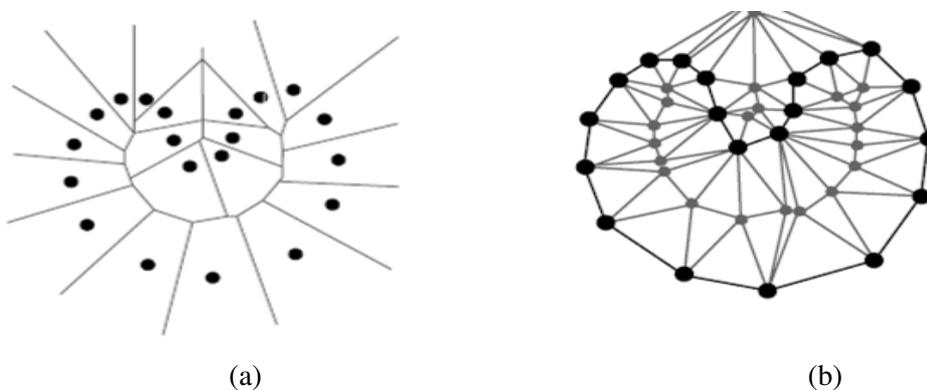


Figura 2.4. Método da família crust

1. Calcula-se Diagrama de Voronoi de S , em que S é um conjunto de pontos da amostra;
2. Sendo V o conjunto de vértices do Diagrama de Voronoi calculado em (1), calcula se a Triangulação de Delaunay do conjunto $S \cup V$;
3. Defina-se aproximação da curva-chamada *Crust* como um conjunto de arestas da triangulação que possuem p e q tais que $p \in S$ e $q \in S$.

A figura 2.4 fornece elementos gráficos do funcionamento do Algoritmo de *Crust* em duas dimensões. A figura 2.4(a) ilustra o Diagrama de Voronoi do conjunto de pontos da amostra S , enquanto que a figura 2.4(b) traz a Triangulação de Delaunay de $V \cup S$ em que V são os Vértices de Voronoi da figura 2.4(a). A curva reconstruída é representada pelo conjunto das arestas em preto realçado.

A partir do trabalho de Amenta *et al.*, toda uma família de algoritmos com garantias teóricas de reconstrução tem sido proposta. Gois (2004) realizou uma revisão extensa desses métodos que, porém, não serão discutidos neste trabalho. Uma lista dos principais algoritmos da família é fornecida a seguir:

- Crust (Amenta *et al.*, 1998b)
- Power Crust (Mederos *et al.*, 2005)
- Cocone (Amenta *et al.*, 2000)
- Tigh Cocone (Dey & Goswami, 2002)
- Robust Cocone (Dey & Goswami, 2006)
- Eigencrust (Kolluri *et al.*, 2003).

2.2.6 - Métodos implícitos

Estes métodos, também conhecidos como métodos baseados em conjuntos zero de função, consistem em utilizar a amostragem para definir uma função distância com sinal f e calcular o conjunto zero de f , que denotamos por $Z(f)$.

Os pontos em $Z(f)$ devem ser, por hipótese, regulares (pontos que satisfazem as condições de variedade⁶) ou o processo de reconstrução pode não gerar a superfície original do objeto. Dentre as vantagens dos métodos implícitos, podem-se destacar a menor sensibilidade a ruídos e a facilidade de se combinar funções implícitas para fins de modelagem. No entanto, as mesmas características que tornam os métodos implícitos robustos com relação a dados ruidosos e incompletos podem produzir suavizações indesejáveis nas superfícies.

⁶ **Variedade** O conjunto $M \subset \mathbb{R}^k$ é uma variedade de dimensões n classe C^r se para todo $x \in M$ existe uma vizinhança aberta $U \ni x$, com $U \subset \mathbb{R}^k$, V aberto de \mathbb{H}^n e um difeomorfismo de classe C^r $\varphi: U \cap M \rightarrow V$

Em 1982, Blinn propôs uma nova técnica para a modelagem de superfícies implícitas, conhecidas como Modelo *Blobby*, através da combinação de funções primitivas como esferas ou superquádricas. Esse modelo serviu de ponto de partida para uma das principais técnicas de reconstrução de nuvem de pontos usando funções implícitas (Muraki, 1991). A técnica proposta por Muraki consiste na definição de um conjunto de *blobbies* que, combinados, aproximam a superfície subjacente à nuvem de pontos. O algoritmo funciona de forma a minimizar uma função de energia que mede a distância entre a superfície gerada e a nuvem de pontos (figura 2.5).

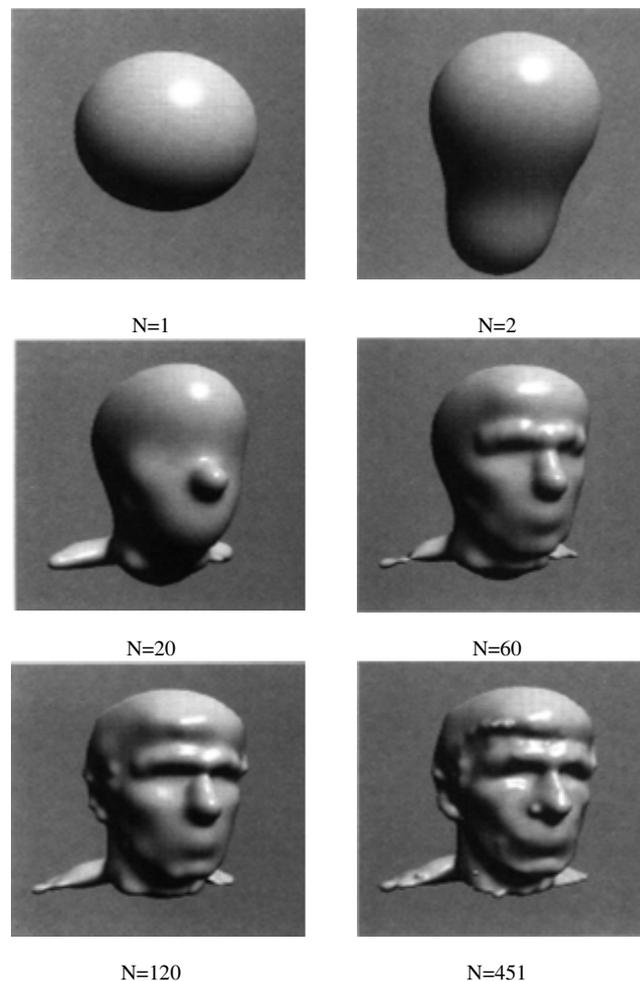


Figure 2.5. A transformação do modelo *Blobby* com o número de N primitivos para a imagem Cyberware 4020/PS.

O trabalho de Hoppe *et al.* (1992) é um dos mais citados na área de reconstrução, pois algumas das técnicas e idéias introduzidas são aplicadas ainda hoje. O algoritmo descrito por Hoppe *et al.* (1992) é executado em dois estágios.

O primeiro consiste em definir uma função $f: D \subset R^3 \rightarrow R$ em que f estima uma distância com sinal para a superfície S procurada. O conjunto $Z(f)$ é a estimativa para S . A segunda fase consiste em aproximar $Z(f)$ por um complexo simplicial.

A função distância definida por Hoppe é dada por $f_{dista_i}(p) = (p - o_i) \cdot \hat{n}_i$ sendo o_i o centro (centróide) de uma região plana e \hat{n}_i a normal unitária dessa região. O centro do plano o_i é definido pelos k pontos mais próximos de x_i (x_i pertence à nuvem de pontos), denotada por $V(x_i)$. O cálculo do vetor normal é feito pela matriz de covariância simétrica e semidefinida positiva por:

$$Cov = \sum_{y \in V(x_i)} (y - o_i)^t (y - o_i) \quad (2.1)$$

Pelo fato da matriz ser simétrica e semi-definida positiva (Gois, 2004) tem-se três autovalores reais e maiores ou iguais a zero, sejam eles $\lambda_1 \leq \lambda_2 \leq \lambda_3$. Então, escolhe-se \hat{n} como o autovetor associado a λ_3 ; v_3 ou $-v_3$, de modo a manter os planos orientados .

Para obter a isosuperfície, Hoppe utiliza uma variação do algoritmo *Marching Cubes* em que cada cubóide é decomposto em tetraedros.

Figura (2.6) ilustra o processo de reconstrução através do algoritmo proposto por Hoppe (1992):

Hoppe *et al.* (1994) estenderam o trabalho descrito anteriormente através da adoção de funções suaves para realizar a aproximação local. Uma técnica conhecida como superfície de subdivisão foi empregada. Nessa técnica, funções suaves foram aproximadas pela combinação de entalhes definidos por um número reduzido de parâmetros irrestritos.

O método descrito no trabalho pode ser dividido em três passos:

1. Geração de uma malha que aproxime o objeto usando a técnica descrita em Hoppe *et al.* (1992).
2. Simplificação da malha
3. Uso da superfície de subdivisão para aproximar regiões locais. O esquema de subdivisão proposto é capaz de detectar e representar curvas de difícil desenho.

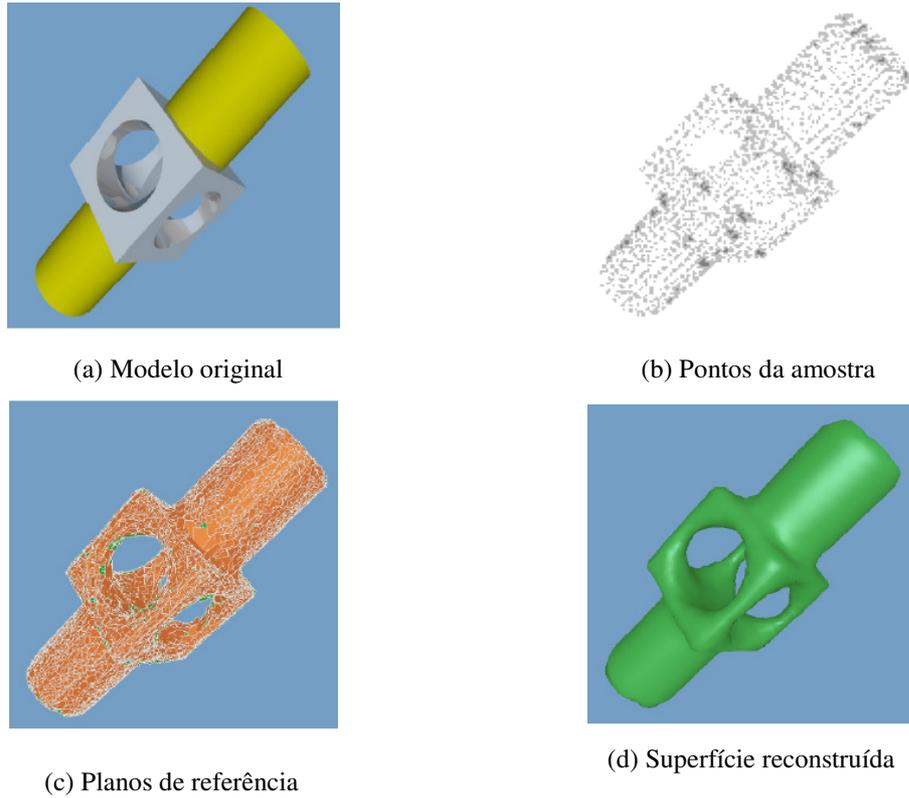


Figura 2.6. Método de reconstrução da superfície pelo método de Hoppe *et al.* (1992).

Boissonnat & Cazal (2000) demonstraram uma função de distância baseada na interpolação de vizinhos naturais⁷. Sejam $h_{x_i}(p)$ e $\lambda_{x_i}(p)$, respectivamente, uma função de distância e uma função de peso associado ao ponto x_i da amostra X . A distância de um ponto p à superfície S subjacente à nuvem de pontos é dada por:

$$f(p) = \sum_{p \in X} \lambda_{x_i}(p) h_{x_i}(p) \quad (2.2)$$

A função peso $\lambda_{x_i}(p)$ é dada pela relação de vizinhança natural.

De maneira equivalente, a vizinhança natural de p pode ser definida, em termos do Diagrama de Voronoi, como o conjunto de pontos $\{x_i\}$ que tem suas células afetadas pela inserção de p no diagrama. Essa situação pode ser visualizada na figura 2.7.

Logo, uma forma de definir a função de peso $\lambda_{x_i}(p)$ para um ponto da amostra x_i e através da medida da inserção entre a célula de x_i no Diagrama de Voronoi de X e a célula de p no Diagrama de Voronoi de $X \cup \{p\}$. Na figura 2.7 está representado o valor de $\lambda_{x_2}(p)$.

⁷ **Vizinhança natural.** Sejam $X = \{x_0, \dots, x_n\}$ uma amostra obtida sobre uma superfície S e p um ponto qualquer. A vizinhança natural de p é dada pelos pontos interligados a p na Triangulação de Delaunay de $X \cup p$.

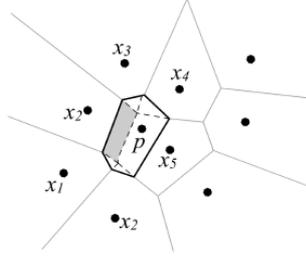


Figura 2.7. Vizinhança natural de p .
A área cinza representa o valor de $\lambda_{x_2}(p)$.

Carr *et al.* (2001) utilizaram RBFs⁸ na aproximação de um conjunto de pontos equipados com normais orientadas. Dada uma RBF apropriada, no caso $\phi(x) = |x|$, a função de interpolação para um conjunto de amostras S é de forma

$$f(x) = \sum_{j=1}^N \lambda_j \phi(x - s_j) + P(x) \quad (2.3)$$

Em que N é o tamanho da amostra, λ_j são os pesos e $P(x)$ é um polinômio de grau m .

Uma restrição colocada sobre os coeficientes é:

$$\sum_{j=1}^N \lambda_j P(x_j) = 0 \quad (2.4)$$

Sejam $\{p_1, \dots, p_l\}$ uma base de polinômios p_i de grau no máximo m e $\{c_1, \dots, c_l\}$ os coeficientes que descrevem P na base, as equações (2.3) e (2.4) podem ser descritas através do sistema linear:

$$\begin{pmatrix} A & P \\ P^T & 0 \end{pmatrix} \begin{pmatrix} \lambda \\ c \end{pmatrix} = \begin{pmatrix} f \\ 0 \end{pmatrix} \quad (2.5)$$

Em que f é o valor da função a ser determinada nos pontos da amostra.

Todos os pontos da amostra S estão localizados sobre a superfície, ou seja, apresentam $f(x_i) = 0$. Portanto é necessária a adição de novos pontos ao sistema para evitar a solução trivial $\lambda_i = 0$ para qualquer i . Esses pontos são gerados a partir de cada ponto x_i caminhando-se na direção do vetor normal.

Turk & O'Brien (2002) também trabalharam com RBFs. No entanto, o foco dessa pesquisa estava na modelagem e não no processo de reconstrução. A função de base radial utilizada nesse trabalho foi $\phi(x) = |x^3|$.

⁸ RBFs (Função-base radial)

Uma função-base radial é uma função simétrica em relação à vizinhança radial de um ponto (centro).

Uma das vantagens dos métodos baseados em RBFs é a capacidade de lidar com dados incompletos e ruidosos. Porém, tais métodos apresentam uma desvantagem quanto ao número de pontos que podem ser tratados devido à complexidade de resolução do sistema linear.

2.2.7 - Métodos baseados em modelos deformáveis

Os métodos baseados em modelos deformáveis constituem um grupo mais reduzido de algoritmos cujo funcionamento consiste em obter um modelo inicial que aproxime o conjunto de pontos. Essa aproximação é então deformada até que a superfície se ajuste ao conjunto de pontos. Esses métodos são rápidos, porém dependem de uma boa aproximação inicial derivada de algum conhecimento prévio sobre o formato do objeto a ser construído.

Os algoritmos dessa classe podem ser subdivididos em duas categorias:

- **Métodos baseados em Equações Diferenciais Parciais:** são métodos que empregam Equações Diferenciais Parciais para a deformação dos modelos iniciais (Zhao *et al.*, 2000). Apesar de produzir bons resultados, essas técnicas dependem de discretizações e de resoluções de grandes sistemas lineares;
- **Métodos baseados em funções paramétricas:** esses métodos realizam deformações na superfície através da variação de parâmetros. A partir disso surgem dois problemas principais nesse tipo de abordagem: a necessidade de se conhecer como a alteração de parâmetros influencia a deformação e a dificuldade de representar detalhes finos.

No trabalho de Bardinnet *et al.*(1995) foram utilizadas superquádricas⁹. A vantagem do emprego de superquádricas está no número reduzido de parâmetros. No entanto, isso se torna uma desvantagem uma vez que a técnica é incapaz de representar detalhes finos. Para esse problema, uma etapa final de pós-processamento, chamada *free form deformation*, deve ser aplicada.

Recentemente, o artigo de Chui *et al.* (2008) descreveu o método para organizar amostras por meio da deformação que experimenta uma malha quando a nuvem de pontos é nela projetada.

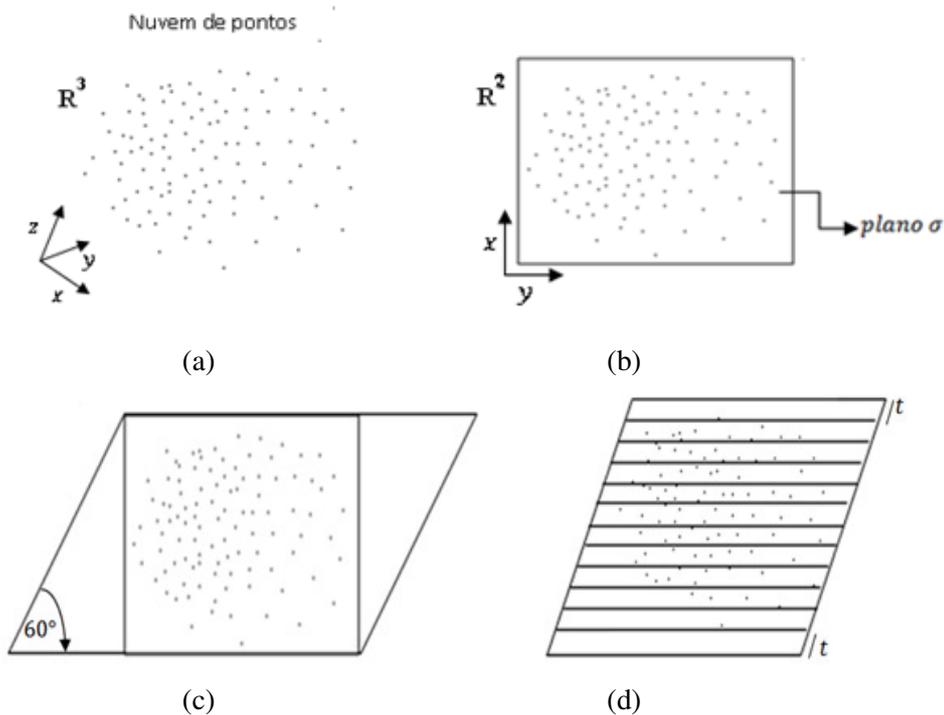
O algoritmo está dividido em três etapas e cada etapa, por sua vez, está subdividida em várias subetapas:

⁹ Superquádricas (Apendice E)

- Projeção da nuvem de pontos sobre a malha;
- Reposicionamento de pontos nodais;
- Reconstrução da malha triangular 3D a partir da malha triangular 2D.

2.2.7.1 - Projeção da nuvem de pontos sobre a malha;

- Projeta-se uma nuvem de pontos de R^3 para R^2 sobre o plano σ (figura 2.8(a) e figura 2.8(b)).
- Ajusta-se a nuvem de pontos dentro de um paralelogramo de ângulo de 60° (figura 2.8(c)). Por meio deste polígono garante-se que todos os pontos da amostra sejam analisados em todos os passos posteriores do algoritmo.
- Divide-se o polígono em linhas horizontais e verticais separadas por uma distância t . Esse valor influencia a resolução da reconstrução tridimensional da superfície, como observa-se nas figuras 2.8(d) e 2.8(e).
- Projetam-se linhas diagonais da esquerda à direita do polígono, formando triângulos equiláteros (figura 2.8(f)).



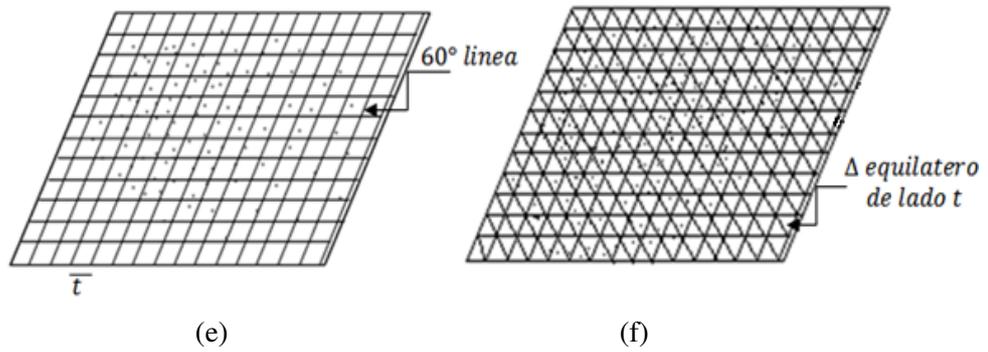


Figura 2.8. Organização da nuvem de pontos dentro de uma malha.

2.2.7.2 - Novo posicionamento de pontos nodais;

O reposicionamento do ponto nodal é a essência de um processo de filtragem. Em cada nó da malha (também chamado de ponto nodal) desenha-se um círculo do tamanho que ligue os pontos nodais contíguos, a fim de selecionar um só ponto entre muitos que se encontram dentro da área fechada por círculo. O ponto escolhido é aquele que está mais próximo do nó da malha inscrita no círculo correspondente (figura 2.9).

O procedimento é repetido para os outros nós e pontos. Aqueles que não tenham sido selecionados serão apagados.

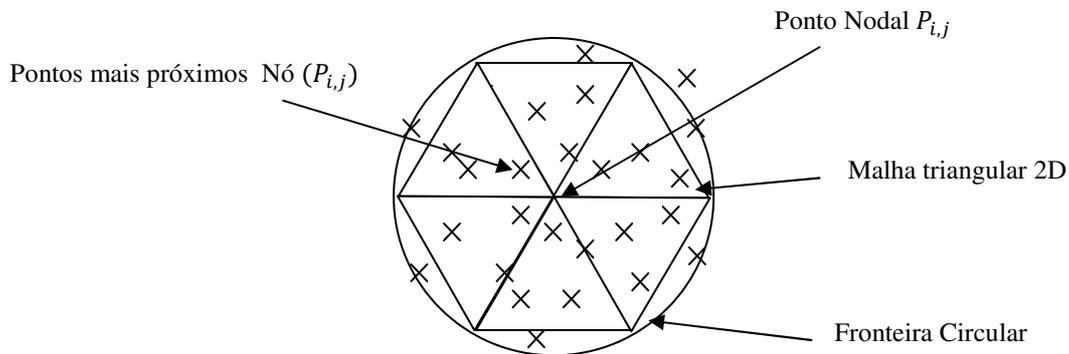


Figura 2.9. Seleção de pontos

Por exemplo, na figura 2.9, o ponto mais próximo projetado será considerado como a nova posição para o ponto nodal $P_{i,j}$. No entanto, se houver dois ou mais pontos que satisfaçam distâncias mais próximas, deverão ser estabelecidos os seguintes critérios:

Critério A. Se dois ou mais pontos projetados são iguais em distância para $P_{i,j}$, mas estão localizados em diferentes triângulos, o ponto projetado na parte superior (ou seja, maior y) será o escolhido.

Por exemplo, na figura 2.10, cinco pontos projetados estão em igual distância do ponto nodal $P_{i,j}$. Os pontos que se encontram na parte superior do ponto nodal serão

selecionados. A escolha da nova posição do ponto nodal será justificada utilizando-se o Critério B.

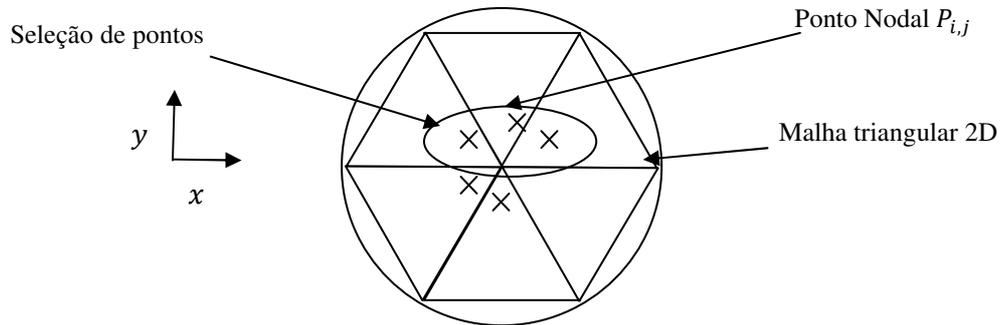


Figura 2.10. Seleção de pontos utilizando o critério A

Critério B. Se dois ou mais pontos projetados são iguais em distância na mesma linha horizontal (figura 2.10), mas em vários triângulos diferentes, o ponto projetado no meio triângulo é escolhido como a nova posição do ponto nodal $P_{i,j}$.

Critério C. Se dois ou mais pontos projetado são iguais em distância na mesma linha horizontal e no mesmo triângulo, o ponto projetado com maior valor x será escolhido (figura 2.11).

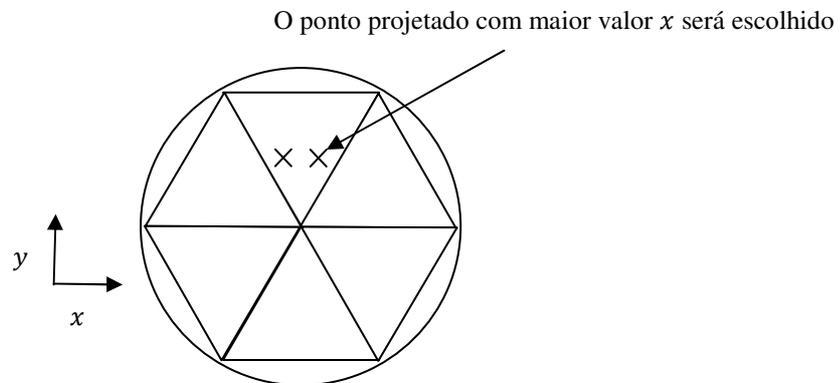


Figura 2.11. Seleção de ponto utilizando o critério C

2.2.7.3 - Construção da malha triangular 3D a partir da malha triangular 2D.

A malha triangular 3D pode ser reconstruída a partir do processo de deslocamento dos nós da malha 2D (figura 2.12) e dos valores de profundidade de cada ponto nodal que foi influenciado pelo ponto selecionado: critério A, critério B ou critério C (figura 2.13).

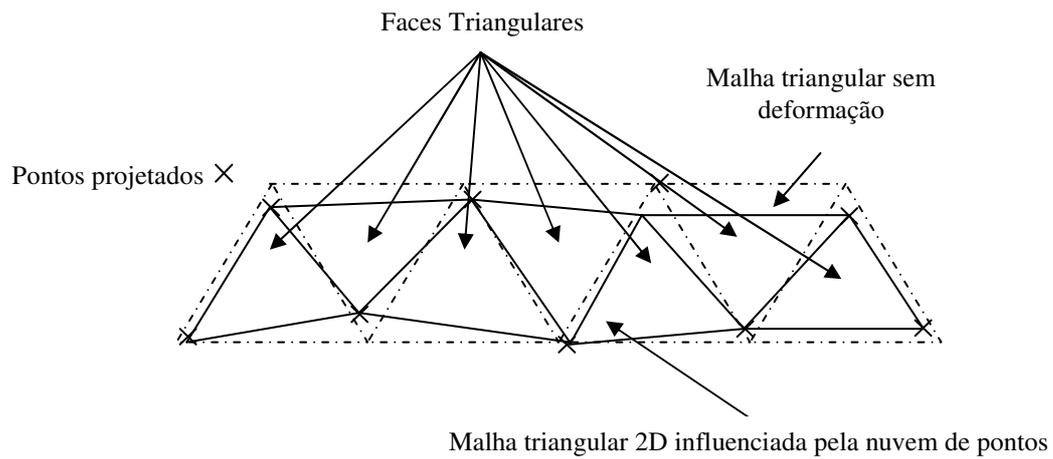


Figura 2.12. Desenho de uma malha triangular 2D.

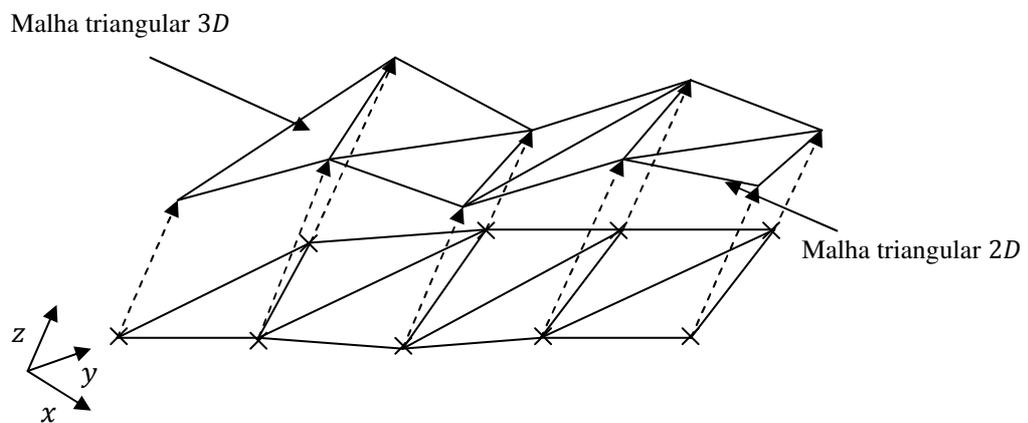


Figura 2.13. Formação da malha triangular 3D construída em base de uma malha triangular 2D.

A figura (2.14) ilustra o processo de reconstrução de superfícies através de métodos baseados em modelos deformáveis, proposto por Chui *et al.* (2008).

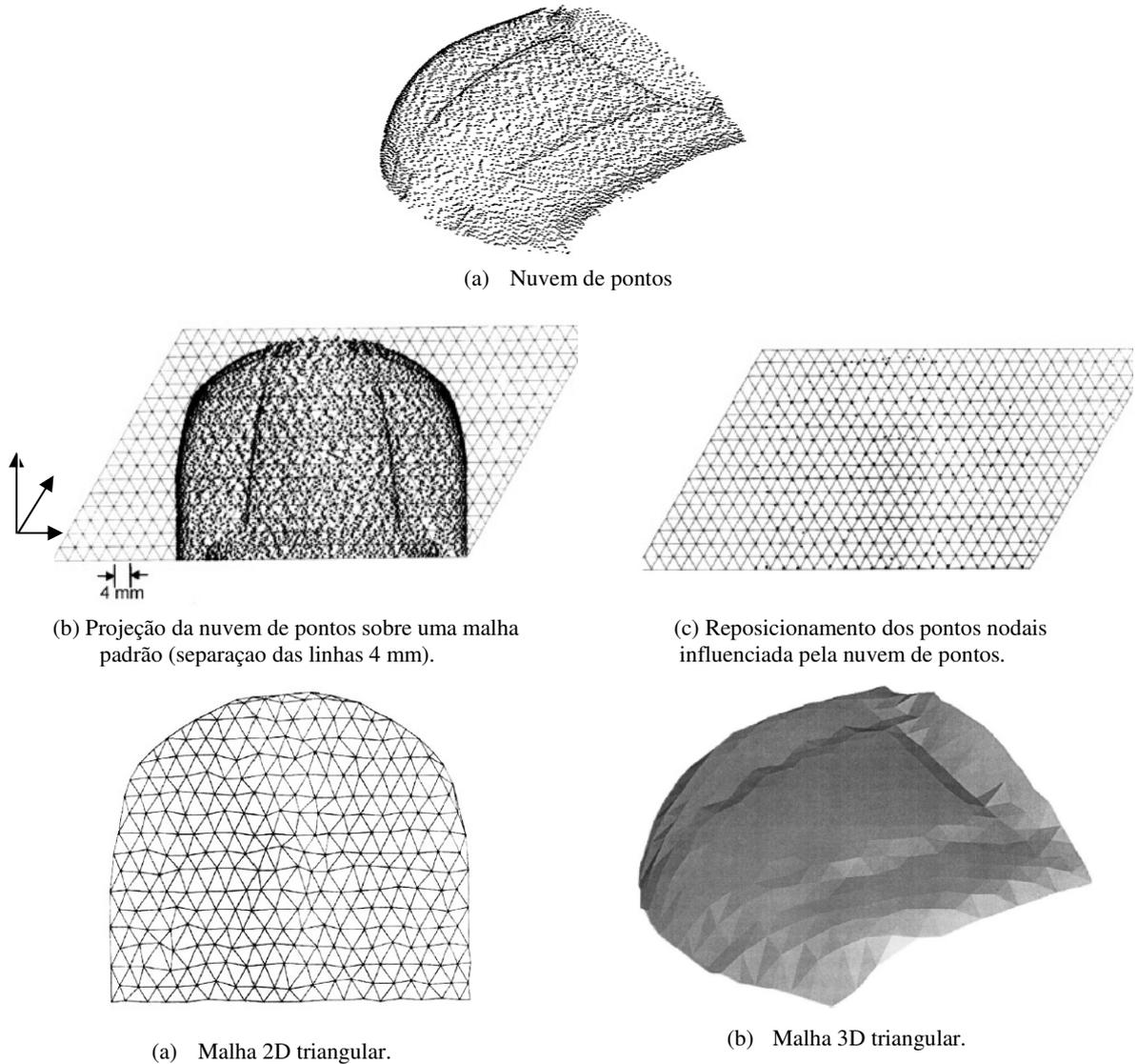


Figura 2.14. Processo de reconstrução de superfícies através de Modelos deformáveis.
Fonte: Chui *et al.*, 2008.

2.3 - CONSIDERAÇÕES FINAIS

O presente capítulo introduziu o problema da reconstrução de superfícies a partir de nuvem de pontos e apresentou uma revisão dos principais métodos empregados.

Dentre os tipos de algoritmos introduzidos foi enfatizada a classe dos métodos baseados em modelos deformáveis, já que esse é foco principal do presente trabalho.

O método de Modelos deformáveis foi escolhido por que apresenta uma poderosa metodologia que permite organizar uma nuvem de pontos em uma malha estruturada, através de um número reduzido de passos.

Nos capítulos seguintes serão abordadas as principais dificuldades que atingem o problema geral de reconstrução, como é o tratamento de *Amostras com ruídos* e *Amostras Incompletas*.

Adicionalmente será apresentada a descrição de funções paramétricas, já que as mesmas são utilizadas na poligonalização de superfícies obtidas através de modelos deformáveis.

3 - FILTRAGEM DA NUVEM DE PONTOS

3.1 - INTRODUÇÃO

A técnica de filtragem de nuvem de pontos tem o objetivo de processar os pontos de modo que a amostra resultante seja mais adequada que a nuvem de pontos original para uma aplicação específica.

Um problema fundamental na restauração de uma amostra ruidosa é a remoção de ruídos sem o desaparecimento dos detalhes.

Para tentar resolver esse problema, algumas estratégias de filtragem são baseadas num conhecimento *a priori* da distribuição do ruído presente na amostra e da forma em que o ruído “entra” nos dados (Fisher *et al.*, 2000).

Existem inúmeras classificações para as operações de filtragem. Dentre as mais importantes, tem-se: a Filtragem seletiva por frequência e a adaptativa (Fisher *et al.*, 2000).

3.2 - PROCESSOS ESTOCÁSTICOS E DETECÇÃO DE MUDANÇA

A idéia central da detecção de mudança está no próprio nome da técnica. Mediante o acompanhamento dos dados da nuvem de pontos, qualquer mudança no seu comportamento (seja em magnitude, frequência ou picos abruptos), deverá ser percebida pelo sistema. Essas perturbações poderão ser características de defeitos na superfície gerada pela nuvem de pontos.

3.2.1 - Processos estocásticos

É uma família de variáveis aleatória em função do tempo ou do espaço. Alguns fenômenos da natureza podem ser modelados por processos estocásticos, uma vez que se trata de uma variável aleatória, sob uma certa distribuição, que se propaga ao longo do tempo ou do espaço. Encontram-se abaixo exemplos de comportamentos do sinal segundo determinadas distribuições para as variáveis aleatórias, dados pelas Figuras 3.1, 3.2 e 3.3:

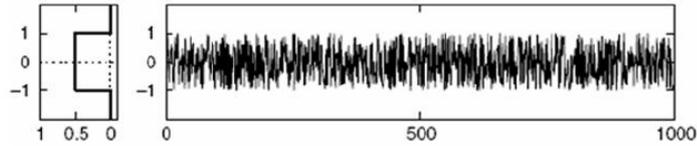


Figura 3.1. Distribuição uniforme

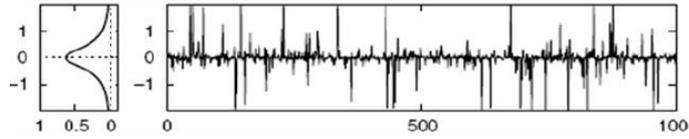


Figura 3.2. Distribuição Cauchy

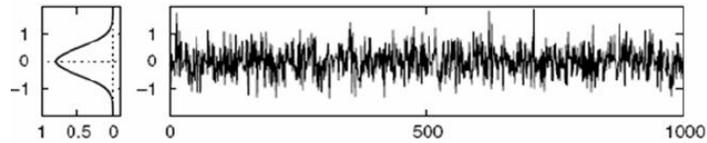


Figura 3.3. Distribuição Gaussiana

3.2.2 - Detecção de mudança

Sabendo-se que um sinal se comporta segundo um processo estocástico é possível determinar um modelo probabilístico e aplicar alguns algoritmos para processar a sinal.

Exemplos de cálculos são: filtros, estimação do sinal, soma de sinais, reconhecimento de padrão e algoritmos diversos para inúmeras tarefas.

Inicialmente, é necessário considerar um modelo para a aplicação das técnicas. No caso deste trabalho, o processo envolvido pode ser encarado como um sistema linear em unidades discretas (Basseville & Nikiforov, 1993):

$$d(n) = hI_p(n) + s(n) \quad (3.1)$$

Em que $d(n)$ é o sinal lido proveniente da nuvem de pontos, é h a matriz de medição de uma imagem I_p e $s(n)$ é o ruído. De modo similar ao ruído do processo, esse último segue distribuição gaussiana com média zero, porém matriz de covariância dada por (Basseville & Nikiforov, 1993) conhecido como ruído branco:

$$E\{s_n s_n^T\} = \begin{cases} R_n & \text{para } m = n \\ 0 & \text{para } m \neq n \end{cases} \quad (3.2)$$

3.3 - CLASSIFICAÇÃO DE FILTROS

3.3.1 - Filtragem seletiva de frequência

Os filtros seletivos de frequência são fixos, ou seja, seus coeficientes não variam durante seu funcionamento.

As características mais comuns do filtro de frequência são (Fisher *et al.*, 2000):

- Os filtros são lineares e invariantes no tempo ou espaço;
- O processo de projeto utiliza informações sobre a banda de frequência em que o filtro atuará.
- Como os filtros são seletivos de frequência, eles funcionam melhor quando os vários componentes de frequência do sinal de entrada ocupam bandas não sobrepostas.

Os tipos mais comuns de filtros seletivos de frequência são: filtros passa-altas, filtros passa-baixas e filtros passa-faixas. Os comportamentos desses filtros são apontados na figura 3.4:

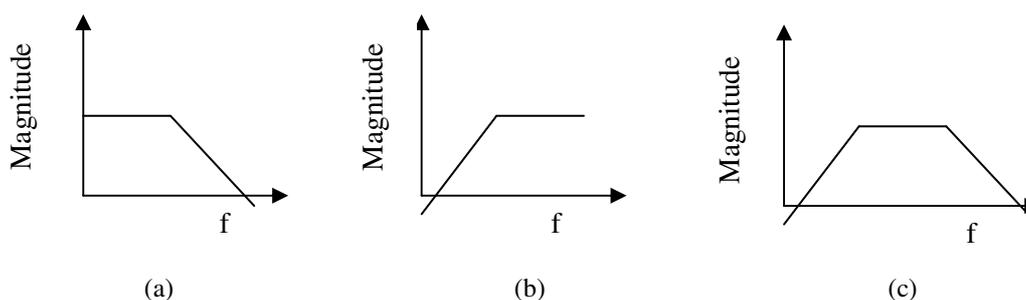


Figura 3.4. Filtros seletivos de frequência:
(a) filtro passa-baixa, (b) filtro passa-alta e (c) filtro passa-faixa

Os filtros passa-baixas, figura 3.4(a), permitem a passagem apenas de baixas frequências. Eles atuam eliminando ou atenuando os componentes de alta frequência do sinal de entrada.

Os filtros passa-altas, figura 3.4(b), permitem a passagem apenas de altas frequências. O funcionamento dos filtros passa-altas permite que eles eliminem ou atenuem os componentes de baixa frequência do sinal de entrada.

Já os filtros passa-faixa, figura 3.4(c), permitem apenas a passagem de sinais dentro de uma determinada faixa de frequências. Eles eliminam ou atenuam os componentes de frequência que estejam fora dessa faixa.

3.3.2 - Filtros adaptativos

Existem diversas aplicações práticas que não podem ser resolvidas com sucesso utilizando-se filtros digitais fixos. As dificuldades devem-se à falta de informação para projetar o filtro com coeficientes fixos ou à mudança nos critérios do projeto durante a operação. Essas dificuldades podem ser parcialmente resolvidas por um tipo especial de filtro chamado de filtro adaptativo. A característica marcante dos filtros adaptativos é que eles podem modificar sua resposta automaticamente para melhorar seu desempenho durante a operação (Basseville & Nikiforov, 1993).

A decorrência do acréscimo nos resultados da operação de filtragem é o aumento da complexidade de tais filtros (Haykin, 2001). Geralmente, os filtros adaptativos são formados por três módulos (Manolakis *et al.*, 2000) que são mostrados na figura 3.5:

Os filtros adaptativos se baseiam na minimização do erro entre a diferença que existe entre o sinal gerado pela estrutura e o sinal gerado pelo algoritmo adaptativo.

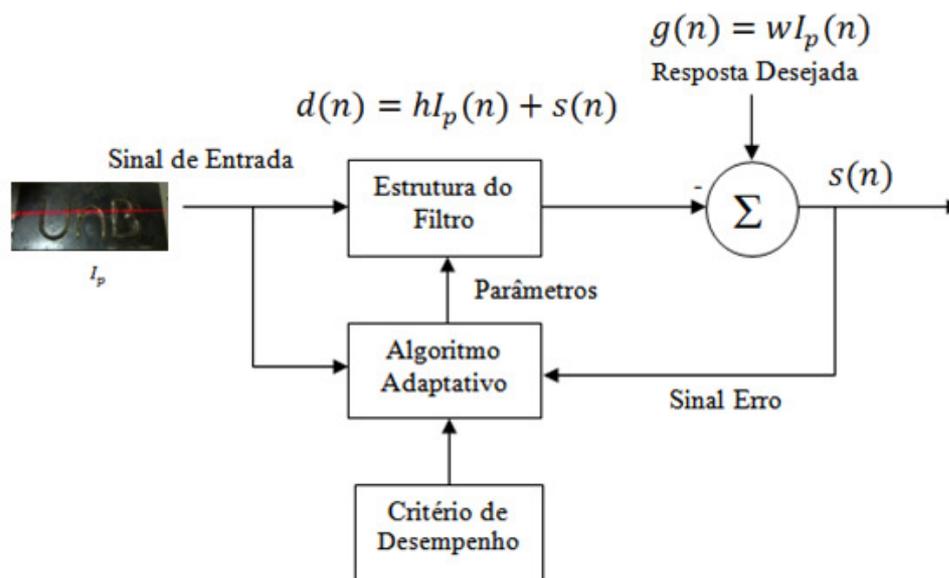


Figura 3.5. Elementos que conformam um filtro adaptativo.

Fonte: Born, 2000.

Em que:

- $d(n)$ Sinal gerado pelo sistema VISCAN3D
- $g(n)$ Sinal gerado pelo algoritmo adaptativo
- h, w Matriz de medição
- $I_p(n)$ Imagem de entrada

- **Estrutura do filtro:** Esse módulo produz a saída do filtro a partir de medições do sinal de entrada. A estrutura é fixa e seus parâmetros são ajustados pelo polígono adaptativo;
- **Crítérios de desempenho:** A saída do filtro adaptativo e a resposta desejada são processadas pelo módulo de critério de desempenho para avaliar sua qualidade em relação aos requisitos de aplicação;
- **Algoritmo adaptativo:** O algoritmo adaptativo utiliza o valor do critério de desempenho (ou alguma função dele) e medições da entrada e da resposta desejada para decidir como modificar os parâmetros do filtro a fim de melhorar seu desempenho.

3.3.2.1 - Aplicações

Existem várias aplicações onde os filtros adaptativos podem ser empregados. Essas aplicações podem ser classificadas em quatro classes: identificação de sistemas, inversão de sistemas, predição de sinal e cancelamento de ruídos (Manolakis *et al.*, 2000).

Dentre os tipos de modelo de aplicação existentes foi enfatizado o cancelamento de ruído, já que esse é foco principal do presente trabalho.

Cancelamento de ruído

Um sistema de captação de imagens (escâner) além de capturar a deformação que sofre a luz projetada sobre objeto, também captura ruídos produzidos pela mesma câmara ou pelo mesmo ambiente.

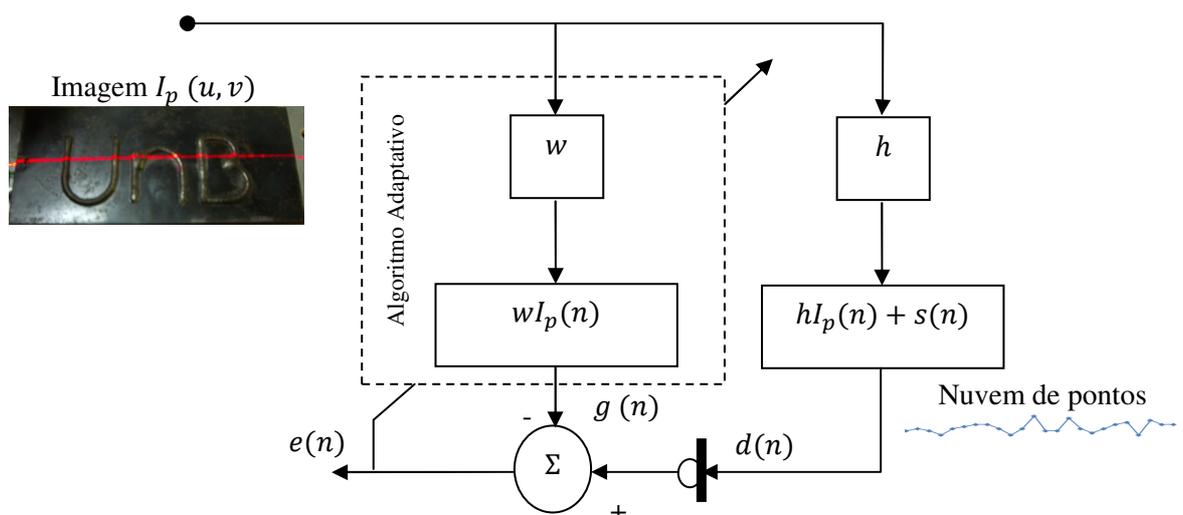


Figura 3.6. Diagrama do processo de cancelamento de ruído

A nuvem de pontos adquiridos $d(n)$ de uma linha *laser* pode ser modelada como a saída de uma imagem de entrada I_p (figura 3.6), que contém a informação tridimensional da superfície em estudo. A imagem I_p é descomposta através de critérios de filtragem FIR (*Finite-duration Impulse Response*) de resposta impulsional e de duração finita N (equação 3.3):

$$h = [h_0 \ h_1 \ \dots \ h_{N-1}] \quad (3.3)$$

A saída $d(n)$ é:

$$d(n) = hI_p(n) + s(n) \quad (3.4)$$

Em que $s(n)$ representa o ruído aleatório que contamina a nuvem de pontos, h matriz de transformação e I_p a imagem de entrada.

Para atenuar o ruído $s(n)$ deve-se estimar o valor h através de critérios de desempenho (w) gerados pelo algoritmo adaptativo:

$$g(n) = wI_p(n) \quad (3.5)$$

Em que w é matriz de transformação:

$$w = [w_0 \ w_1 \ \dots \ w_{N-1}] \quad (3.6)$$

A estimativa do ruído $s(n)$ pode ser subtraída pela diferença entre a equação (3.4) e a equação 3.5:

$$e(n) = d(n) - g(n) = (h - w)I_p(n) + s(n) \quad (3.7)$$

Em um caso ideal $w = h$, conseqüentemente, $e_i(n) = s_i(n)$, o ruído seria completamente eliminado. No entanto, aproximação de h por um filtro w de comprimento e precisão finitos impede que o percurso do ruído seja perfeitamente modelado (Bruno, 2007).

3.3.2.2 - Algoritmos adaptativos

A escolha do algoritmo adaptativo (Haykin, *et al.*, 1991) frequentemente é selecionada considerando-se: a velocidade de convergência e precisão, a complexidade numérica e a estabilidade da estrutura.

A velocidade de convergência é definida como o número de iterações necessárias para que o algoritmo, mediante sinais estacionários, aproxime-se suficientemente da solução

ótima. Quanto maior a velocidade de convergência, mais rapidamente o algoritmo se adaptará a um ambiente estacionário de características desconhecidas. A complexidade numérica refere-se ao número de operações a serem efetuadas para completar uma iteração do algoritmo. Dependendo do algoritmo de adaptação, poderá existir instabilidade numérica ao ser concretizado em um processador. O acúmulo de erros de quantificação devidos à precisão finita poderá fazer com que o algoritmo se desvie da situação ideal. É possível que o desvio seja de uma natureza tal que os erros se acumulem sem limite. Se essa situação persistir sem qualquer controle, o algoritmo será conduzido em última instância à situação de *overflow*, dizendo-se que o algoritmo é numericamente instável.

3.3.2.3 - Filtro Kalman - Fundamentos do modelo dinâmico do sistema

Filtros de Kalman são baseados em álgebra linear e no modelo oculto de Markov (Haykin, 2001). A base de sistemas dinâmicos é modelada como uma cadeia de Markov construída por operadores lineares perturbados por um ruído gaussiano. O estado do sistema é representado como um vetor de números reais. A cada incremento estado de tempo discreto, um operador linear é aplicado ao estado para gerar um novo estado, com algum ruído agregado a ele.

Pra modelar o filtro de Kalman é necessário especificar as matrizes F_k , H_k , Q_k , R_k e B_k para cada índice de tempo k . O modelo usado pelo filtro de Kalman assume que o estado verdadeiro no tempo k evolui a partir do estado em $(k - 1)$:

$$x_k = F_k x_{k-1} + B_k u_k + W_k \quad (3.8)$$

Em que:

- F_k é o modelo de transição de estado, que é aplicado ao estado anterior x_{k-1} ;
- B_k é o modelo de controle de entrada que é aplicado ao vetor de controle u_k ;
- W_k é o ruído do processo, que corresponde a uma variável aleatória normal multivariada de média zero e matriz de variância Q_k , $W_k \sim N(0, Q_k)$

No tempo k uma observação (ou medida) x'_k do estado verdadeiro x_k é realizada de acordo com: $x'_k = H_k x_k + v_k$, em que H_k é o modelo de observação que transforma o espaço de estados reais no espaço de observações e v_k é o ruído da observação, suposto um ruído branco gaussiano com média zero e matriz de variância R_k , $v_k \sim N(0, R_k)$

Vários modelos dinâmicos reais não se encaixam exatamente neste modelo; porém, como o filtro de Kalman foi construído para funcionar na presença de ruído, um modelo aproximado costuma ser suficientemente bom para que o filtro seja útil.

Para o tratamento dos ruídos sobre a nuvem de pontos, foram escolhidos três filtros baseados no filtro Kalman, devido à complexidade numérica e estabilidade do algoritmo que ofereceu quando foram implementados no *software* (Apêndice C), a continuação se descreve cada uno deles.

A - Mínima média quadrática (LMS)

O algoritmo LMS (Least Mean Squares), é um algoritmo adaptativo que utiliza o método da descida (*Steepest Descent*) baseado no uso do gradiente. Esse algoritmo busca a minimização do erro quadrático médio através de suas rotinas que incorporam um procedimento iterativo fazendo as correções nos coeficientes a cada elemento do vetor de dados $d(n)$. O LMS, resumidamente, é obtido a partir da equação abaixo (Bruno, 2007):

$$w(n) = w(n - 1) - \mu_{LMS} e(n)$$

Em que:

$$e(n) = d(n - 1) - g(n) \tag{3.9}$$

Em que:

$w(n)$	Coeficiente atual
$w(n - 1)$	Coeficiente passado
μ_{LMS}	Critério desempenho
$e(n)$	Vetor erro
$d(n)$	Vetor de dados (Sinal lido proveniente da nuvem de pontos)
$g(n)$	Vetor de dados (Sinal gerado pelo algoritmo adaptativo)

B - Normal mínima média quadrática (NLMS)

Para aumentar a velocidade de convergência do algoritmo LMS propôs-se no trabalho de Haykin (1991) uma versão modificada deste: o *Normalized Least Mean Squares*-NLMS. O algoritmo modificado continua a ser pouco exigente em termos computacionais, incluindo agora um passo de adaptação variável e obtido pela expressão:

$$\mu_{NLMS} = \frac{\ell}{P(n)}$$

$$P(n) = d^T(n)d(n) \tag{3.10}$$

Em que ℓ é uma constante positiva de valor inferior à unidade e $P(n)$ é a estimativa da potência do sinal de referência. Então a equação (3.9) se transforma em (Haykin,1991):

$$w(n) = w(n - 1) + \mu_{NLMS}e(n)$$

Em que

$$e(n) = d(n - 1) - g(n) \quad (3.11)$$

Em que:

$w(n)$	Coeficiente atual
$w(n - 1)$	Coeficiente passado
μ_{NLMS}	Critério desempenho
$e(n)$	Vetor erro
$d(n)$	Vetor de dados (Sinal lido proveniente da nuvem de pontos)
$g(n)$	Vetor de dados (Sinal gerado pelo algoritmo adaptativo)

A expressão (3.11) introduz, contudo, um novo problema. Se o vetor de dados de entrada $d(n)$ tomar valores muito baixos será feita a divisão por valor muito pequeno originando problemas numéricos. Para eliminar esse problema, Haykin (1991) determinou que o cálculo do passo de adaptação fosse determinado pela expressão:

$$\mu_{NLMS} = \frac{\ell}{a + P(n)} \quad , \text{com } a > 0 \quad (3.12)$$

A constante a foi escolhida de modo a evitar o *overflow* quando a potência do sinal de entrada tomar valor inferior a um dado limiar.

C - Mínimos quadrados recursivo (RLS)

O algoritmo RLS é semelhante ao LMS por ser um algoritmo que obtém seus coeficientes ótimos de um filtro transversal, que minimiza o erro médio quadrático entre o valor do sinal desejado e seu valor na saída do filtro. Porém, no RLS, o estimador médio é atualizado com base em um conjunto de valores previamente simulados. Enquanto que, na técnica utilizada no LMS, o estimador é atualizado com um único valor. Além disso, outra característica importante é a instabilidade numérica que o mesmo possui. É possível a obtenção dos coeficientes através de:

$$w(n) = w(n - 1) + Q_{RLS}(n)e(n)$$

Em que:

$$e(n) = d(n - 1) - g(n) \quad (3.13)$$

$w(n)$	Coeficiente atual
$w(n - 1)$	Coeficiente passado
Q_{RLS}	Critério desempenho
$e(n)$	Vetor erro
$d(n)$	Vetor de dados (Sinal lido proveniente da nuvem de pontos)
$g(n)$	Vetor de dados (Sinal gerado pelo algoritmo adaptativo)

3.4 – CONSIDERAÇÕES FINAIS

O presente capítulo introduziu o problema do tratamento do ruído. A idéia fundamental foi apresentar a metodologia matemática que permite processar os pontos de modo que a amostra resultante seja mais adequada do que a nuvem de pontos original para uma aplicação específica.

O modelo matemático a trabalhar para o processamento dos dados provenientes da nuvem de pontos está expressa através de $d(n) = hI_p(n) + s(n)$ (equação 3.1), em que $d(n)$ representa a nuvem de pontos provenientes do sistema VISCAN3D, h é a matriz de transformação que seleciona os pontos de uma imagem I_p e $s(n)$ é ruído aleatório associada no processo de aquisição de dados.

Entre vários modelos, foram selecionados os filtros adaptativos que permitem trabalhar com sinais provenientes de ambientes desconhecidos. De acordo com o modelo da equação 3.1, os algoritmos que melhor eliminam ruídos são os baseados no filtro de Kalman, a dizer NLMS e RLS, escolhidos por sua baixa complexidade computacional e alta velocidade de convergência .

No capítulo de modelamento de superfícies (capítulo 5) serão abordadas as dificuldades que envolvem o problema geral da reconstrução e serão apresentados alguns exemplos para indicar de que forma deve-se selecionar o valor de critério desempenho para obter uma construção ótima da superfície.

4 - CURVAS E SUPERFÍCIES PARAMÉTRICAS

4.1 - INTRODUÇÃO

As *B-Splines* racionais não uniformes NURBS (*Non Uniform Rational B-Spline*) são frequentemente usadas para a parametrização de superfícies de forma livre e geometrias comuns, pois permitem gerar representações robustas e flexíveis das mesmas. As curvas e superfícies NURBS tornaram-se de fato padrão de indústria para representação, projeto e troca de informações de dados geométricos processados por computadores. Muitos padrões internacionais, IGES, STEP, PHIGS e OpenGL reconhecem as NURBS como ferramenta poderosas para projeto geométrico (Yang *et al*, 2005).

O sucesso enorme atrás das NURBS deve-se, em grande parte, aos seguintes fatos (Teixeira, 2003):

- Provem de uma base matemática unificada para representar formas analíticas como seções cônicas e superfícies quádricas, além de entidades de forma livre, como corpos de carros de navios ou aviões;
- Suas curvas permitem representar formas arbitrárias enquanto mantem exatidão e independência de resolução matemática;
- Seus algoritmos são rápidos e numericamente estáveis;
- Suas curvas e superfícies são invariantes sob transformações geométricas comuns, como: translação, rotação, projeções de perspectiva;
- São generalizações de curvas e superfícies *B-Splines* racionais e não racionais e Curvas e Superfícies de Bezier não racionais.
- São capazes de proporcionar grande controle sobre a forma de uma curva. Um conjunto de pontos de controle de nós, que guiem a forma da curva pode ser diretamente manipulado para controlar sua suavidade e curvatura;
- Podem representar formas complexas através de poucos dados. Por exemplo: é possível aproximar um círculo por entremeio de uma sequência de segmentos de reta, os quais exigiram dezenas de milhares de segmentos para fazer parecer que é um círculo, ao invés de um polígono. Definir o mesmo círculo com uma representação NURBS comporta somente sete pontos de controle.

O presente capítulo exhibe uma referência dos aspectos matemáticos das funções NURBS, para posteriormente descrever as propriedades importantes que permitem manipular estas funções em *OpenGL*.

4.2 - CURVAS E SUPERFÍCIES DE BEZIER

Um dos métodos mais comuns de representação de curvas e superfícies em modelamento geométrico são as equações implícitas e as funções paramétricas (Wayne, 1997). A equação implícita de uma curva no plano xy tem a forma $C(x, y) = 0$. Essa equação descreve uma relação implícita entre as coordenadas x e y dos pontos sobre a curva. Na forma paramétrica, cada uma das coordenadas de um ponto sobre a curva é representada como função explícita de um parâmetro independente:

$$C(u) = (x(u), y(u)) \quad a \leq u \leq b \quad (4.1)$$

Assim, $C(u)$ é uma função avaliada pela variável independente u no intervalo paramétrico $[a, b]$, o qual usualmente está normalizado entre $[0, 1]$. Similarmente, uma superfície definida por uma equação implícita $f(x, y, z) = 0$ tem uma representação paramétrica da forma:

$$S(u, v) = (x(u, v), y(u, v), z(u, v)) \quad (4.2)$$

A parametrização esquematizada por Bezier em 1972 (*apud* Farin, 2002) baseia-se nos Polinômios de Bernstein de grau n para criar curvas e superfícies paramétricas que dependem de alguns pontos de controle (p_i). Uma Curva de Bezier de grau n tem a forma:

$$C(u) = \sum_{i=0}^n B_{i,n}(u) p_i \quad (4.3)$$

Em que $B_{i,n}(u)$ são funções-base. Essas têm a forma característica dos Polinômios de Bernstein de grau n (Anderson & Crawford-Hines, 2000). Sua formulação é a seguinte:

$$B_{i,n}(u) = \frac{n!}{i!(n-i)!} u^i (1-u)^{n-i} \quad (4.4)$$

As Superfícies de Bezier se definem como:

$$S(u, v) = \sum_{i=0}^n \sum_{j=0}^m B_{i,n}(u) B_{j,m}(v) p_{i,j} \quad (4.5)$$

Das equações 4.3 e 4.5, pode-se depreender que o grau das funções-base que geram as Curvas e Superfícies de Bezier depende do número de pontos de controle com os quais se gera a representação. Essa forma paramétrica apresenta problemas de cálculo em funções-base quando existem muitos pontos de controle (equação 4.4). As Curvas e Superfícies de Bezier podem-se estender em diferentes formas, onde cada ponto de controle p_i tem associado um peso w_i . As *B-Splines* apresentam uma solução para esse problema, ao fazer com que as funções-base existam apenas em uma parte do intervalo paramétrico.

4.3 - FUNÇÕES-BASE *B-SPLINES*

A ordem das funções-base na expansão polinomial de Bernstein usada por Bezier (equação 4.4) depende do número de pontos de controle do modelo. Por isso, são necessárias funções-base de grau $(n - 1)$ para passar uma curva polinomial de Bezier através de n pontos. No entanto, o cálculo de funções de ordem elevada é ineficiente e numericamente instável. Esse tipo de curva tem problemas para ajustar-se em algumas funções complexas. Além disso, o controle obtido sobre a representação não é suficientemente local.

A solução para esse problema é a utilização de funções-base polinomiais por partes conhecidas como *Splines*. Esse tipo de função existe apenas em uma parte do intervalo paramétrico e depende de alguns pontos de controle, os quais permitem ter um controle local da representação. Adicionalmente, a ordem dessas funções pode ser convenientemente escolhida. As funções-base *Splines* são definidas da seguinte forma:

Seja $U = \{u_0, \dots, u_m\}$ uma sequência não decrescente de números reais, por exemplo, $u_i \leq u_{i+1}, i = 0, \dots, m - 1$. Onde u_i são os chamados nós, e U é o vetor de nó.

A i - ésima função base de grau p_{grau} , denotada $N_{i,p_{grau}}$, se define como (Minetto, 2003) :

$$N_{i,0}(u) = \begin{cases} 1 & \text{se } u_i \leq t \leq u_{i+1} \\ 0 & \text{em caso contrario} \end{cases}$$

$$N_{i,p_{grau}}(u) = \frac{u - u_i}{u_{i+p_{grau}} - u_i} N_{i,p_{grau}-1}(u) + \frac{u_{i+p_{grau}+1} - u}{u_{i+p_{grau}+1} - u_{i+1}} N_{i+1,p_{grau}-1}(u)$$

(4.6)

Na figura 4.2, observa-se as funções-base de ordem três geradas a partir do vetor de nós não uniformes $U = \{0, 0, 0, 0, 0.6, 0.6, 0.7, 1, 1, 1, 1\}$.

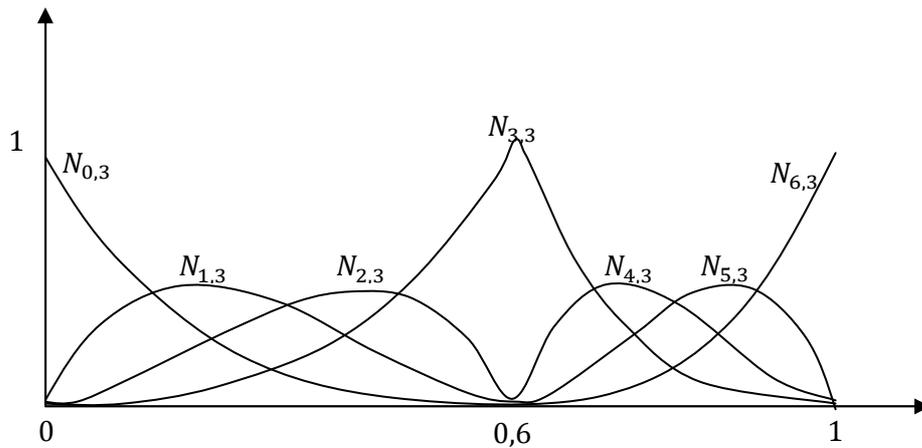


Figura 4.2. Funções-base com nó não uniforme.

4.3.2 - Propriedades das funções-base Splines

As funções *B-Splines* tem uma série de propriedades importantes que determinam as características geométricas desejáveis sobre funções NURBS (curvas e superfícies).

- $N_{i,0}(u)$ é uma função passo que é igual a zero exceto pelo intervalo semi-aberto $u [u_i, u_{i+1})$.
- Se $N_{i,p_{grau}}(u) = 0$ Se u está fora do intervalo $u [u_i, u_{i+p_{grau}+1})$. *Propriedade de suporte local*.
- Em qualquer expansão de nós dada $[u_i, u_{i+1})$, pelo menos $p_{grau} + 1$ das $N_{i,p_{grau}}$ funções-base são diferentes de zero. *Distribuição piramidal*
- $N_{i,p_{grau}}(u) \geq 0$ Para todo i, p_{grau} e u . *Não negatividade*.
- Em um intervalo arbitrário de nós $[u_i, u_{i+1})$, $\sum_{j=i-p_{grau}}^i N_{j,p_{grau}}(u) = 1$, para todo $u \in [u_i, u_{i+1})$. *Partição de unidade*.
- Todas as derivadas de $N_{i,p_{grau}}(u)$ existem dentro do intervalo fornecido pelo vetor de nós U (equação (4.8)). Num nó n , $N_{i,p_{grau}}(u)$ é $p_{grau} - k$ vezes continuamente diferenciável, em que k é a multiplicidade de nó n . *Diferenciação contínua*.
- $N_{i,p_{grau}}(u)$ Tem exatamente um valor máximo (exceto no caso em que $p_{grau} = 0$). *Máximo absoluto*.

4.4 - B-SPLINE

As curvas e superfícies surgem a partir de uma combinação das funções-base *Splines* (equação 4.6) e os pontos de controle. As propriedades das funções-base $N_{i,p_{grau}}(u)$ transferem-se às curvas e superfícies paramétricas *B-splines* com as quais essa representação soluciona vários dos problemas que tem aproximação com funções de Bezier.

4.4.1 - Curvas *B-Splines*

Uma curva *B-Spline* de grau p é definida por:

$$C(u) = \sum_{i=0}^n N_{i,p_{grau}}(u)p_i \quad (4.9)$$

Em que p_i são os pontos de controle e $N_{i,p_{grau}}(u)$ são as funções-base *B-splines* de grau p_{grau} .

As *B-Splines* são uma generalização das funções de Bezier, já que se $p_{grau} = n$ e $U = \{0, \dots, 0, 1, \dots, 1\}$, então, $C(u)$ é uma Curva de Bezier (equação 4.3).

$C(u)$ tem um esquema de modificação local de forma tal que a movimentar o ponto de controle p_i , a curva mudará só no intervalo $[u_i, u_{i+p_{grau}+1})$, o qual permite que as *B-splines* tenham um ajuste mais fino sob aproximação do que as Curvas de Bezier. As propriedades de continuidade e derivação das funções *B-Splines* estão presentes em $C(u)$, já que esta é uma combinação linear de $N_{i,p_{grau}}$ (equação 4.9). Dessa maneira, $C(u)$ é infinitivamente diferenciável dentro dos intervalos definidos pelo vetor de nós U (equação 4.7) e tem pelo menos $p_{grau} - k$ derivadas contínuas num nó de multiplicidade k .

4.4.2 - Superfícies *B-Splines*

Uma superfície *B-Spline* é obtida pela combinação linear de uma rede bidirecional de pontos de controle $p_{i,j}$ e dos produtos das funções *B-Spline* $N_{i,p_{grau}}(u)$ e $N_{j,q_{grau}}(v)$ definidas nos vetores nó U e V , respectivamente.

$$S(u, v) = \sum_{i=0}^n \sum_{j=0}^m N_{i,p_{grau}}(u)N_{j,q_{grau}}(v)p_{i,j} \quad (4.10)$$

As superfícies *B-splines* conformam uma representação unificada que tem todas as propriedades de continuidade e derivação das funções *B-Splines* em ambas as direções paramétricas. Nas superfícies *B-Splines* é gerado um produto tensor das funções-base que apresentam capacidade de ajuste local e alta continuidade. O produto tensor das funções-base $N_{i,p_{grau}}(u) N_{j,q_{grau}}(v)$ com ordem p_{grau} na direção u e q_{grau} na direção v é definido pelos vetores de nós normalizados.

$$U = \left\{ \underbrace{0, \dots, 0}_{p_{grau}+1}, u_{p+1}, \dots, u_{n-1}, \underbrace{1, \dots, 1}_{p_{grau}+1} \right\}$$

$$V = \left\{ \underbrace{0, \dots, 0}_{q_{grau}+1}, v_{p+1}, \dots, v_{m-1}, \underbrace{1, \dots, 1}_{q_{grau}+1} \right\}$$

4.5 - B-SPLINES RACIONAIS NÃO UNIFORMES (NURBS)

Apesar de a representação polinomial oferecer muitas vantagens, existe certos tipos de curvas e superfícies cônicas importantes que não podem ser representadas com *B-Spline*. A solução para esse problema é a utilização de funções racionais, já que estas possibilitam a definição matemática de todas as curvas e superfícies cônicas.

Uma curva polinomial em um espaço $n - dimensional$ tem uma representação racional num espaço $(n + 1)$ dimensional. Sendo $p = (x, y)$ um ponto no espaço euclidiano bidirecional (X, Y) , sua representação racional se estende a um espaço tridimensional (X, Y, W) onde w é nova dimensão.

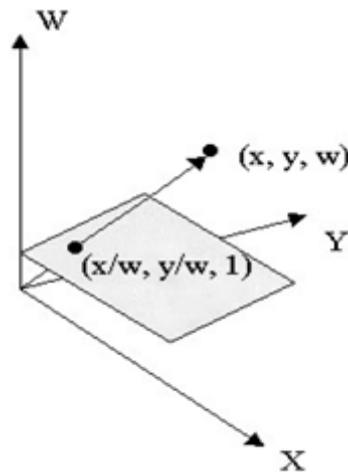


Figura 4.3. Projeção do plano racional W .

4.5.1 - Propriedades da representação racional com *B-Spline*

As Curvas e Superfícies *NURBS* contam com todas as propriedades de continuidade e derivação das *B-Splines*, mas apresentam uma capacidade de aproximação local maior. Isso é observado ao modificar-se o peso w_i em uma curva *NURBS* associado ao ponto de controle p_i , já que este somente afeta a curva no intervalo $u \in [u_i, u_{i+p_{grau}+1})$.

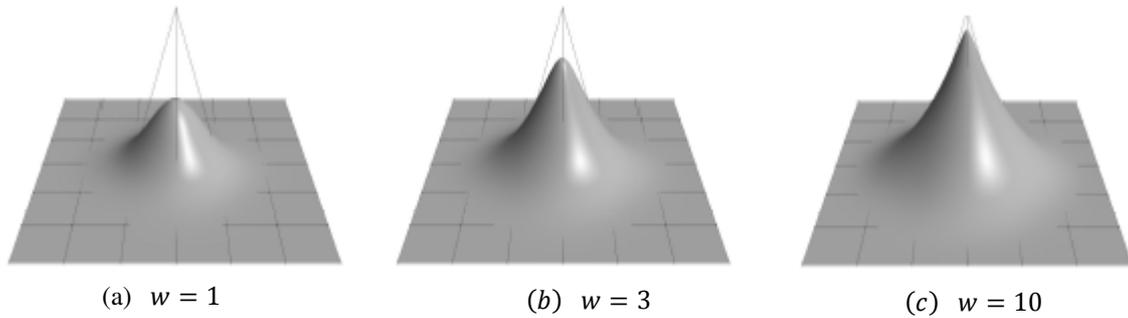


Figura 4.4. Superfícies *NURBS* com diferentes pesos.

Nas figuras 4.4(a), 4.4(b) e 4.4(c) observa-se o efeito da aproximação local numa superfície *NURBS* com a variação do peso associado a um ponto de controle.

As funções *NURBS* tem um efeito de filtro passa-baixa ao aproximar um conjunto de dados. A curva ou superfície paramétrica tem um efeito suave na aproximação que depende da ordem de suas funções-base. Quanto maior for a ordem das funções-base, maior será o efeito passa-baixa da parametrização sobre os dados da faixa e, portanto, maior será o erro da aproximação final. A representação racional não está definida para o espaço w_{neg} , logo, deve-se $w_i \geq 0$ para todo i .

4.5.2 - Curvas e superfícies *NURBS*

Uma curva *NURBS* de grau p é definida pela seguinte equação:

$$C(u) = \frac{\sum_{i=0}^n N_{i,p_{grau}}(u)w_i p_i}{\sum_{i=0}^n N_{i,p_{grau}}(u)w_i} \quad (4.11)$$

Onde $N_{i,p_{grau}}(u)$ são as funções-base *Splines* de grau p_{grau} , p_i são os pontos de controle e w_i são os pesos associados a cada ponto de controle.

A equação geral para as superfícies paramétricas *NURBS* é dada pela equação abaixo:

$$S(u, v) = \frac{\sum_{i=0}^n \sum_{j=0}^m N_{i,p_{grau}}(u) N_{j,q_{grau}}(v) w_{i,j} p_{i,j}}{\sum_{i=0}^n \sum_{j=0}^m N_{i,p_{grau}}(u) N_{j,q_{grau}}(v) w_{i,j}} \quad (4.12)$$

Onde $N_{i,p_{grau}}(u) N_{j,q_{grau}}(v)$ é o produto tensor das funções-base *B-splines* com grau p na direção paramétrica u e de grau q_{grau} na direção v . $p_{i,j}$ é a malha de pontos de controle e $w_{i,j}$ são os pesos associados a cada ponto de controle .

4.5.3 - Graus de liberdade

As funções NURBS permitem adaptar curvas e superfícies polinomiais tanto para a representação de forma livre como para as formas geométricas comuns. Isso se logra pela manipulação dos graus de liberdade que tem as funções NURBS em sua formulação matemática. Essa manipulação permite um controle de aproximação local por meio dos pesos e pontos de controle e um mapeamento não-uniforme da parametrização por meio do vetor paramétrico de nós. Os graus de liberdade das funções NURBS a serem otimizados são:

- Pontos de controle
- Pesos associados aos pontos de controle
- Vetores paramétricos de nós

Para aperfeiçoar a aproximação mediante curvas e superfícies NURBS deve-se modificar seus graus de liberdade a fim de minimizar a distância entre a representação paramétrica e o conjunto de dados (Krishnamurthy & Levoy, 1996).

A seleção do vetor paramétrico de nós tem grande influência no processo de aproximação mediante funções NURBS (Krishnamurthy & Levoy, 1996; Randrianarivony & Brunnett, 2002; Cox *et al.*, 2002).

Esse problema será estudado na seção 4.7, a qual apresenta uma técnica que busca modificar os vetores paramétricos de nós mediante a iteração de Newton.

4.6 - TÉCNICAS PROPOSTAS DE RECONSTRUÇÃO DE SUPERFÍCIES ATRAVÉS DE FUNÇÕES PARAMÉTRICAS

Uma malha representa uma amostra organizada de algumas superfícies fundamentais decompostas em uma coleção ordenada de polígonos. No entanto, essas amostras geralmente não proporcionam um ajuste desejável para todas as aplicações.

Nos últimos anos, pesquisadores no tema tem dado ênfase aos trabalhos que incluem uma combinação de técnicas de ajuste de superfícies NURBS.

Em 1994, Hoppe *et al.* indicaram a primeira solução ao problema de ajuste de superfícies *B-Splines* de superfícies de topologia arbitrária a partir de pontos dispersos não ordenados. O método consistiu na parametrização inicial de uma malha gerada sobre os pontos de dados para construir um domínio quadrilateral. Sobre o domínio quadrilateral realizou-se o ajuste de entalhes de *B-Splines*. O processo de refinamento adaptativo da malha, utilizado para reduzir o erro de ajuste do entalhe a uma tolerância especificada, teve custo computacional alto pela quantidade de passos e processos para reconstruir os entalhes *B-Splines* sobre a superfície de ajuste.

Também em 1996, Krishnamurthy & Levoy propuseram um novo enfoque para o ajuste de entalhes de superfícies NURBS sobre pontos dispersos. O método consistiu na construção de uma malha poligonal sobre o conjunto de pontos. Sobre essa malha realizou-se uma reamostragem para gerar uma malha regular sobre a qual se poderiam ajustar os entalhes de superfícies NURBS.

Elsäesser (1998) descreveu um método para aproximar superfícies NURBS a pontos digitalizados. Esse método será posteriormente aqui descrito e consiste em uma técnica de otimização que modifica os pesos e os pontos de controle mediante uma otimização linear para em seguida encontrar as posições ótimas do vetor paramétrico de nós mediante a iteração de Newton.

Em 1999, Park *et al.* demonstraram um algoritmo de duas fases. Na fase inicial, realizou-se um agrupamento dos pontos mediante um algoritmo de *K-mens*. Uma vez estabelecidos os grupos, construiu-se uma malha poliedral sobre cada um deles para logo reduzi-lo à malha triangular, a qual, por sua vez, se convertia numa malha de quadriláteros. que se utilizava para construir uma rede de entalhes NURBS sobre a totalidade da superfície.

A proposta desses autores presumiu que a nuvem de pontos a ser ajustada era fechada, de modo que sempre seria possível a construção do modelo quadrilateral que dá lugar à rede de entalhes NURBS.

Gregorski *et al.* (2000) elaboraram um algoritmo que decompunha um conjunto de pontos de dados numa estrutura de dados similar a um *quad - tree*, conhecida como *strip - tree*. O *strip - tree* foi utilizado no ajuste de um conjunto de superfícies quadráticas de mínimos quadrados a pontos de dados. Realizou-se uma elevação no grau

das superfícies quadráticas para superfícies bicúbicas. Em seguida, foram misturadas para formar um conjunto de superfícies *B – Splines* que aproxima o conjunto de pontos dados. Essa proposta não pode ser aplicada às superfícies fechadas nem às superfícies que “se curvem” sobre si mesmas.

Kahlesz *et al.* (2002) demonstraram um método que determinava automaticamente os limites comuns das superfícies NURBS. Cada superfície NURBS foi convertida em uma representação poligonal antes do renderização (comumente conhecido como subdivisão de superfícies NURBS). O procedimento de conversão produziu uma aproximação linear contínua da superfície a modelar e posteriormente controlou o erro entre a superfície original e a modelada. Esse processo de modelamento ocasionou dificuldades no tratamento de detalhes finos.

4.7 - AJUSTE ESPACIAL DE SUPERFÍCIES PARAMÉTRICAS

A parametrização de um modelo 3D a partir de uma malha estruturada no espaço R^3 , mediante superfícies paramétricas é um processo que requer um ajuste de aproximação entre estas estruturas geométricas.

Esta seção trata do desenvolvimento matemático necessário para gerar a aproximação de ajuste espacial entre os nós da malha estruturada 3D e as superfícies paramétricas, alterando os graus de liberdade dos entalhes paramétricos NURBS. A descrição do processo de ajuste primeiramente trata de curvas e logo sua formulação se estende às superfícies.

4.7.1 - Formulação do problema

Dada uma seqüência de amostras (u_i, M_i) com $(i = 0, \dots, m)$, pretende-se encontrar a curva $C(u) = C_{W,D,T}(u)$ que melhor se adapte às amostras M_i , minimizando o erro quadrático médio:

$$\min_{W, D, T} \sum_{i=0}^m \|C_{W,D,T}(u_i) - M_i\|^2 \quad (4.13)$$

Onde W são os pesos, D são os pontos de controle e T é o vetor paramétrico de nós.

O problema é difícil de ser abordado, já que não existe uma dependência linear C com os parâmetros W, D, T . Randrianarivony & Brunnett (2002) solucionaram o problema ao

simplificar a função e adicionar algumas penalizações que garantem a formulação matemática das funções NURBS.

Dado um vetor de nós T , o problema pode ser reduzido a:

$$\min_{W, D} \sum_{i=0}^m \|C_{W,D}(u_i) - M_i\|^2 \quad (4.14)$$

A (equação 4.14) pode ser resolvida como um sistema linear que aperfeiçoe os pesos e os pontos de controle para uma seqüência T de nós subministrados (Elsäesser, 1998).

Onde W e D pode definir-se como função de T :

$$(W, D) = (W(T), D(T)) \quad (4.15)$$

A equação (4.14) pode ser escrita como:

$$\min_T \sum_{i=0}^m \|C_T(u_i) - M_i\|^2 \quad (4.16)$$

Pode-se modificar o problema até obter uma expressão vetorial da forma:

$$\min_T \sum_{i=0}^m \|r_i(T)\|^2 \quad (4.17)$$

A equação 4.17 apresenta uma forma típica de otimização por mínimos quadrados que pode ser resolvida por um método não linear como: Iteração de Newton, ou Levenberg-Marquardt dentre outros (Elsäesser, 1998; Xie & Qin., 2001).

4.7.2 - Projeção de um ponto a uma curva

O método utilizado para minimizar a distância entre um ponto p e a curva $C(u)$ é a Iteração de Newton. A convergência do método de Newton depende de um ponto de partida u_0 (Elsäesser, 1998).

Considerando a propriedade de controle local (seção 4.3.2), a curva pode ser avaliada nos intervalos do vetor de nós, nos quais cada ponto de controle p_i tem efeito.

Seja o produto escalar entre a deriva da curva $C(u)$ e a projeção do ponto p sobre a curva tem-se (Nakamura, 2005):

$$f(u) = C'(u) \cdot (C(u) - p) \quad (4.18)$$

Em que, $C'(u)$ é derivada da curva em um valor de u determinado.

A distância entre p e $C(u)$ é mínima quando $f(u) = 0$.

A Iteração de Newton minimiza a equação (4.18) através da equação (4.19):

$$u_{i+1} = u_i - \frac{f(u_i)}{f'(u_i)} = u_i - \frac{C'(u_i) \cdot (C(u_i) - p)}{C''(u_i) \cdot (C(u_i) - p) + |C'(u_i)|^2} \quad (4.19)$$

4.7.3 - Implementação

O algoritmo implementado para achar a projeção das curvas NURBS tem critérios de parada para deter a Iteração de Newton em cada projeção (Piegl & Tiller, 1997):

- *Distância mínima* do ponto p ao ponto da curva $C(u_i)$ - compara a magnitude da projeção com um limiar de erro fornecido;

$$|C(u_i) - P| \leq \varepsilon_1 \quad (4.20)$$

- *Cosseno Zero* - Indica se o vetor da derivada sobre a curva de projeção é perpendicular;

$$\frac{|C'(u_i) \cdot (C(u_i) - P)|}{|C'(u_i)| \cdot |C(u_i) - P|} \leq \varepsilon_2 \quad (4.21)$$

- *Controle local* - estabelece se uma partição paramétrica serve para buscar o valor inicial da Iteração de Newton

$$\begin{aligned} \text{Se } (u_{i+1} < a) \quad u_{i+1} &= a \\ \text{Se } (u_{i+1} > b) \quad u_{i+1} &= b \end{aligned} \quad (4.22)$$

A propriedade de controle local diminui o tempo de cálculo do algoritmo devido, a que reduz o tamanho do intervalo de pesquisa do valor inicial u_0 para a Iteração de Newton.

4.8 – CONSIDERAÇÕES FINAIS

O presente capítulo descreveu a funções NURBS que serão utilizadas para a parametrização de superfícies com base em malha estruturada. O método consiste na construção de uma malha poligonal, usada para preencher superfícies paramétricas sobre o conjunto de pontos obtidos do sistema VISSCAN 3D.

O preenchimento das superfícies NURBS é um processo de minimização de distância entre os pontos de controle da malha estruturada e os entalhe paramétricos, e que pode ser expressa para curvas através da equação (4.16) e ampliada para superfícies na equação (4.23)

$$\min_T \sum_{i=0}^m \|S_T(u_i) - M_i\|^2 \quad (4.23)$$

em que T são os vetores paramétricos do nós, M_i são os pontos nodais da malha estruturada e S a superfície parametrizada.

A equação 4.23 é uma expressão não linear que pode ser resolvida através da otimização de mínimos quadrados. Dentre vários, foi escolhido o método iterativo de Newton pela alta convergência e baixa complexidade computacional.

No seguinte capítulo (capítulo 5) serão abordadas as dificuldades que atingem o problema do processo de preenchimento e adaptabilidade destas estruturas.

5 - PROCESSO DE RECONSTRUÇÃO DE SUPERFÍCIES

5.1 - INTRODUÇÃO

Este capítulo aborda o processo de reconstrução de superfícies baseado na nuvem de pontos obtida com digitalizador óptico.

Apresentam-se os resultados obtidos de:

- Tratamento de ruídos através algoritmos adaptativos: NLMS¹⁰ e RLS¹¹;
- Reconstrução de uma malha triangular 3D, através de método baseados em modelos deformáveis;
- Reconstrução de áreas desconhecidas através de interpolação.
- Reconstrução de superfícies mediante uso de uma rede de entalhes paramétricos bicúbicos, desenhada a partir da malha triangular 3D;
- Suavização da superfície através ajuste contínuo dos entalhes paramétricos;
- Adaptação dos entalhes paramétricos sobre malha triangular.

5.2 - SENSOR ÓPTICO LASER

O escâner VISSCAN-3D está baseado no princípio da triangulação de luz (Ginani & Motta, 2007). Uma das vantagens do escâner VISSCAN-3D está no fato dele não depender de sensores de posicionamento para medir o deslocamento angular da fonte *laser*. Entretanto, a determinação desse deslocamento angular é necessária para a obtenção do mapa de profundidade da superfície a ser digitalizada. Contudo, para realizar esta medição é necessário substituir o sensor de posicionamento por uma fonte *laser*, sendo o sistema como um todo composto por dois diodos *laser* e uma câmera, conforme ilustrado no esquemático simplificado na figura 5.1.

¹⁰ NLMS Normal Mínima Média Quadrática

¹¹ RLS Mínimos Quadrados Recursivo

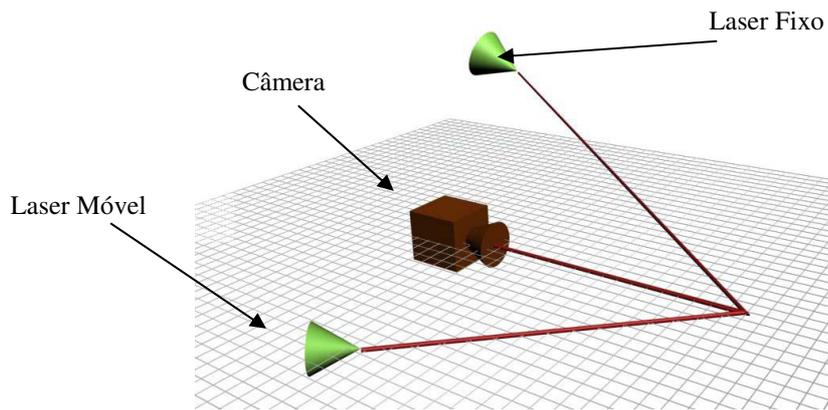


Figura 5.1. Esquemático do Sistema VISSCAN-3D.
Fonte: Ginani & Motta, 2008.

A figura 5.2 ilustra a versão de campo do *hardware* do Sistema VISSCAN-3D:

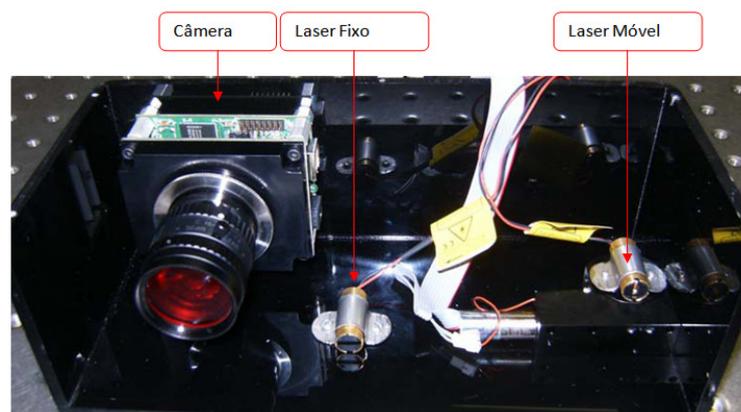


Figura 5.2. Versão de campo do *hardware* do sistema VISSCAN-3D.
Fonte: Ginani & Motta, 2008.

As figuras 5.3 e 5.4 apresentam o processo de escaneamento e nuvem de pontos obtida através do sistema VISSCAN3D.

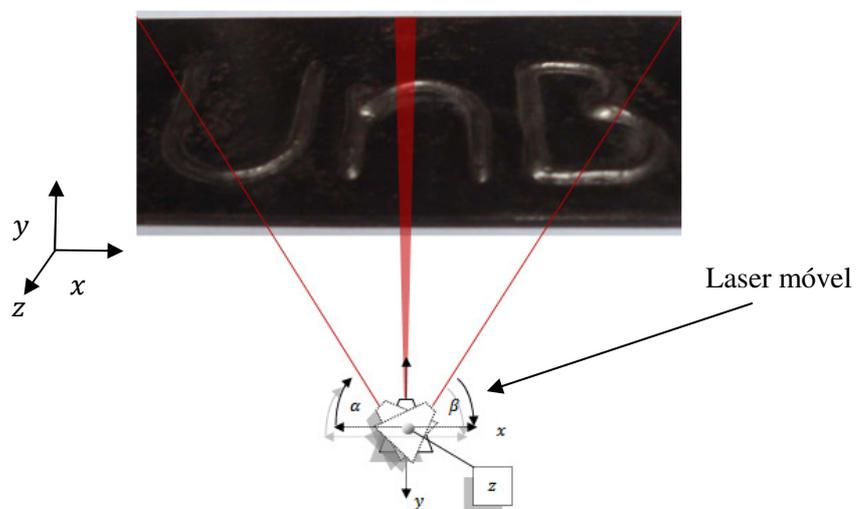


Figura 5.3. Processo de escaneamento. (α e β são os ângulos de rotação inicial e final).

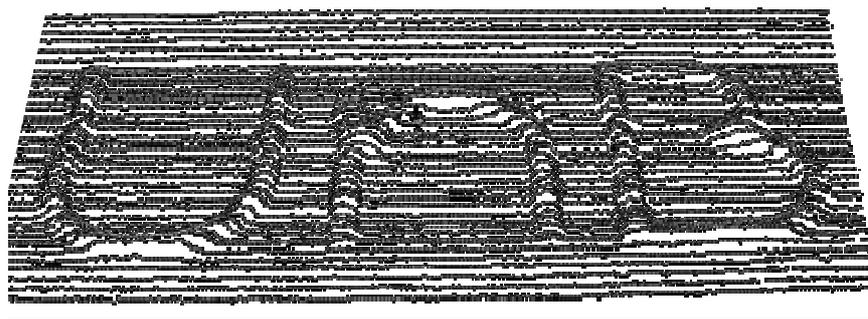


Figura 5.4. Imagem de profundidade (Nuvem de pontos).

5.3 - TRATAMENTO DE RUÍDOS ATRAVÉS ALGORITMOS ADAPTATIVOS

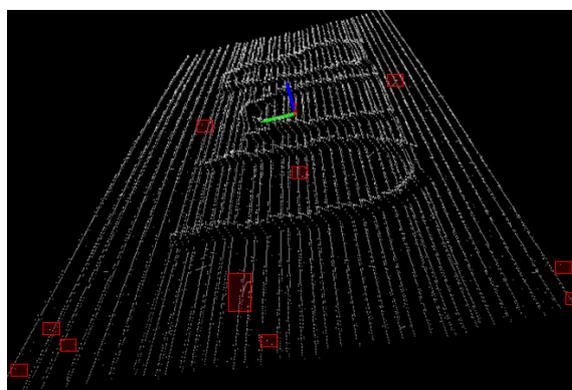


Figura 5.5. Identificação visual dos ruídos sobre a nuvem de pontos.

Para eliminar o ruído adquirido no processo de aquisição¹² de uma nuvem de pontos é preciso realizar uma filtragem de dados. O processo de filtragem permite que a amostra resultante seja mais adequada para trabalhar do que a nuvem de pontos original. Em geral, os ruídos que contaminam a nuvem de pontos aparecem de forma aleatória a cada novo processo de aquisição e, portanto, não podem ser atenuados com sucesso através de filtros digitais convencionais (passa-baixas, passa-altas, passa-faixas).

Devido a incerteza associada à natureza dos dados, técnicas adaptativas surgiram para a atenuar ou até mesmo eliminar ruídos por meio de um critério de desempenho¹³.

A figura 5.5 apresenta alguns ruídos identificados visualmente e que estão associados à nuvem de pontos.

¹² Ruídos em Imagens Digitais(apêndice E)

¹³ Critérios de desempenho:coeficientes dos filtros adaptativos que permitem definir a resposta desejada.

5.3.1 - Formato de arquivo da nuvem de pontos

O formato de arquivo "**IMG.txt*" foi o padrão de organização da informação tridimensional criado e adotado para a representação das superfícies digitalizadas pelo sistema VISSCAN 3D. Esse arquivo armazena o resultado final do processo de digitalização contendo todas as informações sobre a nuvem de pontos obtida.

O arquivo "**IMG.txt*", é um arquivo de texto que apresenta as informações de coordenadas em unidade S.I. (mm). Esse formato proporciona uma arquitetura de organização baseada em blocos ou linhas digitalizadas.

Uma vez conhecendo todas as informações do formato em que se encontra armazenada a nuvem de pontos, o passo seguinte consiste em realizar a filtragem dos dados em toda a superfície geométrica (x, y, z) .

As linhas digitalizadas da nuvem de pontos podem ser modeladas como a saída de uma imagem de entrada (I_p), que contém a informação tridimensional de uma região específica da superfície (figura 5.6). Esta imagem é descomposta em três componentes $[x, y, z]$ através de critérios de filtragem FIR¹⁴ de resposta impulso e de duração finita N (equação 5.1).

$$h = [h_0 \ h_1 \ \dots \ h_{N-1}] \quad (5.1)$$

A saída das componentes será:

$$\begin{aligned} d_x(n) &= hI_p(n) + s(n) \\ d_y(n) &= hI_p(n) + s(n) \\ d_z(n) &= hI_p(n) + s(n) \end{aligned} \quad (5.2)$$

Em que $s(n)$ representa o ruído aleatório que contamina a nuvem de pontos.

Para atenuar o ruído $s(n)$ deve-se estimar o valor h através de critérios de desempenho (w) gerados por algoritmos adaptativos:

$$\begin{aligned} g_x(n) &= wI_p(n) \\ g_y(n) &= wI_p(n) \\ g_z(n) &= wI_p(n) \end{aligned} \quad (5.3)$$

Em que:

$$w = [w_0 \ w_1 \ \dots \ w_{N-1}] \quad (5.4)$$

¹⁴ **FIR** (*Finite-duration Impulse Response*)

A estimativa do ruído pode ser subtraída do sinal lido proveniente da nuvem de pontos (equação 5.2) menos o sinal de gerado pelo algoritmo adaptativo (equação 5.3)

$$e_x(n) = d_x(n) - g_x(n) = (h - w)I_p(n) + s(n) \quad (5.5)$$

$$e_y(n) = d_y(n) - g_y(n) = (h - w)I_p(n) + s(n) \quad (5.6)$$

$$e_z(n) = d_z(n) - g_z(n) = (h - w)I_p(n) + s(n) \quad (5.7)$$

Para calcular a estimativa do erro médio $\bar{e}(n)$ do processo de convergência foram utilizadas as equações (5.5), (5.6) e (5.7).

$$\bar{e}(n) = \frac{e_x(n) + e_y(n) + e_z(n)}{3} \quad (5.8)$$

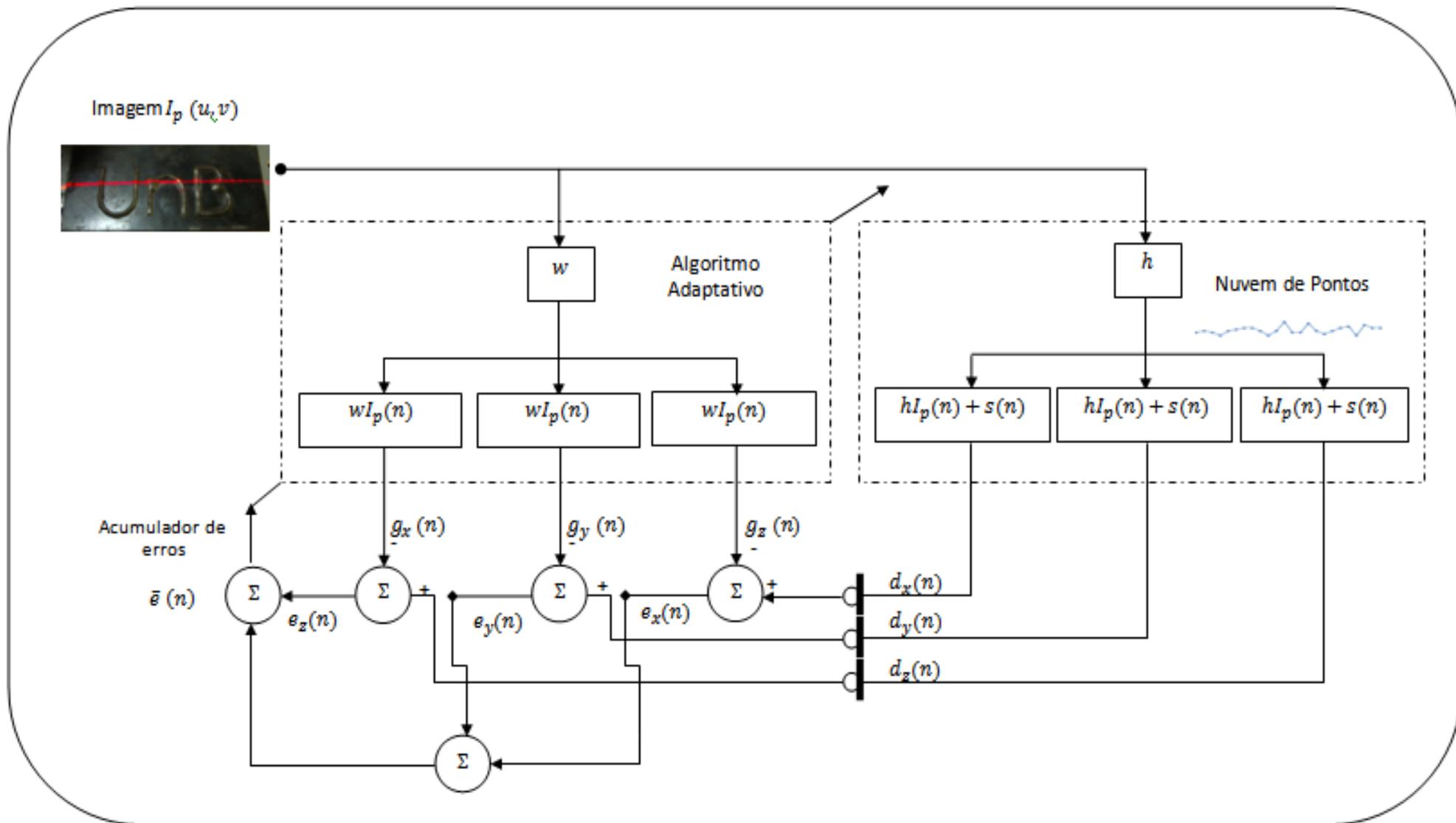


Figura 5.6. Diagrama do processo de cancelamento de ruídos nas componentes $[x, y, z]$ para uma linha digitalizada da imagem de entrada (I_p)

5.3.2 - Seleção do Critério de desempenho em um algoritmo adaptativo.

Para ilustrar a utilização do algoritmo adaptativo sobre uma nuvem de pontos foram selecionados 22 pontos da imagem de profundidade (figura 5.5) que estão descritos nas figuras 5.7 a 5.10 sob diferentes perspectivas.

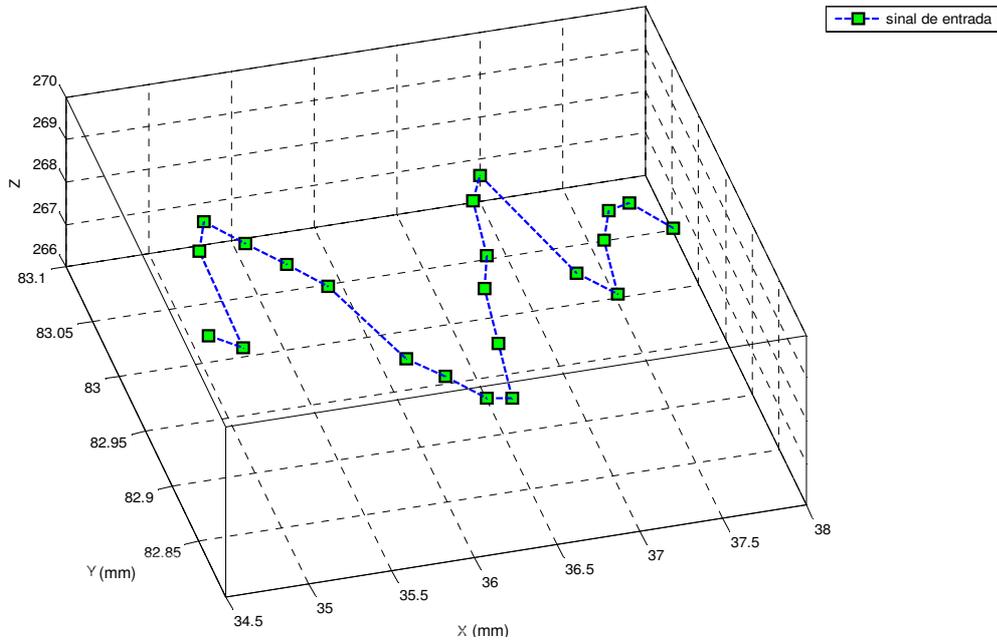


Figura 5.7. Plotagem dos pontos observados no plano [X, Y, Z].

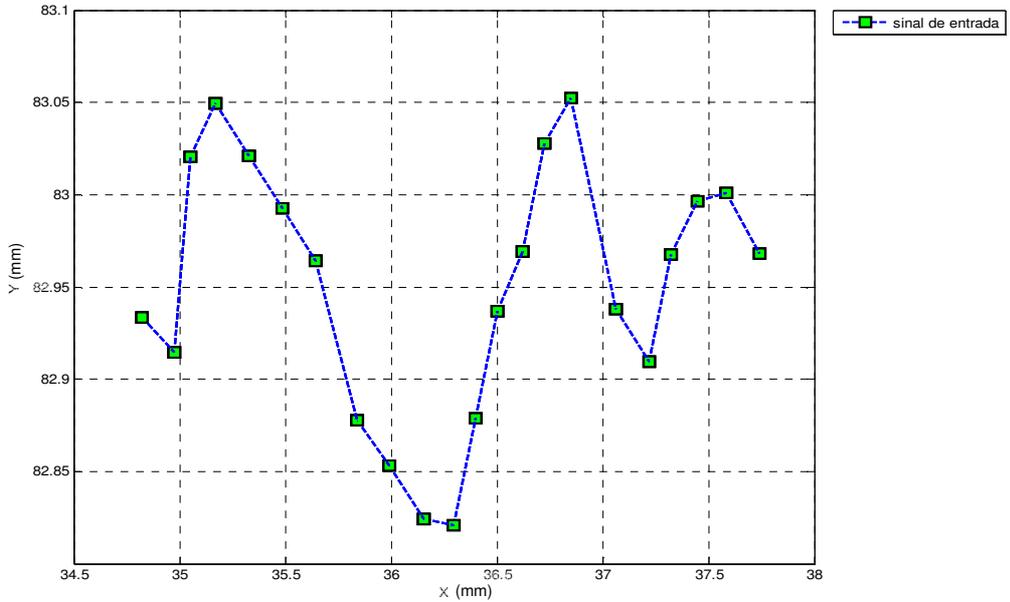


Figura 5.8. Plotagem dos pontos observados no plano [X, Y].

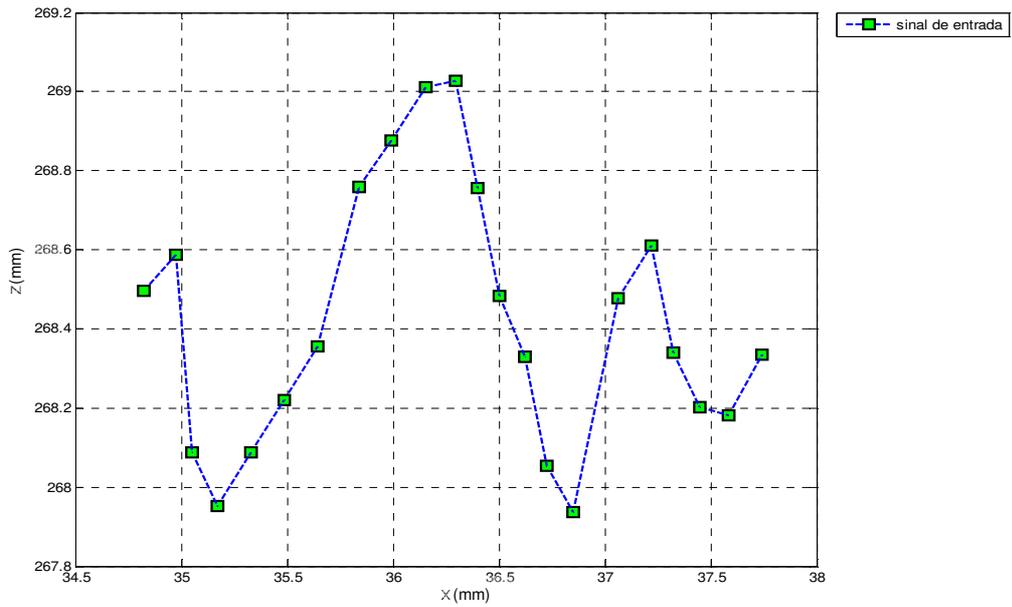


Figura 5.9. Plotagem dos pontos observados no plano [X, Z]

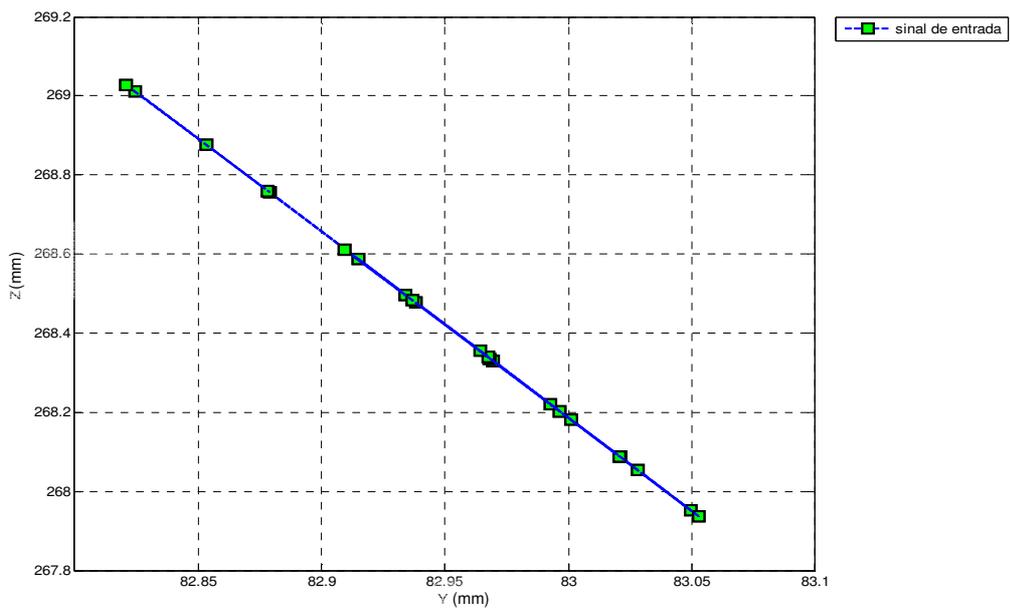


Figura 5.10. Plotagem dos pontos observados no plano [Y,Z].

Para modelar o sinal lido e proveniente da nuvem de pontos $d(n)$ deve-se gerar um sinal $g(n)$ pelo algoritmo adaptativo, de tal forma que w se aproxime de h (equação 5.5, 5.6 e 5.7).

No caso específico para um algoritmo NLMS¹⁵ a expressão $w(n)$ é:

$$w(n) = w(n - 1) - \mu_{LMS} e(n)$$

Em que:

$$e(n) = d(n - 1) - g(n)$$

(5.9)

Em que:

$w(n)$	Coeficiente atual
$w(n - 1)$	Coeficiente passado
μ_{NLMS}	Critério desempenho
$e(n)$	Vetor erro
$d(n)$	Vetor de dados (Sinal lido proveniente da nuvem de pontos)
$g(n)$	Vetor de dados (Sinal gerado pelo algoritmo adaptativo)

O valor do critério de desempenho depende exclusivamente da forma visual em que foi construída a superfície. Por exemplo, para um valor $\mu_{NLMS} = 0.1$ o sinal gerado nos três componentes é:

Nota: No eixo das abscissas se encontram os valores dos índices dos pontos e no eixo das ordenadas a magnitude das componentes em unidades milimétricas.

- **Para X**

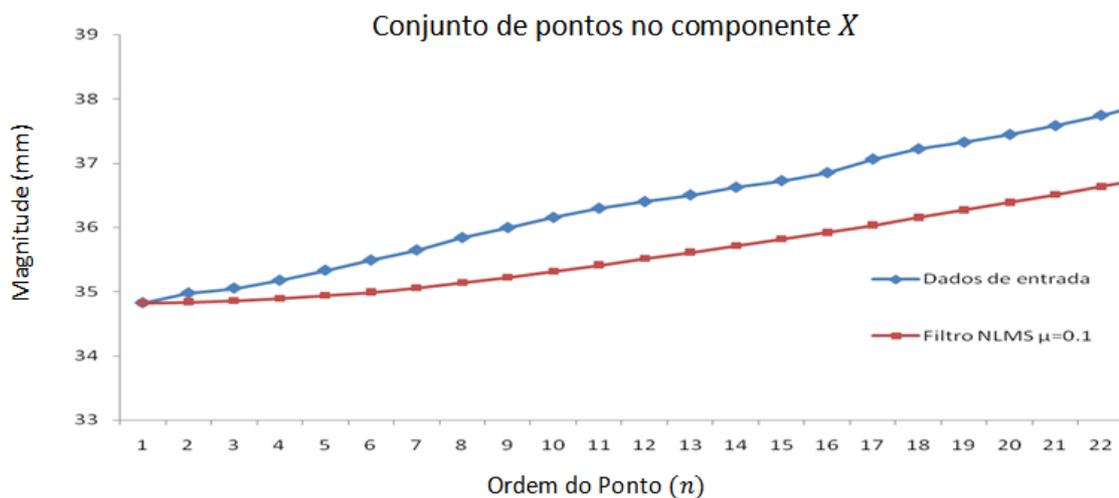


Figura 5.11. Plotagem dos pontos no componente X.

¹⁵ NLMS Normal Mínima Média Quadrática

Na Figura 5.11, o sinal azul representa os dados provenientes da nuvem de pontos d_1 (equação 5.2) o sinal vermelho representa os dados gerados pelo algoritmo adaptativo g_1 (equação 5.3) para $\mu_{NLMS} = 0,1$.

- **Para Y**

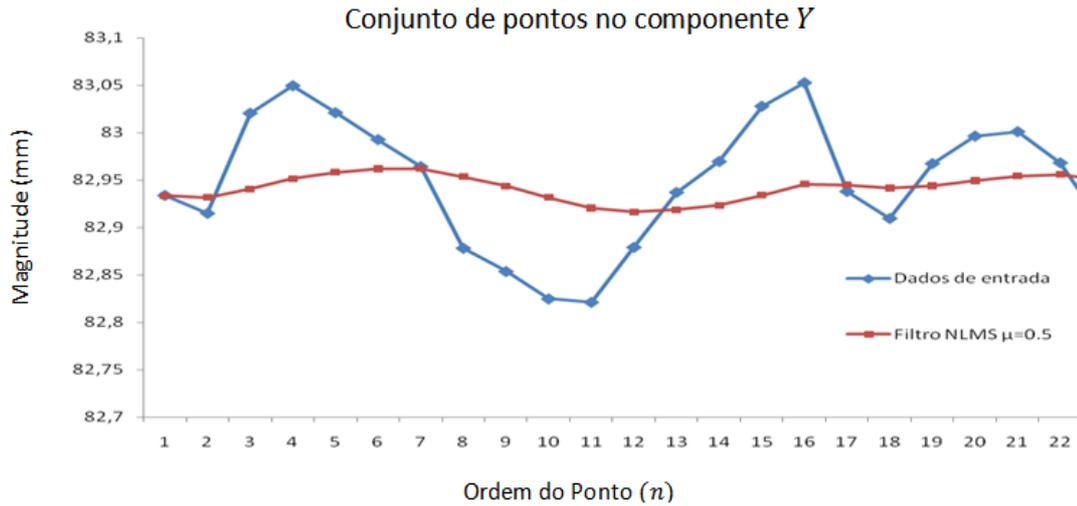


Figura 5.12. Plotagem dos pontos no componente Y.

Na Figura 5.12, o sinal azul representa os dados provenientes da nuvem de pontos d_2 (equação 5.2) o sinal vermelho representa os dados gerados pelo algoritmo adaptativo g_2 (equação 5.3) para $\mu_{NLMS} = 0,1$.

- **Para Z**

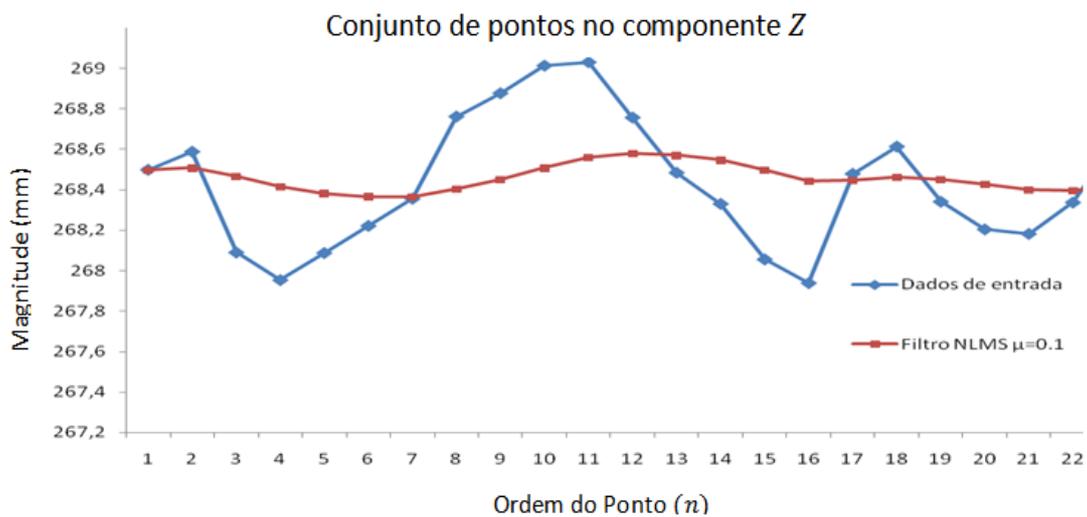


Figura 5.13. Plotagem dos pontos no componente Z

Na Figura 5.13, o sinal azul representa os dados provenientes da nuvem de pontos d_3 (equação 5.2) o sinal vermelho representa os dados gerados pelo algoritmo adaptativo g_3 (equação 5.3) para $\mu_{NLMS} = 0,1$.

Utilizando o mesmo gráfico para visualização do comportamento do sinal gerado pelo algoritmo adaptativo (cor vermelha) e o sinal lido proveniente da nuvem de pontos de entrada (cor azul) tem-se (figura 5.14):

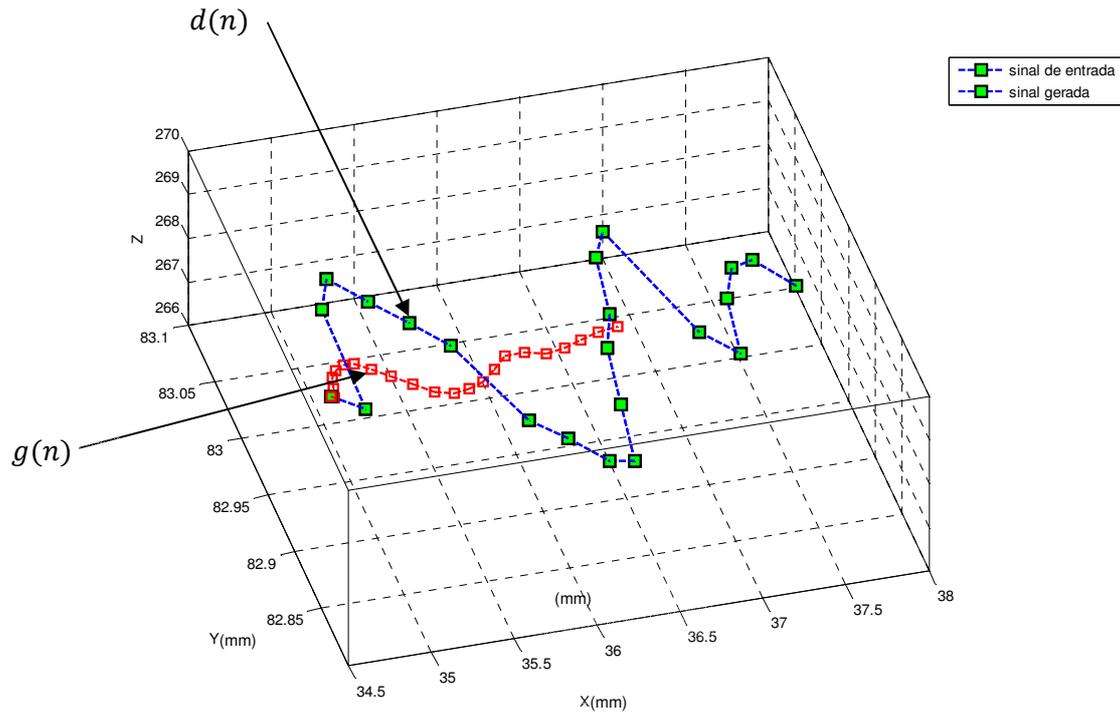


Figura 5.14. Plotagem dos pontos observados no plano [X, Y, Z]

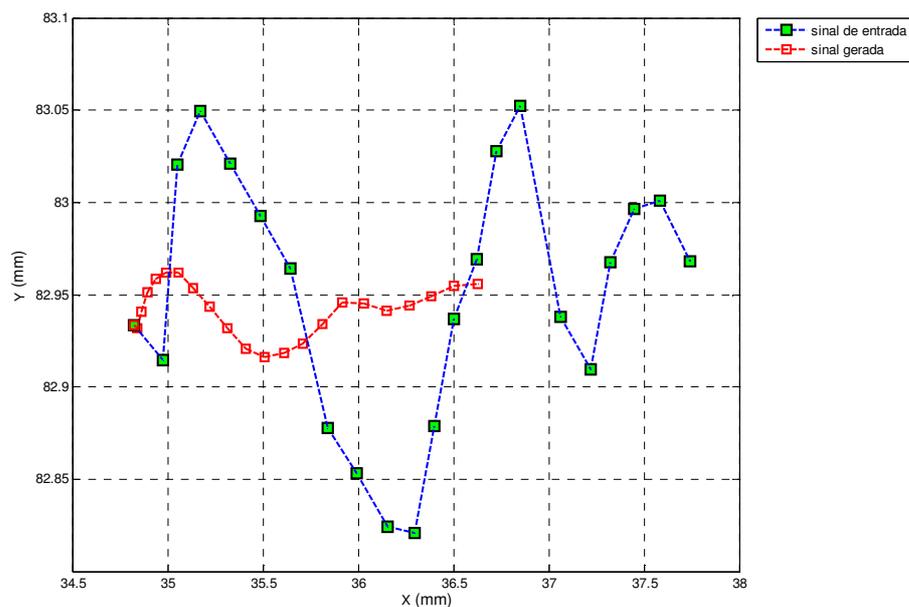


Figura 5.15. Plotagem dos pontos observados no plano [X, Y].

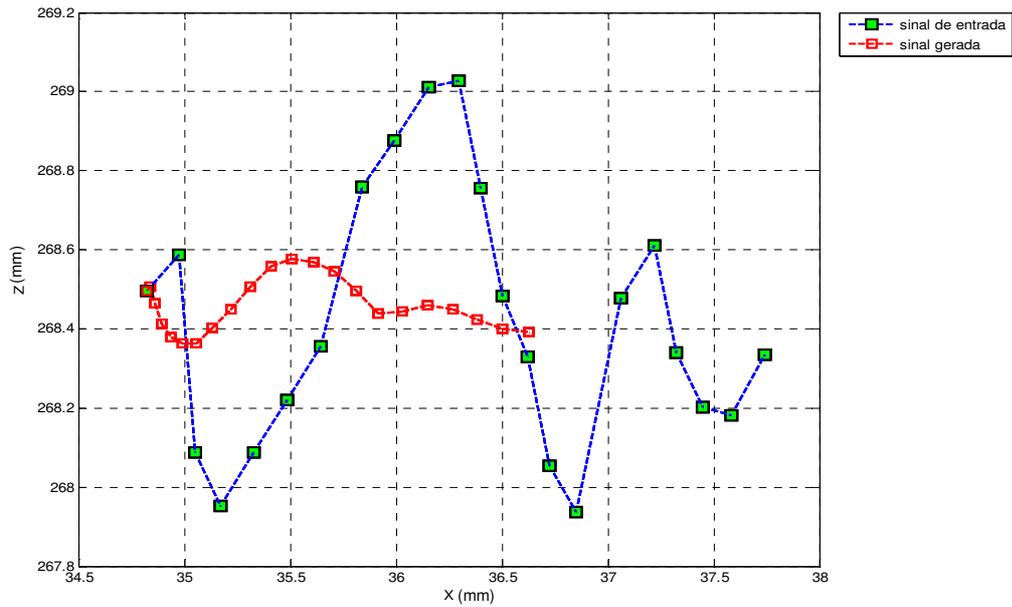


Figura 5.16. Plotagem dos pontos observados no plano [X, Z].

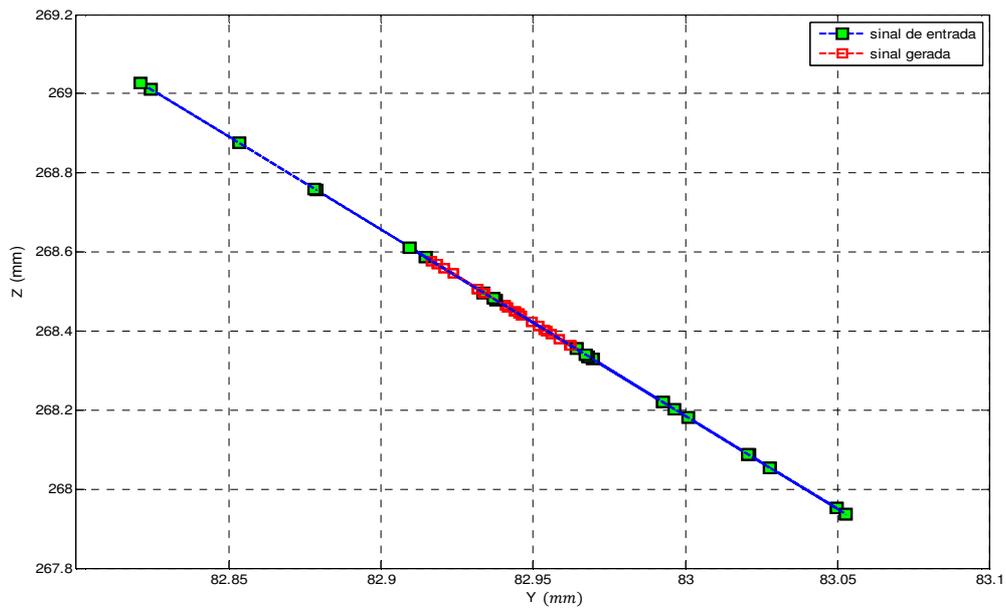


Figura 5.17. Plotagem dos pontos observados no plano [Y, Z].

Como mostram as figuras 5.14 a 5.17, o valor $\mu_{NLMS} = 0,1$ não permite a convergência entre o sinal gerado pelo algoritmo adaptativo e o sinal proveniente da nuvem de pontos, entretanto, observa-se que existe uma diminuição da magnitude de seus componentes. Por outro lado, ao utilizar um valor de critério mais alto para $\mu_{NLMS} = 0,7$, ocorrem os seguintes resultados:

- **No componente X:**

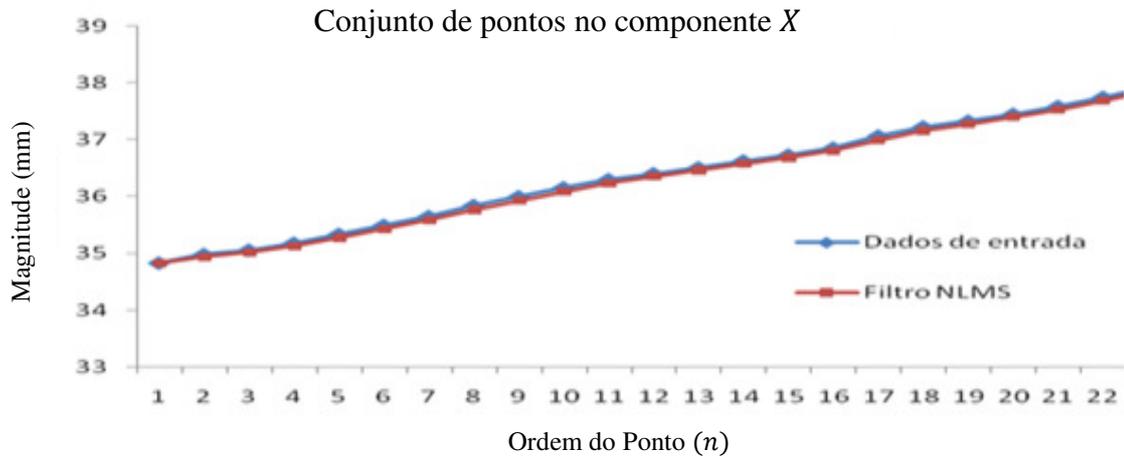


Figura 5.18. Plotagem dos pontos no componente X.

Na Figura 5.18, o sinal azul representa os dados provenientes da nuvem de pontos d_1 (equação 5.2) o sinal vermelho representa os dados gerados pelo algoritmo adaptativo g_1 (equação 5.3) para $\mu_{NLMS} = 0,7$.

- **No componente Y:**

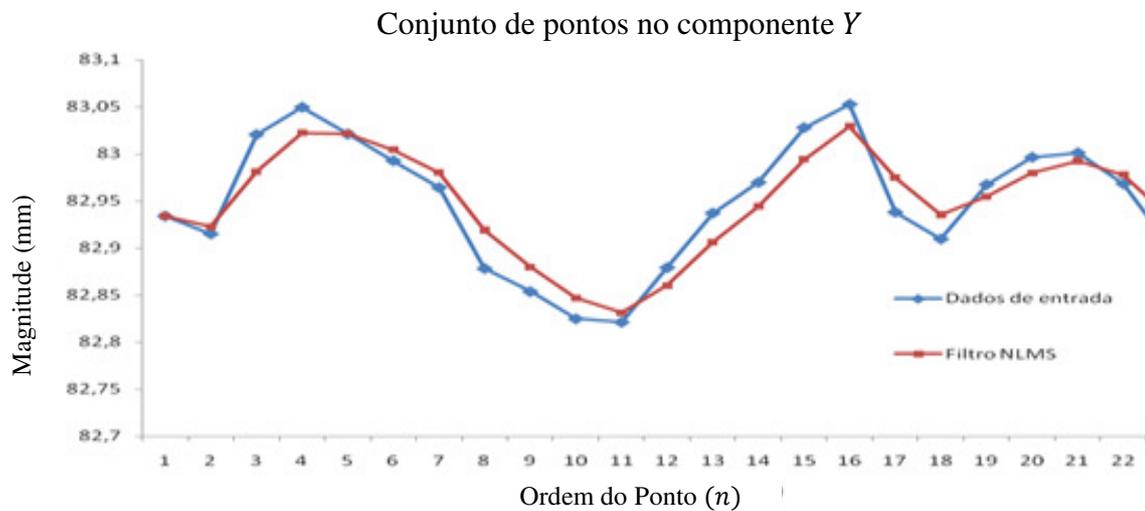


Figura 5.19. Plotagem dos pontos no componente Y

Na Figura 5.19, o sinal azul representa os dados provenientes da nuvem de pontos d_2 (equação 5.2) o sinal vermelho representa os dados gerados pelo algoritmo adaptativo g_2 (equação 5.3) para $\mu_{NLMS} = 0,7$.

- **No componente Z:**

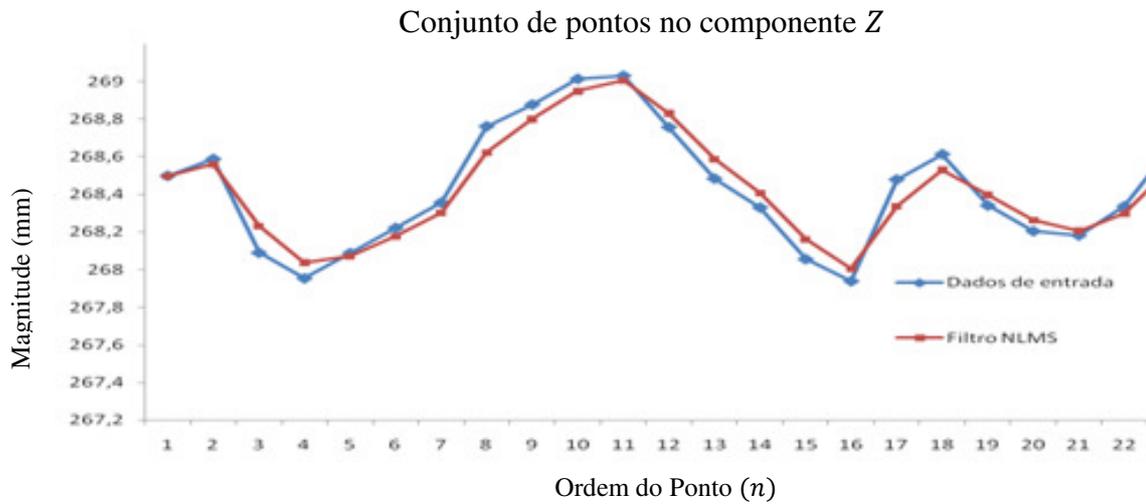


Figura 5.20. Plotagem dos pontos no componente Z.

Na Figura 5.20, o sinal azul representa os dados provenientes da nuvem de pontos d_3 (equação 5.2) o sinal vermelho representa os dados gerados pelo algoritmo adaptativo g_3 (equação 5.3) para $\mu_{NLMS} = 0,7$.

Ploteando na mesma figura o sinal gerado pelo algoritmo adaptativo $g(n)$ (cor vermelha) e o sinal proveniente da nuvem de pontos $d(n)$ (cor azul) tem-se:

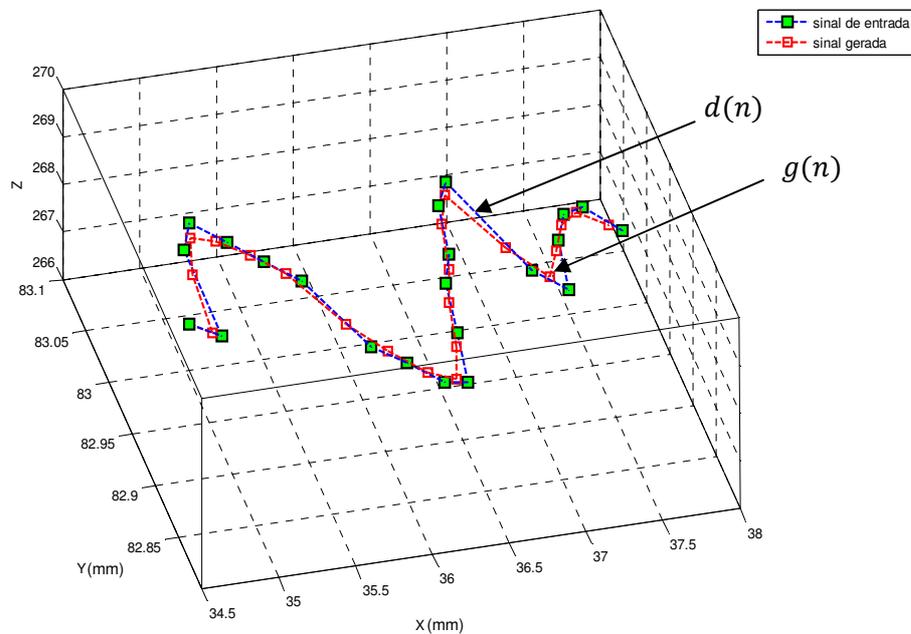


Figura 5.21. Plotagem dos pontos observados no plano [X, Y, Z].

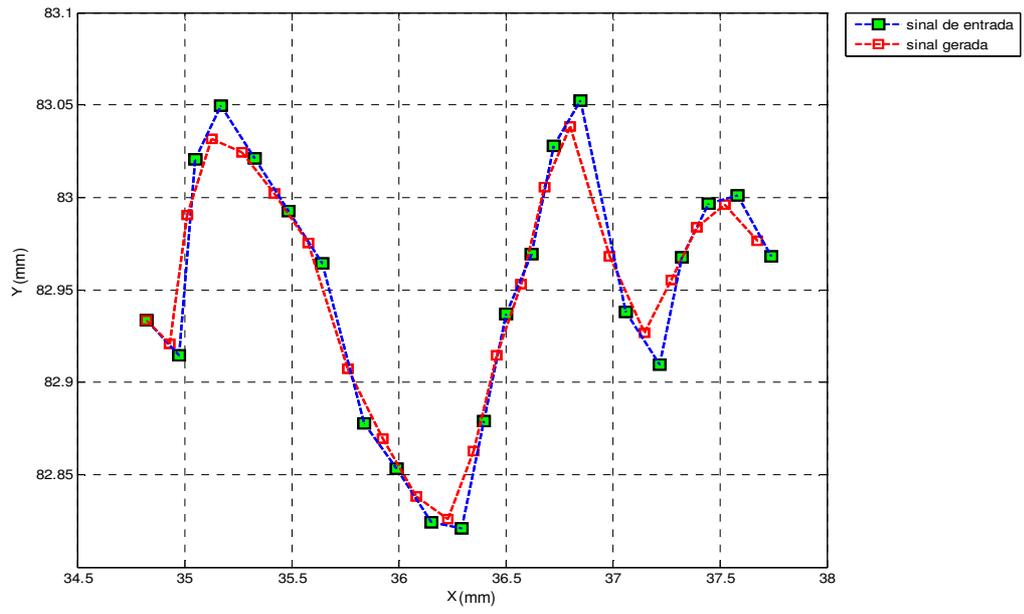


Figura 5.22. Plotagem dos pontos observados no plano [X, Y].

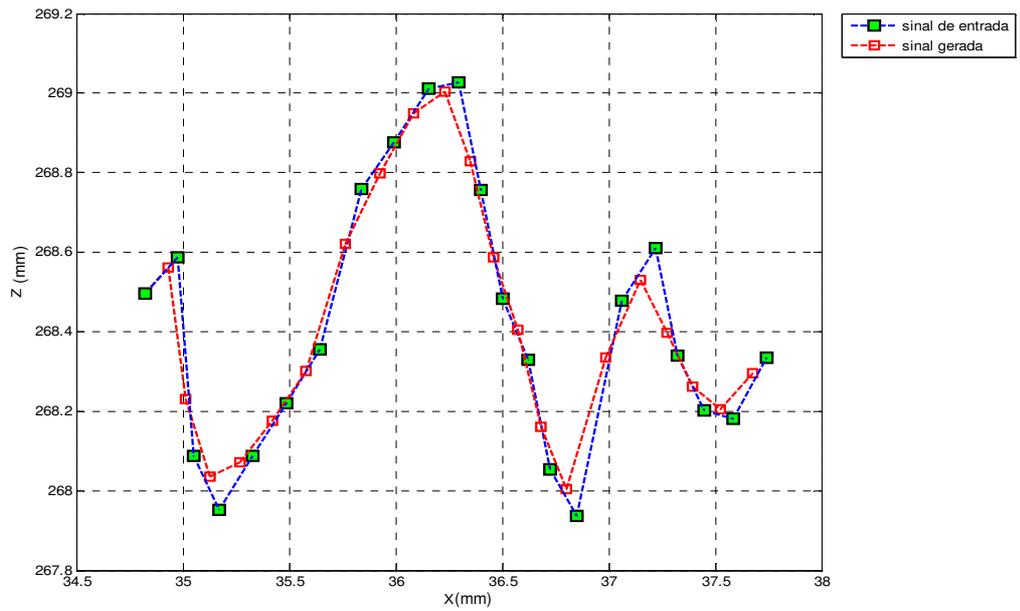


Figura 5.23. Plotagem dos pontos observados no plano [X, Z].

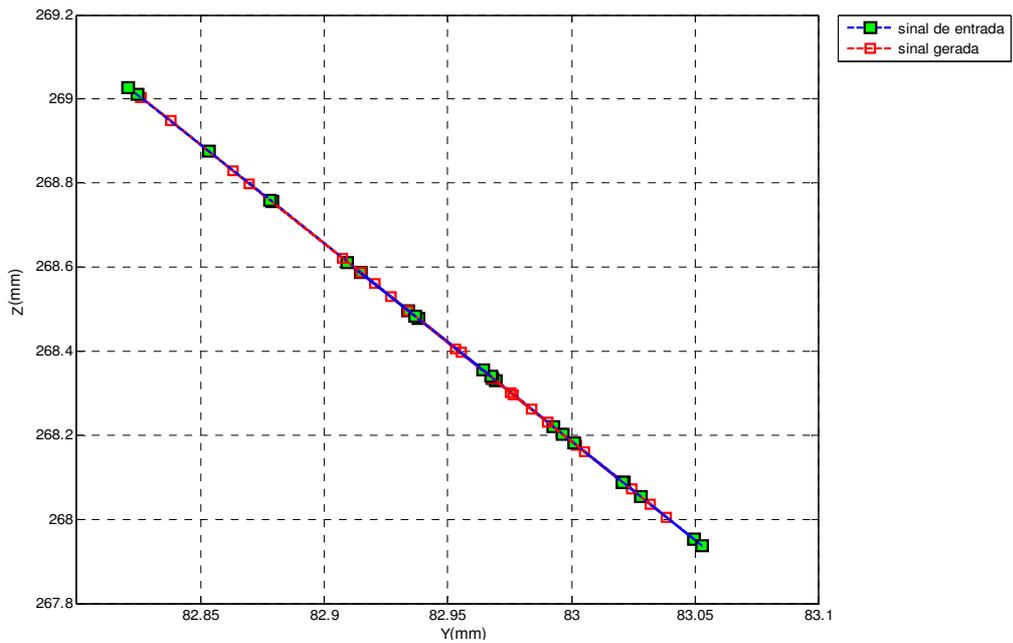


Figura 5.24. Plotagem dos pontos observados no plano [Y, Z].

Como demonstram as figuras 5.21 a 5.24, o valor $\mu_{NLMS} = 0.7$ permite uma maior convergência entre o sinal gerado pelo algoritmo adaptativo e o sinal de entrada proveniente da nuvem de pontos.

A faixa de operação do algoritmo NLMS pode ser configurada no *software* (Apêndice C) entre [0,1 e 1,0] com 10 posições de seleção. O valor (0,1) corresponde ao estágio em que o filtro elimina a maioria dos ruídos, todavia causa uma demasiada degradação das informações obtidas da nuvem de pontos (figura 5.25 a). Por outro lado, o valor (1,0) corresponde ao estágio em que a saída da nuvem de pontos gerada pelo algoritmo converge na nuvem de pontos de entrada sem praticamente eliminar a presença de ruídos (figura 5.25 c).

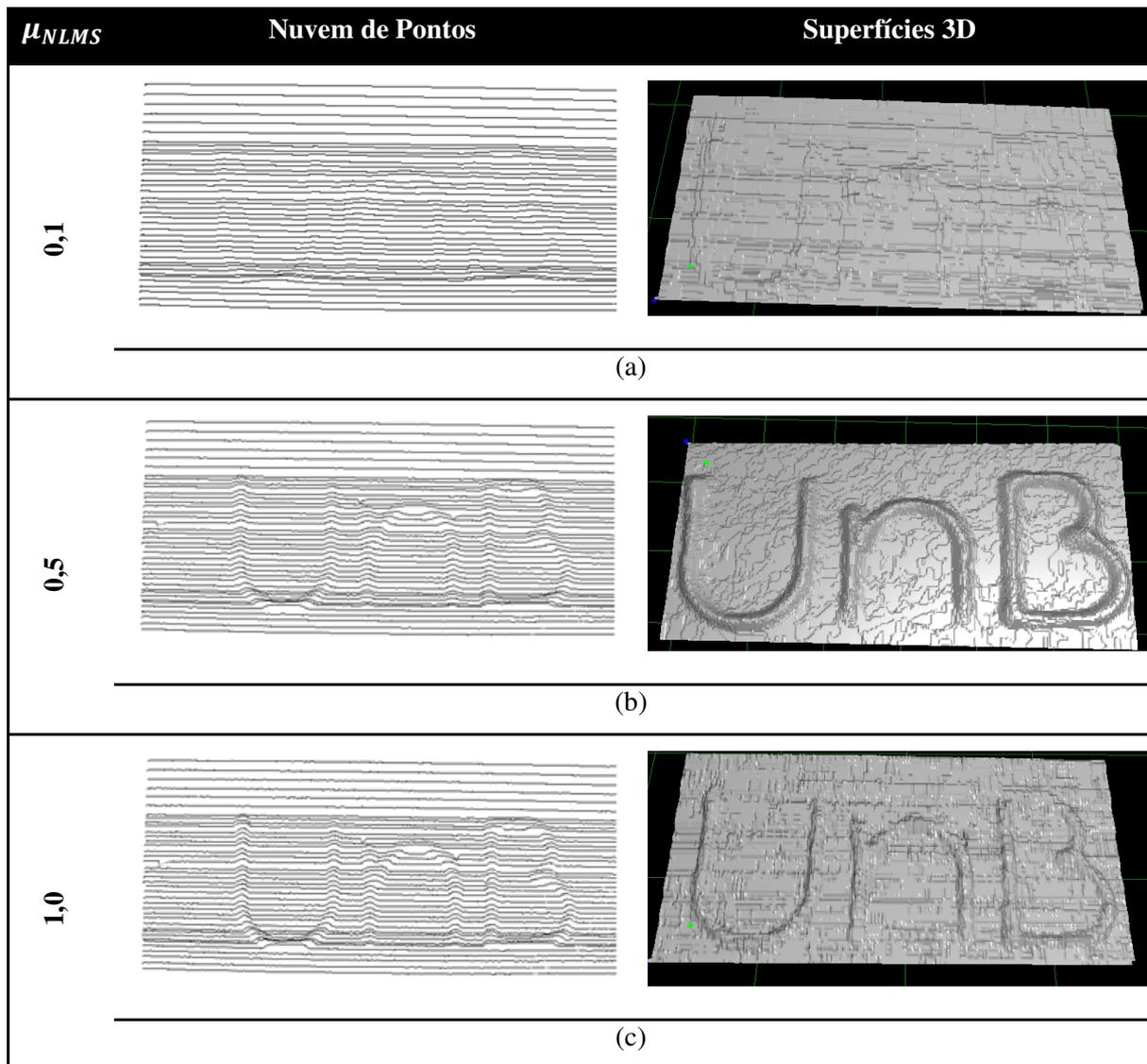


Figura 5.25. Apresentação da nuvem de pontos aplicando diferentes valores de μ_{NLMS} . Para todas as superfícies 2D e 3D foram utilizadas as mesmas condições de construção (Iluminação e Textura).

5.3.3 - Qualidade de aproximação dimensional através do algoritmo adaptativo

Quando se filtra a nuvem de pontos através de algoritmos adaptativos existe uma diminuição dimensional do objeto nas coordenadas em que se aplica o algoritmo.

Através do acumulador de erro médio (equação 5.8) foi possível determinar o valor aproximado da diminuição dimensional que experimenta uma superfície da nuvem de pontos original. A tabela 5.1 apresenta os valores de aproximação obtidos para nuvem de pontos selecionada, utilizando os dois algoritmos adaptativos NLMS¹⁶ e RLS¹⁷ inseridos no *software* (Apêndice C).

¹⁶ NLMS *Normal Mínima Média Quadrática*

Repetindo as equações tem-se,

- Equação (5.11)

Algoritmo NLMS

$$w(n) = w(n - 1) - \mu_{NLMS} e(n)$$

Em que:

$$e(n) = d(n - 1) - g(n)$$

- Equação (5.13)

Algoritmo RLS

$$w(n) = w(n - 1) + Q_{RLS} e(n)$$

Em que:

$$e(n) = d(n - 1) - g(n)$$

- Valor de erro no componente x (equação 5.5),

$$e_x(n) = d_x(n) - g_x(n) = (h - w)I_p(n) + s(n)$$

- Valor de erro no componente y (equação 5.6),

$$e_y(n) = d_y(n) - g_y(n) = (h - w)I_p(n) + s(n)$$

- Valor de erro no componente z (equação 5.7)

$$e_z(n) = d_z(n) - g_z(n) = (h - w)I_p(n) + s(n)$$

- Acumulador de erro médio (equação 5.8)

$$\bar{e}(n) = \frac{e_x(n) + e_y(n) + e_z(n)}{3}$$

Em que:

$w(n)$	Coeficiente atual
$w(n - 1)$	Coeficiente passado
μ_{NLMS}, Q_{RLS}	Critério desempenho
$e(n)$	Vetor erro
$d(n)$	Vetor de dados (Sinal lido proveniente da nuvem de pontos)

Tabela 5.1. Acumulador de erro médio em função do critério desempenho de um algoritmo adaptativo

Algoritmo Adaptativo	Critério Desempenho	$e_1(n)$ (mm)	$e_2(n)$ (mm)	$e_3(n)$ (mm)	$\bar{e}(n)$ (mm)	
NMLS	μ	0,1	1,56	1,74	1,63	1,64
		0,2	1,54	1,48	1,55	1,52
		0,3	1,49	1,38	1,47	1,44
		0,4	1,31	1,21	1,37	1,29
		0,5	1,22	1,18	1,32	1,24
		0,6	1,13	1,07	1,22	1,14
		0,7	0,84	0,9	1,11	0,95
		0,8	0,57	0,45	0,91	0,62
		0,9	0,29	0,40	0,75	0,48
		1,0	0,14	0,21	0,47	0,27
RLS	Q	0,1	1,64	1,72	1,75	1,70
		0,2	1,56	1,64	1,57	1,59
		0,3	1,32	1,35	1,38	1,35
		0,4	1,27	1,25	1,22	1,24
		0,5	1,18	1,17	1,19	1,18
		0,6	1,01	1,04	1,11	1,05
		0,7	0,75	0,81	1,01	0,90
		0,8	0,44	0,32	0,85	0,53
		0,9	0,32	0,22	0,58	0,37
		1,0	0,12	0,17	0,32	0,20

Para maior compreensão dos dados descritos na tabela 5.1, apresenta se os gráficos 5.26 e 5.27, estes gráficos mostram o erro médio em função aos valores de critério de desempenho em cada uma de suas componentes $[x, y, z]$.

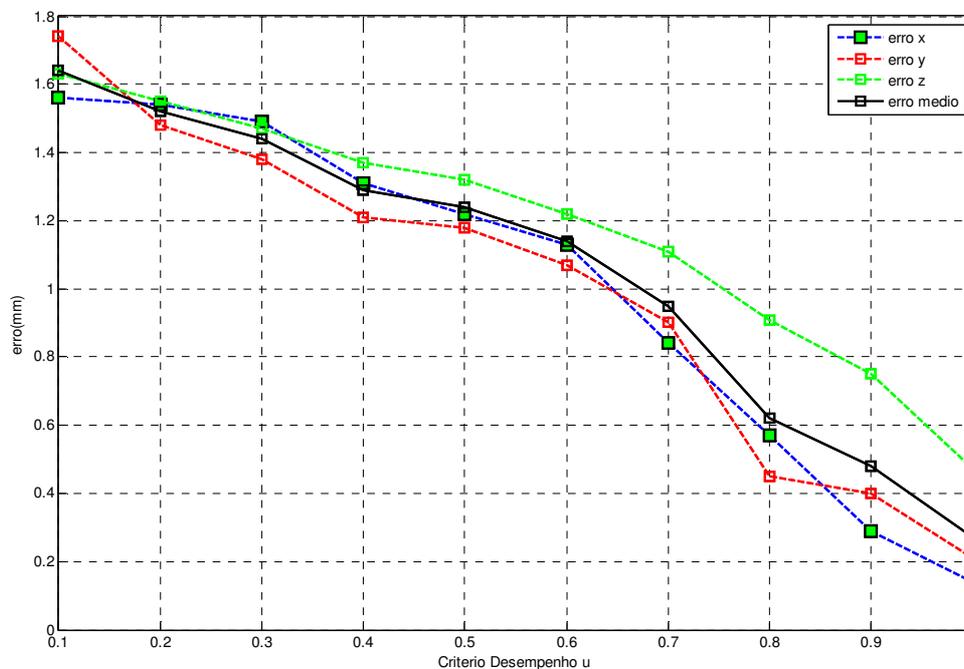


Figura 5.26. Relação entre o erro associado e o critério de desempenho no algoritmo NLMS.

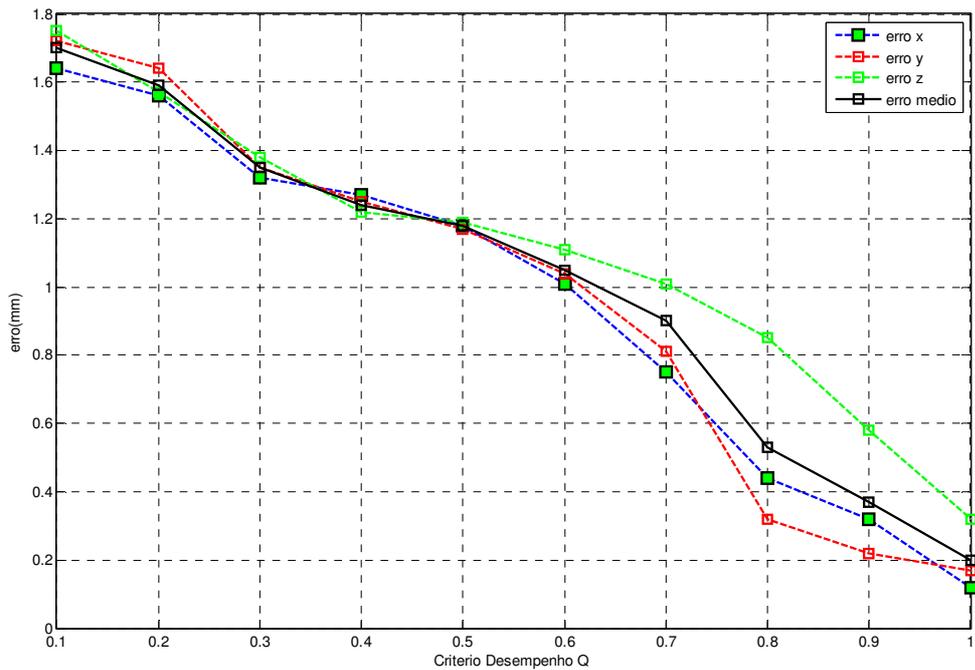


Figura 5.27. Relação entre o erro associado e o critério de desempenho no algoritmo RLS.

No gráfico 5.26 e 5.27, observa-se que a medida que se aumenta o valor de critério (μ ou Q) diminui o erro. Este comportamento permite ao usuário selecionar o valor de critério de desempenho mais adequado a os interesses específicos, para cada nuvem de pontos processados pelo *software* desenvolvido (apêndice C).

5.4 - CONSTRUÇÃO DE UMA MALHA ESTRUTURADA

Depois de se processar a nuvem de pontos através do filtro adaptativo selecionado como critério de desempenho ($\mu_{NLMS} = 0.5$) procedeu-se a construção de uma malha estruturada através do método baseado em modelos deformáveis .

Continuando o processo do algoritmo proposto por Chui *et al.*(2008), projetou-se a nuvem de pontos de R^3 para R^2 sobre o plano σ (figura 5.28).

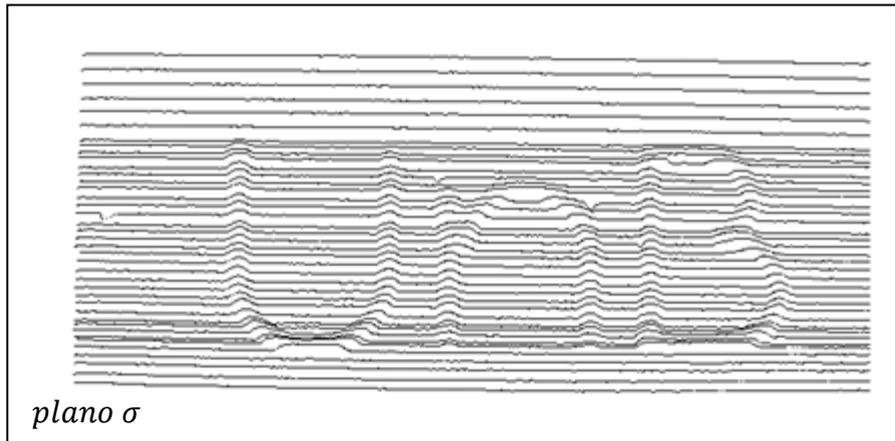


Figura 5.28. Projeção da nuvem de pontos de R^3 para R^2 sobre plano σ .

Posteriormente ajustou-se a nuvem de pontos dentro de um paralelogramo (ângulo de 60°). Por meio desse polígono garantiu-se que todos os pontos da amostra fossem analisados em todos os passos posteriores do algoritmo. Em seguida, dividiu-se o polígono em linhas horizontais e verticais separadas a uma distância t^{18} de 4 mm . Esse valor influencia a resolução da reconstrução tridimensional da superfície. Por último, projetaram-se linhas diagonais da esquerda à direita do polígono, formando triângulos equiláteros (figura 5.29).

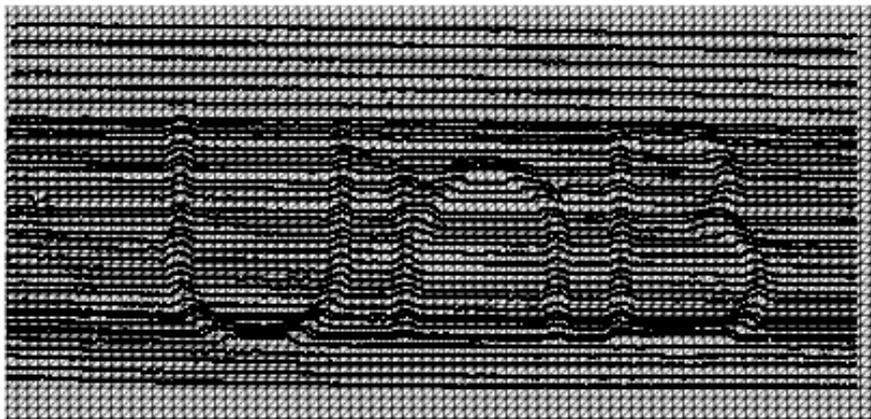


Figura 5.29. Projeção da nuvem de pontos sobre uma malha padrão.
(separação entre as linhas horizontais e verticais 4 mm).

Depois de se projetar a nuvem de pontos no plano σ , realizou-se um reposicionamento de pontos nodais e uma interpolação de dados para encontrar e ajustar a informação daquelas regiões incompletas (seção 5.5) (figura 5.29).

A malha triangular 3D pode ser reconstruída a partir dos valores de profundidade de cada ponto nodal, que foi influenciado pelo ponto selecionado através dos critérios A, B

¹⁸ t . Distância de separação entre linhas de uma malha padrão

ou C da seção (2.2.7). Nas figuras (5.30), (5.31) e (5.32) é apresentada a reconstrução da superfície 2D e 3D.

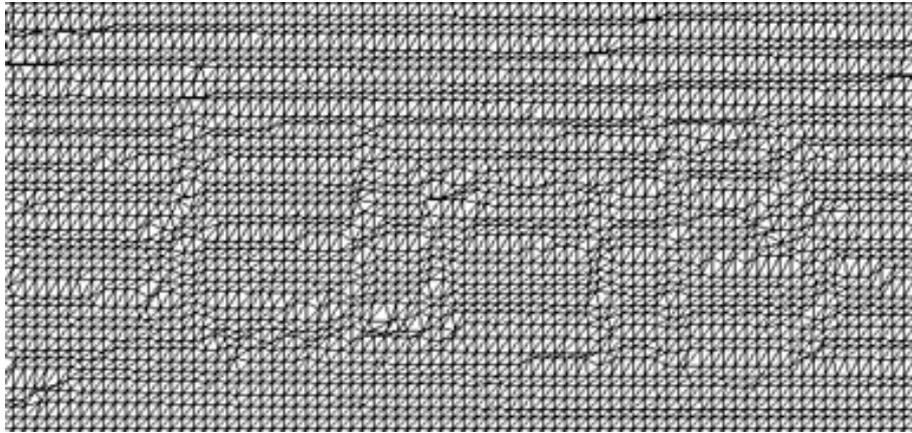


Figura 5.30. Reposicionamento dos pontos nodais influenciado pela nuvem de pontos.

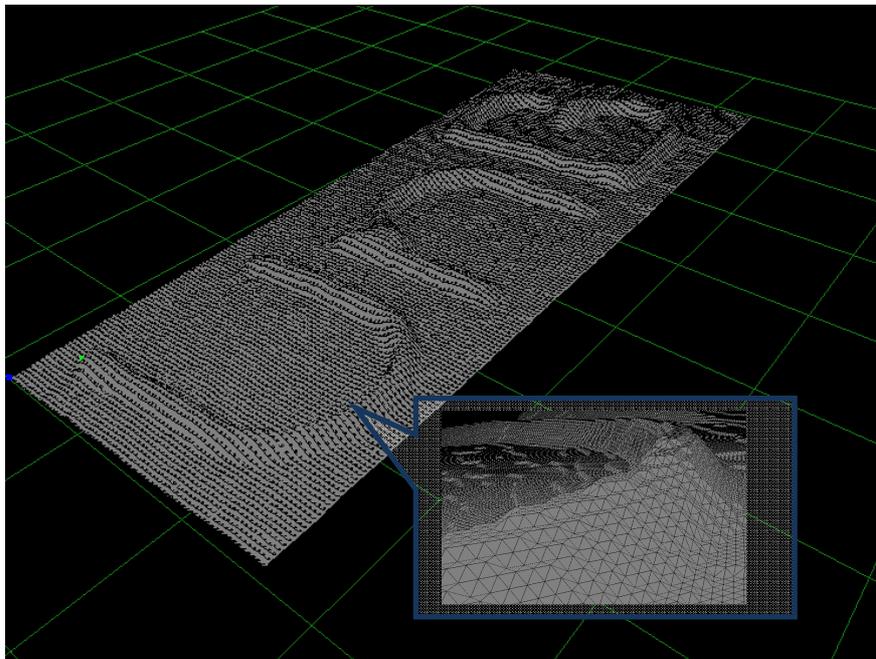


Figura 5.31. Levantamento tridimensional da superfície (cor cinza- Apêndice A).

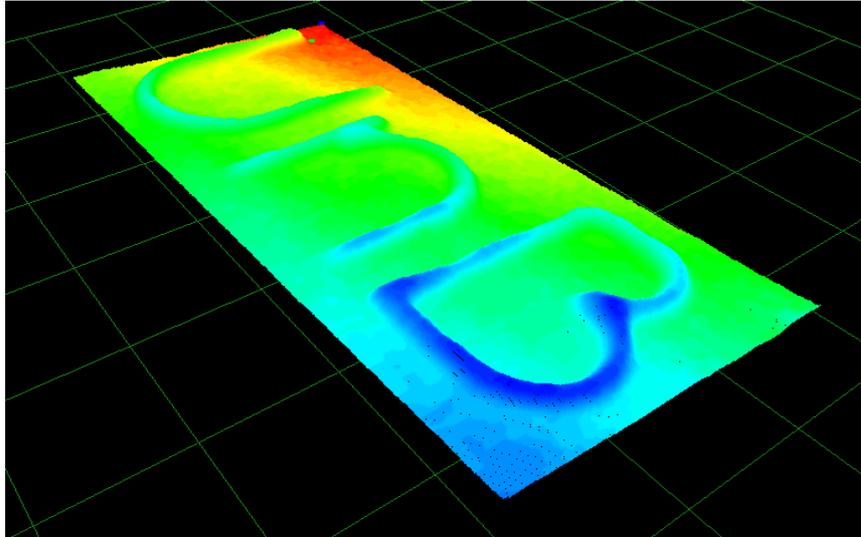


Figura 5.32. Levantamento tridimensional da superfície (cor RGB- Apêndice A).

5.5 - TRATAMENTO DA INFORMAÇÃO EM NUVEM DE PONTOS COM REGIÕES DESCONHECIDAS

A utilização de abordagens adaptativas para tratamentos de ruídos e poligonalização da nuvem de pontos melhora consideravelmente a apresentação dos objetos com uma carga computacional menor. Todavia, nem sempre a amostragem tratada é suficiente para reconstruir a superfície. Frequentemente, algumas nuvens de pontos tem regiões desconhecidas que podem ser de grande interesse para análises posteriores.

Através da interpolação de *C-Splines* é possível encontrar e ajustar a informação daquelas regiões desconhecidas através de uma curva contínua com um grande número de pontos (Nakamura, 2005). A essência da interpolação de *C-Splines* consiste em aplicar um polinômio cúbico a cada intervalo entre os pontos de dados consecutivos como ilustra a figura 5.33.

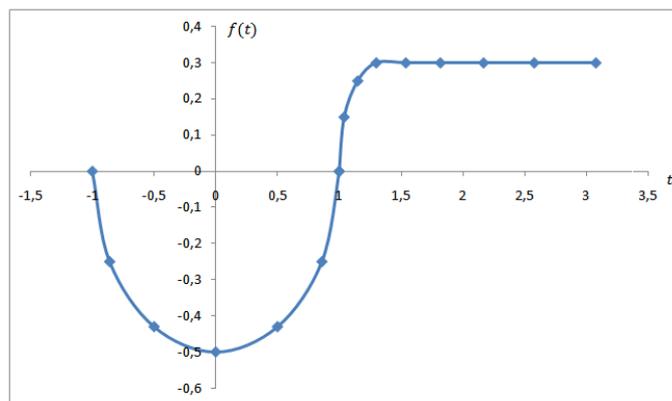


Figura 5.33. Curva de *C-Splines* no plano $\{X, Y\}$.

A equação que caracteriza a curva é:

$$f(t) = a + bt + ct^2 + et^3 \quad (5.10)$$

Nakamura (2005) apresenta a demonstração e o desenvolvimento computacional efetuados para a interpolação de *C-Splines*.

Na figura 5.34, observa-se graficamente a interpolação que se realiza dos dados quando existem sobre a nuvem de pontos regiões desconhecidas.

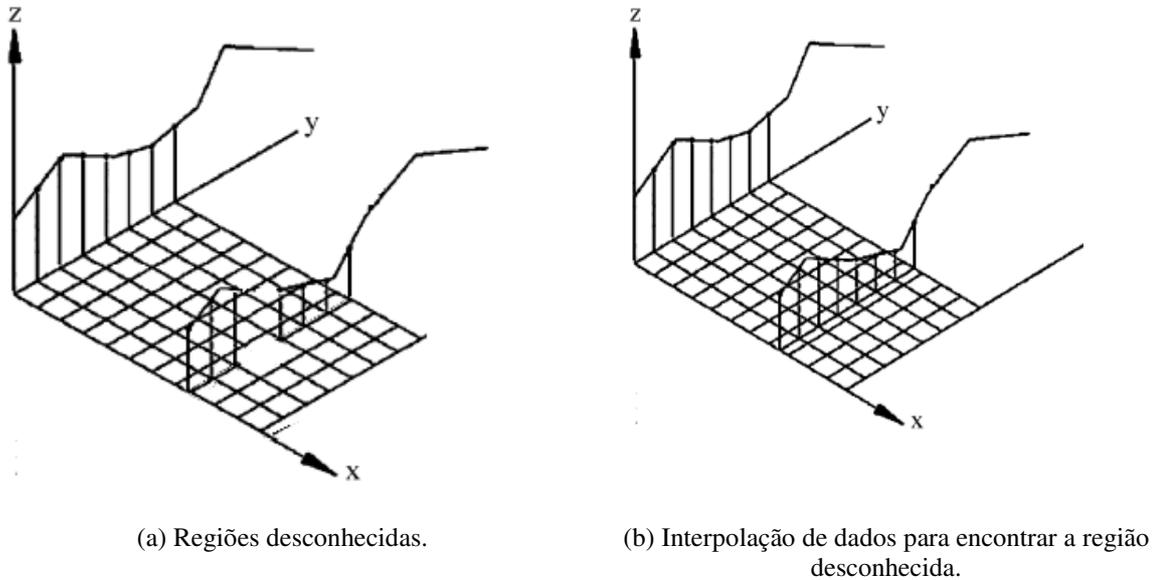


Figura 5.34. Procedimento gráfico de eliminação de regiões desconhecidas.

5.6 - CONSTRUÇÃO DA SUPERFÍCIE MEDIANTE UMA REDE DE ENTALHES PARAMÉTRICOS BICÚBICOS

Construindo a malha estruturada, realiza-se o ajuste dos entalhes paramétricos sobre a estrutura. Cada entalhe ocupa 16 pontos de controle (Apêndice B) como se observa na figura 5.35:

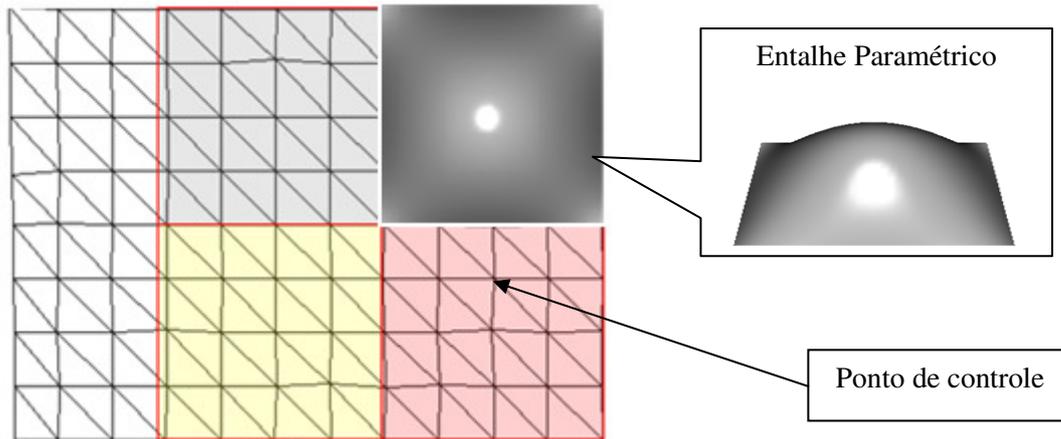


Figura 5.35. Reconstrução de superfícies através dos entalhes paramétricos.

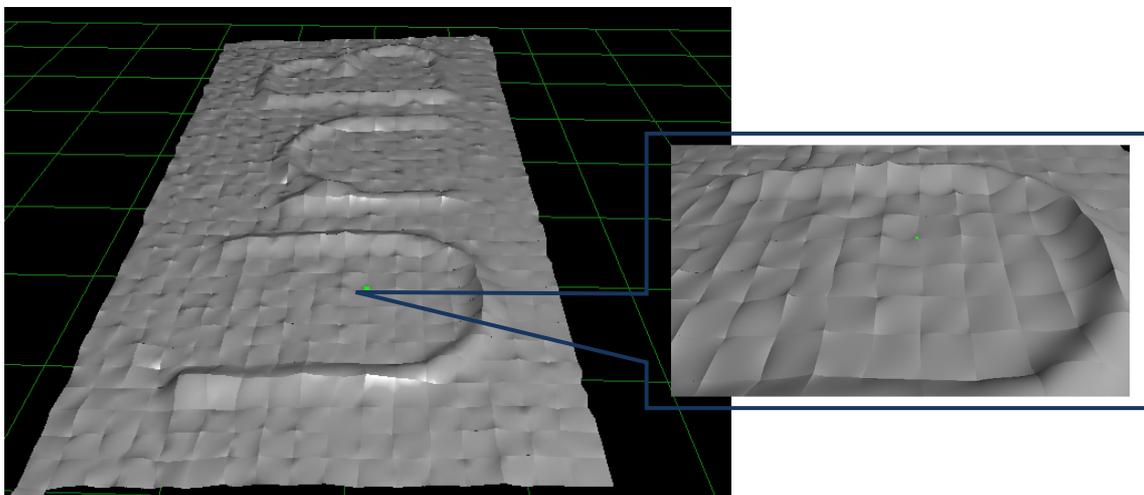


Figura 5.36. Detalhamento visual da superfície paramétrica.

Para se obter uma representação semelhante à superfície real (figura 5.36) é preciso suavizar a superfície virtual através de uma análise de continuidade entre os entalhes paramétricos. A seguir é apresentado o modelo utilizado para suavizar essa superfície.

5.7 - ANÁLISES DE CONTINUIDADE DE ENTALHES PARAMÉTRICOS

A reconstrução de modelos 3D mediante coleções de funções paramétricas introduz o problema da análise de continuidade entre entalhes adjacentes. O trabalho de Hoppe et al. (1994) é o mais completo na área de reconstrução de superfícies, empregando um conjunto de funções parametrizadas. A metodologia de Hoppe é baseada numa divisão volumétrica efetiva que permite gerar vizinhanças quadriláteras sobre modelos tridimensionais irregulares. O processo de ajuste do método de Hoppe usa o esquema de *G-Splines* de Peters (1994) que utiliza funções *Splines* com bases Bezier.

Mediante o esquema de Peters, a continuidade das fronteiras é garantida como somente uma construção quadrilateral geral que pode ser aplicada a todas as vizinhanças dos entalhes paramétricos.

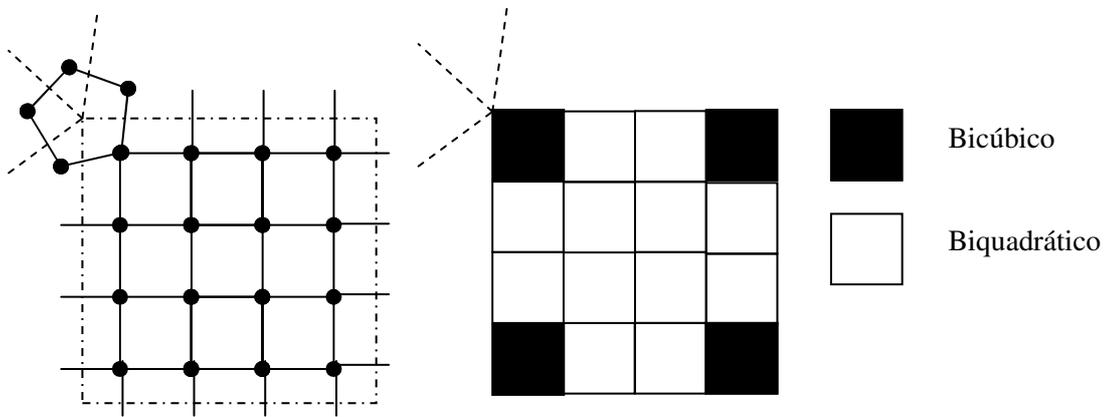


Figura 5.37. *G-Splines* de Peters.

A figura 5.37 mostra o esquema das funções *G – Splines* de Peters, as quais estão compostas por: modelos de Bezier bicúbicos nas esquinas e biquadráticos nas demais vizinhanças.

5.7.1 - Formulação matemática

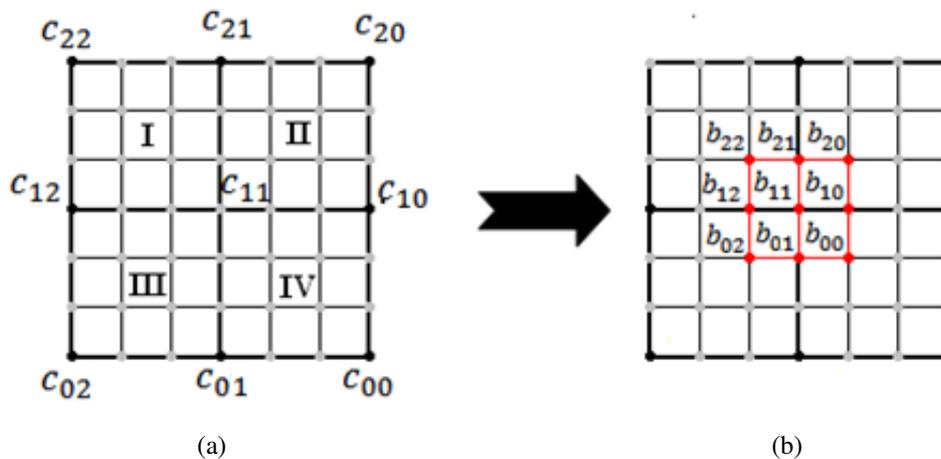


Figura 5.38. Suavização da superfície por meio da análise de continuidade dos entalhes paramétricos (número de lados dos polígonos $m = 4$).

Para suavizar a superfície através da continuidade dos entalhes paramétricos bicúbicos, é preciso modificar alguns pontos de controle pertencentes aos entalhes I, II, III e IV, como mostram as figuras 5.38(a) e 5.38(b). Apresentam-se em seguida os valores que adotam os pontos de controle da malha estruturada.

$$b_{00} = \frac{(c_{00} + c_{10} + c_{01} + c_{11})}{4} \quad (5.11)$$

$$b_{10} = \frac{(c_{11} + c_{10})}{2} \quad (5.12)$$

$$b_{01} = \frac{(c_{11} + c_{01})}{2} \quad (5.13)$$

$$b_{11} = c_{11} \quad (5.14)$$

O processo se repete para os demais parâmetros pela simetria dos entalhes. A figura 5.39 apresenta a reconstrução da superfície através das análises de continuidade dos entalhes paramétricos

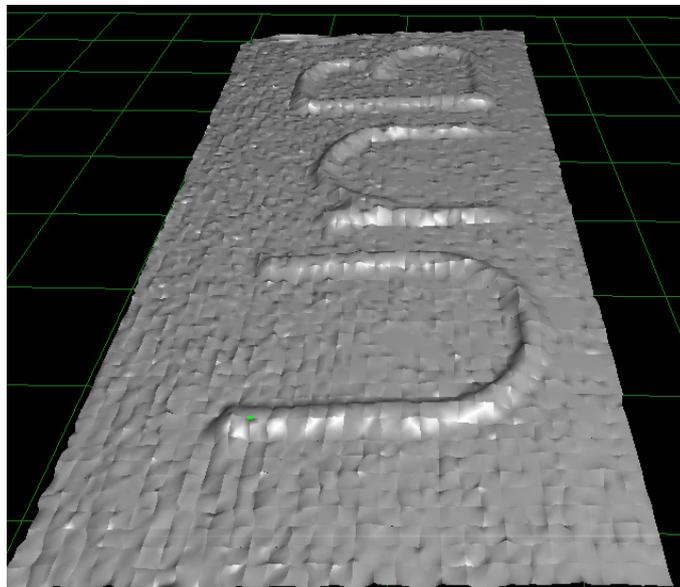


Figura 5.39. Suavização da superfície.

Para os entalhes que não são simétricos pela forma (número de lados do polígono diferentes), o modelo matemático a seguir é o seguinte:

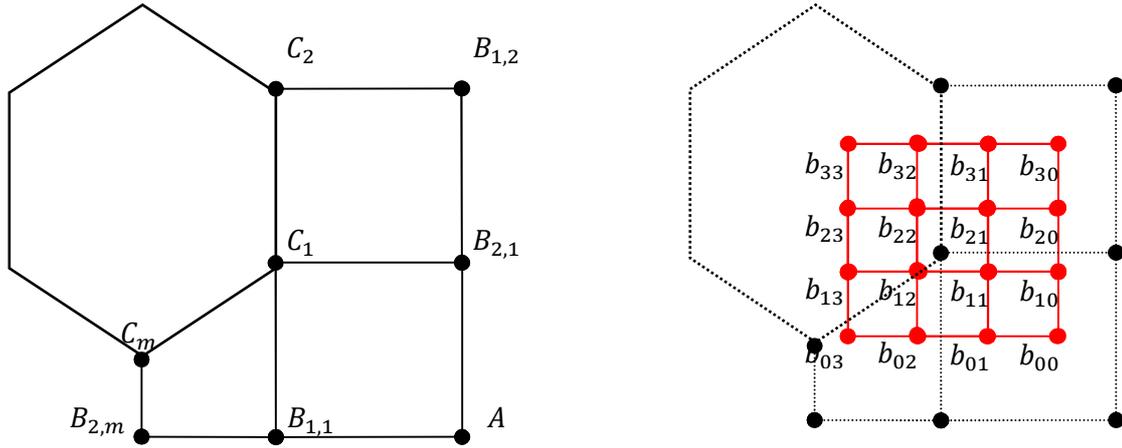


Figura 5.40. Suavização da superfície por meio da análise de continuidades dos entalhes paramétricos (diferentes números de lados dos polígonos).
 $m = \text{número de lados do polígono}$

$$b_{20} = \frac{5B_{2,1} + B_{1,2} + 5C_1 + C_2}{12}$$

$$b_{30} = \frac{B_{2,1} + B_{1,2} + C_1 + C_2}{4}$$

$$b_{00} = \frac{B_{2,1} + B_{1,1} + C_1 + A}{4}$$

$$b_{10} = \frac{5B_{2,1} + B_{1,1} + 5C_1 + A}{12}$$

$$b_{20} = \frac{5B_{2,1} + B_{1,2} + 5C_1 + C_2}{12}$$

$$b_{30} = \frac{B_{2,1} + B_{1,2} + C_1 + C_2}{4}$$

$$b_{11} = \frac{5B_{2,1} + 5B_{1,1} + (25 + 4a)C_1 + (1 - 4a)A}{36}$$

$$b_{21} = ((5 - 10a)B_{2,1} + (1 + 2a)B_{1,2} + (25 + 6a)C_1 + (5 + 2a)C_2)/36$$

$$b_{31} = h_{1,1}$$

$$b_{22} = \left\{ \begin{array}{ll} -\sum_{i=1}^m (-1)^i h_{3,i} & \text{se } m \text{ é ímpar} \\ -\frac{2}{m} \sum_{i=1}^m (-1)^i (m-i) h_{3,i} & \text{se } m \text{ é par} \end{array} \right\}$$

Em que,

$$c = \cos\left(\frac{2\pi}{m}\right)$$

$$a = \frac{c}{1-c}$$

$$h_{1,i} = ((1-2a)B_{2,1} + (1-2a)B_{1,2} + (5+2a)C_1 + (5+2a)C_2)/12$$

$$h_{2,i} = \frac{1}{m} \sum_{i=1}^m C_i + \frac{2a}{3c} \cos\left(\frac{2\pi l}{m}\right) (C_{i+l} + C_{i+l+1})$$

$$h_{3,i} = \left(1 - \frac{2}{3}c\right) h_{2,i} + \frac{2}{3} c h_{1,i}$$

5.8 - ADAPTAÇÃO DOS ENTALHES PARAMÉTRICOS SOBRE MALHA ESTRUTURADA

A construção de um modelo 3D através de superfícies paramétricas é um processo que requer um ajuste de aproximação entre as estruturas geométricas de uma malha estruturada no espaço R^3 (Eck & Hoppe, 1996), (Bertram *et al.*, 2003) seção (4.6) e apêndice (B).

5.8.1 - Seleção do tamanho e valores de um vetor nó

A função `gluNURBSSurface` (da biblioteca de OpenGL) tem como argumentos dois vetores nó na direção u e na direção v (figura 5.41). Esses vetores têm por finalidade controlar a forma da superfície paramétrica.

Durante a realização do presente trabalho não foi encontrada nenhuma referência bibliográfica que explicasse o funcionamento da função `gluNURBSSurface`, mais especificamente a forma de se modificar o tamanho e os valores dos vetores nó contidos na função de OpenGL (ver apêndice B).

Desse modo, optou-se por desenvolver um modelo para facilitar a utilização da mesma. Para tanto, foram geradas duas curvas utilizando as expressões (5.15 e 5.16) desenhadas na figura 5.41. Através do ajuste da curva em referencia aos pontos de controle (malha estruturada) foi possível extrair o tamanho e os valores do vetor nó. Estas informações serão posteriormente utilizadas como argumentos da função `gluNURBSSurface`.

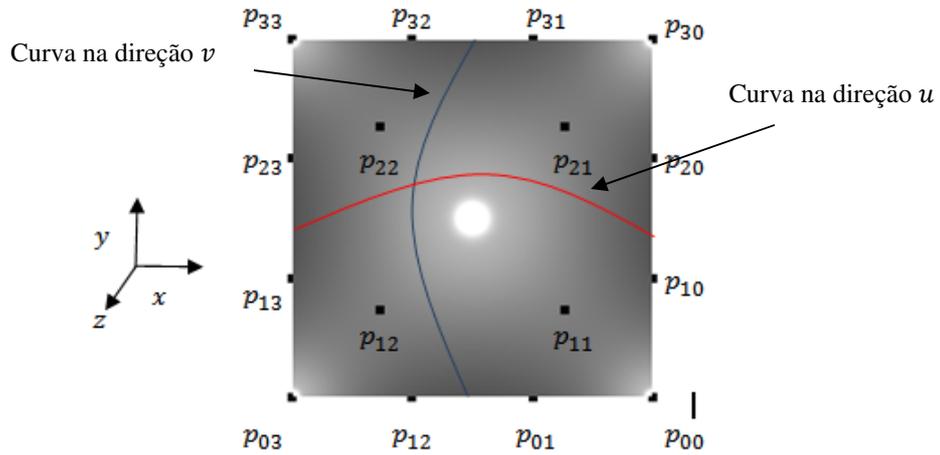


Figura 5.41. Ajuste dos vetores paramétricos através de curvas geradas pelos pontos médios.

- Pontos da Curva na direção u :

$$P_{Au} = \left(\frac{(P_{23,x} + P_{13,x})}{2}, \frac{(P_{23,z} + P_{13,z})}{2} \right)$$

$$P_{Bu} = \left(\frac{(P_{22,x} + P_{12,x})}{2}, \frac{(P_{22,z} + P_{12,z})}{2} \right)$$

$$P_{Cu} = \left(\frac{(P_{21,x} + P_{11,x})}{2}, \frac{(P_{21,z} + P_{11,z})}{2} \right)$$

$$P_{Du} = \left(\frac{(P_{20,x} + P_{10,x})}{2}, \frac{(P_{20,z} + P_{10,z})}{2} \right)$$

(5.15)

- Pontos da curva na direção v :

$$P_{Av} = \left(\frac{(P_{32,x} + P_{31,x})}{2}, \frac{(P_{32,z} + P_{31,z})}{2} \right)$$

$$P_{Bv} = \left(\frac{(P_{22,x} + P_{21,x})}{2}, \frac{(P_{22,z} + P_{21,z})}{2} \right)$$

$$P_{Cv} = \left(\frac{(P_{12,x} + P_{11,x})}{2}, \frac{(P_{12,z} + P_{11,z})}{2} \right)$$

$$P_{Dv} = \left(\frac{(P_{02,x} + P_{01,x})}{2}, \frac{(P_{02,z} + P_{01,z})}{2} \right)$$

(5.16)

- Curva gerada na direção u :

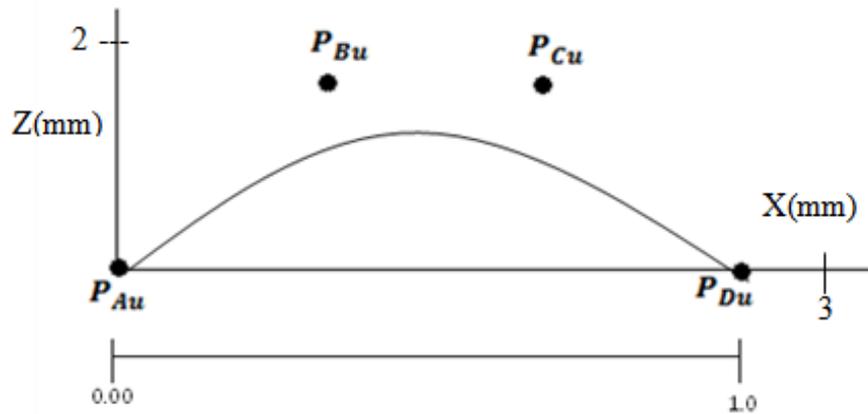


Figura 5.42. Curva de ajuste na direção u - Vetor nó = {0,0,0,0,1,1,1,1}.

- Curva gerada na direção v :

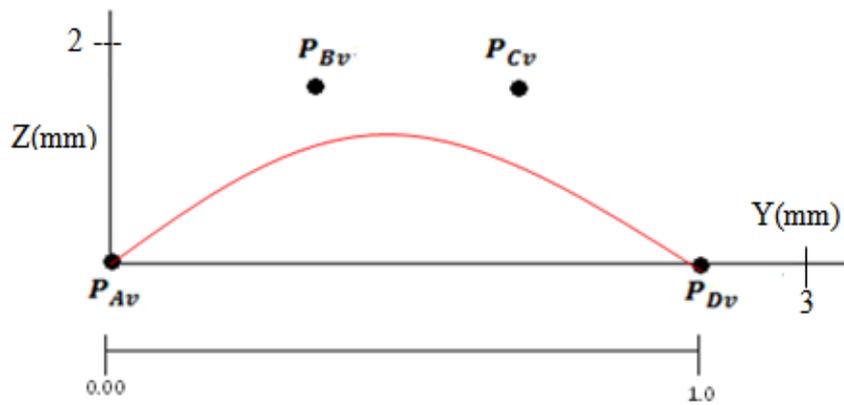


Figura 5.43. Curva de ajuste na direção v - Vetor nó = {0, 0, 0, 0, 1, 1, 1, 1}.

De acordo com a seção 4.7, para minimizar a distância entre a curva e os pontos de controle deve-se empreender uma mudança dos parâmetros do vetor nó através da Iteração de Newton.

Na direção u tem-se a seguinte curva de ajuste:

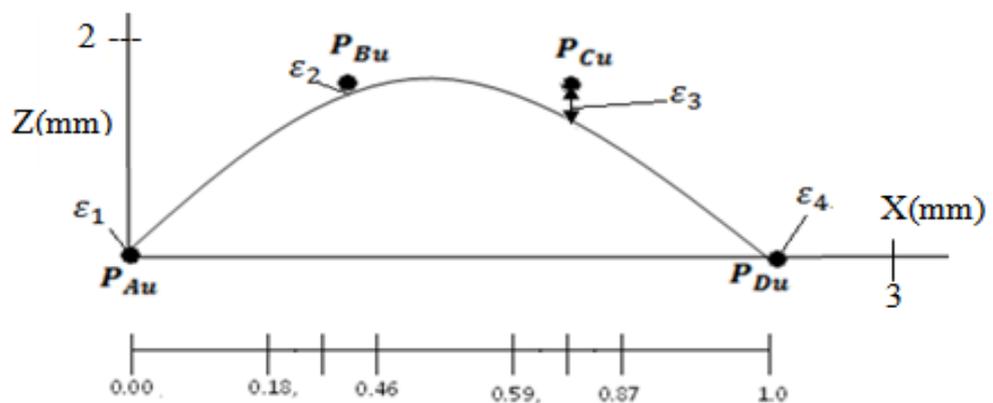


Figura 5.44. Curva de ajuste na direção u - Vetor nó = {0.00, 0.18, 0.32, 0.46, 0.59, 0.73, 0.87, 1.0}

Na direção v , a curva de ajuste é:

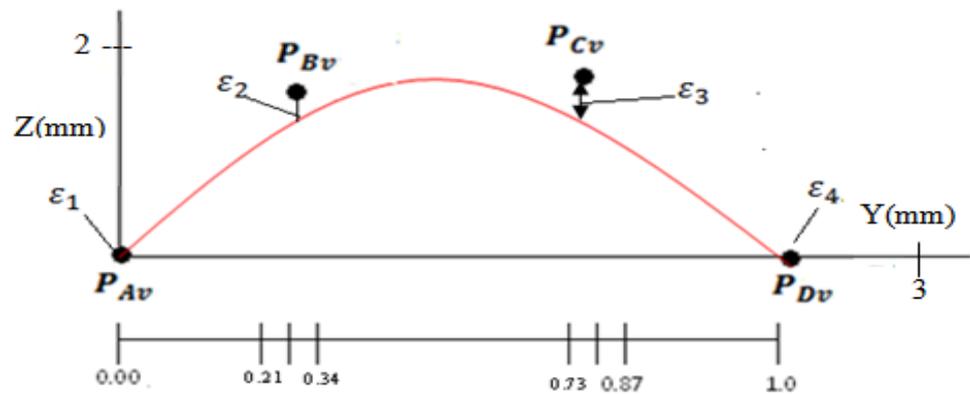


Figura 5.45. Curva de ajuste na direção v – Vetor nó = { 0.00, 0.21, 0.27, 0.34, 0.73, 0.80, 0.87, 1.0}.
 $\varepsilon_1, \varepsilon_2, \varepsilon_3$ e ε_4 são os valores do erro de ajuste.

Para cada ajuste, o algoritmo implementado utiliza um número máximo de 20 iterações. Através da equação 5.17 foi possível determinar o erro médio de ajuste ($\bar{\varepsilon}_{ajuste}$) obtido na direção u e na direção v do entalhe paramétrico:

$$\bar{\varepsilon}_{ajuste} = \frac{\frac{(\varepsilon_{1,u} + \varepsilon_{2,u} + \varepsilon_{3,u} + \varepsilon_{4,u})}{4} + \frac{(\varepsilon_{1,v} + \varepsilon_{2,v} + \varepsilon_{3,v} + \varepsilon_{4,v})}{4}}{2} \quad (5.17)$$

A figura 5.46 mostra o ajuste entre a malha estruturada e o conjunto de entalhes paramétricos. Na tabela 5.2 encontram-se consignados os valores estatísticos encontrados durante as análises para diferentes valores t .

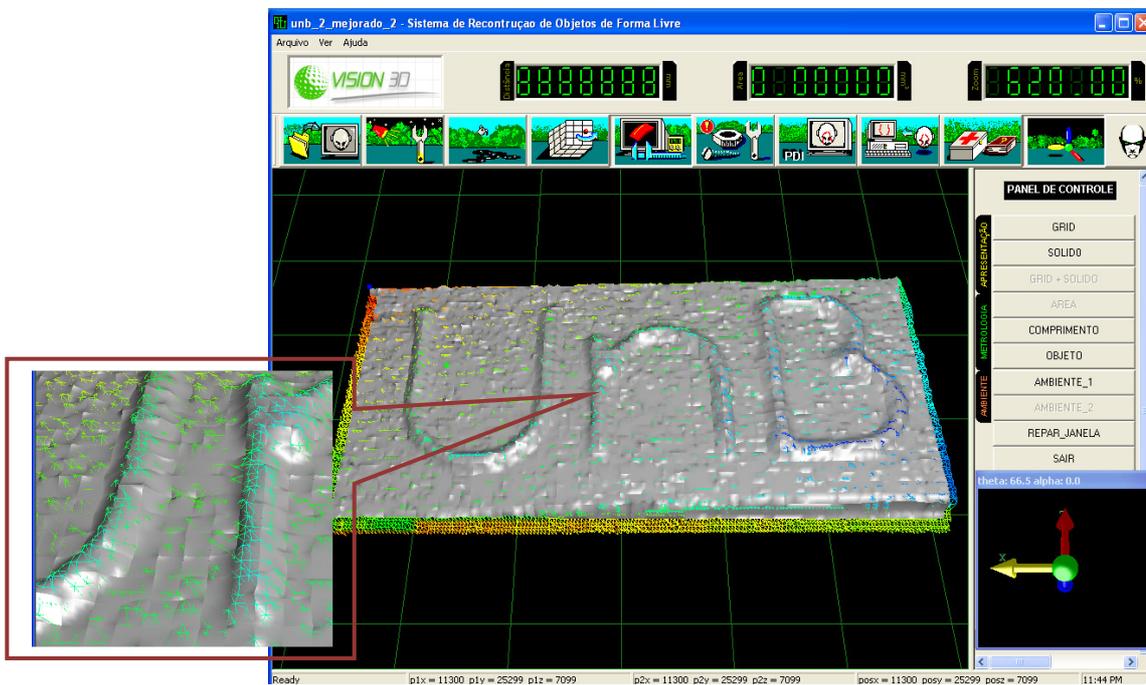
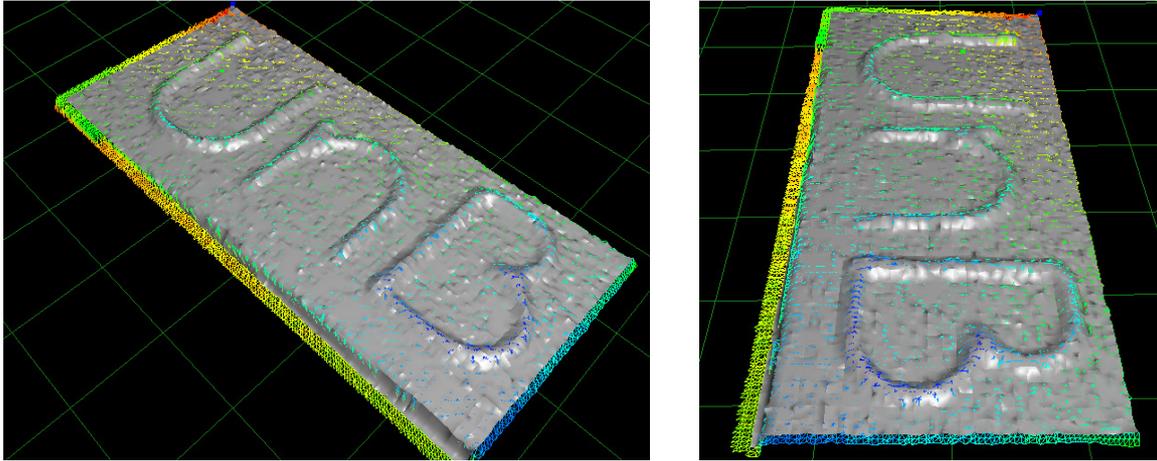


Figura 5.46. Acoplamento da superfície. Malha estruturada e entalhes paramétricos.

Tabela 5.2. Resumo dos valores obtidos do erro de ajuste de toda a superfície através do erro acumulado $\bar{\epsilon}_{ajuste}$ (equação 5.17), para diferentes valores de t .

Tamanho da Grade Triangular (t)	Número de entalhes Paramétricos	Erro de ajuste ($\bar{\epsilon}_{ajuste}$)	
		Média aritmética $\bar{\mu}(mm)$	Desvio padrão $\sigma(mm)$
2	9800	0,458	0,049
4	4900	1,034	0,077
7	1633	1,661	0,082
10	1088	2,581	0,123

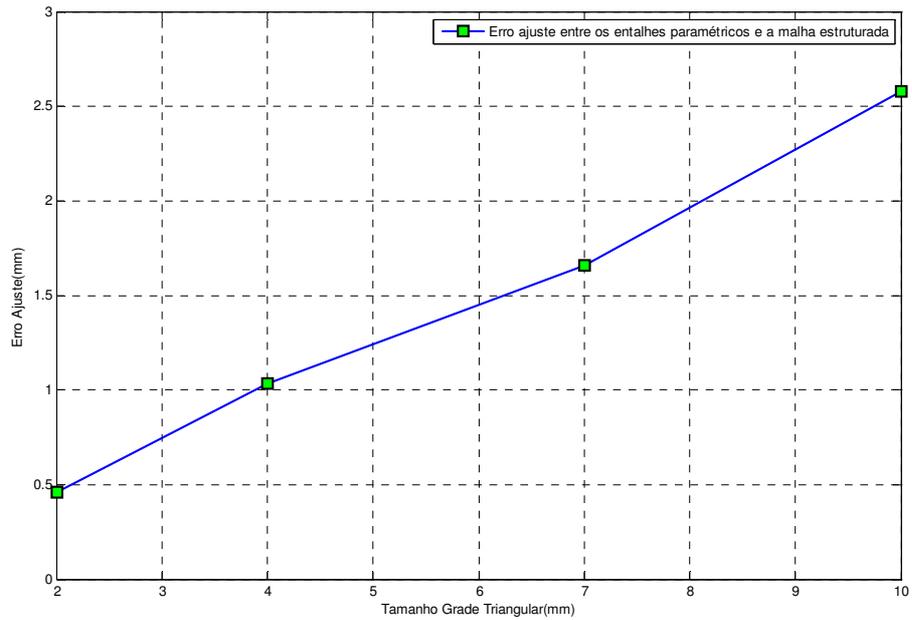


Figura 5.47. Tendência do valor do erro de ajuste com a variação de t .

5.9 - QUALIDADE DA APROXIMAÇÃO

Observa-se no gráfico (figura 5.47) que o erro médio de ajuste entre os entalhes paramétricos e a malha estruturada aumenta proporcionalmente ao tamanho da grade triangular (t).

Para a obtenção de medidas da superfície paramétrica é recomendável realizar o tratamento da nuvem de pontos com uma malha triangular pequena.

6 - MÉTODO DE VERIFICAÇÃO DA EXATIDÃO DO SISTEMA LASER SCANNER UTILIZANDO ALVOS ESFÉRICOS

6.1 - INTRODUÇÃO

É de extrema importância o conhecimento prévio da resolução de um equipamento de varredura *laser* em determinadas distâncias (Wutke, 2006). Assim, dependendo do detalhamento desejado da superfície a ser construída, é possível definir a distância de operação e utilização do equipamento.

Este capítulo descreve o experimento que permitiu determinar a variação da resolução do equipamento *VISSCAN 3D laser* em função da distância de operação.

Para fornecer confiança nos dados do experimento, foram empregados elementos padronizados como uma mesa óptica de isolamento de vibração e uma esfera de rolamento de um gerador elétrico de ferro fundido de diâmetro $36,50 \pm 0,005 \text{ mm}$.

6.2 - DESCRIÇÃO DO EXPERIMENTO

O primeiro passo para o desenvolvimento do experimento foi situar a esfera de cor cinza a uma distância aproximada de 250 mm a partir da lente da câmera P_0 (figura 6.1). A partir da posição D_1 (figura 6.1) digitalizou-se a esfera (captura da nuvem de pontos) para 10 posições diferentes. A distância de separação entre cada registro digital foi de $(25,0 \pm 0,5 \text{ mm})^{19}$ (figura 6.1).

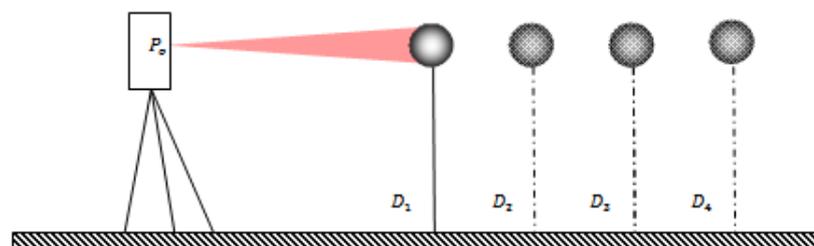


Figura 6.1. Montagem esquemática da captura da nuvem de pontos da esfera em diferentes posições.

¹⁹ A resolução é dada pela mesa onde as esferas foram posicionadas.

O deslocamento da esfera ocorreu sobre uma mesa de isolamento de vibração empregada na montagem de experimentos ópticos da série (63-500) da empresa *Technical Manufacturing Corporation*. A figura 6.2 apresenta a montagem experimental.

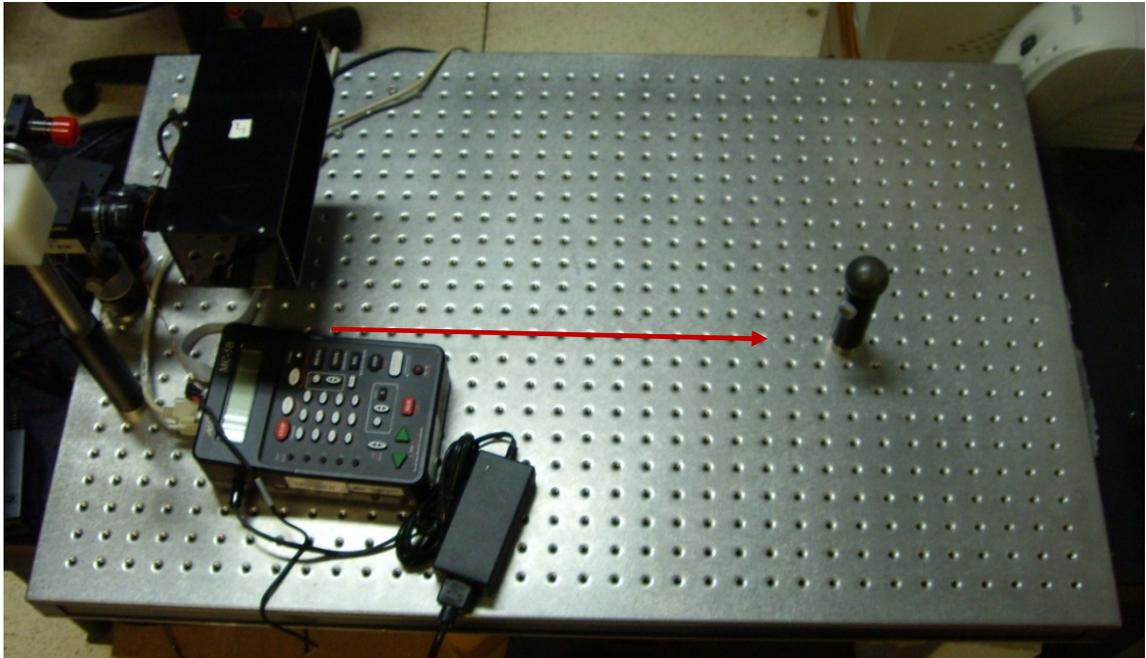


Figura 6.2. Montagem experimental:
Deslocamento da esfera através dos furos da mesa ótica.
(A distância entre furo e furo é de $25,0 \pm 0,5$ mm).

Depois de digitalizar a esfera para diferentes distâncias (figura 6.3), calculou-se o raio da esfera para cada uma das nuvens de pontos capturadas, conforme descrito na seção 6.3.

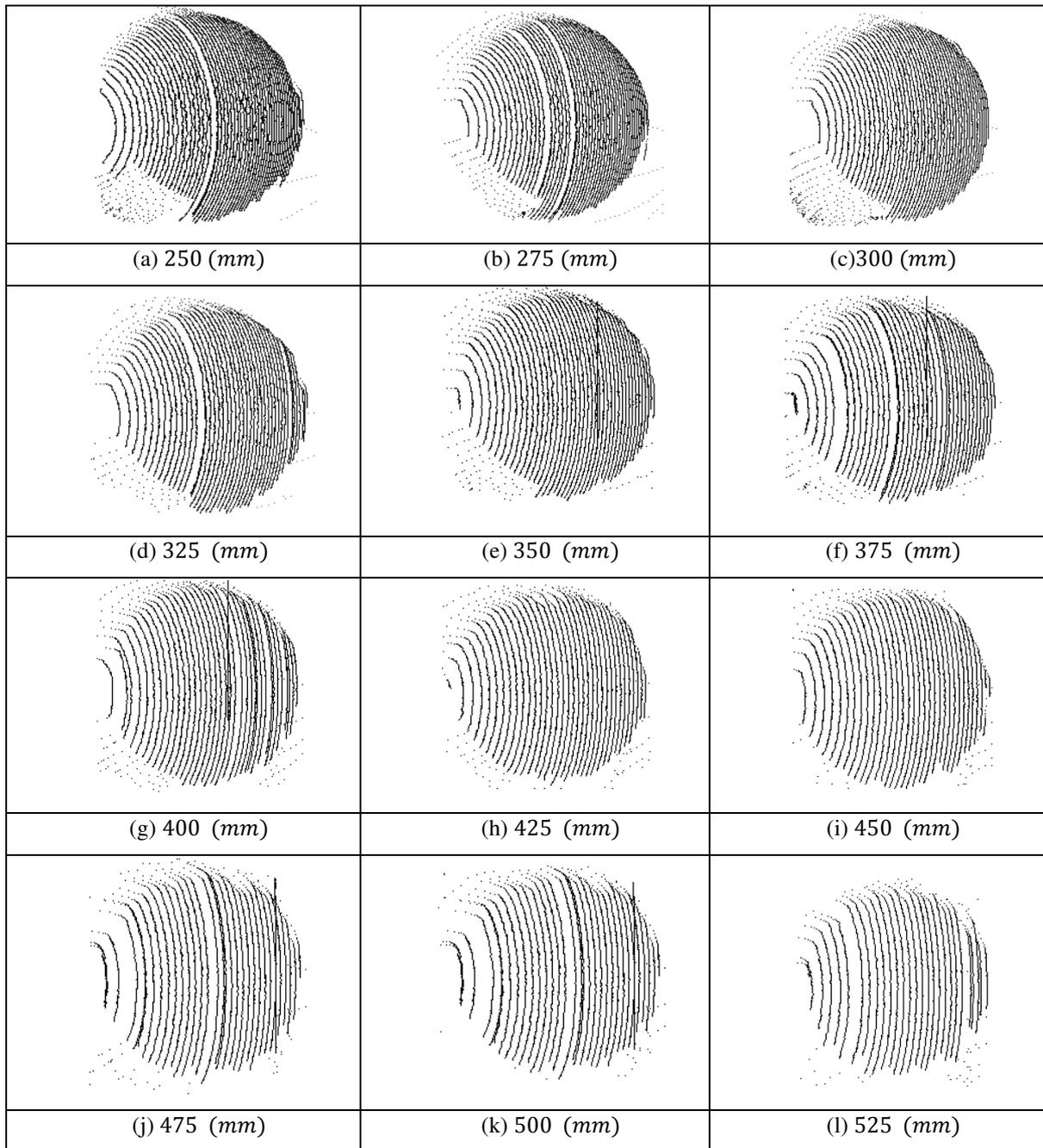


Figura 6.3. Registro das nuvens de pontos em R^2 .
O número em baixo representa a distância em que a nuvem de pontos foi capturada

6.3 - CÁLCULO DO RAIOS DA ESFERA

Para este cálculo matemático utilizaram-se ferramentas computacionais desenvolvidas na geração de malhas triangulares 3D.

A figura 6.4 exibe a construção da superfície da esfera a partir da nuvem de pontos da figura 6.3(a):

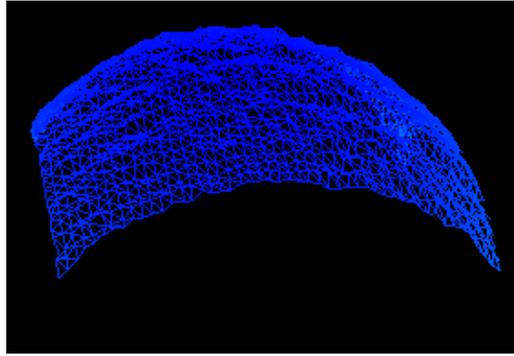


Figura 6.4. Construção da malha estruturada da esfera.
Parâmetros de construção: $\mu_{NLS} = 0.6$ e $t = 2 \text{ mm}$

Para cada triângulo que conforma a superfície estruturada foi calculado o vetor normal (figura 6.5):

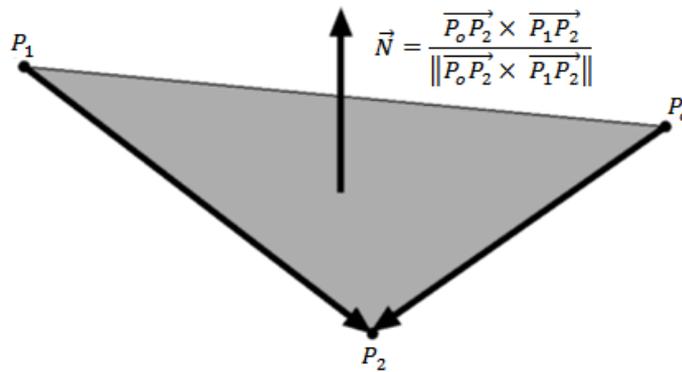


Figura 6.5. Cálculo do vetor normal de um triângulo.
 P_3, P_1, P_2 são os pontos nodais da malha estruturada e \vec{N} é o vetor normal calculado.

Após o cálculo de todos os vetores normais da malha procedeu-se a projeção de uma reta desde o incentro de cada triângulo até o centro da esfera (figura 6.6):

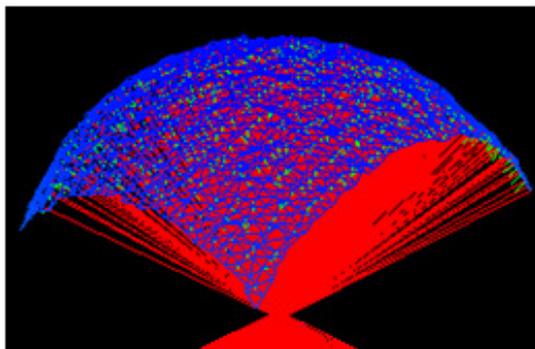


Figura 6.6. Projeção de retas desde o incentro de cada triângulo ao centro da esfera.

Para calcular o ponto de interseção das retas projetadas utilizou-se a triangulação de pontos, conforme esta descrita na (seção 6.4). As duas retas selecionadas projetam-se dos incentros das áreas contíguas e opostas a um ponto nodal (figura 6.7).

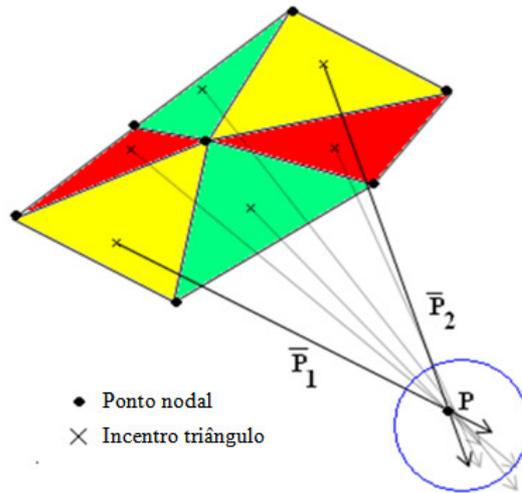


Figura 6.7. Projeção de duas retas contíguas e opostas a um ponto nodal

Após o cálculo do ponto médio de interseção entre as duas retas projetou-se uma linha ao ponto nodal com magnitude r_i (equação 6.2), como mostra a figura 6.8:

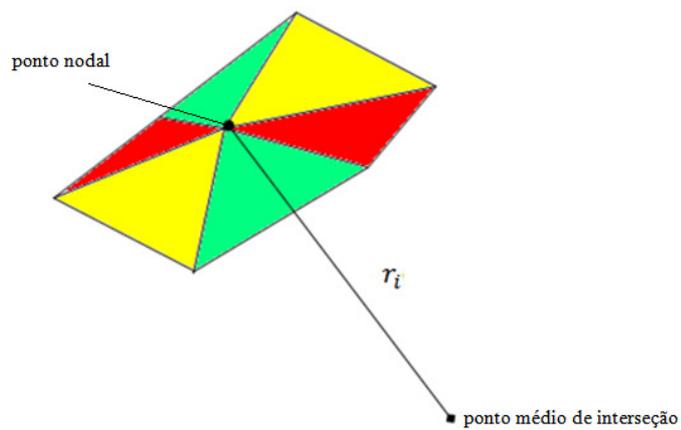


Figura 6.8. Distância do ponto nodal até o ponto médio de interseção (r_i).

Esse mesmo processo repetiu-se em todos os pontos nodais e triângulos da esfera (figura 6.9):

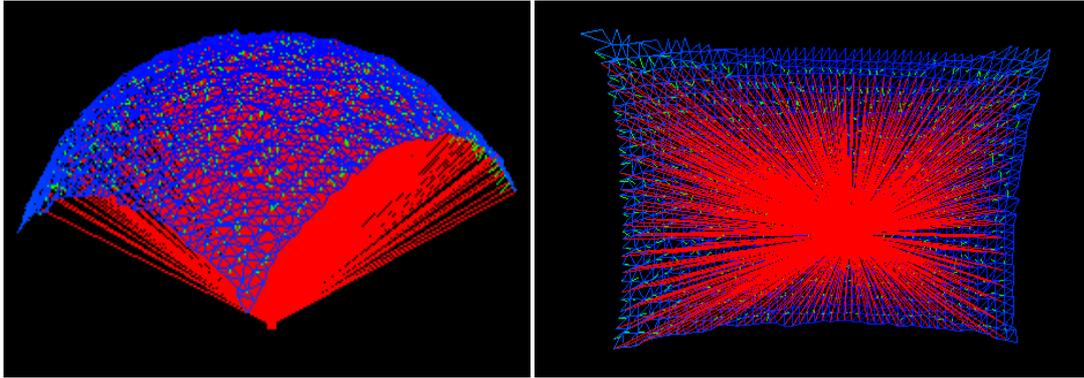


Figura 6.9. Projeção de retas r_i desde todos os pontos nodais.

6.4 - TRIANGULAÇÃO DE PONTOS

A triangulação de pontos é uma técnica matemática comumente utilizada em geometria para encontrar o ponto de interseção $p(x, y, z)$ de dois vetores unitários \hat{P}_2 e \hat{P}_1 . Esses vetores surgem dos pontos $p_1(x, y, z)$ e $p_2(x, y, z)$, como apresenta a figura (6.10).

Frequentemente, devido à precisão finita dos cálculos e ao ruído das medidas, não é possível a definição desse ponto, sendo necessário situá-lo no ponto médio ($p_{medio}(x, y, z)$) de um segmento perpendicular a ambas. Para cada par de pontos em correspondência, $p_1(x, y, z)$ e $p_2(x, y, z)$, emprega-se o seguinte modelo matemático (Forsyth & Ponce, 2003):

$$a\hat{P}_1 + c(\hat{P}_1 \times R^T \hat{P}_2) = \bar{T} + bR^T \hat{P}_2 \quad (6.1)$$

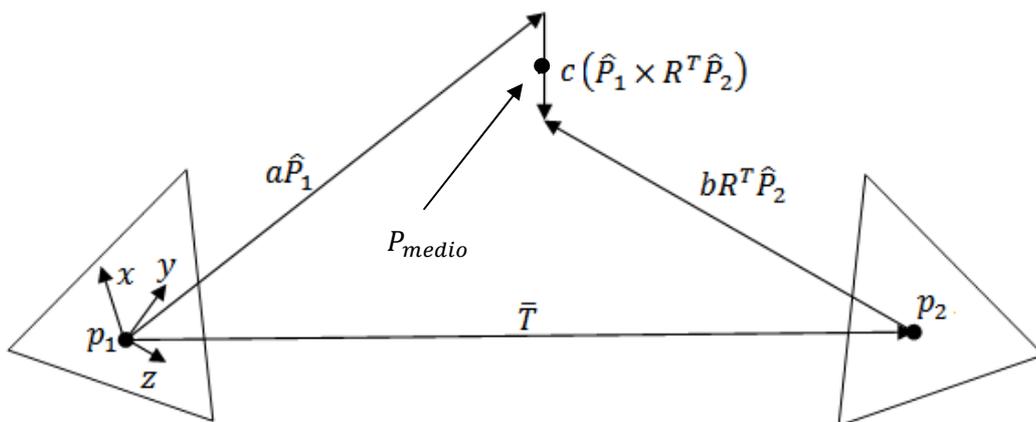


Figura 6.10. Triangulação de pontos.

Em que:

a, b, c	Valores escalares
$\bar{T}(i, j, k)$	$p_2(x, y, z) - p_1(x, y, z)$
R	Matriz de rotação

6.4.1 - Distância entre um ponto e uma reta

Conhecendo os valores de b e c obtidos com a equação (6.1) e o vetor gerado T' entre os pontos $p_k(x, y, z)$ e $p_2(x, y, z)$ (figura 6.11), pode-se determinar a distância (r_i) entre o ponto $p_k(x, y, z)$ e a reta $\frac{c}{2}(\hat{P}_1 \times R^T \hat{P}_2)$ com a seguinte equação (Forsyth & Ponce, 2003):

$$r_i \hat{D} + \frac{c}{2}(\hat{P}_1 \times R^T \hat{P}_2) = \bar{T}' + bR^T \hat{P}_2 \quad (6.2)$$

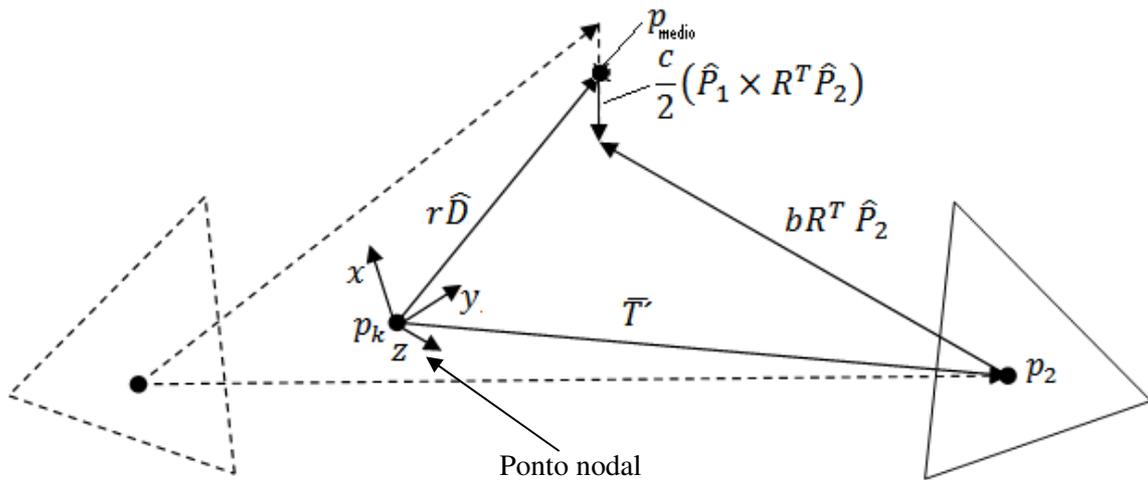


Figura 6.11. Triangulação de pontos (Ponto médio).

Em que,

a, b, c	Valores escalares
$\bar{T}'(i, j, k)$	$p_2(x, y, z) - p_k(x, y, z)$
R	Matriz de rotação

6.5 - OBTENÇÃO DE DADOS

Depois de construir a malha triangular para todas as nuvens de pontos, mantendo as mesmas condições de construção ($\mu_{NLMS} = 0.6$ e $t = 2 \text{ mm}$), registraram-se os seguintes dados estatísticos (Tabela 6.1):

Tabela 6.1. Dados estatísticos obtidos de r_i para cada nuvem de pontos registrada.

Nuvem de Pontos	Distância de Trabalho (mm)	Número de Amostras N	Raio r_i (mm)		E_M (mm)
			Média Aritmética $\bar{\mu}$	Desvio Padrão σ	
a	250	472	17,93	0,287	0,32
b	275	452	17,93	0,346	0,32
c	300	424	17,84	0,333	0,41
d	325	436	17,84	0,411	0,41
e	350	450	17,73	0,412	0,52
f	375	472	17,69	0,512	0,56
g	400	448	17,67	0,528	0,58
h	425	426	17,60	0,628	0,65
i	450	388	17,54	0,684	0,71
j	475	324	17,48	0,762	0,77
k	500	326	17,39	0,821	0,86

Ploteando a informação da tabela (6.1), Distância do trabalho versus raio médio da esfera se obtém:

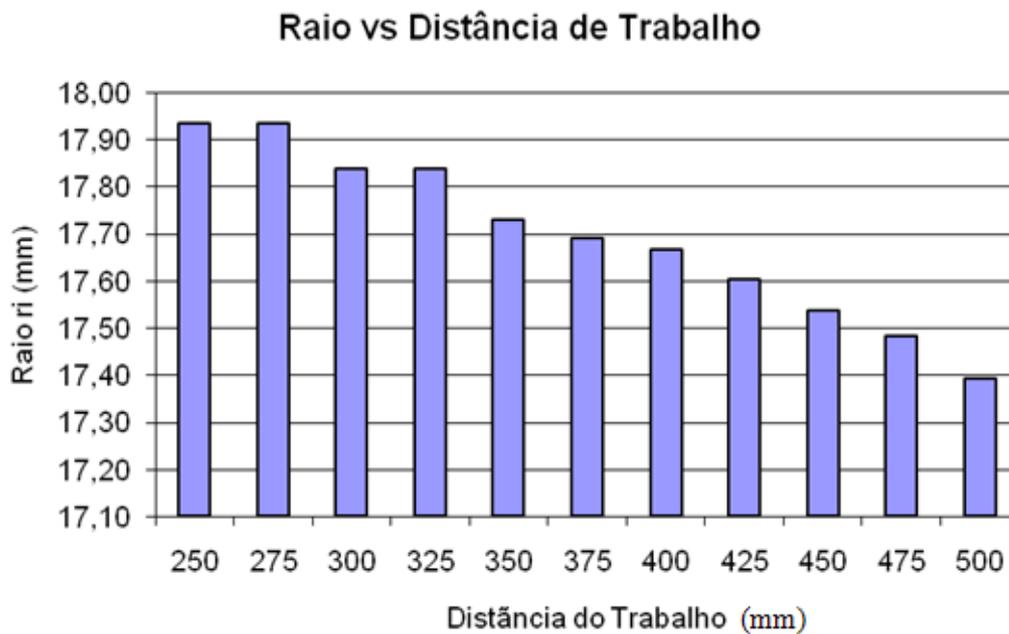


Figura 6.12. Variação do raio médio r_i versus distância de trabalho.

Para calcular a exatidão do sistema E_M se realiza a diferença entre o valor do raio obtida pelo paquímetro ($\frac{36,50 \text{ mm}}{2}$) e o valor do raio médio (r_i) obtido experimentalmente (equação 6.3) (os valores se encontram depositados na tabela 6.1).

$$E_M = \frac{36,50 \text{ mm}}{2} - r_i \quad (6.3)$$

Plotando os dados da tabela 6.1 (Distancia do trabalho versus erro (E_M)) obtém-se,

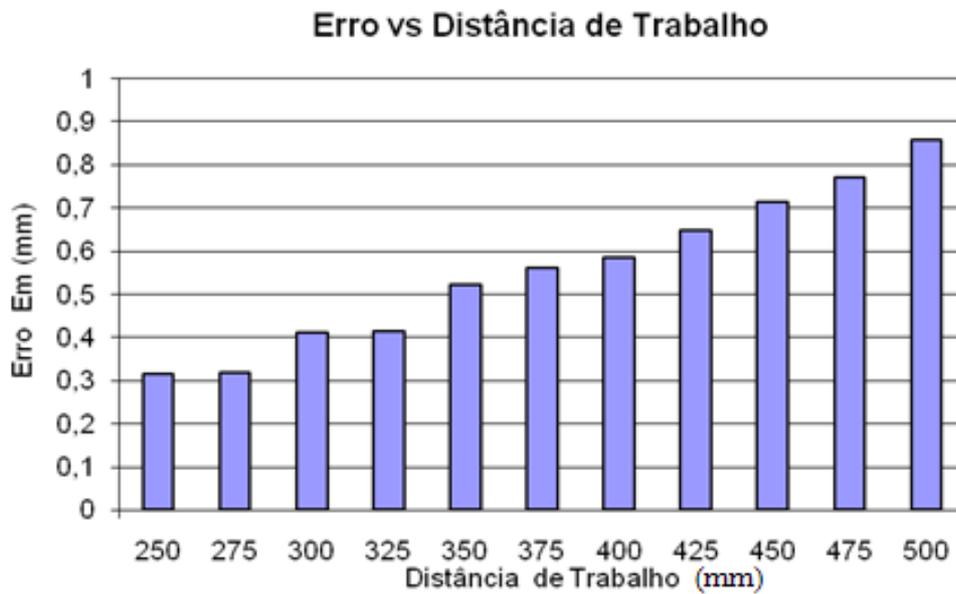


Figura 6.13. Exatidão da versão de campo do SistemaViscan-3D.

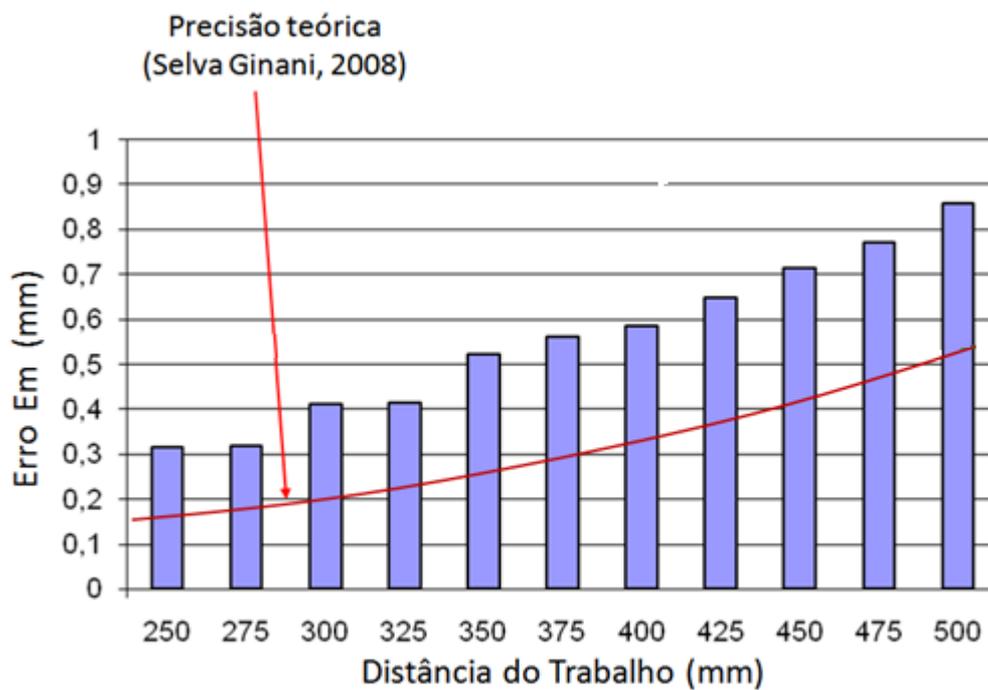


Figura 6.14 Comparação de medidas dos sistemas de medição

6.6 - ANÁLISE E DISCUSSÃO DOS RESULTADOS

Para realizar uma análise estimativa da precisão do experimento é necessário comparar os resultados obtidos experimentalmente (figura 6.14) com os dados teóricos calculados e descritos no trabalho de Ginani, 2008. Na figura (6.14) se destaca graficamente a informação experimental (barras) e a informação teórica (Curva vermelha). O comportamento dos valores é considerado próximo devido à ordem de grandeza que as separa, com um valor máximo de 0,5mm.

Em relação a os coeficientes de desvio padrão alcançados em cada experimento feito para cada nuvem de pontos (Tabela 6.1), depreende-se que a dispersão das medidas aumenta em proporção à distância em que se digitaliza a esfera.

De acordo ao resultado da equação (6.4) e figura 6.14, pode-se afirmar que para uma distância de 250 mm o escâner VISSCAN 3D terá uma resolução menor a que 1.0 mm para cada medida que se realiza através do *software* (Apêndice C).

7 - CONCLUSÕES

Para cumprir os objetivos propostos, realizou-se uma revisão bibliográfica sobre as técnicas e metodologias sobre reconstrução de superfícies através de nuvem de pontos. A partir desses trabalhos, foi selecionando o método baseado em modelos deformáveis, porque neste as superfícies modelo não são distantes, em forma, das superfícies a serem construídas.

Todos os métodos de reconstrução apresentam dificuldades para uma representação suave da superfície quando a nuvem de pontos está contaminada com ruídos. Devido a esse problema, estudou-se uma abordagem para atenuação dos ruídos que prejudicam a nuvem de pontos gerada pelo sistema de visão VISSCAN 3D. O ruído gerado pelo ambiente e pela câmera é aleatório e descreve um padrão gaussiano. Uma solução para atenuá-lo foi a filtragem adaptativa. Essa filtragem permite, através da modificação dos critérios de desempenho, a melhoria da qualidade da amostra da nuvem de pontos original para uma aplicação específica.

Realizar a construção de uma malha estruturada 3D é, em essência, um processo de filtração. Esse processo determina a quantidade de pontos a ser representada e depende exclusivamente do número de linhas em que está dividido o polígono modelo, o que ocasiona uma diminuição da carga computacional quando a superfície é representada.

Visualizar a superfície através da rampa de cor permite realizar uma distinção dos efeitos de profundidades do objeto. A partir da ferramenta desenvolvida podem-se distinguir de forma rápida e fácil as zonas de interesse para qualquer nível de resolução em que é construída a superfície. De outra forma, seria impossível realizar um monitoramento da peça em estudo, devido ao fato da nuvem de pontos em que se trabalha exibir informação da geometria, mas não a informação de sua textura.

Além de deformar uma malha triangular influenciada pela nuvem pontos, foi gerada, em cima da superfície estruturada, uma rede de entalhes paramétricos bicúbicos para dar realismo à superfície representada.

As funções NURBS apresentam uma formulação matemática unificada que permite a fácil implementação e digitalização. Além disso, o alto grau de adaptação e capacidade de aproximação local faz das superfícies NURBS uma ferramenta poderosa para gerar modelos computacionais muito próximos aos pontos de controle.

O esquema utilizado para suavizar a superfície, através da continuidade dos entalhes paramétricos bicúbicos, permite uma representação agradável aos olhos humanos, sempre que se realize um tratamento adequado de ruído sobre a nuvem de pontos a ser trabalhada.

A interpretação dos resultados obtidos na avaliação da exatidão do sistema laser permite concluir que o método analítico é válido e confiável, pois através desse método foi possível definir a resolução de operação do equipamento de varredura *laser* em função da distância. Esse dado facilita a elaboração de um levantamento tridimensional específico. O conhecimento da resolução em que o equipamento opera permite o planejamento do nível de detalhamento a ser obtido em função da distância de instalação do equipamento para a obtenção dos dados.

Considerando os objetivos do presente trabalho, pode-se afirmar que os resultados obtidos foram satisfatórios em cada etapa. O desenvolvimento de um conjunto de ferramentas computacionais independentes foi alcançado. Que em conjunto, permitiram manipular e modificar as amostras capturadas, aptas para a sua análise. Essas mesmas ferramentas permitem criar modelos de dificuldade baixa e média, mantendo o compromisso entre o nível de detalhe da representação e a carga mínima computacional requerida pelo sistema para sua representação. Atualmente, a ferramenta computacional permite criar modelos bi e tridimensionais de ordem de dificuldade, desde um cubo até um crânio humano.

A metodologia implementada no *software* pode ser adequada para qualquer sistema digitalizador *laser*, não somente para o sistema VISSCAN 3D.

8 - TRABALHOS FUTUROS

Muitas das técnicas empregadas nesta dissertação podem ser aprofundadas em futuras pesquisas a fim de melhorar o desempenho e o potencial de aplicação. A seguir são apresentadas sugestões para futuros trabalhos:

1. Desenvolvimento de metodologia que permita de forma automática selecionar o critério desempenho para uma dada aplicação.
2. Em termos de processamento de dados na etapa de filtragem e na etapa de interpolação, poder-se-ia desenvolver uma versão multiprocessada em hardware ou em software que trabalhasse as três coordenadas $[x, y, z]$ ao mesmo tempo, melhorando o desempenho da aplicação.
3. Realizar novos experimentos que permitissem avaliar exatidão e precisão do sistema de medição do *software* desenvolvido, testando ferramentas de medição como: distância, áreas e volumes.

9 - REFERÊNCIAS BIBLIOGRÁFICAS

- Amenta, N., Bern, N. & Eppstein, D. (1998a). “The crust and the beta-skeleton: combinatorial curve reconstruction.” *Graphical Models and Image Processing*. Academic Press, Berkeley, Vol.1, pp125-135.
- Amenta, N., Bern, N. & Kamvyselis, M. (1998b) “A new Voronoi-based surface reconstruction algorithm.” Proceedings of *SIGGRAPH '98*. Vol. 2. pp 415–421.
- Amenta, N., Choi, S., Dey, T.K. & Leekha, N. (2000) “A simple algorithm for homeomorphic surface reconstruction.” *International Journal of Computational Geometry and Applications*. WSPC Journals, EUA, Vol 2. pp 213-222.
- Anderson, C.W. & Crawford-Hines, S. (2000). “Fast generation of NURBS surfaces from polygonal mesh models of human anatomy”. *Computer Science Technical Report*. URL <<http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.43.1290>>
- Bajaj, C., Bernardini, F. & Xu, G. (1995) “Automatic reconstruction of surfaces and scalar fields from 3D scans.” In: *Computer Graphics Proceedings*. ACM SIGGRAPH, Nova Iorque, EUA, Vol. 1, pp. 109-118.
- Bardinet, E., Cohen, L. D. & Ayache, N. (1995). “A Parametric Deformable Model To Fit Unstructured 3d Data.” *Computer Vision and Image Understanding*. Vol.1 pp.39-54.
- Basseville, M., & Benveniste, A. (1983) “*Design and comparative study of some sequential jump detection algorithms for digital signals*”. In: *IEEE Transactions on Acoustics, Speech and Signal Processing*, Vol 31, issue 3, pp. 521-535.
- Bebiano, D. F. V. (2008). *Monitoração e Localização de Defeitos na Soldagem TIGU utilizando Técnicas de Espectrometria [Distrito Federal] 2008*. Dissertação de Mestrado em Sistemas Mecatrônicos, Publicação ENM.DM-024A/08, Departamento de Engenharia Mecânica, Universidade de Brasília, Brasília, DF, 141p.
- Bernardini, F. & Bajaj, C. L. (1997) “Sampling and Reconstructing Manifolds using Alpha-Shapes.” *Proceedings of the Ninth Canadian Conference on Computational Geometry*, Vol 1. pp 193-198.
- Bernardini, F., Bajaj, C. L., Chen, J. & Schikore, D. (1999) “Automatic Reconstruction of 3D CAD Models form Digital Scans.” *International Journal on Computational Geometry and Applications*, Vol 2. pp 327-370.

- Bertram, M., Tricoche, X. & Hagen, H. (2003). "Adaptive Smooth Scattered-Data Approximation for Large-Scale Terrain Visualization." In: *Proceedings of the symposium on Data visualisation 2003*. Eurographics Association, Aire-la-Ville, Vol. 1, pp 177-184.
- Blinn, J. F. (1982) "A Generalization of Algebraic Surface Drawing", *ACM Transactions on Graphics (TOG)*, Vol.1, No.3, pp.235-256.
- Boissonnat, J.-D. (1984). "Geometric Structures for Three-Dimensional Shape Representation." *ACM Transactions on Graphics*, Vol. 3., pp 266-286.
- Boissonnat, J. D. & Cazals, F. (2000). "Smooth Surface Reconstruction Via Natural Neighbour Interpolation of Distance Functions" In: *Proceedings of the Sixteenth Annual Symposium on Computational Geometry 2000*. Vol 2. pp 223-232.
- Bruno, C. & Netto, S. (2007). "Comparação de algoritmos adaptativos para cancelamento de eco acústico". In: XXV Simpósio Brasileiro de Telecomunicações. URL.<<http://www.lps.ufrj.br/profs/sergioln/papers/BC14.pdf>>
- Carr, J. C., Beatson, R. K., Cherrie, J. B, Mitchell, T. J. & Fright, W. R. (2001). "Reconstruction and Representation of 3D Objects with Radial Basis Functions". In: *Proceedings of SIGGRAPH 2001*. ACM Press, Nova Iorque, pp.67-76.
- Chui, K. L., Chiu, W. K. & Yu, K. M. (2008). "Direct 5-Axis Tool-path Generation from Point Cloud Input Using 3D Biarc Fitting." *Robotics and Computer-Integrated Manufacturing*. Department of Industrial and Systems Engineering, Vol.1, pp 270–286.
- Cox, M. G., Harris, M. & Kenward, P. (2002). "Fixed and Free Knot Univariate Least-squares Data Approximation by Polynomial Splines." *Technical Report*. National Physical Laboratory, Teddington, Reino Unido. URL <http://handle.dtic.mil/100.2/ADP013744>
- Dey, T. K. & Goswami, S. (2003) "Tight Cocone: A Watertight Surface Reconstructor." *Journal of Computing and Information Science in Engineering*. ASME Journals, Arizona, pp 302-307.
- Dey, T. K. & Goswami, S. (2006) "Provable Surface Reconstruction from Noisy Samples." *Computational Geometry: Theory and Applications*. Elsevier Science, Holanda, pp.124-141.
- Diniz, P. S. R. (2002). *Adaptive Filtering: Algorithms and Practical Implementations*. Kluwer Academic Publishers, Boston, 2^a. edição.

- Eck, M. & Hoppe, H. (1996). "Automatic Reconstruction of B-spline Surfaces of Arbitrary Topological Type." *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*. ACM, Nova Iorque, Vol 2, pp 325–334.
- Edelsbrunner, H. & Mücke, E. P. (1994). "Three-dimensional Alpha Shapes." *ACM Transactions on Graphics*, Vol.13, pp 43-72.
- Edelsbrunner, H. (2004). "Surface Reconstruction by Wrapping Finite Point Sets in Space". In: *Discrete Computational Geometry: The Goodman-Pollack Festschrift*. Springer-Verlag, Berlim, Vol.3, pp.379-404.
- Elsässer, B. (1998). "Approximation with Rational B-spline Curves and Surfaces". *Proceedings of the International Conference on Mathematical Methods for Curves and Surfaces II Lillehammer*. Vol.1, pp.87-94.
- Farin, G. (2001). "Shape". *Math. Unlimited-2001 and Beyond*, 3a. edição, pp 463-466.
- Farin, G. (2002). *Curves and Surfaces for Computer-Aided Geometric Design: A Practical Guide*. Morgan Kaufmann Publishers (Academic Press), EUA, 5a. edição.
- Forsyth, D. & Ponce, J. (2003). *Computer Vision: A Modern Approach*. Prentice-Hall, Nova Jersey.
- Ginani, L. & Motta, J. M. S. T. (2007). "A Laser Scanning System for 3D Modeling of Industrial Objects Based on Computer Vision". In: *Proceedings of 19th International Congress of Mechanical Engineering, Vol. 1*. Brasília, Brasil.
- Ginani, L.S. (2008). Desenvolvimento de um Sistema de Digitalização Tridimensional de Superfícies Baseado em Visão Computacional com Varredura a Laser para Uso em Metrologia Dimensional. Dissertação de Mestrado em Sistemas Mecatrônicos, Publicação ENM.DM-18A/2008, Departamento de Engenharia Mecânica, Universidade de Brasília, Brasília, DF, 187p.
- Gois, J. P. (2004) *Reconstrução de Superfícies a Partir de Nuvens de Pontos*. Dissertação de Mestrado. Instituto de Ciências Matemáticas e de Computação - Universidade de São Paulo, São Carlos, 155 pp.
- Gregorski, B., Hamann, B. & Joy. D. (2000). "Reconstruction of B-spline Surfaces from Scattered Data Points." In: *Proceedings of Computer Graphics International 2000*. pp 163-170.
- Haykin, S. (2001). "Kalman Filtering and Neural Networks". In: John Wiley & Sons.
- Hoppe, H., DeRose, T., McDonald, J. & Stuetzle, W. (1992). "Surface Reconstruction from Unorganized Points." In: *ACM Computer Graphics Conference Proceedings*, Vol.1, pp. 71-78.

- Hoppe. H., DeRose, T., Duchamp, T., Halstead, M., Jin, H., McDonald, J., Schweitzer, J. & Stuetzle, W. (1994). "Piecewise Smooth Surface Reconstruction." In: *ACM SIGGRAPH 1994 Conference Proceedings*. ACM Press, Vol.1, pp. 295-302.
- Fisher, M., Mandic, D., Bangham, J. A. & Harvey, R. (2000). "Visualising Error Surfaces for Adaptive Filters and Other Purposes." *Proceedings of the Acoustics, Speech, and Signal Processing, 2000. on IEEE International Conference - Volume 06*. IEEE Computer Society, Washington, pp. 3522-3525
- Kahlesz. F., Balázs, A. & Klein, R. (2002). "Multiresolution Rendering by Sewing Trimmed NURBS Surfaces." In: *7th ACM Symposium on Solid Modeling and Applications*. ACM, Nova Iorque, Vol. 1, pp 281-288.
- Kolluri, R. K., Shewchuk, J. R. & O'Brien, J. F. (2003). "Spectral Watertight Surface Reconstruction." In: *Proceedings of the SIGGRAPH 2003*. ACM Press, Nova Iorque. Vol 8, pp 1022-1032.
- Kolluri, R. K. (2005). "Provably Good Moving Least Squares." In: *Proceedings of the 2005 ACM-SIAM Symposium on Discrete Algorithms*. Society for Industrial and Applied Mathematics, Vancouver, pp 1008-1018.
- Krishnamurthy, V. & Levoy, M. (1996). "Fitting Smooth Surfaces to Dense Polygon Meshes". In: *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*. ACM SIGGRAPH, Nova Iorque, Vol. 2. pp 313-324.
- Manolakis, D. G., Ingle, V.K. & Kogon, S. M. (2000). *Statistical and Adaptive Signal Processing: Spectral Estimation, Signal Modelig, Adaptive Filtering and Array Processing*. McGraw-Hill, EUA.
- Marques, P. A. (2003). "Introdução à Filtragem Adaptativa." In: *Apostila do curso de Processamento Digital de Sinal II - Engenharia Informática e de Computadores*. Instituto Superior de Engenharia de Lisboa, Lisboa. URL <<http://www.deetc.isel.ipl.pt/sistemastele/STDS/Bibliografia/TextoFiltragemAdaptativa.pdf>>
- Mederos, B., Velho, L. & de Figueiredo, L. H. (2003). "Moving Least Squares Multiresolution Surface Approximation." In: *Anais do Simpósio Brasileiro de Computação Gráfica e Processamento de Imagens (SIBGRAPI 2003)*. São Carlos, Vol.1, pp.19-26.
- Mederos, B., Amenta, N., Velho, L. & De Figueiredo, L. H. (2005). "Surface Reconstruction from Noisy Point Clouds." In: *Eurographics Symposium on Geometry Processing*. Eurographics Association, Aire-la-Ville, Vol.1, pp. 53-62.

- Minetto, C. (2003). “Um estudo sobre curvas NURBS”. Dissertação de mestrado - Universidade Federal do Rio Grande do Sul. Programa de Pós-graduação em Matemática Aplicada. URL<<http://biblioteca.universia.net/ficha.do?id=38064776>>
- Muller, H. (1992). “Surface Reconstruction - An Introduction.” In: Scientific Visualization Conference, Vol. 1, pp. 239- 239.
- Muraki, S. (1991). Volumetric Shape Description of Range Data Using “Blobby Model.” *ACM SIGGRAPH Computer Graphics*. ACM, Vol 1, pp 227-235.
- Nakamura, S. (2005). *Numerical Analysis and Graphic Visualization with MATLAB*. Prentice Hall Hispanoamericana S.A.
- Oppenheim, A. V. & Schafer, R. W. (1993). *Digital Signal Processing*. Prentice-Hall, EUA.
- Park, K., Lee, S. U. & Yun, I. D. (1999). “Constructing NURBS Surface Model from Scattered and Unorganized Range Data.” In: *Second International Conference on 3D Digital Imaging and Modeling (3DIM'99)*. Vol 2, pp 312-317.
URL < <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=805361> >
- Pender, T. (2003). *UML Bible*. Wiley Publishing, EUA.
- Peters, J. (1994). “Constructing Cl Surfaces of Arbitrary Topology Using Bicubic and Bicubic Splines.” *Designing Fair Curves and Surfaces*. Vol 2. pp 277–293.
- Piegl, L. A. & Tiller, W. (1997). *The NURBS Book*. Springer, Nova Iorque, 2a. edição.
- Randrianarivony, M. & Brunnett, G. (2002). “Parallel Implementation of Surface Reconstruction from Noisy Samples.” In: Preprintreihe des Chemnitzer SFB Vol 393, pp 02-16.
- Rodríguez, A., Espadero, J M., López, D. & Pastor, L. (2000). “Delaunay Surface Reconstruction from Scattered Points.” In : *Proceedings of the 9th International Conference on Discrete Geometry for Computer Imagery*. Vol. 1953, PP 272-283.
- Roth, G. & Boulanger, P. (1998). “CAD Model Building from Multiple Range Images”. In: *Proceedings of Vision Interface 98*. Vancouver, Canadá, pp. 274-281.
- Teixeira, F. G. (2003). *Modelamento Paramétrico e Geração de Malha em Superfícies para Aplicações em Engenharia*. Tese de Doutorado, Departamento de Doutorado em Engenharia Mecânica, Universidade Federal do Rio Grande do Sul.
- Turk, G. & O'Brien, J. F. (2002). “Modelling with Implicit Surfaces that Interpolate.” In: *ACM Transactions on Graphics (TOG)*. ACM, Nova Iorque, pp.855-873.
- Wayne T. (1997), *The NURBS Book*, Springer-Verlag, New York, 2nd edition.

- Wutke, J. D. (2006). *Métodos para Avaliação de um Sistema Laser Scanner Terrestre*. Tese de Mestrado, Departamento de Geomática - Universidade Federal do Paraná. p140
- Xie, H. & Qin, H. (2001). “Automatic Knot Determination of NURBS for Interactive Geometric Design.” In: *Proceedings of International Conference on Shape Modelling and Applications (SMI 2001)*. IEEE Computer Society, Washington, p 267.
- Yang, J., Speidel, U., Zany, E. A. & Girgis, M. R. (2005). “Collect the Fitted Surfaces into Complex Based on C0 Continuity”. *Journal of Universal Computer Science*. Faculty of Science, Minia University, Egipto, Vol. 11, pp 1102 – 1115.
<http://www.jucs.org/jucs_11_6/collect_the_fitted_surfaces>
- Yvart, A., Hahmann, S. & Bonneau, G.-P. (2005). “Smooth Adaptive Fitting of 3D Models Using Hierarchical Triangular Splines”. In: *Proceedings of the International Conference on Shape Modeling and Applications*. Vol.1, pp. 13-22.
- Zhao, H. K., Osher, S., Merriman, B. & Kang, M. (2000). “Implicit and Nonparametric Shape Reconstruction from Unorganized Data Using a Variational Level Set Method.” *Computer Vision and Image Understanding*. Elsevier Science, Nova Iorque, EUA, Vol.2, pp. 295-314.
- Zhao, H. K., Osher, S. & Fedkiw, R. (2001). “Fast Surface Reconstruction Using the Level Set Method”. *Proceedings of IEEE Workshop on Variational and Level Set Methods in Computer Vision*. IEEE Computer Society, Los Alamitos, EUA. pp 194-201.

APÊNDICES

APÊNDICE A - VISUALIZAÇÃO DA MALHA TRIANGULAR ATRAVÉS DE RAMPAS DE COR

Este apêndice descreve a elaboração do mapa de cor com base no sistema *RGB*, como é frequentemente exigido na visualização de objetos num programa de desenho *CAD*.

Essa ferramenta foi utilizada porque os dados provenientes da nuvem de pontos apresentam informação apenas da geometria, mas não de sua textura. A ilustração de toda a informação proveniente da malha com uma só cor faz com que objeto virtual perca aos olhos do usuário uma boa representação. Diante disso, foi preciso realizar um estudo de coloração que mudasse as cores dos triângulos da malha, dependendo da profundidade em que se encontram os pontos desde um marco de referência conhecido.

A.1 - ESCALA DE CINZA

Inicialmente, foi detalhado o mapeamento de cor de uma só grade de intensidade, chamada escala de cinza, cuja faixa está compreendida entre o preto e o branco.

Para ilustrar essa escala foi utilizado um cubo como meio de representação, chamado cubo *RGB* (Amenta *et al.*, 2000), que se observa na figura (A.1). A escala de cinza foi determinada por uma linha diagonal que cruza os pontos $(0,0,0)$ que corresponde cor preto e $(1,1,1)$ que corresponde cor branco.

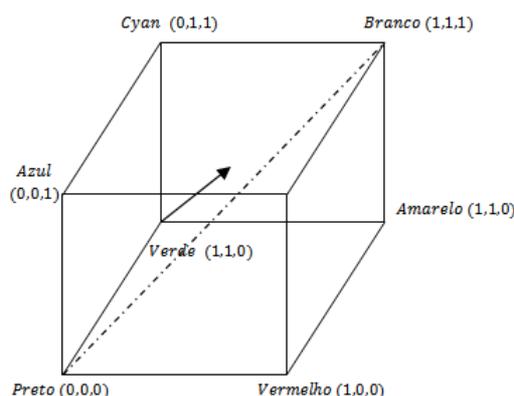


Figura A.1. Cubo de cor *RGB* – escala Cinza.

Para identificar quantitativamente a cor cinza correspondente a um valor de profundidade (z), dentro de uma faixa de dados $[z_{min} z_{max}]$, o valor foi determinado através da seguinte expressão:

$$vermelho = verde = azul = \frac{(z - z_{min})}{(z_{max} - z_{min})} \quad (A.1)$$

A.2 - ESCALA DE COR

Essa escala é também chamada escala termográfica, onde a cor azul é escolhida para representar os valores mais baixos, a verde para valores médios e a vermelha para valores altos. A trajetória que segue entre o azul e o vermelho é mostrada no cubo *RGB*. Essa trajetória tem como fim introduzir mais valores de variação de cor ao passar de um valor ao outro.

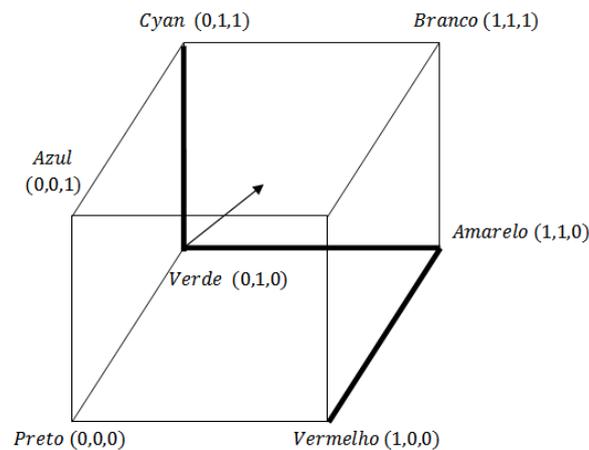


Figura A.2. Cubo de cor *RGB* – escala Termográfica.

A faixa de cor é indicada abaixo, com as transições de cor:

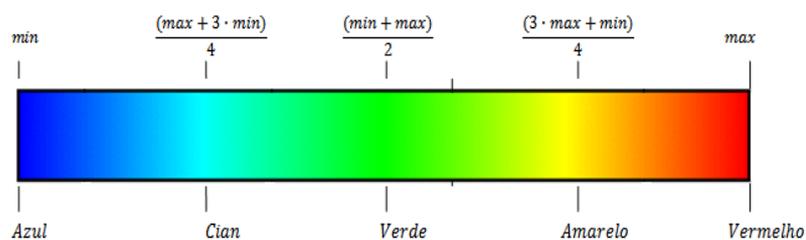


Figura A.3. Faixa de cor Termográfica.

As figuras A.4 e A.5 exibem em perspectivas diferentes os resultados obtidos, além do algoritmo implementado:

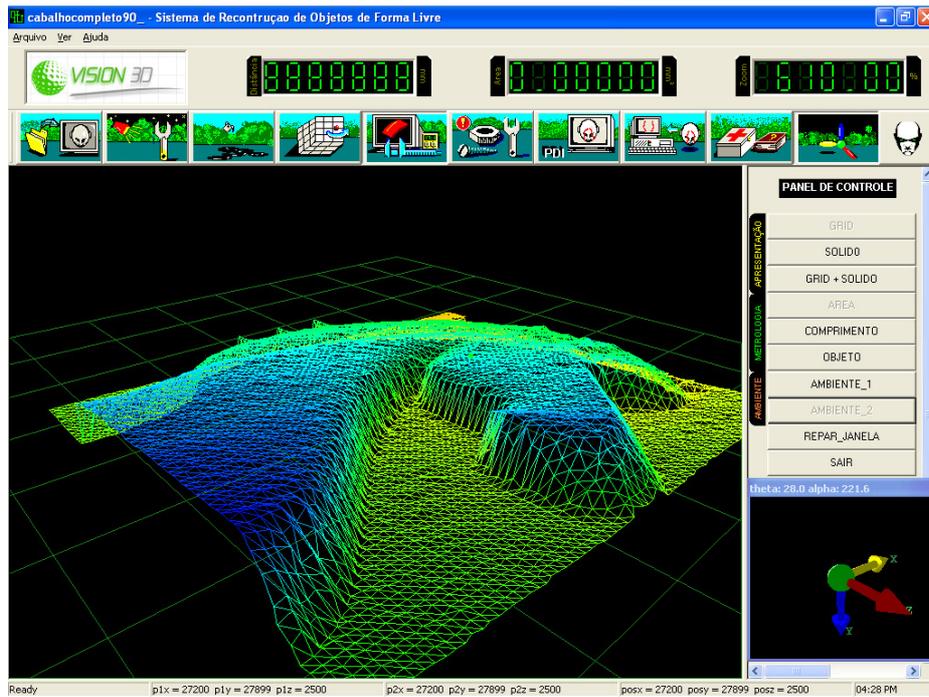


Figura A.4. Representação da figura através da escala termográfica – malha 3D. Perspectiva A.

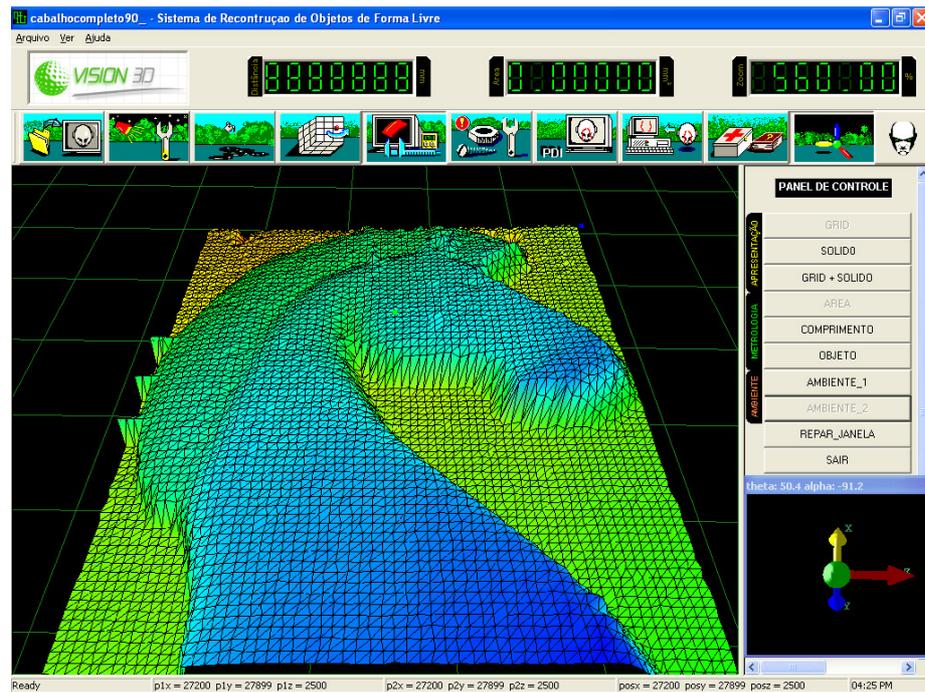


Figura A.5. Representação da figura através da escala termográfica – malha 3D sólida. Perspectiva B.

/*CÓDIGO EM C - ESCALA COR

Esse programa retorna um valor RGB depois de receber três valores que representam a altura do objeto z , em relação a um ponto de referência conhecido, que varia $[z_{min}, z_{max}]$.

```

typedef struct {
    double r, g, b;
} COLOR;

COLOR RGBCor(double z, double zmin, double zmax)
{
    COLOR c = {1.0,1.0,1.0}; // BRANCO
    double dz;
    if (z < zmin)
        z = zmin;
    if (z > zmax)
        z = zmax;
    dz = zmax - zmin;

    if (z < (zmin + 0.25 * dz)) {
        c.r = 0;
        c.g = 4 * (z - zmin) / dz;
    } else if (z < (zmin + 0.5 * dz)) {
        c.r = 0;
        c.b = 1 + 4 * (zmin + 0.25 * dz - z) / dz;
    } else if (z < (zmin + 0.75 * dz)) {
        c.r = 4 * (z - zmin - 0.5 * dz) / dz;
        c.b = 0;
    } else {
        c.g = 1 + 4 * (zmin + 0.75 * dz - z) / dz;
        c.b = 0;
    }
    return(c);
}

```

APÊNDICE B - SUPERFÍCIES NO ESPAÇO ATRAVÉS DE OPENGL

O propósito deste apêndice é mostrar a geração do espaço das superfícies paramétricas NURBS utilizando OpenGL.

B.1 - SUPERFÍCIES EM OPENGL

A biblioteca Glut de OpenGL contém funções para desenhar Superfícies e Curvas *B-Splines*. A biblioteca OpenGL desenha as superfícies dividindo-as em triângulos. Ele permite o estabelecimento de propriedades de materiais, possibilitando a visualização da superfície com um alto grau de realismo.

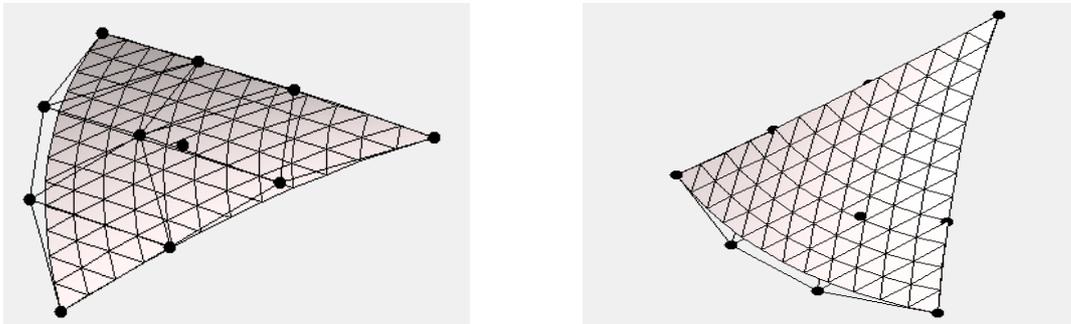


Figura B.1. Representação da superfície NURBS em diferentes perspectivas.

Para utilizar as funções NURBS é preciso definir as estruturas de dados para conter a descrição da superfície em OpenGL, além de inicializar os parâmetros de visualização e armazenamento.

Os parâmetros a inicializar são:

- Matriz de pontos de controle;
- Vetores de nós;
- Objeto superfície.

As matrizes de pontos de controle e os vetores de nós são do tipo GLfloat e o objeto da superfície é de tipo GLUNURBS.

Observa-se a seguir um exemplo da definição das matrizes geradas:

```
GLfloat ctrlpoints[4][4][3]; // 4x4 pontos de controle 3D
GLfloat knots_u[8] = {0.0, 0.0, 0.0, 0.0, 1.0, 1.0, 1.0, 1.0}; //vetor nós na direção u
GLfloat knots_v[8] = {0.0, 0.0, 0.0, 0.0, 1.0, 1.0, 1.0, 1.0}; //vetor nós na direção v
GLUnurbsObj *theNurb; //objeto
```

Na inicialização deve-se gerar o objeto NURBS e inicializar os parâmetros de visualização como:

- Tolerância de amostragem. Indica o numero de linhas em que se subdividirá a superfície.
- Modo de desenho (representação do objeto em linhas ou de modo sólido);
- Geração de vetores normais.

```
theNurb = gluNewNurbsRenderer();
gluNurbsProperty(theNurb, GLU_SAMPLING_TOLERANCE, 5.0); //Tolerância de amostragem
gluNurbsProperty(theNurb, GLU_DISPLAY_MODE, GLU_FILL); // Modo de desenho
glEnable(GL_AUTO_NORMAL); //vetores normais
```

Por último, se desenha a superfície através das funções `gluBeginSurface`, `gluNURBSSurface` e `gluEndSurface`.

```
gluBeginSurface(theNurb);
gluNurbsSurface(theNurb,
                8,
                knots_u,
                8,
                knots_v,
                4 * 3,
                3,
                &ctlpoints[0][0][0],
                4, 4,
                GL_MAP2_VERTEX_3);
gluEndSurface(theNurb);
```

Na figura B.2, se observa o entalhe NURBS em duas diferentes perspectivas.

A figura B.2(b) mostra com maiores detalhes que o entalhe não envolve quatro dos pontos de controle. Para que exista uma aproximação entre estas duas estruturas, é preciso realizar uma valoração dos parâmetros do vetor paramétrico nó mas detalhes são descritos no capítulo 5.

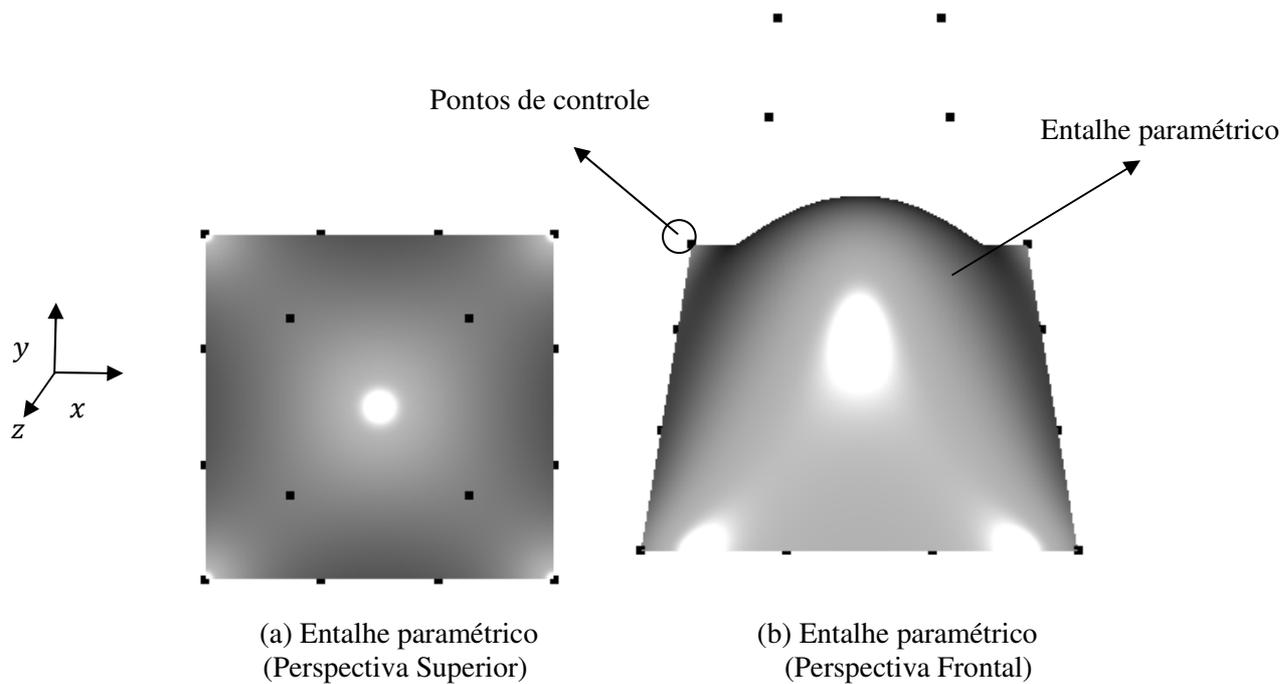


Figura B.2. Entalhe NURBS.

Vetor $\text{Knot}_u = \{0,0,0,0,1,1,1,1\}$ e com vetor $\text{Knot}_v = \{0,0,0,0,1,1,1,1\}$;

APÊNDICE C - SOFTWARE DESENVOLVIDO

Durante o desenvolvimento desta dissertação foram necessárias a construção e programação de uma quantidade considerável de algoritmos. As soluções apresentadas ao longo deste trabalho têm um desenvolvimento prático importante. Essas soluções necessitam de uma base de software que permita a adaptação de forma mais eficiente de todos os algoritmos desenvolvidos.

Quando o *software* foi inicialmente desenhado, levou-se em conta uma série de fatores que afetariam o resultado final. O que se buscou foi um sistema que fosse fácil de estender e atualizar.

Os aspectos que definem a estrutura do programa desenvolvido foram os seguintes:

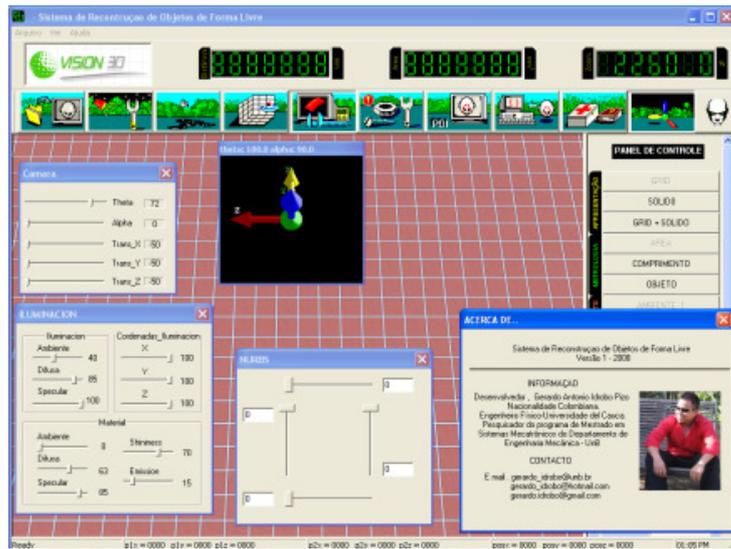
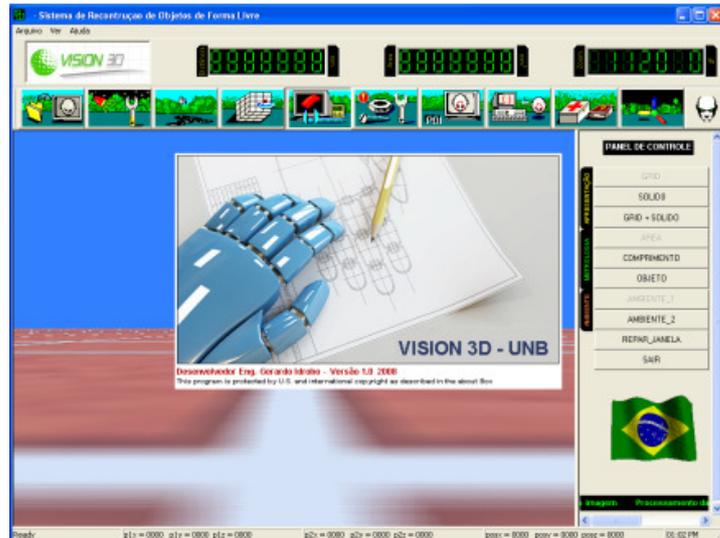
- O *software* deve ser o mais eficiente possível. O acesso ao *hardware* gráfico deve ser garantido, tanto em *API*²⁰ como de *shaders*²¹.
- A interface deve ser suficientemente elaborada para permitir uma utilização rápida e cômoda pelo usuário.
- O *software* deve estar preparado para agregar funcionalidade, novas estruturas e algoritmos dentro de um limite aceitável.
- Graficamente, deve ser possível a visualização interativa dos resultados através de ferramentas com barra de menus.
- Uma quantidade suficiente de formatos gráficos deve ser aceitar como entrada.
- Uma estrutura de dados que suporte a visualização (pontos, linhas, triângulos, malhas estruturadas e superfícies paramétricas) deve ser fornecida ao usuário.

Em função dos critérios anteriores, a programação foi totalmente realizada em código C++. Esse código permite que o projeto seja orientado a objetos estruturados e facilmente modificáveis (Pender, 2003). Da mesma maneira, o código permite o emprego de recursos de baixo e alto nível. A figura C.1 exibe a plataforma de desenvolvimento que apresenta as superfícies digitalizadas.

²⁰ *API* Interface de Programação de Aplicativos

²¹ *Shader* Instruções de *software* usados nos processadores gráficos para renderizar efeitos.

Entorno Visual



Superfície 3D

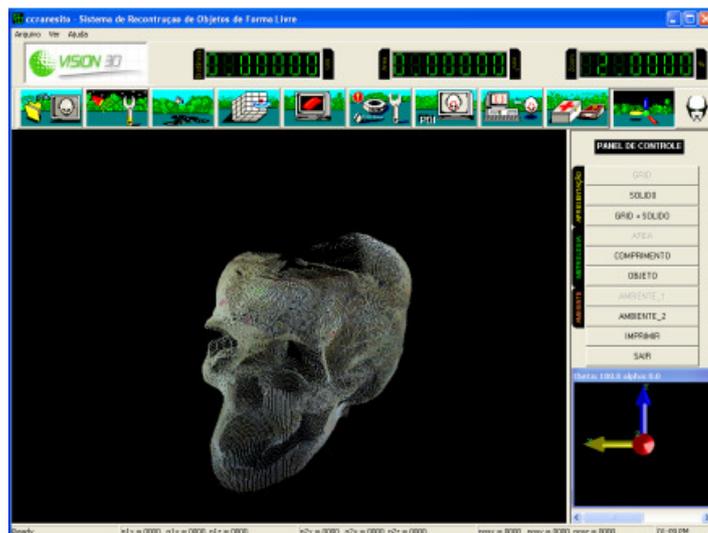


Figura C.1. Entorno visual em que é apresentada a superfícies virtuais

C. 1- ESTRUTURA DO SOFTWARE

O ambiente de desenvolvimento utilizado durante foi o *Visual Studio - Microsoft*. A versão em que foi projetado o *software* é a *Visual C++ 6.0* sobre a plataforma *Windows XP*. Essa escolha foi motivada pelos seguintes fatores: possibilidade de criação da interface de usuário, execução e depuração completa dentro do ambiente, assim como o aperfeiçoamento de códigos disponíveis (*SSE2*, multitarefa, dentre outros). Para o acesso da *CPU* programável utilizou-se a *Nvidia-Cg*. Para a programação dos *Shaders* e *OpenGL* utilizou-se *API 3D*.

Durante o processo de pesquisa foi necessária a realização de algoritmos individuais para testar a informação de experimentos específicos. No entanto, em determinado momento, faz-se necessário encadear toda a informação para criar um método que permita cumprir o objetivo geral.

As classes em que está desenhado o *software* foram construídas de forma a possibilitar a sua utilização em futuras aplicações. Desde o início deu-se uma clara separação entre as classes relacionadas com a interface e as classes consideradas parte do núcleo do programa.

Como é conhecido, o C++ introduz uma série de perdas em rendimento devido a fatores como as conexões dinâmicas, construtores, sobrecarga de operadores, *RTTI (Run Time Type Information)*, entre outros. A linguagem foi utilizada como meio e não como fim. Foi evitado o uso de conexões dinâmicas (funções virtuais) e funções *inline* quando possível.

C.2 - ORGANIZAÇÃO DO SOFTWARE

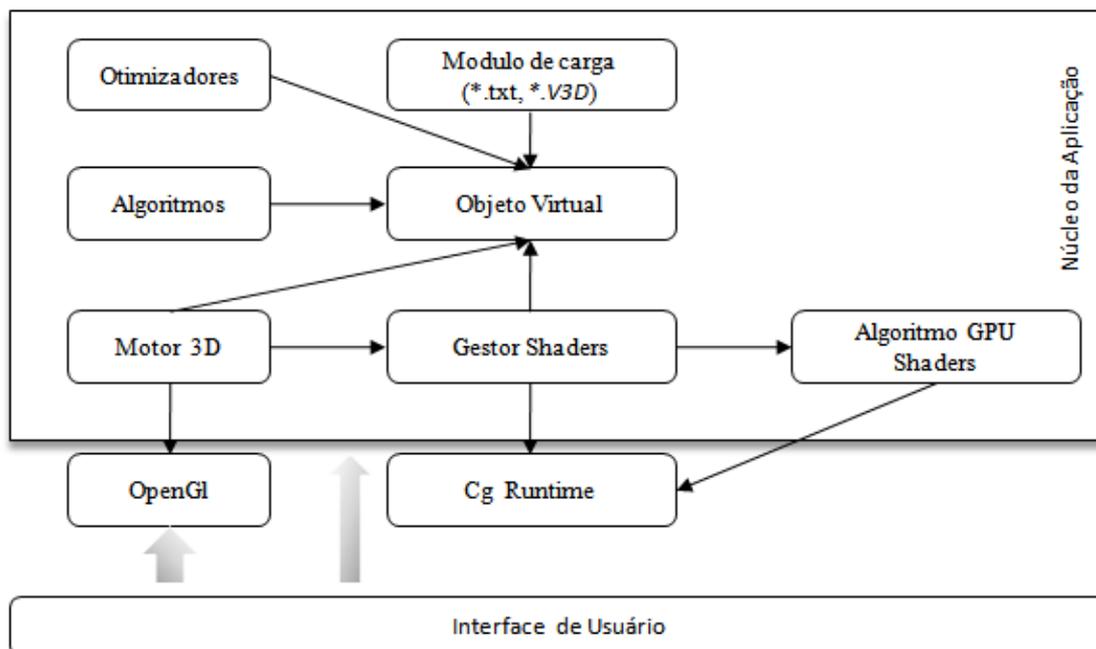


Figura C.2. Diagrama básico do Núcleo *software* que indica as unidades funcionais mais importantes.

A Figura C.2 mostra as unidades funcionais mais importantes do *software* e seus elementos mais significativos. Cada unidade compõe-se de uma ou várias classes inter-relacionadas de forma conveniente. O programa foi construído de forma a apresentar um núcleo e uma interface totalmente separados. A fim de oferecer uma breve descrição do *software*, serão descritas as unidades funcionais em vez de toda a hierarquia de classes.

As classes do núcleo são responsáveis por todas as tarefas relacionadas com os algoritmos geométricos implementados, os otimizadores espaciais, a recarga de dados, a gestão de *shaders* e as operações básicas sobre os objetos virtuais. Essas classes foram programadas em *ANSI C++* e não dependeram de nenhuma biblioteca própria do sistema operacional de nenhum *software* proprietário. O objetivo foi manter o núcleo com portabilidade a outros sistemas operativos e compiladores.

As classes da interface estão intimamente relacionadas com o sistema operacional. De forma concreta, estão baseadas em *MFC (Microsoft Foundation Classes)*. Essa é a biblioteca que proporciona uma abstração de recursos do sistema operacional através da API do *Windows*, o que permite o controle de todos os aspectos da interface de usuário. Devido à existência de uma adaptação das classes da interface em uma biblioteca própria do *Windows* (o que impede sua portabilidade a outros sistemas) esse grupo de classes se mantém bem separado do núcleo.

1. **Painel de controle:** é a caixa de diálogo principal que controla o ambiente ou janela em que está representado o objeto de estudo (figura C.7). Nessa mesma caixa de diálogo é também administrada a representação do objeto (pontos, malha triangular 3D em cor de cinza ou RGB, superfícies cobertas por entalhes paramétricos bicúbicos e representações surgidas através de combinações anteriores). O painel de controle também permite a comunicação com outras caixas de diálogo como utilização de ferramentas que permitam extrair medidas de distancias, áreas e volumes de uma superfície virtual.

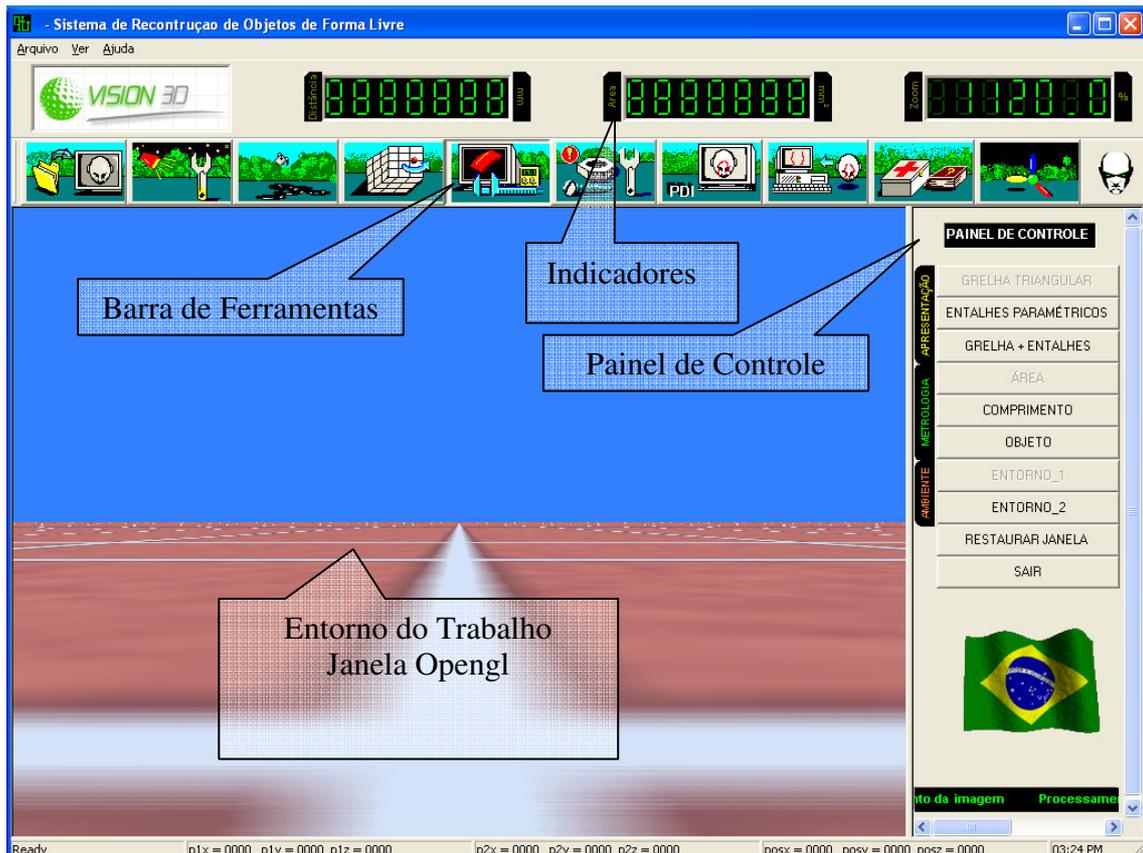


Figura C.7. Interface de usuário.

2. **Barra de Ferramentas:** permite a comunicação com as caixas de diálogo. Os elementos da Barra de Ferramentas são:
 - **Abrir:** permite selecionar a nuvem de pontos a ser representada. Para o carregamento da superfície virtual foram implementados vários módulos que permitem importar dados de arquivos com extensão **IMG.txt* ou **.V3D*. Todos os dados carregados são traduzidos para uma nuvem de pontos ou para uma malha de polígonos, dependendo da escolha do usuário.
 - **Iluminação:** contem ferramentas de iluminação, simulando ambientes reais.
 - **Cor:** permite a manipulação da cor da cena e da superfície a ser representada.

- **Câmera:** controla uma câmera virtual simulando a visualização do objeto em qualquer posição.
 - **Metrologia:** metrologia dimensional que permite através de geometria computacional extrair medidas como comprimentos, áreas e volumes de áreas específicas ou do objeto geral.
 - **Propriedades:** permite controlar alguns recursos dos ambientes em que é apresentado o objeto em estudo.
 - **PDI (Processamento Digital de Imagens):** processa a nuvem de pontos que chega do sistema VISSCAN-3D através dos algoritmos adaptativos (capítulo 3). Posteriormente, a função *PDI* organiza os dados em uma malha triangular 2D e permite ao usuário selecionar a área do objeto que deseja representar na tela 3D.
 - **Superfície:** compõe-se de uma só classe que organiza e administra a malha de polígonos que representa superfície do objeto, incluindo seu controle e sua representação em superfícies paramétricas
 - **Sistema Referência:** é uma janela em *OpenGL* que trabalha independentemente da janela principal em que é representado o objeto. Foi construída como um ponto de referência para a câmera virtual anteriormente citada.
3. **Algoritmos GPU:** nessa unidade estão incluídas as classes que contêm os algoritmos geométricos apresentados neste trabalho. Esse algoritmo não se inclui como parte da classe que administra os objetos virtuais. São denominados algoritmos GPU por que estão implementados em C++ para serem compilados e executados na CPU.

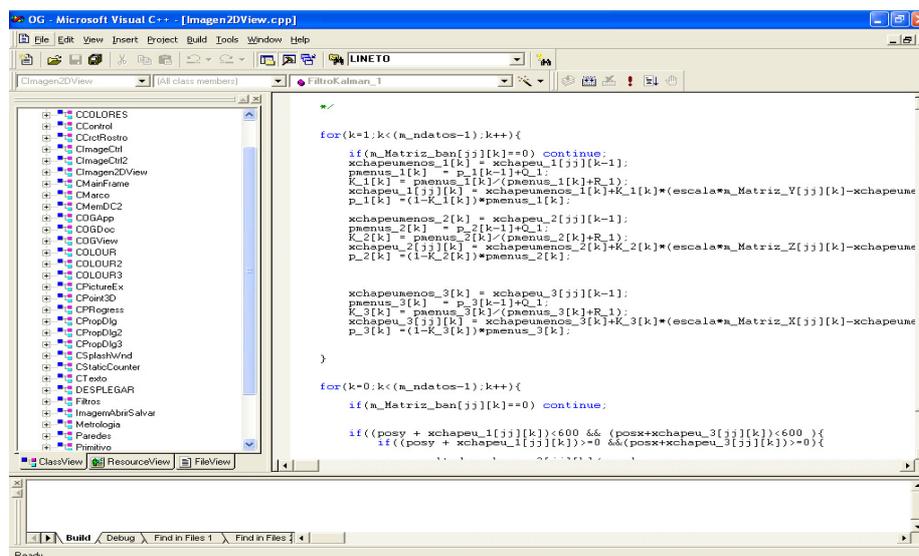


Figura C.8. Plataforma de visual C++ na qual se desenvolveu o sistema 3D.

4. **Gestor de *shaders***: administra, compila e executa os programas para GPU. Está sempre a serviço do motor *3D*.
5. **Motor 3D**: tem a responsabilidade de melhorar a cena, assim como o controle dos *shaders* ou o programa para *GPU*. É a unidade funcional mais complexa.

C.4 MODELAGEM UML DO SISTEMA DE PROCESSAMENTO DE NUVEM DE PONTOS

Esta unidade apresenta a modelagem UML de toda aplicação proposta, através dos diagramas de casos de uso.

C.4.1 – DIAGRAMA DE CASOS DE USO

A modelagem de um diagrama de caso de uso é uma técnica usada para descrever e definir os requisitos funcionais de um sistema. Eles são escritos em termos de atores externos, casos de uso e o sistema modelado. Os atores representam o papel de uma entidade externa ao sistema como um usuário, um hardware, ou outro sistema que interage com o sistema modelado. Os atores iniciam a comunicação com o sistema através dos *use-cases* (casos de uso). Estes representam uma seqüência de ações executadas pelo sistema e recebem do ator dados tangíveis de um tipo ou formato já conhecido.



Figura C.9 Diagrama de caso uso do sistema de reconstrução de superfícies

C.4.1 – DIAGRAMA DE CLASSE

Diagramas de classes mostram as relações entre as diferentes classes que fazem parte do sistema de construção de superfícies através de uma nuvem de pontos. O Diagrama de classes da figura c.10 apresenta os métodos e atributos e os relacionamentos estáticos entre elas.

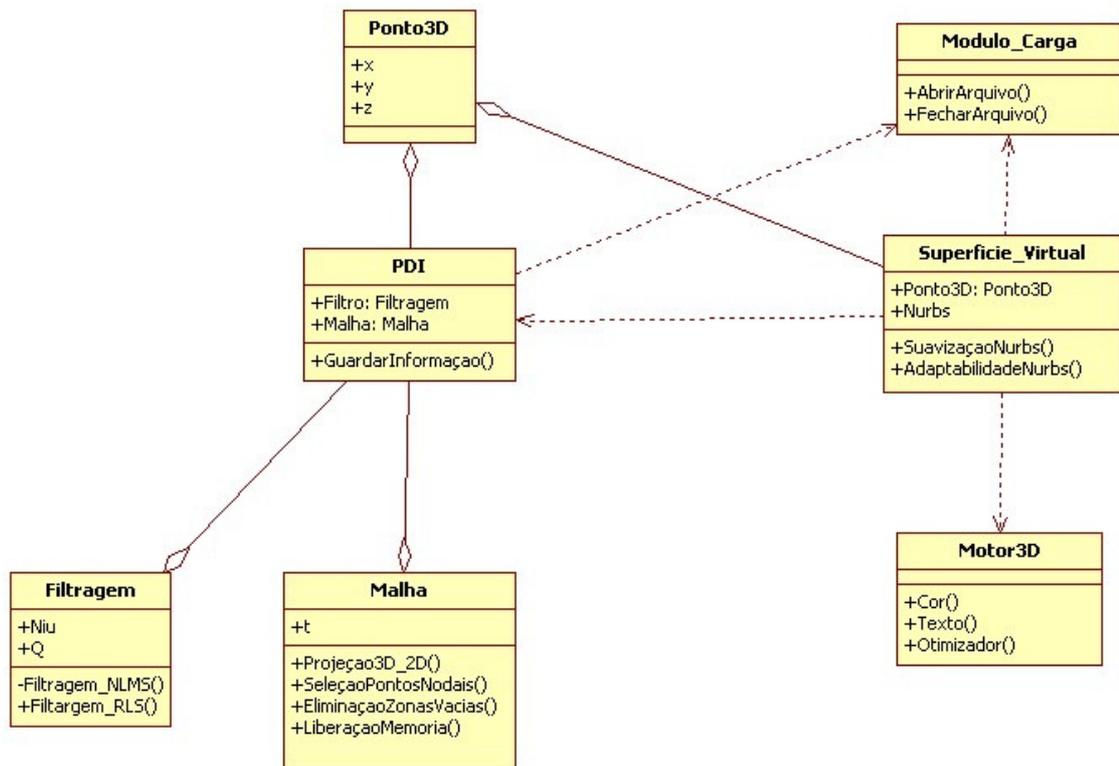


Figura C.10 Diagrama de classes do sistema de reconstrução de superfícies

Classe Filtragem permite realizar o seguimento da informação proveniente do sistema VISCAN3D, eliminando as interferências que não permitem uma representação ótima da superfície a ser analisada.

Classe Malha, permite organizar a nuvem de pontos em uma matriz através do método de modelos deformáveis.

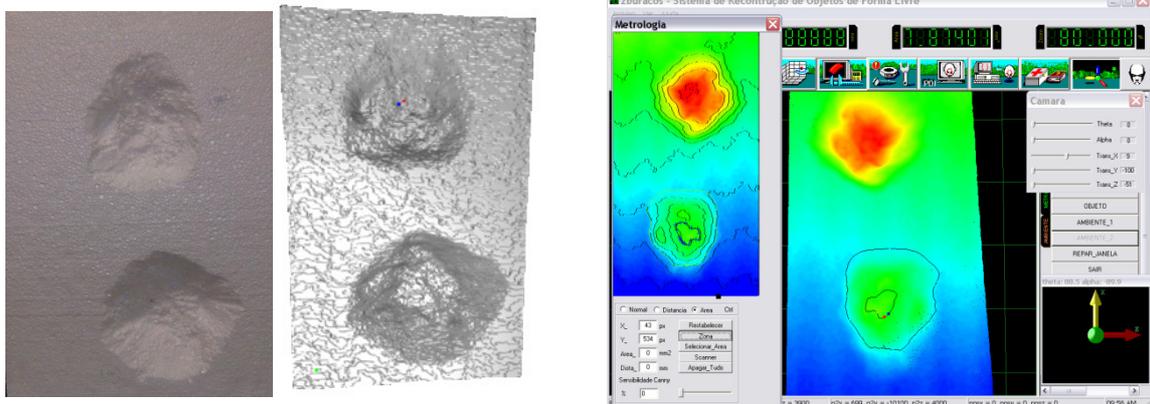
Classe PDI permite realizar o processamento da nuvem de pontos e modelar estas em uma superfície que é armazenada em um arquivo de texto.

Classe superfície_virtual, através da classe modulo de carga e classe Motor3D permite representar e dar realismo a superfície em 3D.

APÊNDICE D – OUTROS RESULTADOS

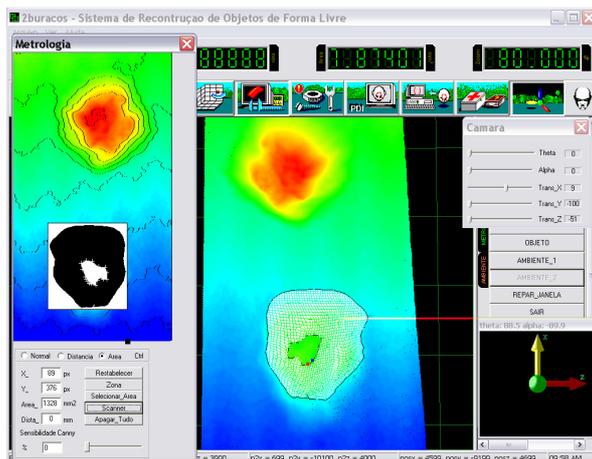
D.1 - AVALIAÇÃO GEOMÉTRICA DE UMA SUPERFÍCIE DANIFICADA POR CAVITAÇÃO

As turbinas hidráulicas de grande porte das usinas de geração de energia elétrica apresentam erosão por cavitação e necessitam ser restauradas pela deposição de material por soldagem. A seguir, observa-se a digitalização de um modelo com forma similar à superfície a ser trabalhada (figura D.1).



(a) Esquerda - imagem original.
Direita - imagem construída através de entalhes paramétricos.

(b) Representação em malha triangular 2D (imagem esquerda) e 3D (imagem direita).
Escala de cores RGB.



(d) Monitoramento da superfície construída.
Obtenção de uma área selecionada

T. Entrada	T. Saída	t	μ_{NLMS}	$\bar{\epsilon}_{ajuste}$
8.41(MB)	1.99 (MB)	5 mm	0.4	2.5 mm

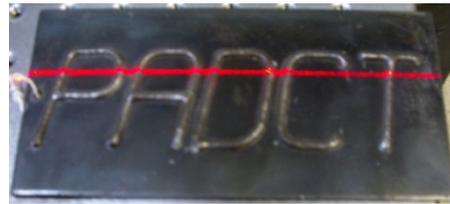
Tabela D.1. Parâmetros de construção da superfície danificada por cavitação (simulação da peça).

Figura D.1. Avaliação geométrica de uma superfície danificada por cavitação (simulação da peça).

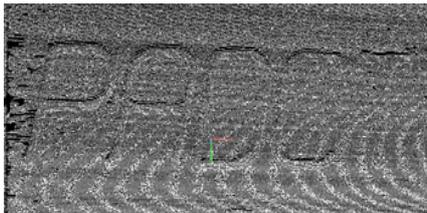
D.2. CONSTRUÇÃO PASSO A PASSO DE UMA CORDA DE SOLDA



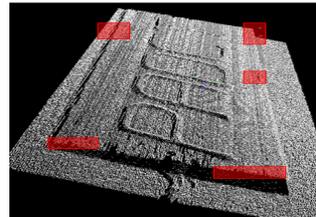
(a) Cordão de solda.



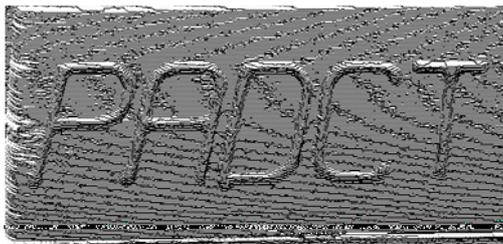
(b) Processo de varredura *laser* sobre o cordão de solda.



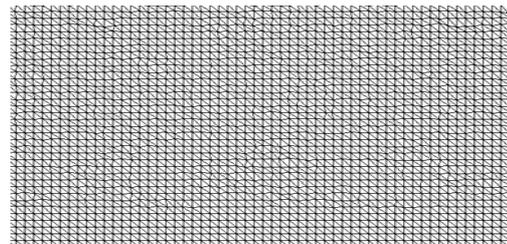
(c) Nuvem de pontos capturados do cordão de solda.



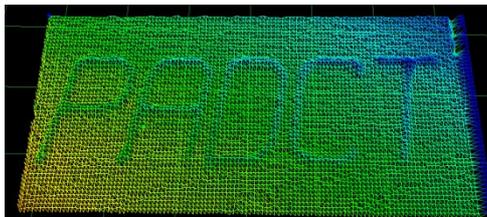
(d) Identificação de ruídos sobre a nuvem de pontos.



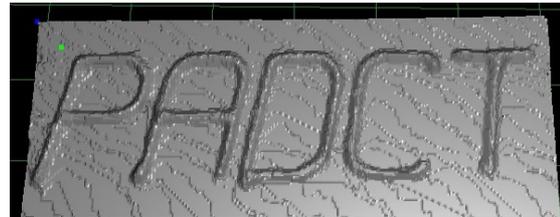
(g) Projeção da nuvem de pontos sobre o plano 2D.



(h) Malha triangular 2D.



(i) Malha triangular 3D.



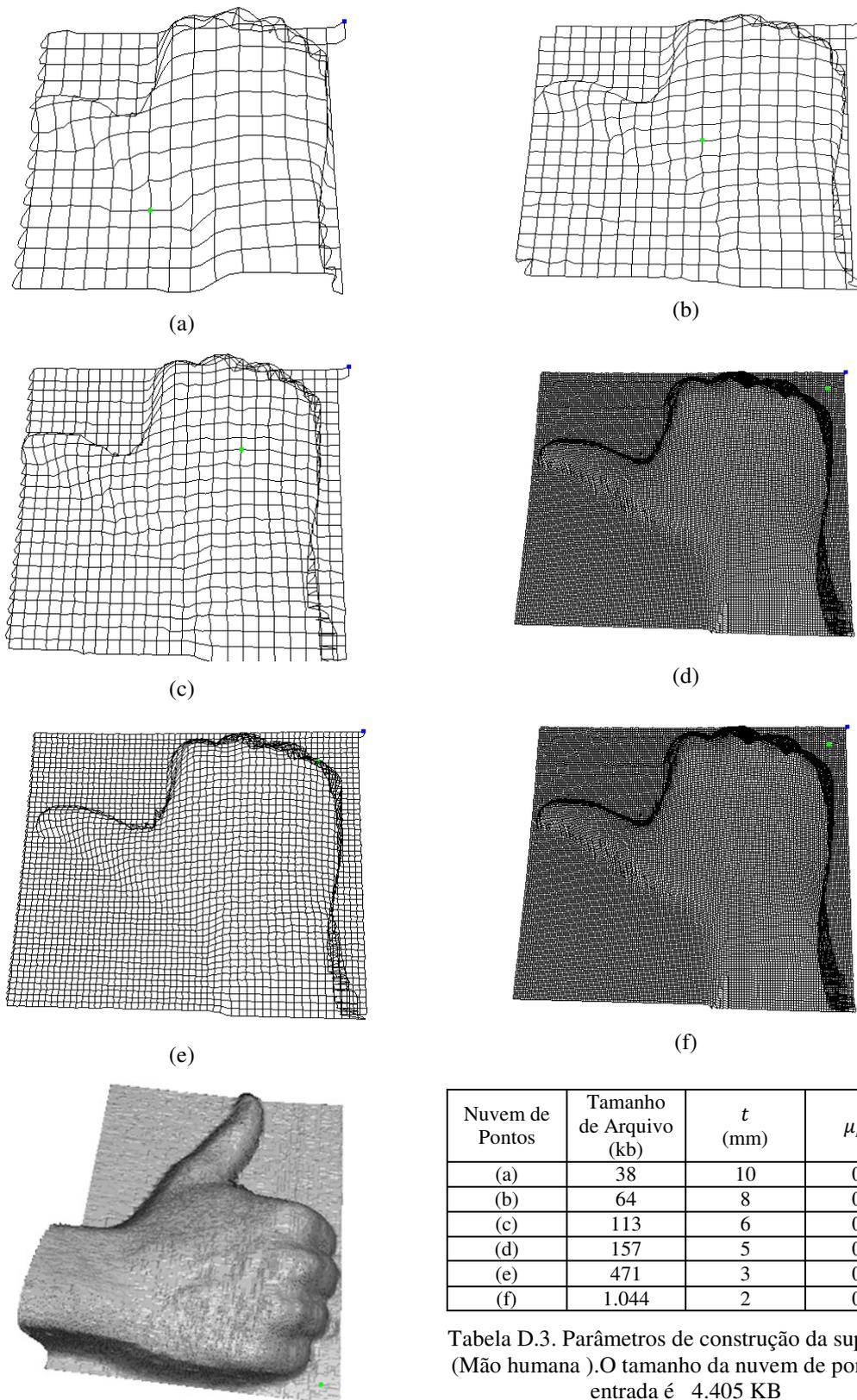
(j) Superfícies paramétricas sobre a malha triangular 3D.

Figura D.2. Procedimento geral para reconstrução de superfícies através de nuvens de pontos.

Tabela D.2. Parâmetros de construção da superfície (corda de solda)

Tamanho do arquivo da nuvem de pontos	Tamanhode arquivo de saída da nuvem de pontos	t	Q_{RLS}	$\bar{\epsilon}_{ajuste}$
3.76 (MB)	2.42(MB)	2 mm	0.5	1.7 mm

D.3. RESOLUÇÃO DA SUPERFÍCIE



Reconstrução da superfície com tamanho de grade triangular (t) de 10 mm

Figura D.3. Construção de uma superfície em diferentes valores de resolução.

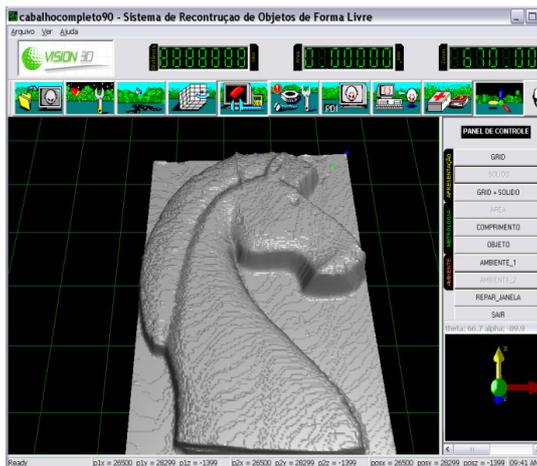
D.4. CONSTRUÇÃO DE UMA SUPERFÍCIE DE FORMA LIVRE



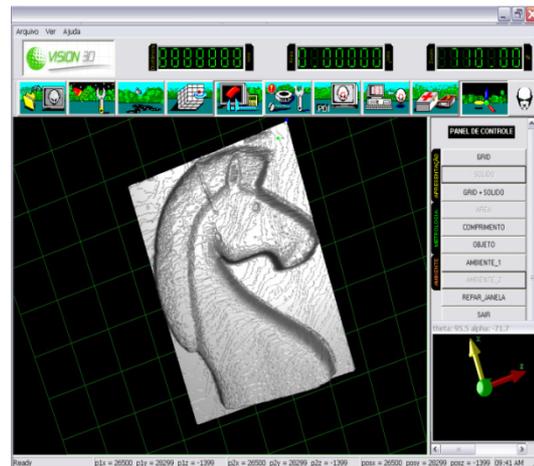
(a) Imagem Original

Tamanho Entrada	Tamanho de arquivo de Saida	t	μ_{NLMS}	$\bar{\epsilon}_{ajuste}$
9.26 MB	4.88MB	3 mm	0.5	1.2 mm

Tabela D.4. Parâmetros de construção da de uma superfície de forma livre.



(b) Imagem virtual A



(c) Imagem Virtual B

Figura D.4. Construção de uma superfície de forma livre

APÊNDICE E – DEFINIÇÕES

Nesta seção se apresenta um conjunto definições e nomenclaturas

Amostragem

Uma amostragem de uma variedade M é um conjunto finito de pontos

Amostragem uniforme

Uma amostragem é dita ε -uniforme se para todo q pertencente à superfície original existe um ponto p da amostra tal que $\|p - q\| < \varepsilon$.

API(Application Programming Interface (ou Interface de Programação de Aplicativos))

É um conjunto de rotinas e padrões estabelecidos por um software para a utilização das suas funcionalidades por programas aplicativos isto é, programas que não querem envolver-se em detalhes da implementação do software, mas apenas usar seus serviços

Diagrama de Voronoi

Seja $P \subset R^n$ um conjunto de pontos. Para cada p e $q \in P$, seja $B(p, q) = \{x: \|x - p\| \leq \|x - q\|\}$, contendo p , e $D(p, q)$ é o semi-espaco que contém q . Defina se por $VR(p, P) = \cap_{q \in P, q \neq p} D(p, q)$ a região de Voronoi.

O Diagrama de Voronoi de P é definido por: $V(P) = \cup_{p, q \in P, p \neq q} VR(p, P) \cap VR(q, P)$.

A figura E.1 apresenta um exemplo de Diagrama de Voronoi:

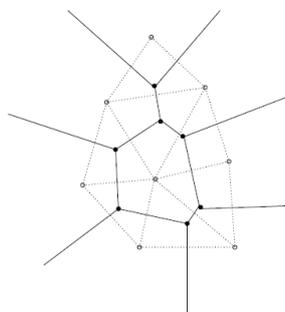


Figura E.1. Diagrama de Voronoi (dois pontos sem preenchimento) em linhas contínuas. Em linhas pontilhadas, seu dual, a Triangulação de Delanuy.

ε – amostragem :

Uma ε – amostragem é definida a partir do local-feature size $f(x)$ da seguinte forma: seja um ponto x sobre a superfície original, sempre deve existir um ponto p na amostragem tal que $\|p - x\| \leq \varepsilon f(x)$. Ou seja, para cada ponto da superfície deve existir um ponto na amostra a uma distância proporcional ao local-feature size.

Eixo medial

Define se como eixo medial de uma superfície M o conjunto de pontos que definem os centros de esferas que tangenciam M em ao menos dois pontos.

Entalhe paramétrico

Patches NURBS

Face

Se σ é um n -simplexo formado pelos pontos $X = \{x_0, \dots, x_n\}$ qualquer l -simplexo dado por um subconjunto de $l+1$ com $l < n$ elementos de X é face de σ .

Fecho convexo

Dado um conjunto de pontos $P \subset R^n$, o conjunto de todas as combinações lineares convexas, dada pela seguinte equação, exprime a definição de fecho convexo (figura E.2)

$$\text{Conv}(P) = \left\{ \sum_{i=0}^n \lambda_i p_i, \sum_{i=0}^n \lambda_i = 1, \lambda_i \geq 0, \lambda_i \in R, p_i \in P \right\} \quad (\text{E.1})$$

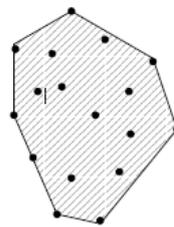


Figura E.2. Exemplo bidimensional do fecho convexo de um conjunto de pontos.

O fecho convexo $\text{Conv}(P)$ pode ser visto como o menor conjunto convexo que contém os pontos de P .

Homeomorfismo

Sejam os conjuntos $U \subset R^n$ e $V \subset R^m$. Afirma-se que uma função $f: U \rightarrow V$ é um homeomorfismo, se f e f^{-1} são contínuas. Nesse caso U e V são ditos homeomorfos.

ISO

O "ISO" de uma câmera (em inglês "ISO setting" ou "ISO speed") é um padrão que descreve a sua sensibilidade absoluta à luz

Local Feature Size

A local Feature Size é a distância de um ponto na superfície ate o eixo medial.

Quádrica

Conjunto dos pontos do espaço tridimensional cujas coordenada formam um polinômio de segundo grau de no máximo três variáveis denominada de equação cartesiana da superfície:

$$Ax^2 + By^2 + Cz^2 + Dxy + Eyz + Fxz + Gx + Hy + Iz + J = 0 \quad (\text{E.2})$$

Malhas estruturadas

São Malhas que apresentam a mesma conectividade para todos os vértices internos. Seus nós internos formam tipicamente quadriláteros ou hexaedros, seus algoritmos geralmente empregam métodos complexos de suavização na tentativa de alinhar os elementos internos com os elementos das bordas do domínio.

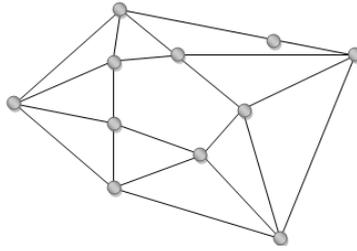


Figura E.3 Uma malha estruturada, todos os nos internos têm grau quatro.

Malhas não estruturadas

São malhas que apresentam maior liberdade na conectividade dos vértices internos, podendo apresentar qualquer número de arestas incidentes. Os elementos de uma malha não estruturada são tipicamente triângulos ou tetraedros

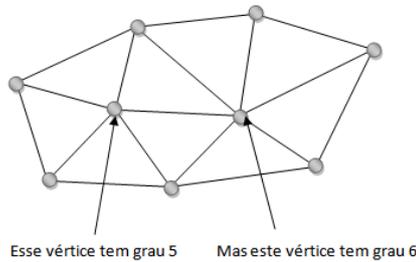


Figura E.4 Uma malha não estruturada, os nos internos possuem graus diferentes

Triangulação de Delaunay

Seja $X = \{x_0, \dots, x_k\} \subset R^n, n \leq k$, afirma-se que uma triangulação de X é de Delaunay se a circunferência de todo n -simplexo não contém nenhum outro ponto de X em seu interior.

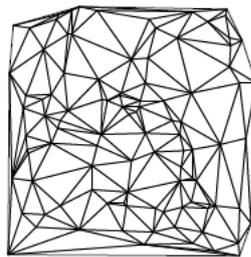


Figura E.5. Exemplo de uma Triangulação de Delaunay de um conjunto de 100 pontos no plano euclidiano.

Simplexo

Um n -simplexo $\sigma \subset R^n$ é o fecho convexo de $n + 1$ pontos em posição geral (figura E.6).

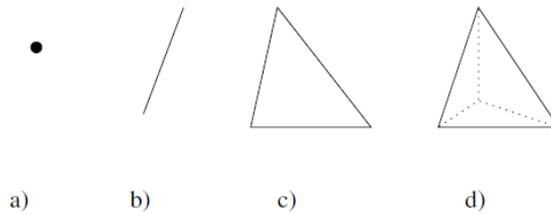


Figura E.6. (a) simplexo (b)1-simplexo (c)2-simplexo (d)3-simplexo

Superquadrica

Trata-se de uma generalização das quádricas donde se adicionam parâmetros que afeitam o grau da equação implícita, com aquilo se obtém variações interessantes da forma. Por exemplo Super-elipse será uma variante do elipse, donde se adiciona um parâmetro n que afeita o grau da equação. Na figura E.5, se pode observar os efeitos das mudanças em n .

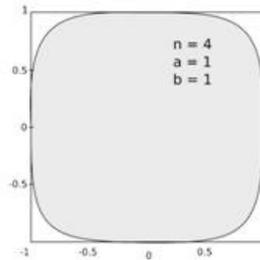


Figura E.7. Uma super elipse para $n = 4$, $a = b = 1$ se aproxima a um quadrado arredondado.

RBFs (Função-base radial)

Uma função-base radial é uma função simétrica em relação à vizinhança radial de um ponto (centro).

Ruído em Imagens Digitais

Para as imagens digitais, o ruído aparece como manchas aleatórias em uma superfície originalmente suave e pode degradar a qualidade da imagem significativamente.

Tipos de ruído

Câmeras digitais produzem três tipos mais comuns de ruído: aleatório, de 'padrão fixo' e de bandas. As três imagens abaixo mostram exemplos qualitativos pronunciados de casos isolados de cada um dos tipos de ruído contra um fundo cinza.



Ruído com Padrão Fixo
Exposição Longa
Baixo ISO

Ruído Aleatório
Exposição Curta
Alto ISO

Ruído em Bandas
Câmera Suscetível
Sombras Clareadas

Figura E.8. Ruído dentro de uma imagem.

Variedade

O conjunto $M \subset \mathbb{R}^k$ é uma variedade de dimensões n classe C^r se para todo $x \in M$ existe uma vizinhança aberta $U \ni x$, com $U \subset \mathbb{R}^k$, V aberto de H^n e um difeomorfismo de classe C^r $\varphi: U \cap M \rightarrow V$ figura (E. 9)

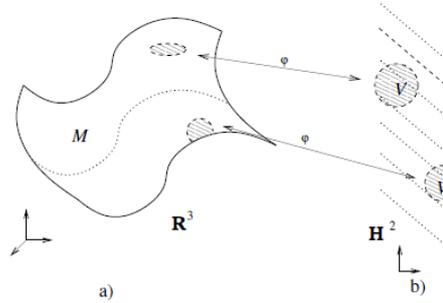


Figura E.9. Exemplo de uma variedade de dimensão 2 imersa em \mathbb{R}^3 .

Vizinhança Natural.

Sejam $X = \{x_0, \dots, x_n\}$ uma amostra obtida sobre uma superfície S e p um ponto qualquer. A vizinhança natural de p é dada pelos pontos interligados a p na Triangulação de Delanuy de $X \cup p$.

Muito Obrigado DEUS

Muchas Gracias DIOS