
Correção de normais para suavização de nuvens de pontos

Paola Tatiana Llerena Valdivia

SERVIÇO DE PÓS-GRADUAÇÃO DO ICMC-USP

Data de Depósito:

Assinatura:

Correção de normais para suavização de nuvens de pontos

Paola Tatiana Llerena Valdivia

Orientador: Prof. Dr. Luis Gustavo Nonato

Dissertação apresentada ao Instituto de Ciências Matemáticas e de Computação - ICMC-USP, como parte dos requisitos para obtenção do título de Mestre em Ciências - Ciências de Computação e Matemática Computacional. *VERSÃO REVISADA*

USP – São Carlos
Dezembro de 2013

Ficha catalográfica elaborada pela Biblioteca Prof. Achille Bassi
e Seção Técnica de Informática, ICMC/USP,
com os dados fornecidos pelo(a) autor(a)

L146c Llerena Valdivia, Paola Tatiana
 Correção de normais para suavização de nuvens de
 pontos / Paola Tatiana Llerena Valdivia; orientador
 Luis Gustavo Nonato. -- São Carlos, 2013.
 64 p.

Dissertação (Mestrado - Programa de Pós-Graduação
em Ciências de Computação e Matemática
Computacional) -- Instituto de Ciências Matemáticas
e de Computação, Universidade de São Paulo, 2013.

1. suavização de superfícies. 2. nuvens de pontos.
3. filtragem de normais. 4. estimativa de normais.
I. Nonato, Luis Gustavo, orient. II. Título.

A mi amada familia:
Liz V., Norma G., Jaime V.,
Neg y Ross.

Agradecimentos

Quero agradecer especialmente ao meu orientador Gustavo Nonato por ter me ajudado durante todo o processo, pela paciência e por todo o apoio e confiança que tornaram possível a conclusão desta dissertação. Quero agradecer também aos meus colegas do Grupo de Visualização e Processamento Geométrico, especialmente ao Douglas, Fabiano e Afonso pelas importantes contribuições neste trabalho.

Vou agradecer em espanhol à minha família e amigos. A mi mami Tatiana por el amor incondicional, los consejos y apoyo, por que siempre esta dispuesta a ayudarme a solucionar los mil y un problemas en que me meto. A mi abuelita Norma, igualmente, por su amor incondicional, por toda la paciencia, rezos, consejos y enseñanzas que incitaron en mí las ganas de siempre querer aprender más. A mi abuelito Jaime, por haber sido un papá para mí, por tratar de enseñarme a siempre encarar la vida con una sonrisa (y una canción o un silbido). A Neg, mi amiga y hermana, por el amor, el apoyo, la paciencia, la compañía, las risas, los gruñidos, por estar siempre cuidando de mí, por estar siempre presente, incluso en los silencios. A toda mi familia, por que cada uno de ellos, a su manera, contribuyeron en esta disertación.

A mis mejores amigos Mayra y Alan, aunque lejos, siempre (siempre!) presentes, dándome ánimos, ayudando a re-enfocarme, demostrando siempre su cariño, con sus consejos y palabras desacertadas. A Frizzi, por la amistad, consejos y tantos momentos compartidos. A Pedro, por la compañía durante las interminables

noches en el laboratorio, los cafés, y por intentar volverme aún más nerd con sus consejos y enseñanzas. A Alfredo, por su cariño, paciencia y por darme la ayuda y sacudida que faltaba para concluir esta disertación. A Yorsh, Erick, Aurea, Andre, toda la banda Meshkalina (oh yeah!).

À Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP), pela concessão da bolsa de mestrado, associada ao processo 2011/04045-3, e pelo apoio financeiro para a realização desta pesquisa.

Por último, mas não menos importante, agradeço também ao Alex e Martha, pela ajuda e as boas dicas. E a todos os demais professores e funcionários do ICMC que, diretamente ou indiretamente, contribuíram para a realização deste trabalho.

Com certeza faltaram muitos nomes aqui, a todos estes muito obrigado.

*Entonces, todos los hombres de la tierra
le rodearon; les vio el cadáver triste, emocionado;
incorporóse lentamente,
abrazó al primer hombre; echóse a andar.*

— Cesar Vallejo

Resumo

Nos anos recentes, suavização de superfícies é um assunto de intensa pesquisa em processamento geométrico. Muitas das abordagens para suavização de malhas usam um esquema de duas etapas: filtragem de normais seguido de um passo de atualização de vértices para corresponder com as normais filtradas. Neste trabalho, propomos uma adaptação de tais esquemas de duas etapas para superfícies representadas por nuvens de pontos. Para isso, exploramos esquemas de pesos para filtrar as normais. Além disso, investigamos três métodos para estimar normais, analisando o impacto de cada método para estimar normais em todo o processo de suavização da superfície. Para uma análise quantitativa, além da comparação visual convencional, avaliamos a eficácia de diferentes opções de implementação usando duas medidas, comparando nossos resultados com métodos de suavização de nuvens de pontos encontrados a literatura.

Palavras-chave: suavização de superfícies; nuvens de pontos; estimativa de normais; filtragem de normais.

Abstract

In the last years, surface denoising is a subject of intensive research in geometry processing. Most of the recent approaches for mesh denoising use a two-step scheme: normal filtering followed by a point updating step to match the corrected normals. In this work, we propose an adaptation of such two-step approaches for point-based surfaces, exploring three different weight schemes for filtering normals. Moreover, we also investigate three techniques for normal estimation, analyzing the impact of each normal estimation method in the whole point-set smoothing process. Towards a quantitative analysis, in addition to conventional visual comparison, we evaluate the effectiveness of different choices of implementation using two measures, comparing our results against state-of-art point-based denoising techniques.

Keywords: surface smoothing; point-based surface; normal estimation; normal filtering.

Sumário

Lista de Figuras	xi
Lista de Tabelas	xv
1 Introdução	1
1.1 Contextualização e Motivação	1
1.2 Objetivos	3
1.3 Organização	4
2 Marcos Teóricos	5
2.1 Considerações Iniciais	5
2.2 Nuvens de Pontos	5
2.2.1 Estimativa de Normais	6
2.2.2 Estruturas de Dados	9
2.3 Filtragem Bilateral	12
2.4 Suavização em Dois Passos de Malhas Triangulares	14
2.4.1 Atualização das Normais	15
2.4.2 Atualização dos Vértices	17
3 Suavização de Superfícies Representadas por Nuvens de Pontos	19
3.1 Considerações Iniciais	19
3.2 Técnicas Espectrais	20

3.3	Técnicas de Projeção	23
3.4	Técnicas Baseadas na Norma l_1	25
3.5	Outras Técnicas	29
4	Filtragem de Nuvens de Pontos por Correção de Normais e suas Variantes	31
4.1	Considerações Iniciais	31
4.2	Estimativa de Normais	33
4.2.1	Análise de componentes principais (PCA)	33
4.2.2	Análise de componentes principais ponderada (WPCA) .	34
4.2.3	Transformada de Hough aleatória (RHT)	35
4.3	Correção de Normais	37
4.3.1	Pesos Limiarizados	38
4.3.2	Peso Bilateral-Gaussiano	39
4.3.3	Peso Bilateral-Misto	40
4.4	Filtragem da Superfície	41
5	Resultados	45
5.1	Métricas	46
5.2	Comparações com Outras Técnicas	48
5.3	Preservação de Feições Afiadas	52
5.4	Modelos Reais	52
6	Conclusões	55
6.1	Contribuições	55
6.2	Limitações	56
6.3	Trabalhos Futuros	57

Lista de Figuras

2.1	Aproximação de um plano à vizinhança do ponto \mathbf{p}_i	7
2.2	Células voronoi para um conjunto de pontos (a) sem ruído e (b) com ruído, onde A é o pólo da célula Voronoi do ponto \mathbf{p}_i	8
2.3	Subdivisão do modelo Elephant usando uma <i>Octree</i> com no mínimo vinte pontos por octante.	10
2.4	A filtragem bilateral consegue suavizar a imagem de entrada preservando as feições (Resultado). Um peso ponderado da vizinhança substitui cada pixel. O peso é composto por um peso espacial, que penaliza a distância, e por um peso de rango, que penaliza a variação da intensidade. Assim, somente pixels próximos com intensidade semelhante contribuem no resultado final. Os pesos mostrados são aplicados ao pixel sob a seta. (Imagen adaptada de Kornprobst & Tumblin (2009))	11
2.5	Tipos de vizinhança da face F_i : (I) conjunto de faces que compartem um vértice com a face F_i e (II) conjunto de faces que compartilham uma aresta com a face F_i (Sun et al., 2007)	14
3.1	Suavização do modelo Armadillo utilizando um filtro passa baixo na decomposição espectral do operador Laplace-Beltrami baseado em SPH. O processo não consegue preservar feições (extraído de Petronetto et al. (2013)).	21

3.2	Processo de projeção do MLS. (a) Primeiro, um plano H é aproximado ao conjunto de pontos usando mínimos quadrados. (b) A projeção do ponto \mathbf{r} no plano H define a origem \mathbf{q} do domínio local de referencia. Utilizando esse domínio, um polinômio g é aproximado, novamente usando mínimos quadrados. (c) Finalmente, o ponto é projetado no polinômio g	22
3.3	Comparação das técnicas baseadas na norma l_1 . De esquerda a direita: nuvem de entrada, resultados de <i>Locally Optimal Projection</i> (LOP), <i>Weighted Locally Optimal Projection</i> (WLOP) e <i>Feature-preserving Locally Optimal Projection</i> (FLOP) (extraído de Liao et al. (2013)).	27
3.4	Suavização de nuvens de pontos usando elementos finitos. Na fila superior mostram-se as nuvens de entrada e na fila inferior, as nuvens suavizadas. A abordagem não consegue preservar feições (extraído de Clarenz et al. (2004a)).	29
4.1	<i>Pipeline</i> do processo de filtragem de nuvens de pontos e as variantes.	31
4.2	Estimação da normal \mathbf{n}_i para o ponto \mathbf{p}_i usando Análise de componentes principais (PCA). São mostradas as componentes principais CP_1 e CP_2	33
4.3	Estimação da normal \mathbf{n}_i para o ponto \mathbf{p}_i usando Análise de componentes principais ponderada (WPCA). São mostradas as componentes principais CP_1 e CP_2	34
4.4	Normais dos planos selecionados aleatoriamente na vizinhança de \mathbf{p}_i	35
4.5	Acumulador esférico de Borrman, com n_θ fatias e $2n_\phi$ bins no equador (extraído de (Boulch & Marlet, 2012)).	36
4.6	Os intervalos de confiança dos bins mais votados (p_{m_1} e p_{m_2}) não se intersetam.	37
4.7	Projeção da normal \mathbf{n}_i em \mathbf{n}_j , dada por $\mathbf{n}_i \cdot \mathbf{n}_j$ para três ângulos diferentes: (a) 80° , (b) 50° e (c) 20°	38

4.8	Distância entre os vetores \mathbf{n}_i e \mathbf{n}_j , e valor da função Gaussiana $W_s(x) = \exp(-x^2/2\sigma_s^2)$ com $\sigma_s = 0.3$ para três ângulos diferentes: (a) 80° , (b) 60° e (c) 20°	40
4.9	Projeção do vetor $\mathbf{n}_i - \mathbf{n}_j$ no vetor \mathbf{n}_j e valor do peso Φ_s com $T = 0.25$ para três ângulos diferentes: (a) 60° , (b) 38° e (c) 15°	41
4.10	(b) Filtragem sem levar em conta a orientação das normais e (c) orientando as normais localmente de forma consistente.	42
5.1	Avaliação quantitativa das métricas usando diferentes combinações do método de suavização no modelo Elefante ($\times 10^{-4}$).	47
5.2	Erro médio de cada alternativa para todos os modelos variando as 27 opções de vizinhança e número de passos por cada iteração.	48
5.3	<i>Scatter plot</i> para as métricas área \times curvatura no modelo Bitorus (ruído aleatório).	49
5.4	Suavização de nuvens de pontos: Bitorus com 4350 pontos, Elefante com 24955 pontos, Fêmur com 15185 pontos and Niccolo with 25239 pontos. De direita a esquerda: modelos originais, modelos com ruído Gaussiano com direção aleatória, e os resultados de LOP, WLOP, <i>Algebraic Point Set Surfaces</i> (APSS), <i>Robust Implicit Moving Least Squares</i> (RIMLS) e nosso método.	50
5.5	Nosso método é capaz de preservar feições afiadas enquanto suaviza as superfícies. De esquerda a direita: modelo original e com ruído; resultados usando LOP, WLOP, APSS, RIMLS e nosso método.	52
5.6	Suavização de um modelo real escaneado.	53
6.1	Um inconveniente com o ponto conjunto de superfícies é determinar a relação de vizinhança de pontos em superfícies com folhas próximas.	56

Lista de Tabelas

5.1 Melhor conjunto de parametros e erros	51
-----------------------------------------------------	----

Capítulo 1

Introdução

1.1 Contextualização e Motivação

O mundo ao nosso redor pode ser capturado e modelado por meio das superfícies que percebemos. Na matemática, as superfícies e as suas propriedades têm sido amplamente estudadas desde os séculos passados. Esses conceitos foram estendidos e aplicados na área de modelagem e processamento geométrico para muitas finalidades como por exemplo, suavização de superfícies.

O processamento geométrico é, principalmente, aplicar algoritmos a modelos geométricos, onde o algoritmo representa a ação e a geometria representa o objeto (Mario Botsch, 2010). Para modelar objetos em três dimensões, muitas pesquisas foram conduzidas, focadas em encontrar uma representação computacional apropriada. Não existe uma representação que seja suficientemente boa para todas as aplicações, pois aplicações distintas requerem representações distintas, visando por exemplo:

- Manipular objetos complexos a partir de objetos simples;
- Permitir editar um modelo para, por exemplo, animar algumas partes dele;
- Demandar um custo de memória ou tempo de processamento baixo.

Algumas abordagens utilizadas para representar as superfícies são: malhas de polígonos, nuvens de pontos, splines hierárquicos e superfícies de subdivisão. A representação mais comum de superfícies em computação gráfica é, provavelmente, *malhas de polígonos*. Uma malha de polígonos é essencialmente um conjunto de vértices (pontos), arestas (conexão entre vértices) e faces (conjunto fechado de arestas) que representam explicitamente a superfície. Na maioria das vezes, as faces são representadas por triângulos, quadriláteros ou outros polígonos simples, para simplificar os cálculos e a renderização. Devido a sua simplicidade, as malhas podem ser processadas eficientemente em computadores modernos e placas gráficas. As vantagens são principalmente a facilidade para a visualização, e também a informação de conectividade explícita.

As nuvens de pontos, a representação usada neste trabalho, são um tipo de representação de superfícies conhecidas como superfícies não estruturadas. As superfícies não estruturadas são as representações explícitas que não possuem um modelo de dados associado que indique a conexão entre os elementos básicos, que podem ser pontos ou polígonos. Quando o elemento básico é o ponto, tem-se uma *nuvem de pontos*, a representação mais simples que existe. A nuvem de pontos é um conjunto não ordenado de pontos (x, y, z em 3D) que approxima a superfície. Esta representação foi inicialmente proposta em 1985 por (Levoy & Whitted, 1985). Recentemente este tipo de representação tem experimentado um ressurgimento devido, principalmente, à popularização dos *scanners 3D*. As principais vantagens da representação por pontos são sua simplicidade e facilidade de renderização. As principais desvantagens são: não se tem informação da conectividade, topologia ou quantidades diferenciais além da limitação da teoria matemática para manipular tal representação.

As nuvens de pontos obtidas por *scanners* geralmente têm ruído, com maior concentração próximo das feições afiadas. Algoritmos de reconstrução de superfície conseguem remover o ruído da superfície mas transformam a representação, por exemplo, em malhas de polígonos. Quando se deseja trabalhar diretamente com a nuvem de pontos, temos que aplicar um algoritmo de suavização para remover o ruído. No trabalho realizado propomos um método de suavização de nuvens

de pontos que preserve feições afiadas.

1.2 Objetivos

O objetivo deste trabalho foi investigar uma metodologia de duas etapas para filtrar superfícies, no contexto de superfícies de pontos. As etapas desta metodologia são filtragem de normais e atualização de pontos (vértices no contexto caso de malhas). Enquanto normais em malhas são diretamente definidas por cada face, em nuvens de pontos precisamos estimar as normais previamente. Investigamos três métodos para estimar normais, três métodos para filtrar normais e propomos um método para atualizar os pontos. O estudo apresentado permite identificar o melhor conjunto de ferramentas, e como implementá-las em cada passo do processo de suavização para obter um método robusto e efetivo para suavizar nuvens de pontos a partir da filtragem de normais.

Em síntese, são objetivos específicos desse projeto:

- A investigação detalhada das alternativas para estender o esquema de suavização de superfícies via filtragem de normais no contexto de nuvens de pontos. Mais especificamente, investigar as possíveis implementações de cada etapa do *pipeline* de suavização proposto e analisar a eficácia de cada combinação.
- Desenvolver um método de atualização de pontos a partir de um campo de normais filtradas de modo a preservar feições.
- Um conjunto abrangente de comparações com técnicas de suavização de nuvens pontos.
- A definição de medidas quantitativas para analisar a eficácia das técnicas de suavização, o que permite avaliar objetivamente a qualidade dos métodos de suavização baseadas em nuvens pontos.

1.3 Organização

Esta dissertação está organizada da seguinte maneira:

Capítulo 2: os principais conceitos que têm orientado esta dissertação são apresentados.

Capítulo 3: apresentamos uma revisão bibliográfica sobre as técnicas de suavização de superfícies representadas por nuvens de pontos.

Capítulo 4: apresentamos a metodologia proposta para remover ruído da nuvem de pontos e as alternativas dos passos que compõem nosso *pipeline*: estimação de normais, filtragem de normais e atualização de pontos.

Capítulo 5: as métricas propostas para comparar a eficácia dos métodos para suavizar nuvens de pontos junto com um estudo comparativo entre o esquema proposto e outras técnicas de suavização para nuvens pontos.

Capítulo 6: apresentamos as conclusões desse projeto de mestrado, ressaltando as contribuições e limitações, além da discussão de trabalhos futuros.

Capítulo 2

Marcos Teóricos

2.1 Considerações Iniciais

Neste capítulo apresentamos alguns dos principais conceitos que têm orientado esta dissertação. Começamos descrevendo a representação de superfícies usando nuvens de pontos, os principais algoritmos para estimar normais e as estruturas de dados mais comuns para as armazenar. Continuamos introduzindo o tema de filtragem bilateral muito utilizado em imagens e que serve como inspiração para métodos de suavização de superfícies. Finalmente, apresentamos brevemente métodos de suavização de malhas em dois passos que inspiraram nosso método de suavização de nuvens de pontos.

2.2 Nuvens de Pontos

A representação mais simples de uma nuvem de pontos consiste somente das coordenadas dos pontos:

$$\mathcal{P} = \{\mathbf{p}_i \in \mathbb{R}^3, i \in \{1, \dots, n\}\}$$

Algoritmos específicos para diversos problemas no contexto de nuvens de pon-

tos tem sido desenvolvidos. Alguns desses problemas são suavização de nuvens de pontos, reconstrução de superfícies, estimativa de normais, dizimação e amostragem (Gross & Pfister, 2007).

Outra representação simples de nuvem de pontos considera, além das coordenadas, a normal de cada ponto e é chamada de nuvem de pontos orientada:

$$\mathcal{P} = \{\mathbf{p}_i \in \mathbb{R}^3, \mathbf{n}_i \in \mathbb{R}^3, i \in \{1, \dots, n\}\}$$

As normais podem ser estimadas utilizando algumas técnicas que geralmente analisam a vizinhança de cada ponto. A seguir são descritas algumas técnicas de estimativa de normais.

2.2.1 Estimativa de Normais

A estimativa de normais em nuvens de pontos é um tópico bastante estudado na área de processamento geométrico. Seja uma superfície \mathcal{M} amostrada por uma nuvem de pontos \mathcal{P} . O problema de estimativa de normais visa ter uma normal \mathbf{n}_i por cada \mathbf{p}_i , definida como um vetor perpendicular a um plano tangente à \mathcal{M} em \mathbf{p}_i . Cada normal \mathbf{n}_i tem origem no ponto \mathbf{p}_i associado e o campo de normais é representado por $N = \{\mathbf{n}_i \in \mathbb{R}^3, \|\mathbf{n}_i\| = 1\}$.

As normais estimadas servem como entrada de alguns algoritmos para remoção de ruído, reconstrução de superfícies ou detecção de feições afiadas. A qualidade das normais estimadas depende de vários fatores, como feições afiadas na superfície, regularidade de amostragem e o nível de ruído na nuvem de pontos. Nem todos os algoritmos existentes para estimar normais conseguem lidar com esses fatores.

Uma das abordagens proposta por Hoppe et al. (1992), aproxima um plano localmente para cada ponto e sua vizinhança usando mínimos quadrados (Figura 2.1). O plano $\mathbf{n}^\top \mathbf{x} = c$ é encontrado minimizando o erro $e(\mathbf{n}, c) = \sum_{i=1}^k (\mathbf{n}^\top \cdot \mathbf{p}_i - c)^2$ com a restrição $\mathbf{n}^\top \cdot \mathbf{n} = 1$. Essa abordagem é equivalente a encontrar o plano usando PCA na vizinhança de cada ponto (Secção §4.2.1). Assim, a normal para cada ponto é a normal ao plano estimado para cada vizinhança. A vizinhança

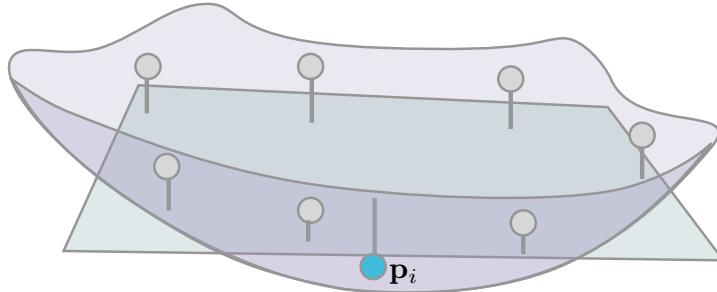


Figura 2.1: Aproximação de um plano à vizinhança do ponto \mathbf{p}_i

pode ser definida como os k vizinhos mais próximos ou como os pontos dentro de um raio r ao redor do ponto. As normais obtidas desse processo não são orientadas, demandando um pos-processamento para obter um campo de normais orientadas de forma consistente.

Pauly et al. (2003b) observaram que os pontos mais próximos na vizinhança de um ponto \mathbf{p}_i deveriam ter maior influencia na minimização do erro de aproximação do plano tangente. Por isso, adicionaram um peso ao erro:

$$e(\mathbf{n}, c) = \sum_{i=1}^k (\mathbf{n}^\top \cdot \mathbf{p}_i - c)^2 \theta(\|\mathbf{p}_i - \mathbf{p}\|),$$

onde θ é o peso Gaussiano: $\theta(t) = \exp(-t^2/h^2)$, na qual h é um parâmetro indicando o tamanho das feições da superfície.

Mitra & Nguyen (2003) investigaram uma forma de escolher o raio r da vizinhança adaptativamente. Eles observaram que o mesmo r para todos os pontos podia produzir resultados errados, devido a que em regiões com alta curvatura o erro da minimização aumenta proporcionalmente de acordo ao r . Observaram também, que a densidade da nuvem de pontos pode afetar o resultado. Assumindo um modelo de ruído aleatório, com média igual a zero e desvio padrão σ_n , propuseram um método analítico para delimitar o erro da normal como função de r , utilizando uma estimativa da curvatura local κ e da densidade local ρ para

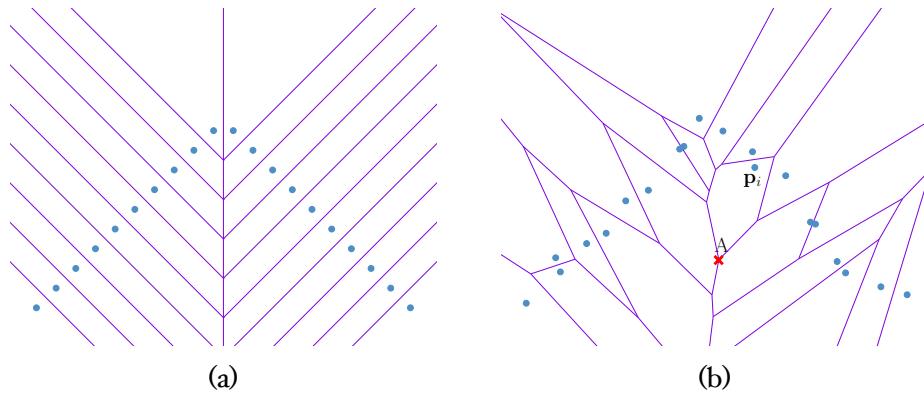


Figura 2.2: Células voronoi para um conjunto de pontos (a) sem ruído e (b) com ruído, onde A é o pólo da célula Voronoi do ponto \mathbf{p}_i

encontrar um r ótimo. O r ótimo pode ser encontrado com probabilidade $1 - \alpha$:

$$r = \left(\frac{1}{\kappa} \left(c_1 \frac{\sigma_n}{\sqrt{\alpha\rho}} + c_s \sigma_n^2 \right) \right)$$

onde c_1 e c_2 são duas constantes.

Amenta & Bern (1998) usaram o diagrama Voronoi de \mathcal{P} para obter uma estimativa das normais. Assumindo que \mathcal{P} tem alta densidade de pontos, a célula de Voronoi para um ponto \mathbf{p}_i é alongada na direção perpendicular à superfície (Figura 2.2). O vetor de um ponto ao pólo da sua célula de Voronoi é uma estimativa da normal, sendo que o pólo da célula de Voronoi é o ponto mais distante ao ponto que define a célula. Uma desvantagem deste método, além de ser custoso, é que é muito sensível ao ruído (Figura 2.2b).

Li et al. (2010), assumindo uma distribuição Gaussiana do ruído, propuseram um método de duas etapas para estimar normais. A primeira etapa é estimar a escala de ruído: para cada ponto \mathbf{p}_i amostram T planos aleatoriamente (selecionando tríades de pontos na vizinhança do ponto). Após, para cada plano calculam a distância do plano aos pontos na vizinhança de \mathbf{p}_i . Baseado nessas distâncias, calculam a escala do ruído. Posteriormente, na segunda etapa, utilizando a escala do ruído, calculam um plano tangente a cada ponto baseado em um núcleo de

densidade. Esse método consegue bons resultados na presença de ruído e feições afiadas. No entanto, a variação de densidades nas bordas afeta o desempenho do *kernel* de densidade, favorecendo os lados com alta densidade.

Recentemente, Boulch & Marlet (2012) propuseram um método baseado na transformada de Hough aleatória (RHT) (Borrmann et al., 2011). Eles utilizam métodos estatísticos para encontrar o número mínimo de amostras de planos a serem consideradas na vizinhança de cada ponto. Além disso, definiram uma condição de parada que consegue acelerar o processo de amostragem. Esse método consegue bons resultados em nuvens de pontos com densidade anisotrópica. Descrevemos o método em maior detalhe na Secção §4.2.3.

2.2.2 Estruturas de Dados

As nuvens de pontos podem ser muito grandes, chegando a milhões de pontos. Por tanto, é desejável dividir o conjunto de pontos em subconjuntos menores.

Dada a natureza não estruturada das nuvens de pontos, precisamos de uma estrutura de dados para armazenar a nuvem de pontos. Além das coordenadas dos pontos, as estruturas de dados, devem fornecer informação da vizinhança de cada ponto. Lembramos que as vizinhanças podem ser definidas de acordo com um raio r : $\mathcal{N}_i = \{\mathbf{p}_j \in \mathcal{P}, \|\mathbf{p}_i - \mathbf{p}_j\| < r\}$, ou podem também ser definidas como os k pontos mais próximos de \mathbf{p}_i .

Uma estrutura de dados muito usada para manipular nuvens de pontos é a *Oc-tree*. Proposta originalmente por (Meagher, 1982), é uma estrutura de dados espacial hierárquica modelada como uma árvore. Ela engloba todos os pontos e particiona a caixa envolvente em oito partes iguais. Cada parte que contenha pontos é recursivamente dividida em oito partes. A recursão continua até atingir um número mínimo de pontos num octante (Figura 2.3). Uma desvantagem dessa estrutura de dados é que geralmente as nuvens de pontos não podem ser divididas de forma uniforme, o que leva a uma estrutura de dados desequilibrada e ineficiente.

Outra estrutura de dados usada para manipular os dados é a *kd-Tree*, proposta

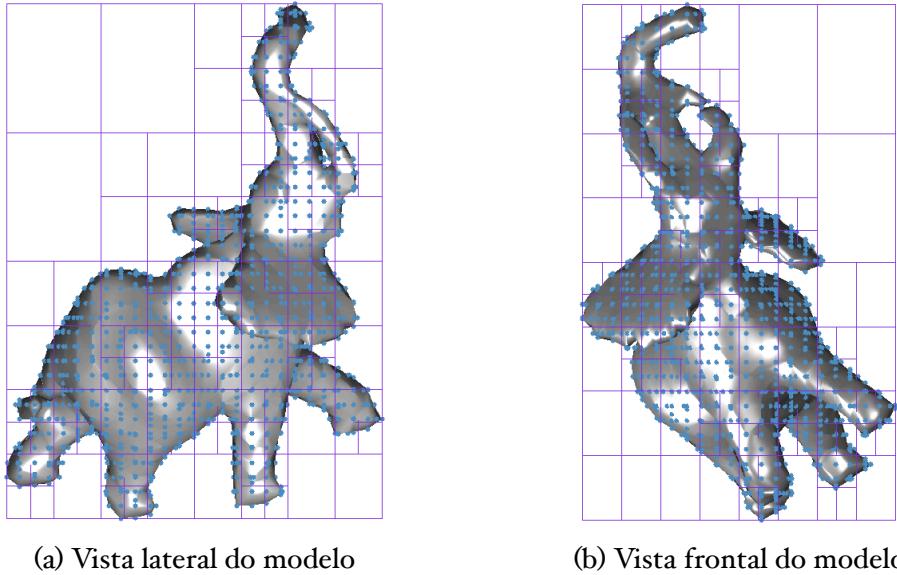


Figura 2.3: Subdivisão do modelo Elephant usando uma *Octree* com no mínimo vinte pontos por octante.

por Bentley (1975). A *kd-Tree* é uma árvore de busca de k dimensões. No caso das nuvens de pontos, a árvore tem 3 dimensões. Partindo da caixa envolvente contendo todos os elementos, recursivamente divide-se as células em regiões com igual número de elementos. A divisão é realizada usando um plano de corte perpendicular a algum eixo coordenado, dividindo a dimensão de maior extensão espacial. Uma vantagem da *kd-Tree* é que consegue gerar uma árvore balanceada de busca. Neste trabalho, usamos uma *kd-Tree* para armazenar os dados e para fazer as consultas das vizinhanças.

Outra estrutura de dados espacial mais genérica é a hierarquia de volumes envolvente (BVH). Proposta por Klosowski et al. (1998), ela agrupa objetos geométricos considerando cada objeto como sendo um nó folha da árvore. Posteriormente, agrupa os nós usando alguma estratégia. Essa estratégia pode ser definida de tal forma que o resultado seja a *Octree* ou a *kd-Tree*.

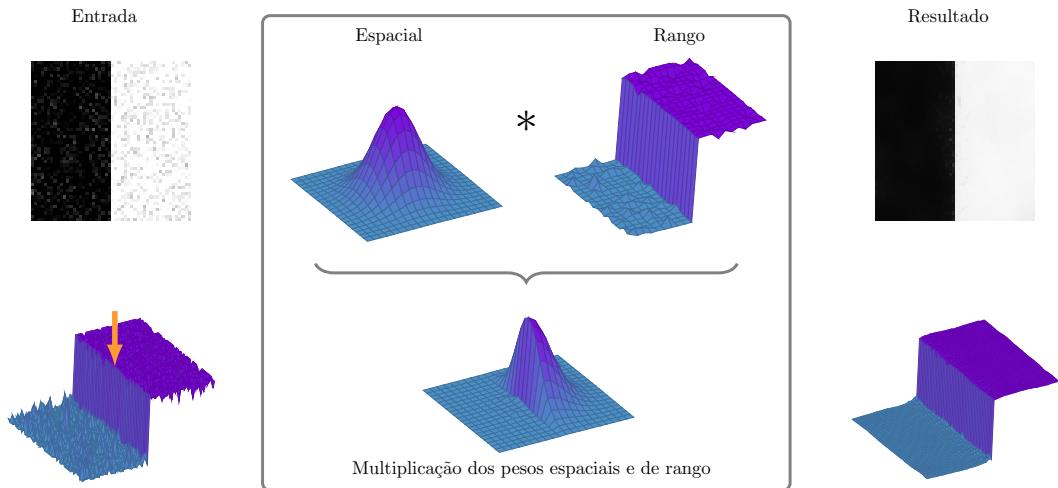


Figura 2.4: A filtragem bilateral consegue suavizar a imagem de entrada preservando as feições (Resultado). Um peso ponderado da vizinhança substitui cada pixel. O peso é composto por um peso espacial, que penaliza a distância, e por um peso de rango, que penaliza a variação da intensidade. Assim, somente pixels próximos com intensidade semelhante contribuem no resultado final. Os pesos mostrados são aplicados ao pixel sob a seta. (Imagen adaptada de Kornprobst & Tumblin (2009))

2.3 Filtragem Bilateral

A filtragem bilateral, originalmente proposta por Tomasi & Manduchi (1998), é um método utilizado para suavizar imagens preservando feições. Esse método baseia-se na filtragem Gaussiana, que substitui cada pixel por uma ponderação dos vizinhos utilizando convolução:

$$p_i = \sum_{p_j \in \mathcal{N}_i} G_\sigma(\|x_i - x_j\|) I_j, \quad (2.1)$$

onde $\|x_i - x_j\|$ representa a distância dos pixels, \mathcal{N}_i é a vizinhança do pixel p_i , I_j é a intensidade da cor do pixel p_j e G_σ é o kernel Gaussiano definido pela função:

$$G_\sigma(x) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{x^2}{2\sigma^2}\right), \quad (2.2)$$

Note-se que a influência de cada pixel p_j vizinho do pixel p_i depende unicamente da distância espacial. A filtragem bilateral trata essa lacuna introduzindo um termo que leva em conta a intensidade da cor. A ideia principal é que um pixel com intensidade similar tenha mais influência que um pixel com intensidade distinta.

Tomasi & Manduchi (1998) definiram a filtragem bilateral:

$$p_i = \frac{1}{W_{p_i}} \sum_{p_j \in \mathcal{N}_i} G_{\sigma_s}(\|x_i - x_j\|) G_{\sigma_r}(|I_i - I_j|) I_j, \quad (2.3)$$

onde G_{σ_s} é o peso espacial, que penaliza a distância entre os pixels. O peso de rango G_{σ_r} penaliza a variação na intensidade dos pixels, representada por $|I_i - I_j|$. Ambos os pesos, são definidos pela função Gaussiana da Equação (2.2). Finalmente, W_{p_i} é o fator de normalização que garante que a soma dos pesos seja 1.0:

$$W_{p_i} = \sum_{p_j \in \mathcal{N}_i} G_{\sigma_s}(\|x_i - x_j\|) G_{\sigma_r}(I_i - I_j) \quad (2.4)$$

e os parâmetros σ_s e σ_r determinam o nível de suavização da imagem. Na Figura 2.4, mostramos como os pesos são calculados para um pixel próximo a uma borda.

A filtragem bilateral foi estendida para malhas simultaneamente por Jones et al. (2003) e por Fleishman et al. (2003). Eles assumem que a malha é localmente plana, ou seja existe um plano que aproxima localmente cada vértice junto com sua vizinhança. Ambos os trabalhos (Fleishman et al., 2003; Jones et al., 2003) definem o peso espacial como a distância entre os vértices da malha. Já o peso de rango é definido de forma diferente em cada trabalho.

Idealmente, o vértice \mathbf{p}_i é igual à projeção dele no plano tangente definido pelo vizinho \mathbf{q}_i , ou seja $\mathbf{p}_i = \pi_{\mathbf{q}_i}(\mathbf{p}_i)$. Por isso, Jones et al. (2003) definem o peso de rango como sendo $G_{\sigma_r}(\|\mathbf{p}_i - \pi_{\mathbf{q}_i}(\mathbf{p}_i)\|)$, assim a influência de um vizinho \mathbf{q}_i cujo plano tangente se encontre longe do ponto \mathbf{p}_i é minimizada. O filtro é definido

como:

$$F(\mathbf{p}_i) = \frac{1}{W_{\mathbf{p}_i} \sum_{\mathbf{q}_i \in \mathcal{N}_i}} a_{\mathbf{q}_i} G_{\sigma_s}(\|\mathbf{p}_i - \mathbf{q}_i\|) G_{\sigma_r}(\|\mathbf{p}_i - \pi_{\mathbf{q}_i}(\mathbf{p}_i)\|), \quad (2.5)$$

no qual $a_{\mathbf{q}_i}$ é a densidade de amostragem.

Outra forma de abordar o fato de que a malha é localmente plana é assumindo que a projeção do vetor diferença entre \mathbf{q}_i e \mathbf{p}_i na normal do vértice \mathbf{p}_i deve ser zero. Ou seja, $\mathbf{p}_i + ((\mathbf{q}_i - \mathbf{p}_i) \cdot \mathbf{n}_i) \mathbf{n}_i = \mathbf{p}_i$. Isto resulta no filtro proposto por Fleishman et al. (2003):

$$F(\mathbf{p}_i) = \frac{1}{W_{\mathbf{p}_i} \sum_{\mathbf{q}_i \in \mathcal{N}_i}} a_{\mathbf{q}_i} G_{\sigma_s}(\|\mathbf{p}_i - \mathbf{q}_i\|) G_{\sigma_r}(|(\mathbf{q}_i - \mathbf{p}_i) \cdot \mathbf{n}_i|) ((\mathbf{q}_i - \mathbf{p}_i) \cdot \mathbf{n}_i). \quad (2.6)$$

Uma discussão sobre as principais diferenças e semelhanças destes dois filtros pode ser encontrada em Kornprobst & Tumblin (2009). Sendo a diferença mais importante que o filtro definido por Fleishman et al. (2003), ao contrário do filtro de Jones et al. (2003), evita que os vértices se movam sobre o plano tangente, já que o vértice \mathbf{p}_i é movido ao longo de sua normal \mathbf{n}_i .

2.4 Suavização em Dois Passos de Malhas Triangulares

As malhas triangulares são compostas de vértices (V) e arestas (A) que formam faces triangulares (F). A suavização de malhas pode ser vista como o processo de deslocar vértices afetados por ruído para uma posição mais próxima da sua posição correta (sem ruído). A atualização dos vértices pode ser feita em um ou dois passos. Os métodos que suavizam a malha em um passo utilizam as posições dos vizinhos para atualizar a posição de cada vértice. Métodos que usam dois passos ajustam primeiro as normais das faces e depois, atualizam os vértices para corresponder com as normais ajustadas.

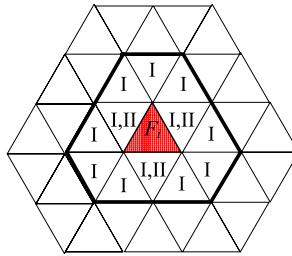


Figura 2.5: Tipos de vizinhança da face F_i : (I) conjunto de faces que compartem um vértice com a face F_i e (II) conjunto de faces que compartilham uma aresta com a face F_i (Sun et al., 2007)

No contexto de malhas, a vizinhança 1-anel de um vértice V_i , denotado \mathcal{N}_{V_i} , é o conjunto de vértices conectados com V_i por uma aresta. E a vizinhança 1-anel da face F_i pode ser definida de duas formas: como o conjunto de faces que compartilham um vértice com a face F_i , denotado \mathcal{N}_{I_i} , ou como o conjunto de faces que compartilham uma aresta com a face F_i , denotado \mathcal{N}_{II_i} (Figura 2.5). O conjunto de faces que compartilham um mesmo vértice V_i é denotado F_{V_i} . E o conjunto de arestas de uma face é denotado ∂F .

As normais em malhas triangulares são estimadas para cada face como o produto vetorial normalizado de duas arestas. A normal da face F_f é:

$$\mathbf{n}_f = \frac{(\mathbf{x}_i - \mathbf{x}_j) \times (\mathbf{x}_k - \mathbf{x}_j)}{\|(\mathbf{x}_i - \mathbf{x}_j) \times (\mathbf{x}_k - \mathbf{x}_j)\|}, \quad (2.7)$$

onde \mathbf{x}_i , \mathbf{x}_j e \mathbf{x}_k são os vértices da face F_f .

A seguir descrevemos os passos envolvidos na suavização em dois passos para malhas triangulares.

2.4.1 Atualização das Normais

Existem vários métodos para atualizar as normais. A maioria deles realizam um processo iterativo, sendo \mathbf{n}_i^l a normal na iteração l , com \mathbf{n}_i^0 a normal estimada como visto na Equação (2.7). Sumarizamos a seguir os principais métodos encontrados na literatura. Uma discussão mais profunda dos métodos pode ser

encontrada em Sun et al. (2007).

Um dos métodos, proposto por Yagou et al. (2002), calcula a nova normal segundo um peso dado pela *média da área das faces*:

$$\mathbf{n}_i^l = \frac{1}{\sum_{j \in \mathcal{N}_{I_i}} A_j} \sum_{j \in \mathcal{N}_{I_i}} A_j \mathbf{n}_j^{l-1} \quad (2.8)$$

onde A_j é a área da face j , e \mathbf{n}_j é a normal da face j .

Outra abordagem, proposta também por Yagou et al. (2002), atualiza a normal da face F_i tomando a mediana dos ângulos entre a normal da face F_i e as normais das faces vizinhas:

$$\mathbf{n}_i^l = \arg \underset{\mathbf{n}_j^{l-1}}{\text{médian}} \{w_j \odot \angle(\mathbf{n}_i^{l-1}, \mathbf{n}_j^{l-1}) : j \in \mathcal{N}_{I_i}\} \quad (2.9)$$

onde $w_j \odot \angle(\mathbf{n}_i^{l-1}, \mathbf{n}_j^{l-1})$, indica que o ângulo no cálculo da mediana se repete w_j vezes, e $\arg \text{médian}_{\mathbf{n}_j^{l-1}} \{\cdot\}$ é a normal \mathbf{n}_j^{l-1} para a qual obteve-se o ângulo médiano. Uma estratégia que pode ser usada no cálculo do peso é atribuir peso 1 a todos os ângulos. Outra estratégia é atribuir peso 2 aos ângulos $\angle(\mathbf{n}_i^{l-1}, \mathbf{n}_j^{l-1}) : j \in \mathcal{N}_{II_i}$ e peso 1 aos demais ângulos de \mathcal{N}_{I_i} .

Outra técnica de atualização de normais, proposta por Yagou et al. (2003), é *alpha-trimming*:

$$\mathbf{n}_i^l = \text{normalise} \left(\sum_{j \in \mathcal{N}_{I_i}} I_\alpha(j) A_j \mathbf{n}_j^{l-1} \right) \quad (2.10)$$

onde $I_\alpha(j)$ é um indicador igual a zero quando o ângulo $\angle(\mathbf{n}_i^{l-1}, \mathbf{n}_j^{l-1})$ encontra-se na proporção α superior ou inferior de todos os ângulos em $\angle(\mathbf{n}_i^{l-1}, \mathbf{n}_j^{l-1})$ e $\text{normalise}(\mathbf{x})$ é o processo de normalizar \mathbf{x} :

$$\text{normalise}(\mathbf{x}) = \frac{\mathbf{x}}{\|\mathbf{x}\|} \quad (2.11)$$

Sun et al. (2007) propõem um método baseado em um *limiar*:

$$\mathbf{n}_i^l = \text{normalise} \left(\sum_{j \in \mathcal{N}_{I_i}} h_j \mathbf{n}_j^{l-1} \right) \quad (2.12)$$

onde h_j é uma função de pesos definida como:

$$h_j = \begin{cases} (\mathbf{n}_i^{l-1} \cdot \mathbf{n}_j^{l-1} - T)^2 & \text{se } \mathbf{n}_i^{l-1} \cdot \mathbf{n}_j^{l-1} > T \\ 0 & \text{se } \mathbf{n}_i^{l-1} \cdot \mathbf{n}_j^{l-1} \leq T \end{cases} \quad (2.13)$$

onde $0 \leq T \leq 1$ é um limiar proposto pelo usuário.

Zheng et al. (2011) apresenta um método baseado na filtragem bilateral para definir o peso das normais na vizinhança \mathcal{N}_{I_i} de F_i :

$$\mathbf{n}_i^l = \text{normalise} \left(\sum_{j \in \mathcal{N}_{I_i}} W_c(\|c_i^{l-1} - c_j^{l-1}\|) W_s(\|\mathbf{n}_i^{l-1} - \mathbf{n}_j^{l-1}\|) \mathbf{n}_j^{l-1} \right) \quad (2.14)$$

onde c_i é o centroide da face F_i , W_c e W_s são funções Gaussianas, W_c é o peso espacial que penaliza a distância entre os centroides e W_s é o peso do ângulo que penaliza a distância das normais das faces vizinhas.

Wang et al. (2012) propõem um método que combina um filtro bilateral com um limiar calculado adaptativamente:

$$\mathbf{n}_i^l = \text{normalise} \left(\sum_{j \in \mathcal{N}_{I_i}} W_c(\|c_i^{l-1} - c_j^{l-1}\|) W_s(\mathbf{n}_i^{l-1}, \mathbf{n}_j^{l-1}) \mathbf{n}_j^{l-1} \right) \quad (2.15)$$

onde W_c é uma função Gaussiana e $W_s(\mathbf{n}_i, \mathbf{n}_j)$ é:

$$W_s(\mathbf{n}_i, \mathbf{n}_j) = \begin{cases} 0 & \text{se } (\mathbf{n}_i - \mathbf{n}_j) \cdot \mathbf{n}_i \geq T \\ [(\mathbf{n}_i - \mathbf{n}_j) \cdot \mathbf{n}_i - T]^2, & \text{caso contrário} \end{cases} \quad (2.16)$$

e T é um limiar calculado adaptativamente:

$$T = \frac{\sqrt{\sum_{j \in \mathcal{N}_{I_i}} [(\mathbf{n}_i - \mathbf{n}_j) \cdot \mathbf{n}_i]}}{\|\mathcal{N}_{I_i}\|} \quad (2.17)$$

2.4.2 Atualização dos Vértices

Algoritmos para atualizar os vértices de modo a corresponder com as normais corrigidas, tem sido propostos. Taubin (2001) propôs um sistema de equações simultâneas baseado na ortogonalidade entre a normal e as arestas de cada face f da malha triangular:

$$\begin{cases} \mathbf{n}_f \cdot (\mathbf{x}_i - \mathbf{x}_j) = 0 \\ \mathbf{n}_f \cdot (\mathbf{x}_k - \mathbf{x}_j) = 0 \quad \forall f = (i, j, k) \\ \mathbf{n}_f \cdot (\mathbf{x}_i - \mathbf{x}_k) = 0 \end{cases} \quad (2.18)$$

Taubin (2001) mostrou que o sistema de equações não tem solução trivial, e propôs resolver as equações no sentido dos mínimos quadrados, minimizando o erro:

$$e(X) = \sum_{k \in F} \sum_{(i,j) \in \partial F_k} (\mathbf{n}_k \cdot (\mathbf{x}_i - \mathbf{x}_j))^2 \quad (2.19)$$

O método do gradiente pode ser utilizado para minimizar o erro, assim, a atualização dos vértices é definida:

$$\mathbf{x}_i^l = \mathbf{x}_i + \lambda \sum_{j \in N_{v_i}} \sum_{(i,j) \in \partial F_k} \mathbf{n}_k^{l-1} (\mathbf{n}_k^{l-1} \cdot (\mathbf{x}_j^{l-1} - \mathbf{x}_i^{l-1})) \quad (2.20)$$

onde $\lambda > 0$ é o tamanho do passo na iteração.

Ohtake et al. (2001) propuseram utilizar o método do gradiente para minimizar

o erro levando em conta a área de cada face:

$$e_1(X) = \sum_{k \in F} \sum_{(i,j) \in \partial F_k} A_k (\mathbf{n}_k \cdot (\mathbf{x}_i - \mathbf{x}_j))^2 \quad (2.21)$$

com passo $\lambda = 1/6 \sum_{k \in \partial F_k} A_k$

Assim, a equação de atualização dos vértices é:

$$\mathbf{x}^l = \mathbf{x}^{l-1} + \frac{1}{3 \sum_{k \in F_{V_i}} A_k} \sum_{j \in \mathcal{N}_{V_i}} \sum_{(i,j) \in \partial F_k} A_k \mathbf{n}_k (\mathbf{n}_k \cdot (\mathbf{x}_j^{l-1} - \mathbf{x}_i^{l-1})) \quad (2.22)$$

Sun et al. (2007), observaram que numa face com área grande geralmente os vértices encontram-se afastados, então, a influência da face na atualização do vértice deve ser menor. Por isso, propuseram uma modificação do algoritmo de Ohtake et al. (2001), substituindo os pesos dados pelas áreas das faces por um:

$$\mathbf{x}^l = \mathbf{x}^{l-1} + \frac{1}{3 \sum_{k \in F_{V_i}} \sum_{j \in \mathcal{N}_{V_i}} (i,j) \in \partial F_k} \sum_{j \in \mathcal{N}_{V_i}} \sum_{(i,j) \in \partial F_k} \mathbf{n}_k (\mathbf{n}_k \cdot (\mathbf{x}_j^{l-1} - \mathbf{x}_i^{l-1})) \quad (2.23)$$

Capítulo 3

Suavização de Superfícies Representadas por Nuvens de Pontos

3.1 Considerações Iniciais

Muitos métodos tem sido propostos para suavizar superfícies tanto para malhas quanto para nuvens de pontos. Apesar disso, a pesquisa nesta área continua ativa pois suavizar superfícies mantendo as feições (curvas nas superfícies que contém as características visuais mais proeminentes) é ainda um problema sem solução definitiva.

O objetivo da suavização de superfícies é remover ruído, mantendo as feições subjacentes, tanto quanto possível. Muitos métodos desenvolvidos para malhas de triângulos inspiraram variantes no contexto de superfícies representadas por nuvens de pontos, que é o foco principal do trabalho aqui desenvolvido. Especificamente, no contexto de superfícies representadas por nuvens de pontos, as técnicas de suavização tem experimentado um progresso importante na última década. As técnicas existentes, variam consideravelmente quanto à base matemática, abrangendo metodologias derivadas da teoria espectral, mapa de difusão, operadores de projeção e filtros bilaterais.

Neste capítulo apresentamos as principais técnicas para suavização de superfí-

cies representadas por nuvens de pontos encontradas na literatura.

3.2 Técnicas Espectrais

Técnicas baseadas em filtros espectrais para imagens inspiraram variantes no contexto de superfícies definidas por nuvens de pontos. Assim, foram propostas variações de técnicas existentes, impulsadas principalmente pelo desenvolvimento de métodos robustos para discretizar o operador Laplace-Beltrami em nuvens de pontos.

Pauly et al. (2002) foram uns dos pioneiros em utilizar os operadores de Laplace para realizar suavização de nuvens de pontos, manipulando coeficientes espectrais e realizando um processo de difusão. O processo de difusão de Pauly et al. pode ser descrito como:

$$\frac{\partial \mathcal{S}}{\partial t} = \lambda \Delta \mathcal{S}, \quad (3.1)$$

onde Δ denota o operador Laplace-Beltrami na superfície. Usando integração Euleriana, chegamos na suavização:

$$\mathbf{p}_i^l = \mathbf{p}_i^{l-1} + \lambda dt \Delta \mathbf{p}_i, \quad (3.2)$$

onde $\Delta \mathbf{p}_i$ é alguma discretização do operador Laplace-Beltrami no ponto \mathbf{p}_i .

A abordagem usada por Pauly et al. (2002) cria uma descomposição da nuvem de pontos usando o operador *umbrella* como mecanismo de discretização:

$$\Delta \mathbf{p}_i = \frac{1}{\Omega} \sum_{j \in \mathcal{N}} w_j (\mathbf{p}_j - \mathbf{p}_i), \quad (3.3)$$

onde w_j define o peso do ponto \mathbf{p}_j na vizinhança de \mathbf{p}_i . Se o peso é um, o operador é conhecido como *umbrella* uniforme. O peso também pode ser definido em relação a distância do ponto \mathbf{p}_i ao vizinho \mathbf{p}_j : $w_j = 1/\|\mathbf{p}_j - \mathbf{p}_i\|$. Ω é o termo de

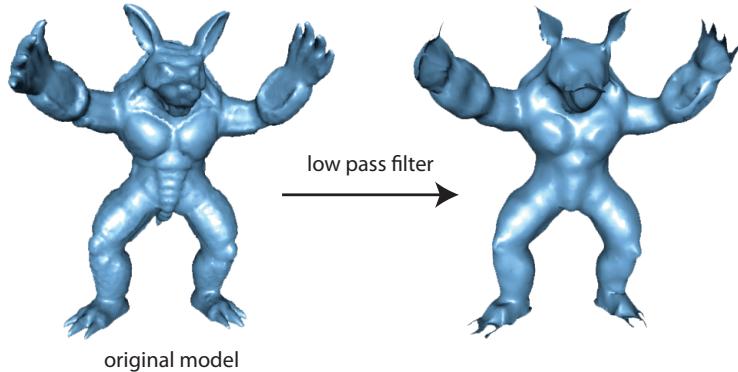


Figura 3.1: Suavização do modelo Armadillo utilizando um filtro passa baixo na decomposição espectral do operador Laplace-Beltrami baseado em SPH. O processo não consegue preservar feições (extraído de Petronetto et al. (2013)).

normalização:

$$\Omega = \frac{1}{\sum_{j \in \mathcal{N}} w_j} \quad (3.4)$$

Um problema conhecido da discretização *umbrella* do operador Laplace-Beltrami é o encolhimento do modelo, ou seja, a perda de volume. Algumas técnicas foram propostas para resolver esse problema, Desbrun et al. (1999) calculam o fator de encolhimento e redimensionam a superfície após cada iteração. Pauly et al. (2002) calculam o fator de encolhimento localmente e deslocam os pontos para compensar a perda de volume.

Lange & Polthier (2005) utilizaram uma versão anisotrópica do operador Laplace para detectar a curvatura média. Seu método soluciona uma equação de derivadas parciais (EDP) com condições de borda. A abordagem consegue suavizar uma nuvem de pontos além de preservar e realçar feições durante o processo de suavização. Uma desvantagem desse método é que precisa de muitas iterações e de um parâmetro definido pelo usuário conhecido como quociente de borda que permite realçar as regiões com alta curvatura.

Belkin et al. (2009) propuseram uma discretização do operador Laplace-Beltrami para nuvens de pontos. Para isso, construíram um *patch* ao redor de cada ponto usando uma triangulação Delaunay local, empregando um kernel de calor em cada *patch*. Provaram a convergência do operador Laplace-Beltrami proposto. Uma limitação desse método é que precisa de uma densidade alta na nuvem de pontos. Além disso, a construção local de uma triangulação aumenta a complexidade do processo.

Recentemente, Petronetto et al. (2013) exploraram as propriedades espectrais da discretização do operador Laplace-Beltrami baseado em SPH. A vantagem desse método é que não precisa de uma malha nem localmente, diferindo de métodos propostos anteriormente. Petronetto et al. conseguiram suavizar nuvens de pontos utilizando a decomposição espectral do operador, mas, sem preservação de feições (Figura 3.1).

3.3 Técnicas de Projeção

As técnicas de projeção estão inspiradas no método de aproximação *Moving Least Squares* (MLS), que está baseado no método de mínimos quadrados (LS). O método LS para superfícies de pontos busca encontrar uma função linear $f(\mathbf{x}_i) = c_0 + c_1 \mathbf{x}$ que aproxime a superfície. Para isso, busca encontrar os coeficientes c_0, c_1 que minimizem a soma dos quadrados das diferenças entre os valores da função e dos dados originais:

$$\min_{c_0, c_1} \left(\sum (\mathbf{p}_i - f(\mathbf{x}_i))^2 \right), \quad (3.5)$$

onde \mathbf{p}_i são os dados originais da nuvem de pontos.

Enquanto o método de mínimos quadrados resulta numa solução global, *Moving Least Squares* (MLS) aproxima os dados localmente. O método MLS foi proposto originalmente por Levin (1998) e surgiram muitas variações que podem ser classificadas em superfícies implícitas e superfícies de projeções.

Proposto inicialmente por Alexa et al. (2003), os operadores de projeção con-

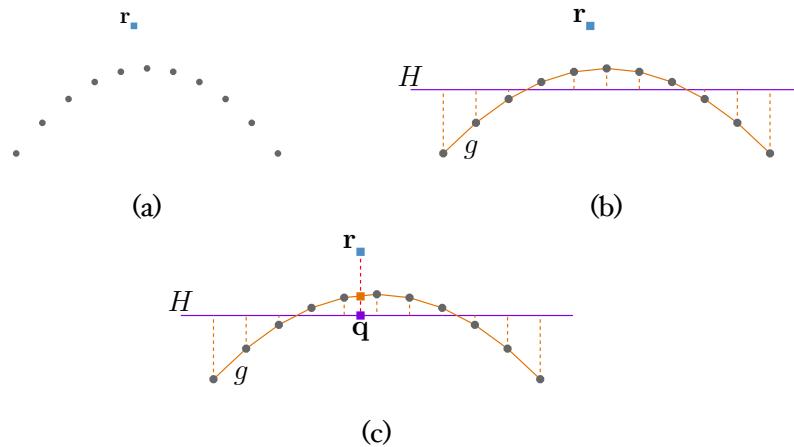


Figura 3.2: Processo de projeção do MLS. (a) Primeiro, um plano H é aproximado ao conjunto de pontos usando mínimos quadrados. (b) A projeção do ponto r no plano H define a origem q do domínio local de referencia. Utilizando esse domínio, um polinômio g é aproximado, novamente usando mínimos quadrados. (c) Finalmente, o ponto é projetado no polinômio g .

seguem suavizar as nuvens de pontos. A ideia principal é definir um esquema de projeção de forma que um ponto próximo à nuvem de pontos seja projetado em uma representação suave da superfície. Tipicamente, o esquema de projeção pode ser dividido em dois passos: encontrar um plano e depois aproximar um polinômio (Figura 3.2).

No primeiro passo, calcula-se um plano $H = \{\mathbf{x} \mid (\mathbf{n} \cdot \mathbf{x}) - D = 0, \mathbf{x} \in \mathbb{R}^3\}$, com origem em D e normal $\mathbf{n} \in \mathbb{R}^3$, $\|\mathbf{n}\| = 1$, associada ao ponto $\mathbf{r} \in \mathbb{R}^3$. H é calculado minimizando a soma ponderada do quadrado das distâncias dos pontos \mathbf{p}_i ao plano. Os pesos pertencentes a \mathbf{p}_i são funções da distância de \mathbf{p}_i à projeção de \mathbf{r} no hiperplano H :

$$\sum_{i=1}^n ((\mathbf{n} \cdot \mathbf{p}_i) - D)^2 w(\|\mathbf{p}_i - \mathbf{q}\|), \quad (3.6)$$

onde \mathbf{q} a projeção de \mathbf{r} em H e w é uma função monotonicamente decrescente.

O sistema local de coordenadas definido por H é usado para calcular uma apro-

ximação polinomial local da superfície na vizinhança de \mathbf{r} . Os coeficientes são calculados minimizando:

$$\sum_{i=1}^n (g(\mathbf{x}_i, \mathbf{y}_i) - f_i)^2 w(\|\mathbf{p}_i - \mathbf{q}_i\|), \quad (3.7)$$

onde $(\mathbf{x}_i, \mathbf{y}_i)$ é a representação de \mathbf{q}_i no sistema local de coordenadas, \mathbf{q}_i é a projeção de \mathbf{p}_i em H e f_i é a altura de \mathbf{p}_i sobre H .

Diferentes enfoques foram propostos para controlar cada passo. Guennebaud & Gross (2007) propuseram usar ajustes não polinomiais no segundo passo do esquema. Para isso, eles propuseram uma abordagem algébrica para ajustar uma esfera em vez do plano H . A solução algebrica para encontrar o vetor de coeficientes $\mathbf{u}(\mathbf{r}) = [u_0, \dots, u_4] \in \mathbb{R}^5$ que definem a 0-isosuperfície de uma esfera para um ponto $\mathbf{r} \in \mathbb{R}^3$ pode ser expressa como:

$$\mathbf{u}(\mathbf{r}) = \arg \min_{\mathbf{u}, \mathbf{u} \neq 0} \|\mathbf{W}^{1/2}(\mathbf{r}) \mathbf{D} \mathbf{u}\|^2, \quad (3.8)$$

onde \mathbf{W} é a matriz diagonal de pesos $n \times n$ e \mathbf{D} é a matriz $n \times 5$ definida:

$$\mathbf{D} = \begin{bmatrix} 1 & \mathbf{p}_0^\top & \mathbf{p}_0^\top \mathbf{p}_0 \\ \vdots & \vdots & \vdots \\ a & \mathbf{p}_{n-1}^\top & \mathbf{p}_{n-1}^\top \mathbf{p}_{n-1} \end{bmatrix}. \quad (3.9)$$

Para evitar a solução trivial $\mathbf{u} = 0$, usaram a restrição de Pratt que fixa a norma do gradiente na superfície da esfera unitária como sendo um.

Pela natureza de suavização do MLS, as técnicas descritas anteriormente não conseguem solucionar o problema de manter as feições afiadas. Porém, as superfícies podem ser vistas como compostas de *patches* separados por feições. Portanto, pontos pertencentes a outros *patches* podem ser considerados *outliers*. Com essa ideia, Fleishman et al. (2005) propuseram uma técnica baseadas em estatística robusta que consegue lidar tanto com feições afiadas quanto com *outliers* reais. Uma desvantagem desse método, além do custo computacional, é que ele requer uma alta densidade de pontos na nuvem. Por outro lado, considerar as fei-

ções afiadas como *outliers* não permite um modelo estatístico específico para modelar as feições. Daniels et al. (2007); Lipman et al. (2007a) conseguiram modelar as feições afiadas com maior flexibilidade, porém no contexto de reconstrução de superfícies e não para suavização.

Öztireli et al. (2009), baseados em técnicas de estatística robusta, conseguiram suavizar as nuvens mantendo as feições sem necessidade de uma segmentação previa. Especificamente, mostraram a relação entre MLS e regressão local via *kernel* (LKR).

3.4 Técnicas Baseadas na Norma l_1

A origem dos métodos baseados na norma l_1 é o algoritmo de Weiszfeld para a solução do problema de localização de pontos de Fermat-Weber. O problema, também conhecido como a mediana multivariada l_1 , é utilizada em dados multivariados para gerar uma representação de grandes conjuntos de amostras na presença de ruído e *outliers*. O problema foi inicialmente conhecido como o problema de Weber & Friedrich (1962) e o objetivo era encontrar uma localização ótima para um local industrial minimizando a soma das distâncias a locais existentes. O problema foi depois rastreado ao trabalho de Fermat no século XVII e assim foi conhecido como o problema de Fermat-Weber.

Dado um conjunto de dados, a mediana l_1 é definida como o ponto \mathbf{q} que minimiza a soma das distâncias Euclidianas aos pontos:

$$\mathbf{q} = \arg \min_{\mathbf{x}} \left\{ \sum_{i=1}^n \|\mathbf{p}_i - \mathbf{x}\| \right\}, \quad (3.10)$$

onde \mathbf{p}_i são os dados originais da nuvem de pontos e \mathbf{x}_i é um conjunto arbitrário de pontos.

Lipman et al. (2007b) desenvolveram um operador de projeção livre de parametrizações localmente ótimo (LOP) que está relacionado com a mediana l_1 . O objetivo é encontrar um conjunto de pontos $\mathcal{Q} = \{\mathbf{q}_j\}$ que aproximem a geome-

tria subjacente da nuvem de pontos \mathcal{P} :

$$\mathcal{Q} = G(\mathcal{Q}), \quad (3.11)$$

onde:

$$\begin{aligned} G(C) &= \arg \min_{x \in \{x_j\}} \{E_1(X, P, C) + E_2(X, C)\}, \\ E_1(X, P, C) &= \sum_{j=1}^m \sum_{i=1}^n \|\mathbf{x}_j - \mathbf{p}_i\| \theta(\|\mathbf{c}_j - \mathbf{p}_i\|), \\ E_2(X, C) &= \sum_{j=1}^m \lambda_j \sum_{i=1, i \neq j}^n \eta(\|\mathbf{x}_j^l - \mathbf{c}_j\|) \theta(\|\mathbf{c}_j^l - \mathbf{c}_j\|), \end{aligned} \quad (3.12)$$

onde θ e η são funções decrescentes:

$$\theta(r) = e^{-r/(h/4)^2}, \quad \eta(r) = \frac{1}{3r^3}, \quad (3.13)$$

onde h é o raio de suporte definido pelo usuário.

Na Equação (3.12), E_1 minimiza a soma ponderada das distâncias dos pontos \mathbf{q}_j aos pontos \mathbf{p}_i em relação aos pesos radiais com centro no conjunto de pontos \mathcal{Q} . E_1 pode ser vista como uma versão local da função de custo 3.10. E_2 busca que os pontos \mathbf{q}_j não estejam muito próximos entre si.

Finalmente, dado um parâmetro de repulsão $\mu \in [0, 1/2)$, definem um processo iterativo para atualizar as posições de cada ponto:

$$\mathbf{x}_{j'}^l = \sum_{i=1}^n \mathbf{p}_i \frac{\alpha_i^{j'}}{\sum_{i=1}^n \alpha_i^{j'}} + \mu \sum_{j'=1, j' \neq j}^m (\mathbf{x}_{j'}^{l-1} - \mathbf{x}_j^{l-1}) \frac{\beta_j^{j'}}{\sum_{j'=1, j' \neq j}^m \beta_j^{j'}}, \quad (3.14)$$

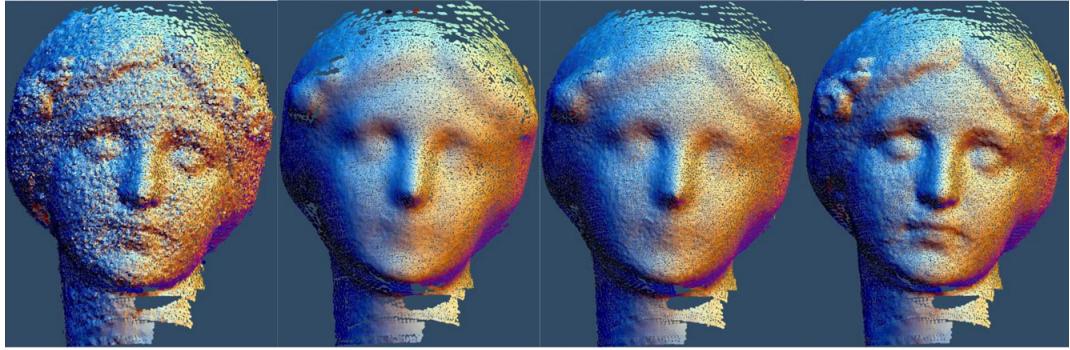


Figura 3.3: Comparação das técnicas baseadas na norma l_1 . De esquerda a direita: nuvem de entrada, resultados de LOP, WLOP e FLOP (extraído de Liao et al. (2013)).

onde $\alpha_i^{j'}$ e $\beta_j^{j'}$ são respectivamente:

$$\begin{aligned} \alpha_i^{j'} &= \frac{\theta(\|\mathbf{x}_{j'}^{l-1} - \mathbf{p}_i\|)}{\|\mathbf{x}_{j'}^{l-1} - \mathbf{p}_i\|}, \\ \beta_j^{j'} &= \frac{\theta(\|\mathbf{x}_{j'}^{l-1} - \mathbf{x}_{j'}^{l-1}\|)}{\|\mathbf{x}_{j'}^{l-1} - \mathbf{x}_{j'}^{l-1}\|} \left| \frac{\partial \eta}{\partial r}(\|\mathbf{x}_{j'}^{l-1} - \mathbf{x}_j^{l-1}\|) \right|. \end{aligned} \quad (3.15)$$

Huang et al. (2009) propuseram uma otimização ponderada localmente ótima (WLOP). Observaram que para nuvens não uniformes, LOP tende a gerar projeções com acumulação de pontos onde a nuvem original é mais densa. Para solucionar esse problema, modificaram o processo iterativo para atualizar os pontos proposto por Lipman et al. (2007b) (3.14):

$$\mathbf{x}_j = \sum_{i=1}^n \mathbf{p}_i \frac{\alpha_i^{j'}/v_j}{\sum_{i=1}^n (\alpha_i^{j'}/v_j)} + \mu \sum_{j'=1, j' \neq j}^m (\mathbf{x}_{j'}^{l-1} - \mathbf{x}_j^{l-1}) \frac{w_{i'}^{l-1} \beta_j^{j'}}{\sum_{j'=1, j' \neq j}^m (w_{i'}^{l-1}, \beta_j^{j'})}, \quad (3.16)$$

onde v_i, w_j^l são os pesos de densidade:

$$\begin{aligned} v_i &= 1 + \sum_{i'=1, i' \neq i}^n \theta \|\mathbf{p}_i - \mathbf{p}_{i'}\| \\ w_j^l &= 1 + \sum_{j'=1, j' \neq j}^m \theta (\|\mathbf{x}_{j'}^{l-1} - \mathbf{x}_j^{l-1}\|) \end{aligned} \quad (3.17)$$

Além disso, propuseram mudar o termo de repulsão (3.13), de modo a produzir uma distribuição mais regular:

$$\eta(r) = -r. \quad (3.18)$$

Liao et al. (2013) observaram que ambos os métodos, LOP e WLOP, além de ser computacionalmente complexos, não conseguem preservar feições afiadas (Figura 3.3). Inspirados nos filtros bilaterais, propuseram uma projeção localmente ótima que consiga preservar feições (FLOP). Para isso, mudaram E_1 utilizado em LOP (3.12):

$$E_1(X, P, C) = \sum_{j=1}^m \sum_{i=1}^n \|\mathbf{x}_j - \mathbf{p}_i\| \theta_s(\|\mathbf{c}_j - \mathbf{p}_i\|) \theta_r(\mathbf{n}_j \cdot (\mathbf{c}_j - \mathbf{p}_i)), \quad (3.19)$$

onde o peso $\theta_r(r) = e^{-r^2/2\sigma_r^2}$ penaliza a variação na geometria, conseguindo preservar feições afiadas.

Outra desvantagem de LOP e WLOP, é que o raio de suporte h tem que ser fornecido pelo usuário por tentativa e erro. Por isso, Liao et al. (2013) propuseram um raio de suporte localmente adaptativo:

$$h_j = \frac{\sum_{i=1}^n \theta_c(\mathbf{n}_j \cdot (\mathbf{c}_j - \mathbf{p}_i)) \theta_s(\|\mathbf{c}_j - \mathbf{p}_i\|)}{S_i} R, \quad (3.20)$$

onde $\theta_c(r) = e^{-r^2/2\sigma_c^2}$ e $\theta_s(r) = e^{-r^2/2\sigma_s^2}$, são as Gaussianas padrão. A constante

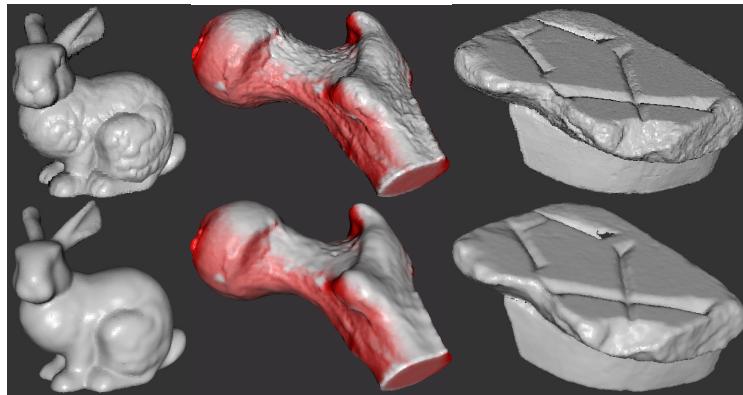


Figura 3.4: Suavização de nuvens de pontos usando elementos finitos. Na fila superior mostram-se as nuvens de entrada e na fila inferior, as nuvens suavizadas. A abordagem não consegue preservar feições (extraído de Clarenz et al. (2004a)).

R define o raio de influência para cada ponto \mathbf{c}_i , S_i é o número de pontos na vizinhança.

3.5 Outras Técnicas

Clarenz et al. (2004a,b) apresentaram uma abordagem para suavizar nuvens de pontos baseada em equações diferenciais parciais (EDPs) via elementos finitos. Propuseram construir matrizes locais baseados em triangulações de Delaunay para, posteriormente, montar uma matriz global que permite a discretização das EDPs. Depois disso, realiza um processo iterativo de difusão para solucionar um sistema de equações anisotrópicas e calcular a superfície suavizada. Porém, no processo, perdem-se as feições afiadas (Figura 3.4).

Outras técnicas para suavizar nuvens de pontos, baseadas na média não-local (Buades et al., 2005), foram propostas por Guillemot et al. (2012); Schall et al. (2007). Schall et al. (2007) propõem um filtro bilateral, onde um dos pesos penaliza a distância dos pontos e o outro peso, chamado peso de similaridade, compara a semelhança entre as vizinhanças geométricas.

Capítulo 4

Filtragem de Nuvens de Pontos por Correção de Normais e suas Variantes

4.1 Considerações Iniciais

A metodologia proposta para remover ruído da nuvem de pontos é composta de três passos: estimação de normais, filtragem de normais e atualização de pontos. O primeiro passo fornece uma normal estimada para cada ponto. Posteriormente realizamos um passo de filtragem de normais, pois a normal estimada pode ser afetada por ruído presente nas coordenadas dos pontos. Finalmente, atualizamos

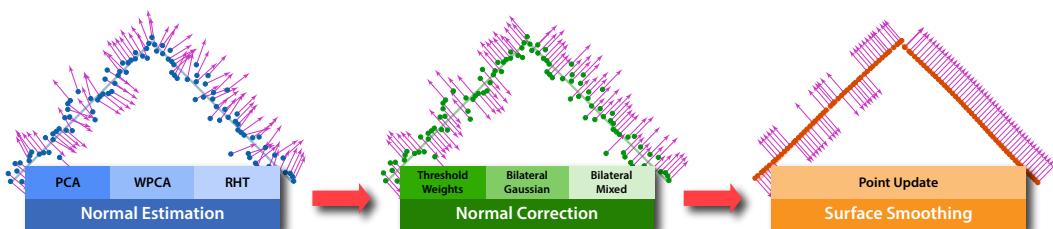


Figura 4.1: *Pipeline* do processo de filtragem de nuvens de pontos e as variantes.

as posições dos pontos para corresponder com as normais filtradas.

Em cada um desses passos, existem vários métodos que podem ser usados. Os passos de nossa abordagem e os métodos usados são ilustrados na Figura 4.1. Realizamos iterações nos passos de filtragem de normais e atualização dos pontos. Temos duas opções para realizar essas duas etapas, podemos realizar l iterações de ambos os passos, um após o outro (Algoritmo 1). Também podemos realizar l iterações no passo de filtragem de normais e posteriormente m iterações no passo de atualização dos pontos (Algoritmo 2). Apesar de aumentar um parâmetro no processo, adotamos a segunda opção, já que desta forma a atualização dos pontos é baseada nas normais corrigidas (que espera-se que sejam ótimas).

Input: l

for $i = 0$ **to** l **do**

 Realizar um passo de filtragem de normais;

 Realizar um passo de atualização dos pontos;

end

Algorithm 1: Alternativa de implementação do algoritmo de dois passos.

Input: l, m

for $i = 0$ **to** l **do**

 Realizar um passo de filtragem de normais;

end

for $i = 0$ **to** m **do**

 Realizar um passo de atualização dos pontos;

end

Algorithm 2: Alternativa de implementação do algoritmo de dois passos.

Neste capítulo apresentamos os detalhes da fundamentação teórica e computacional usadas para implementar cada passo do processo de filtragem. Na Secção §4.2 apresentamos três diferentes mecanismos que podem ser usados para estimar as normais em uma nuvem de pontos. Na Secção §4.3 fornecemos uma descrição das três abordagens usadas para filtrar as normais. E na Secção §4.4 descrevemos o método proposto para realizar a atualização dos pontos.

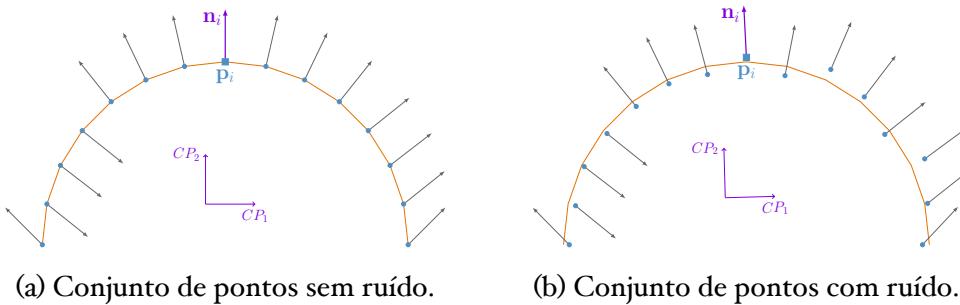


Figura 4.2: Estimação da normal \mathbf{n}_i para o ponto \mathbf{p}_i usando PCA. São mostradas as componentes principais CP_1 e CP_2 .

4.2 Estimativa de Normais

O problema de estimativa de normais, descrito em Secção §2.2.1, visa ter uma normal \mathbf{n}_i para cada \mathbf{p}_i . Investigamos três métodos para estimar o conjunto de normais: análise de componentes principais (PCA), análise de componentes principais ponderada (WPCA) e transformada de Hough aleatória (RHT).

4.2.1 Análise de componentes principais (PCA)

Quando tem-se uma superfície suficientemente densa, a nuvem de pontos reflete a estrutura da superfície. Assim, espera-se que a vizinhança dos pontos esteja distribuída próximo ao plano tangente. Motivado nisso, uma das aplicações do PCA é a estimativa de normais. Pearson (1901) mostrou que a última componente principal de um conjunto de pontos é ortogonal ao plano que melhor aproxima o conjunto. Assim, para estimar a normal \mathbf{n}_i calculamos as componentes principais do conjunto de pontos na vizinhança de \mathbf{p}_i , de modo que a última componente principal corresponde à normal \mathbf{n}_i .

Na Figura 4.2, é mostrado um exemplo de estimativa de normais usando PCA para dois conjuntos de pontos amostrados em uma semi-circunferência: o conjunto sem ruído na Figura 4.2a, e, na Figura 4.2b, o conjunto com ruído.

Para calcular as componentes principais, construímos uma matriz de covariâ-

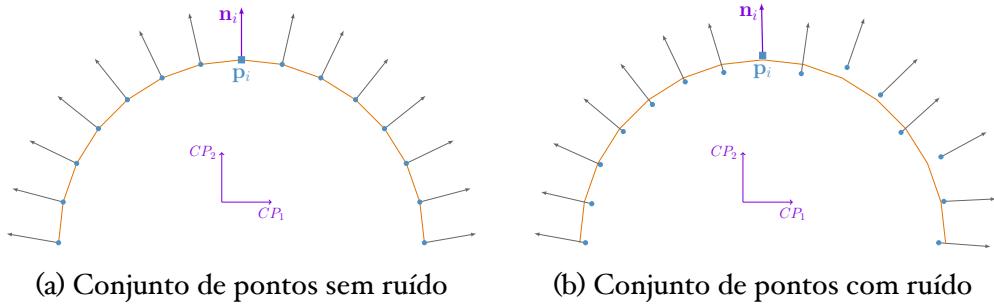


Figura 4.3: Estimação da normal \mathbf{n}_i para o ponto \mathbf{p}_i usando WPCA. São mostradas as componentes principais CP_1 e CP_2 .

cia para os pontos \mathbf{p}_j na vizinhança de \mathbf{p}_i . De modo que, a última componente principal corresponde à menor covariância (o autovetor associado ao menor autovetor).

Os k vizinhos mais próximos de \mathbf{p}_i definem a vizinhança usada para construir a matriz de covariância. O tamanho da vizinhança, dependendo do nível de ruído influencia a normal estimada. Por isso, fazemos testes variando o tamanho da vizinhança, os quais são discutidos no Capítulo 5.

4.2.2 Análise de componentes principais ponderada (WPCA)

WPCA é uma extensão de PCA e também pode ser usado para estimar normais. WPCA é similar a PCA, salvo que uma função de pesos é usada para definir a contribuição de cada ponto na construção da matriz de covariância. Mais especificamente, a matriz de covariância \mathbf{C}_i associada ao ponto \mathbf{p}_i é definida por

$$\mathbf{C}_i = \mathbf{X} \mathbf{W} \mathbf{X}^\top, \quad (4.1)$$

onde \mathbf{X} é a matriz com colunas formadas pelas coordenadas dos pontos na vizinhança de \mathbf{p}_i e \mathbf{W} é uma matriz diagonal com entradas w_{jj} diferentes de zero, correspondendo ao peso associado ao ponto \mathbf{p}_j .

Em nossa implementação, o peso associado a cada ponto \mathbf{p}_j é dado pela inversa da distância euclidiana entre \mathbf{p}_i e \mathbf{p}_j . Apesar do bom desempenho do WPCA em

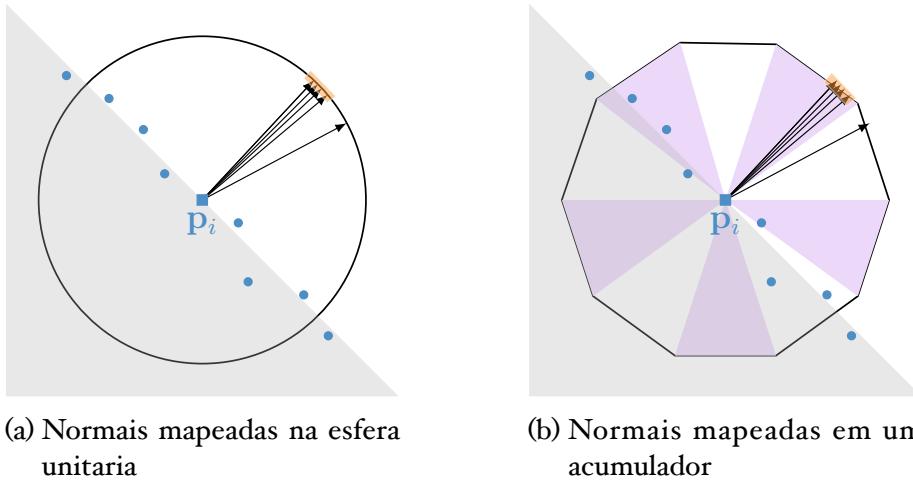


Figura 4.4: Normais dos planos selecionados aleatoriamente na vizinhança de \mathbf{p}_i .

aplicações como agrupamento ou classificação, não foi encontrado na literatura trabalhos que a usem para estimar normais.

Na Figura 4.3, é mostrado um exemplo de estimação de normais usando WPCA. Ilustramos dois conjuntos de pontos amostrando uma semi-esfera: o conjunto sem ruído na Figura 4.3a e o conjunto com ruído na Figura 4.3b.

4.2.3 Transformada de Hough aleatória (RHT)

O uso da RHT foi proposto por Boulch & Marlet (2012) como um mecanismo robusto para estimação de normais. A ideia é selecionar T tríades de pontos na vizinhança de cada ponto \mathbf{p}_i . Cada tríade define um plano e cada plano tem uma normal unitária com origem no ponto \mathbf{p}_i . Assim, para cada ponto temos um conjunto de T normais (Figura 4.4).

As esferas unitárias com centro em cada \mathbf{p}_i são discretizadas em *bins* (Figura 4.4b). Após, contamos (processo de votação) o número de normais que caem em cada *bin*. Finalmente, se seleciona o *bin* com mais votos.

No caso dos pontos em \mathbb{R}^3 , a discretização da normal é obtida transformando-a para o espaço de Hough. Boulch & Marlet (2012) propõem usar o acumulador

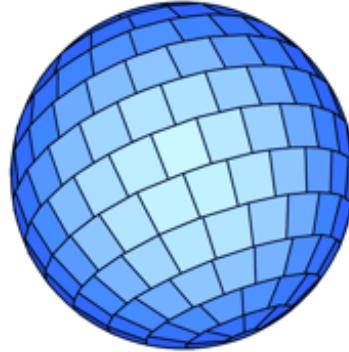


Figura 4.5: Acumulador esférico de Borrman, com n_θ fatias e $2n_\phi$ bins no equador (extraído de (Boulch & Marlet, 2012)).

esférico de Borrman (Figura 4.5).

O número de planos a serem selecionados para que a normal estimada seja confiável é definido por:

$$T = \frac{1}{2\delta^2} \log \left(\frac{2M}{1 - \alpha} \right) ,$$

onde α é a probabilidade mínima tolerada, tal que a distância entre a distribuição teórica e a distribuição observada seja no máximo δ . Assim, por exemplo, sendo $\delta = 0.07$ e $\alpha = 0.95$ e $n_\phi = 5$, temos 23 bins ($M = 23$):

$$T_R = \frac{1}{2(0.07)^2} \ln \left(\frac{2(23)}{1 - 0.95} \right) \approx 700$$

Este número pode ser diminuído, deixando de selecionar planos quando voltamos repetidamente pelo mesmo bin. Ou seja, quando não existir interseção entre os intervalos de confiança de dois bins. Isto é, quando a diferença entre os intervalos de confiança é maior que $2\sqrt{1/T}$ (Figura 4.6).

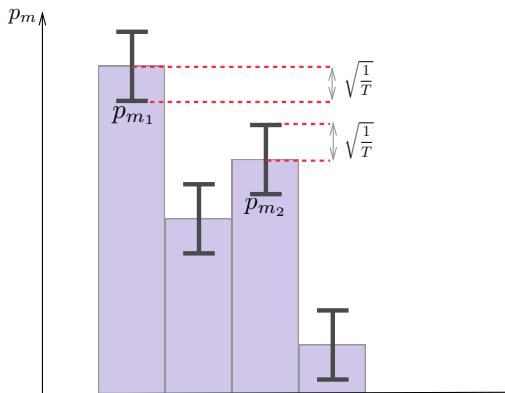


Figura 4.6: Os intervalos de confiança dos bins mais votados (p_{m_1} e p_{m_2}) não se intersetam.

4.3 Correção de Normais

As normais estimadas, usando qualquer um dos métodos na seção anterior, podem não ser suaves devido ao ruído no conjunto de pontos. Por isso, o segundo passo de nosso *pipeline* é o passo de filtragem de normais, com o objetivo de obter um novo campo de normais filtradas $N' = \{\mathbf{n}_i^l \in \mathbb{R}^3, \|\mathbf{n}_i^l\| = 1\}$. Usamos um enfoque iterativo, análogo ao usado em malhas, para filtrar as normais. Mais especificamente, calculamos a média ponderada das normais na vizinhança do ponto \mathbf{p}_i :

$$\mathbf{n}_i^l = \frac{\left(\sum_{j \in \mathcal{N}_i} w_j \mathbf{n}_j^{l-1} \right)}{\left\| \left(\sum_{j \in \mathcal{N}_i} w_j \right) \right\|}, \quad (4.2)$$

onde \mathcal{N}_i considera os k vizinhos mais próximos de \mathbf{p}_i , \mathbf{n}_i^l é a normal obtida na iteração l , sendo $\mathbf{n}_i^0 \in N$ a normal estimada, e w_j é o peso usado para definir a contribuição de cada normal. Sendo $\left\| \left(\sum_{j \in \mathcal{N}_i} w_j \right) \right\|$ o fator de normalização.

Os métodos de filtragem de normais diferem principalmente na escolha dos pesos w_j . Investigamos, três métodos para filtrar as normais, um deles baseado em um esquema de limiarização, outro baseado em um filtro bilateral e o terceiro

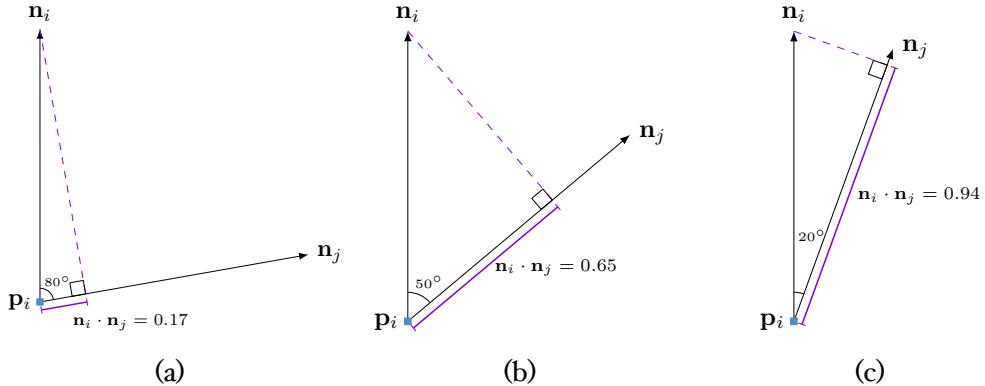


Figura 4.7: Projeção da normal \mathbf{n}_i em \mathbf{n}_j , dada por $\mathbf{n}_i \cdot \mathbf{n}_j$ para três ângulos diferentes: (a) 80° , (b) 50° e (c) 20° .

baseado em um esquema misto, isto é, uma filtragem bilateral usando limiares.

4.3.1 Pesos Limiarizados

Sun et al. (2007) propuseram um esquema iterativo simples para filtrar as normais. Este esquema é baseado em um limiar para definir o peso w_i na Equação (4.2). O peso é calculado da seguinte forma:

$$w_j = \begin{cases} 0 & , \text{if } \mathbf{n}_i \cdot \mathbf{n}_j \leq T \\ (\mathbf{n}_i \cdot \mathbf{n}_j - T)^2 & , \text{caso contrário} \end{cases}, \quad (4.3)$$

onde $0 \leq T \leq 1$ é um limiar definido pelo usuário. Este limiar corresponde a um ângulo máximo tolerado entre as normais \mathbf{n}_i e \mathbf{n}_j . Ou seja, se a projeção de \mathbf{n}_i em \mathbf{n}_j , dada por $\mathbf{n}_i \cdot \mathbf{n}_j$, for menor que T a normal não é considerada no cálculo da normal filtrada (Figura 4.7).

O limiar T usado neste trabalho é 0.65. Para um ângulo grande a projeção de \mathbf{n}_i em \mathbf{n}_j é menor do que o limiar T usado, como é mostrado na Figura 4.7a. Portanto, nesse caso, como expresso na Equação (4.3), o peso w_j é zero. O ângulo maior para o qual w_j é zero, é aproximadamente 50° (Figura 4.7c). Finalmente, a projeção de \mathbf{n}_i em \mathbf{n}_j é maior do que o limiar T para um ângulo pequeno (Fi-

gura 4.7b). Nesse caso, a função decrescente $(\mathbf{n}_i \cdot \mathbf{n}_j - T)^2$ penaliza a divergência das normais \mathbf{n}_i e \mathbf{n}_j .

4.3.2 Peso Bilateral-Gaussiano

Zheng et al. (2011) apresentaram, no contexto de malhas, um esquema de correção de normais baseado em um filtro bilateral. O filtro bilateral, originalmente proposto por Tomasi & Manduchi (1998), é comumente usado em redução de ruído em imagens. O peso do filtro bilateral para reduzir o ruído em imagens é definido:

$$w_j = W_c(\|p - q\|)W_s(\|g(p) - g(q)\|), \quad (4.4)$$

onde q são os pixels na vizinhança do pixel p .

O peso consiste em duas partes: W_c é uma função monotonicamente decrescente da distância entre p e q , enquanto W_s , que representa a diferença de intensidade, é uma função monotonicamente decrescente da diferença de sinais entre p e q . Medir a diferença de intensidade permite que o filtro preserve *features*.

Este esquema serve como base do mecanismo proposto por Zheng et al. (2011), mostrado a seguir:

$$w_j = W_c(\|\mathbf{p}_i - \mathbf{p}_j\|)W_s(\|\mathbf{n}_i - \mathbf{n}_j\|), \quad (4.5)$$

Este esquema, similarmente ao filtro bilateral para suavização de imagens, tem duas partes. W_c corresponde a distância entre os pontos \mathbf{p}_i e \mathbf{p}_j e W_s define a diferença de sinais simplesmente como a distância entre as normais \mathbf{n}_i e \mathbf{n}_j . W_c e W_s são funções Gaussianas definidas como:

$$W_c(x) = \exp(-x^2/2\sigma_c^2), \quad W_s(x) = \exp(-x^2/2\sigma_s^2), \quad (4.6)$$

onde σ_c , σ_s são o desvio padrão das Gaussianas. No contexto de malhas σ_c foi definido como a média do comprimento das arestas incidentes num vértice. Seguindo esse raciocínio, definimos σ_c em relação à distância média de \mathbf{p}_i aos \mathbf{p}_j

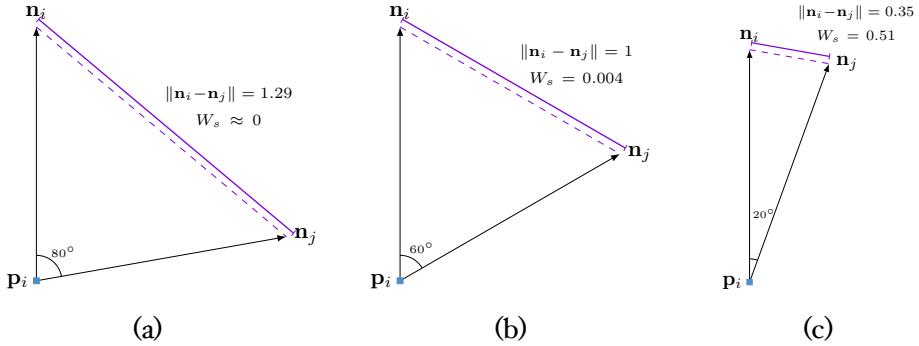


Figura 4.8: Distância entre os vetores \mathbf{n}_i e \mathbf{n}_j , e valor da função Gaussiana $W_s(x) = \exp(-x^2/2\sigma_s^2)$ com $\sigma_s = 0.3$ para três ângulos diferentes: (a) 80° , (b) 60° e (c) 20° .

na vizinhança de \mathbf{p}_i .

Não existe um mecanismo estabelecido para ajustar σ_s , mesmo no contexto de malhas. Neste trabalho, usamos $\sigma_s = 0.3$ (Figura 4.8). Para esse valor, a partir de $x = 1$, o valor da função Gaussiana é próximo a 0 ($W_s(1) < 0.01$). Isto representa um ângulo entre as normais \mathbf{n}_i e \mathbf{n}_j de 60° (Figura 4.9b). Na Figura 4.9a, mostramos normais \mathbf{n}_i e \mathbf{n}_j com angulo de 80° , o valor de $W_s(1) \approx 0$. Ou seja, o valor de W_s é inversamente proporcional ao angulo entre as normais.

4.3.3 Peso Bilateral-Misto

Wang et al. (2012) propuseram um método que combina ambos os esquemas explicados anteriormente, usando um peso bilateral consistindo de um peso Gaussiano e de um limiar.

Adaptamos o mecanismo de Wang et al. (2012) ao contexto de nuvens de pontos como segue:

$$w_j = W_c(||\mathbf{p}_i - \mathbf{p}_j||) \Phi_s(\mathbf{n}_i, \mathbf{n}_j), \quad (4.7)$$

onde W_c é um peso Gaussiano, definido como na Equação (4.6). E Φ_s é a parte que define a diferença de sinais, dada por um peso limiarizado adaptado

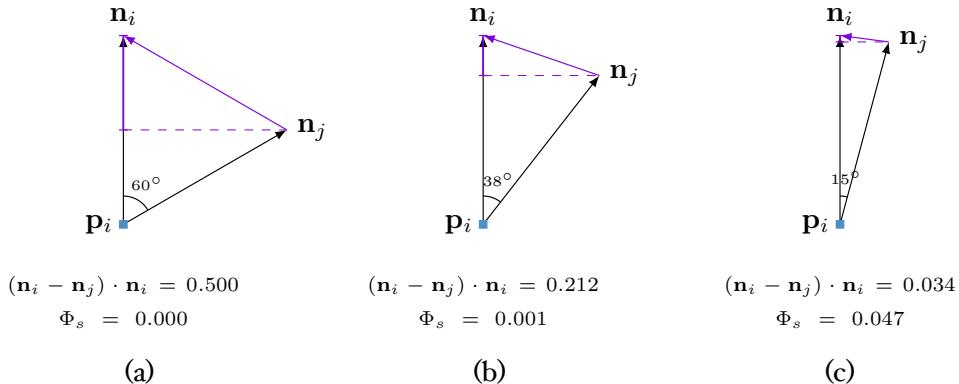


Figura 4.9: Projeção do vetor $\mathbf{n}_i - \mathbf{n}_j$ no vetor \mathbf{n}_j e valor do peso Φ_s com $T = 0.25$ para três ângulos diferentes: (a) 60° , (b) 38° e (c) 15° .

tivamente da seguinte forma:

$$\Phi_s(\mathbf{n}_i, \mathbf{n}_j) = \begin{cases} 0 & , \text{if } (\mathbf{n}_i - \mathbf{n}_j) \cdot \mathbf{n}_i \geq T \\ ((\mathbf{n}_i - \mathbf{n}_j) \cdot \mathbf{n}_i - T)^2 & , \text{caso contrário} \end{cases}, \quad (4.8)$$

onde $T = \sqrt{\sum_{j \in \mathcal{N}_i} ((\mathbf{n}_i - \mathbf{n}_j) \cdot \mathbf{n}_i)^2} / k$, e k é o número de pontos em \mathcal{N}_i .

Os métodos descritos anteriormente não funcionam corretamente quando as normais não estão orientadas de forma consistente, como ilustrado na Figura 4.10. Porém, dado que estes métodos dependem de um procedimento local para filtrar as normais, unicamente necessitamos garantir a consistência local. Em outras palavras, o filtro pode ser aplicado invertendo as normais na vizinhança \mathbf{n}_j se $\mathbf{n}_i \cdot \mathbf{n}_j \leq 0$. Este procedimento não garante um campo de normais orientado de forma consistente no contexto global. No entanto, isso não supõe um problema em nossa formulação.

4.4 Filtragem da Superfície

Depois de obter o campo de normais filtrado, temos que atualizar os pontos para corresponder com o novo campo de normais N' . Sun et al. (2007) propuseram

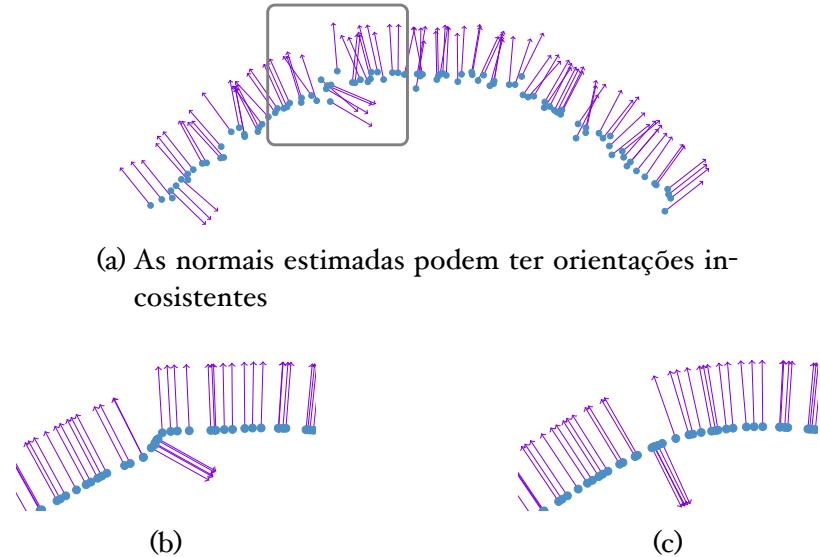


Figura 4.10: (b) Filtragem sem levar em conta a orientação das normais e (c) orientando as normais localmente de forma consistente.

um esquema iterativo para atualizar os vértices de uma malha de acordo com um novo campo de normais. Uma das vantagens dessa abordagem é que não requer cálculos da área da superfície, que é difícil estimar usando nuvens de pontos. Além disso, as normais não precisam estar orientadas de forma consistente.

Baseado neste esquema, propomos um método iterativo para atualizar os pontos.

$$\mathbf{p}_i^l = \mathbf{p}_i^{l-1} + \frac{1}{\sum_{j \in \mathcal{N}_i} w_j} \sum_{j \in \mathcal{N}_i} \mathbf{n}'_j (w_j \mathbf{n}'_j \cdot (\mathbf{p}_j^{l-1} - \mathbf{p}_i^{l-1})), \quad (4.9)$$

onde \mathbf{p}_i^l é o ponto na iteração l^a , e \mathbf{p}_i^0 é o ponto do conjunto original \mathcal{P} .

Este processo de atualização de pontos, Equação (4.9), pode ser visto como um deslocamento baseado em pesos ponderados pelas normais $\mathbf{n}'_j \in \mathcal{N}_i$. Os pesos são definidos como $w_j \mathbf{n}'_j \cdot (\mathbf{p}_j^{l-1} - \mathbf{p}_i^{l-1})$, onde w_j é definida usando um mecanismo

bilateral:

$$w_j = W_c(\|\mathbf{p}_i - \mathbf{p}_j\|)W_s(1 - (\mathbf{n}_i \cdot \mathbf{n}_j)), \quad (4.10)$$

onde W_c e W_s são funções Gaussianas como na Equação (4.6), com desvios padrões $\sigma_c = \max_{j \in \mathcal{N}_i}(\|\mathbf{p}_i - \mathbf{p}_j\|)$ e $\sigma_s = 1/3$. Com W_c , conseguimos penalizar a influência de pontos \mathbf{p}_j afastados de \mathbf{p}_i . Enquanto que com W_s , controlamos a influência de normais \mathbf{n}_j divergentes de \mathbf{n}_i , conseguindo uma difusão anisotrópica.

Neste passo, para lidar com as normais \mathbf{n}_j cuja orientação não é consistente com \mathbf{n}_i , usamos a mesma estratégia de inverter as normais na vizinhança \mathbf{n}_j se $\mathbf{n}_i \cdot \mathbf{n}_j \leq 0$.

No capítulo seguinte mostramos a eficácia de nosso esquema de atualização de pontos combinado com os distintos mecanismos para estimar e filtrar o campo de normais.

Capítulo 5

Resultados

Neste capítulo apresentamos os resultados de cada possível implementação da metodologia proposta neste trabalho. Mais precisamente, examinamos os resultados das seguintes combinações: análise de componentes principais (PCA) + Filtragem de Normais por Limiar + Atualização de pontos (PLA), PCA + Filtragem de Normais por Bilateral Gaussiano + Atualização de pontos (PGA), PCA + Filtragem de Normais por Bilateral Misto + Atualização de pontos (PMA), análise de componentes principais ponderada (WPCA) + Filtragem de Normais por Limiar + Atualização de pontos (WPLA), WPCA + Filtragem de Normais por Bilateral Gaussiano + Atualização de pontos (WPGA), WPCA + Filtragem de Normais por Bilateral Misto + Atualização de pontos (WPMA), Transformada de Hough Aleatória + Filtragem de Normais por Limiar + Atualização de pontos (HLA), Transformada de Hough Aleatória + Filtragem de Normais por Bilateral Gaussiano + Atualização de pontos (HGA), Transformada de Hough Aleatória + Filtragem de Normais por Bilateral Misto + Atualização de pontos (HMA).

Além disso, avaliamos o impacto de variar o tamanho da vizinhança em cada passo do *pipeline*, consideramos vizinhanças de 7, 15 e 21 pontos mais próximos para cada ponto p_i . O número do lado direito do acrônimo indica o tamanho da vizinhança, por exemplo, PLA₇ significa que 7 vizinhos mais próximos são considerados nos cálculos.

Igualmente, consideramos o número de iterações a serem empregadas durante o passo de filtragem de normais e o passo de atualização de pontos. A convergência dos métodos descritos em Equação (4.2) e Equação (4.9) não é garantida mesmo no contexto de malhas. Portanto, na prática, o número de iterações tem que ser determinado a priori. Testamos o passo de filtragem de normais com cinco números de iterações: 2, 4, 8, 16 e 32, e o passo de atualização de pontos com 4 números de iterações: 5, 10, 20 e 40, o que resulta em vinte combinações.

Executamos as combinações em cinco modelos com ruído sintético adicionado aos pontos com desvio padrão σ proporcional a 0.2 da longitude media das arestas da malha do modelo original. Dois tipos de ruído são testados: ruído Gaussiano na direção normal(RGN) e ruído Gaussiano com direção aleatória (RGA). Na Figura 5.4 e na Figura 5.5 são mostrados os modelos usados nos experimentos, os modelos são: Bitorus, Elefante, Fêmur, Niccolo e Fandisk.

5.1 Métricas

Podemos usar a correspondência um-a-um entre os pontos na superfície original \mathcal{M} e na superfície filtrada \mathcal{S} para medir a qualidade do processo de suavização. A saber, definimos uma métrica E_k relacionada com a curvatura: primeiramente, calculamos a diferença da curvatura em cada ponto \mathbf{p}_i em \mathcal{M} e \mathcal{S} , e logo, calculamos a média das diferenças. Quanto mais próximo de zero é E_k , melhor é o processo de suavização. A curvatura pode ser calculada usando o operador de variação superficial descrito em Pauly et al. (2003a).

Comparamos também a área da superfície do modelo original \mathcal{M} e filtrado \mathcal{S} , esta métrica é usada na literatura.

Nos gráficos de *box plot* na Figura 5.1, pode se observar o desempenho de cada uma das vinte e sete alternativas de suavização de nuvem de pontos no modelo Elefante variando o numero de iterações para filtragem de normais e atualização de pontos. O erro varia consideravelmente para todos os métodos, mostrando que o número de iterações afeta significativamente o desempenho. Além disso,

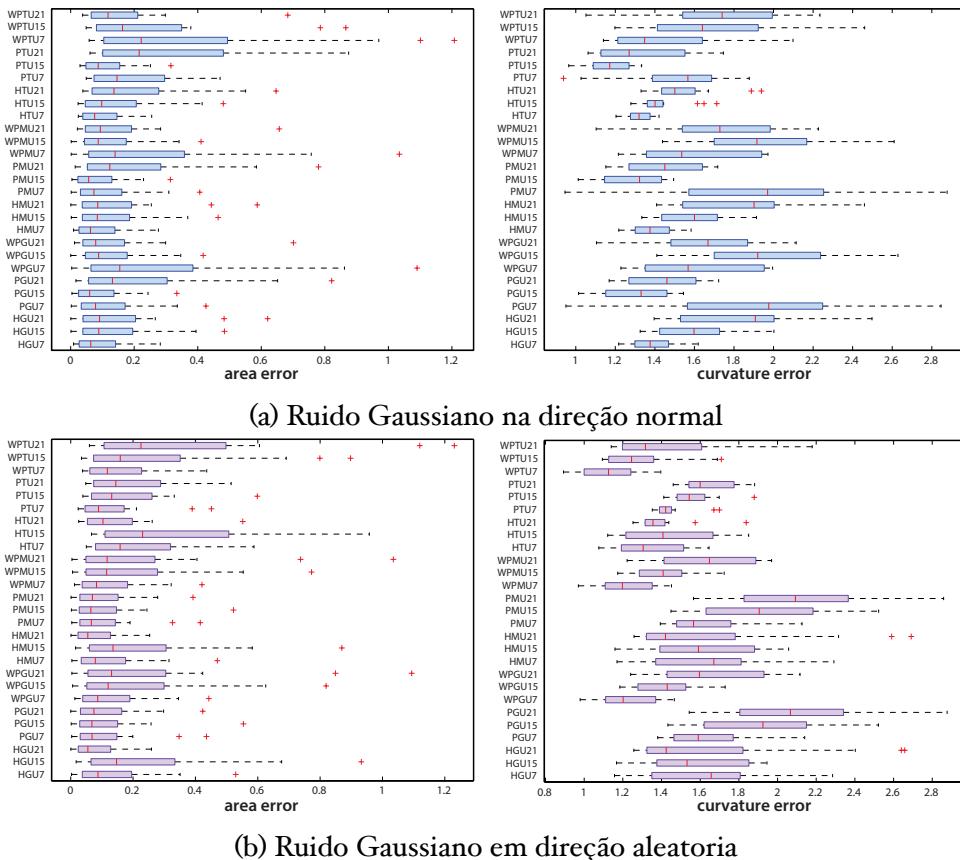


Figura 5.1: Avaliação quantitativa das métricas usando diferentes combinações do método de suavização no modelo Elefante ($\times 10^{-4}$).

podemos ver que os métodos que utilizam PCA e Transformada de Hough aleatória (RHT) com vizinhanças pequenas tendem a gerar erros menores do que outras alternativas para o erro da área. Igualmente, para o erro da curvatura, a estimação de normais usando PCA e WPCA com vizinhanças pequenas têm melhor desempenho.

O erro da área \times curvatura é mostrado no *scatter plot* da Figura 5.3 para todas as 540 possíveis implementações (9 técnicas com 3 tamanhos de vizinhanças e 20 opções da iteração) para suavizar a nuvem de pontos Bitorus. Os pontos em destaque correspondem ao melhor resultado em termos da área (inferior direito)

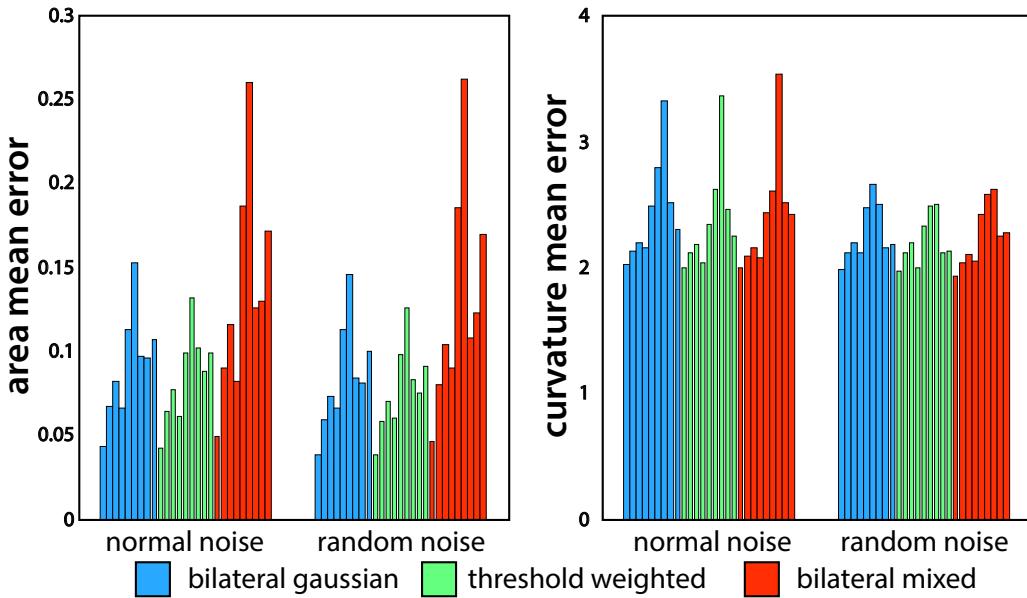


Figura 5.2: Erro médio de cada alternativa para todos os modelos variando as 27 opções de vizinhança e número de passos por cada iteração.

e em termos da curvatura (inferior esquerda) assim como os piores em termos da área (superior direita) e em termos da curvatura (superior direita). O modelo destacado no meio na direita é um caso médio.

Considerando as 60 variações de cada implementação (3 tamanhos de vizinhanças e 20 opções por iteração), na figura 5.2 mostramos os erros da curvatura média e a área para todos os modelos. Pode se observar que o desempenho da abordagem de pesos limiarizados é melhor em relação ao erro da área, enquanto não é possível assinalar qual é o melhor método para o erro da curvatura média.

5.2 Comparações com Outras Técnicas

Comparamos nosso método com LOP, WLOP, APSS e RIMLS. Para cada um desses métodos, testamos três combinações de parâmetros: o conjunto de parâmetros padrão e outras duas combinações que produziram resultados visualmente agradáveis. Desses três resultados, escolhemos o conjunto que deu o me-

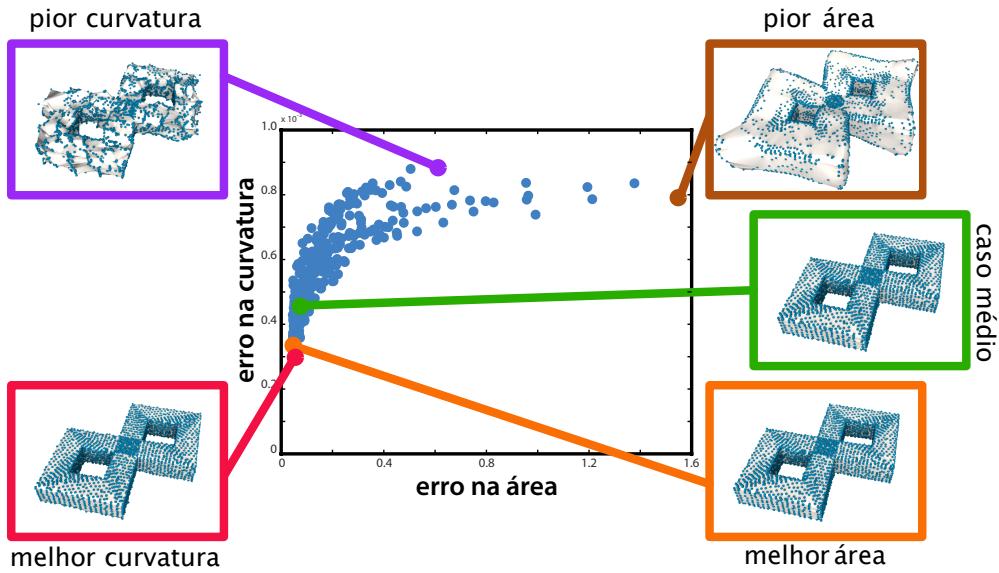


Figura 5.3: *Scatter plot* para as métricas área \times curvatura no modelo Bitorus (ruído aleatório).

lhore resultado em relação aos erros de área e curvatura média. O conjunto de parâmetros para os métodos são: para LOP e WLOP (raio de suporte h , μ controlando a força de repulsão), APSS (escala do filtro, parâmetro esférico), RIMLS (escala do filtro, nitidez) e o nosso (o acrônimo descrito anteriormente, número de iterações para filtragem das normais, número de iterações no passo de atualização dos pontos). Uma comparação qualitativa pode ser vista na Figura 5.4. Na Tabela 5.1 mostramos o resultado das comparações, onde E_A e E_k representam o erro da área e o erro da curvatura. Podemos observar que nosso método é bastante competitivo, superando em alguns casos o método RIMLS (estado da arte em suavização de nuvens de pontos).

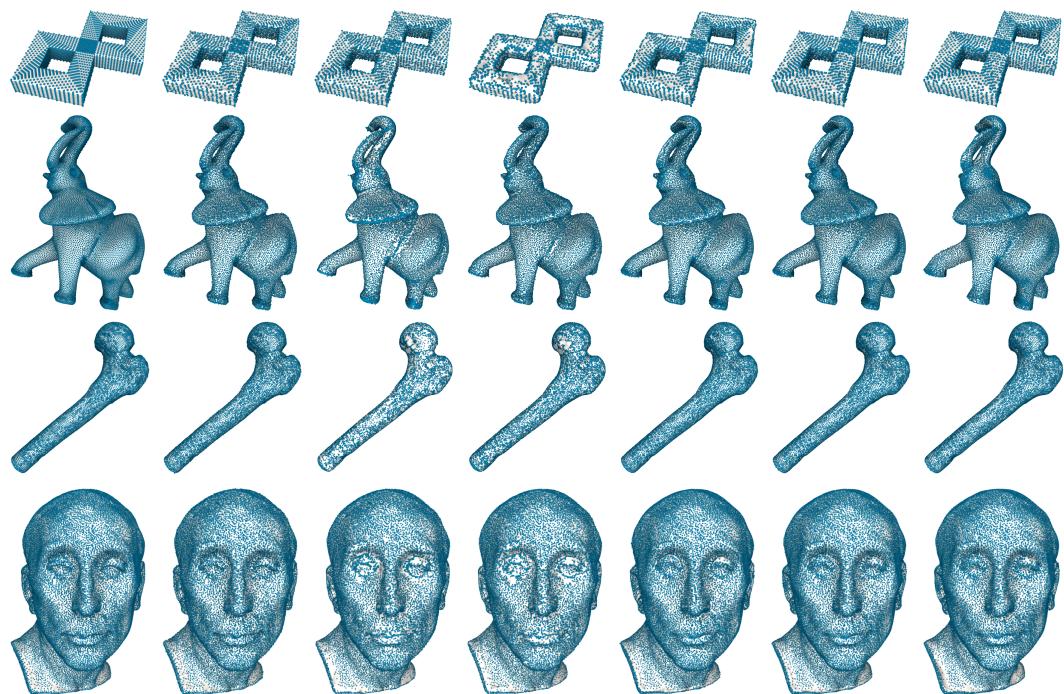


Figura 5.4: Suavização de nuvens de pontos: Bitorus com 4350 pontos, Elefante com 24955 pontos, Fêmur com 15185 pontos and Niccolo with 25239 pontos. De direita a esquerda: modelos originais, modelos com ruído Gaussiano com direção aleatória, e os resultados de *Locally Optimal Projection* (LOP), *Weighted Locally Optimal Projection* (WLOP), *Algebraic Point Set Surfaces* (APSS), *Robust Implicit Moving Least Squares* (RIMLS) e nosso método.

5.2 Comparações com Outras Técnicas

Tabela 5.1: Melhor conjunto de parametros e erros

Modelo	Ruído	Método	Parâmetros	$E_A \times 10^{-3}$	$E_k \times 10^{-3}$
Double Torus ($ S = 4350$)	RGN	LOP	(0.14, 0.3)	53.62	0.78
		WLOP	(0.12, 0.3)	26.45	0.72
		APSS	(3, 0.5)	5.59	0.49
		RIMLS	(5, 0.5)	4.24	0.24
		Nosso	(PTU ₇ , 32, 5)	0.59	0.31
	RGA	LOP	(0.9, 0.4)	95.68	0.45
		WLOP	(0.12, 0.3)	59.42	0.72
		APSS	(3, 0.5)	23.16	0.54
		RIMLS	(3, 0.5)	12.39	0.31
		Nosso	(WPTU ₂₁ , 4, 5)	0.22	0.34
Elephant ($ S = 24955$)	RGN	LOP	(0.07, 0.4)	11.86	0.18
		WLOP	(0.05, 0.5)	5.64	0.24
		APSS	(3, 0.75)	0.70	0.12
		RIMLS	(4, 1)	6.59	0.14
		Nosso	(PGU ₂₁ , 32, 5)	0.24	0.24
	RGA	LOP	(0.05, 0.4)	42.14	0.15
		WLOP	(0.03, 0.4)	38.87	0.13
		APSS	(2, 1)	1.57	0.09
		RIMLS	(2, 0.75)	1.58	0.08
		Nosso	(HG ₂₁ , 32, 5)	0.04	0.22
Fandisk ($ S = 25894$)	RGN	LOP	(0.08, 0.3)	1.22	0.21
		WLOP	(0.07, 0.35)	37.36	0.20
		APSS	(2, 1)	5.76	0.10
		RIMLS	(4, 0.5)	0.59	0.05
		Nosso	(HTU ₂₁ , 16, 5)	0.02	0.08
	RGA	LOP	(0.07, 0.35)	21.82	0.20
		WLOP	(0.07, 0.35)	35.21	0.20
		APSS	(2, 1)	0.58	0.08
		RIMLS	(4, 0.75)	1.13	0.08
		Nosso	(HGU ₇ , 32, 5)	0.04	0.09
Nicolo ($ S = 25239$)	RGN	LOP	(0.06, 0.3)	141.30	0.10
		WLOP	(0.07, 0.25)	10.52	0.10
		APSS	(4, 0.5)	9.74	0.10
		RIMLS	(4, 1)	0.18	0.08
		Ours	(WPTU ₁₅ , 8, 5)	0.05	0.09
	RGA	LOP	(0.04, 0.3)	71.31	0.11
		WLOP	(0.04, 0.25)	25.52	0.10
		APSS	(4, 0.75)	2.67	0.11
		RIMLS	(4, 1)	0.36	0.09
		Nosso	(PGU ₇ , 16, 5)	0.22	0.09
Femur ($ S = 8168$)	RGN	LOP	(0.07, 0.4)	13.92	0.26
		WLOP	(0.06, 0.4)	0.17	0.26
		APSS	(3, 0.75)	1.55	0.22
		RIMLS	(4, 0.75)	0.40	0.23
		Nosso	(HTU ₇ , 32, 5)	9.07	0.22
	RGA	LOP	(0.06, 0.3)	33.19	0.26
		WLOP	(0.06, 0.4)	7.84	0.26
		APSS	(3, 1)	0.98	0.23
		RIMLS	(3, 0.75)	0.60	0.21
		Nosso	(HTU ₇ , 32, 5)	10.20	0.22

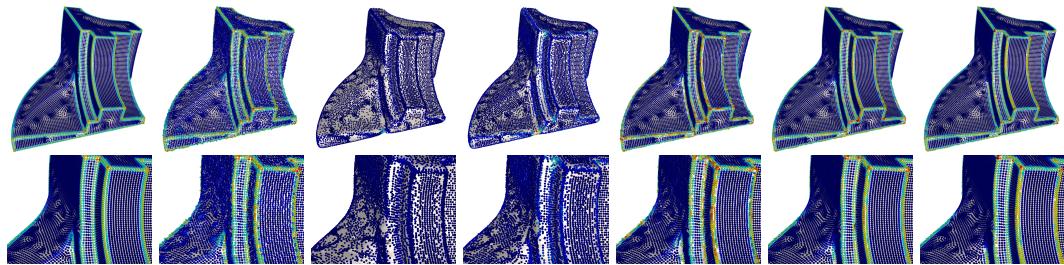


Figura 5.5: Nossa método é capaz de preservar feições afiadas enquanto suaviza as superfícies. De esquerda a direita: modelo original e com ruído; resultados usando LOP, WLOP, APSS, RIMLS e nosso método.

5.3 Preservação de Feições Afiadas

Finalmente, realizamos uma comparação qualitativa mostrando a eficácia de nossa proposta para preservar feições afiadas. Na figura 5.5 mostramos o resultado de suavizar o modelo Fandisk usando os métodos LOP, WLOP, APSS, RIMLS e o nosso. A cor mostra a distribuição da curvatura. Podemos observar que nosso método tem bons resultados, comparáveis com RIMLS.

5.4 Modelos Reais

Além disso, testamos nosso método com modelos escaneados com ruído do *scanner*. Na figura 5.6 mostramos o resultado de suavizar o modelo Busto. No modelo Busto, utilizamos os seguintes parâmetros PGU7 com cinco iterações no passo de filtragem de normais e 7 iterações no passo de atualização de pontos. Observamos que nosso método consegue preservar feições de forma bastante rasoável.

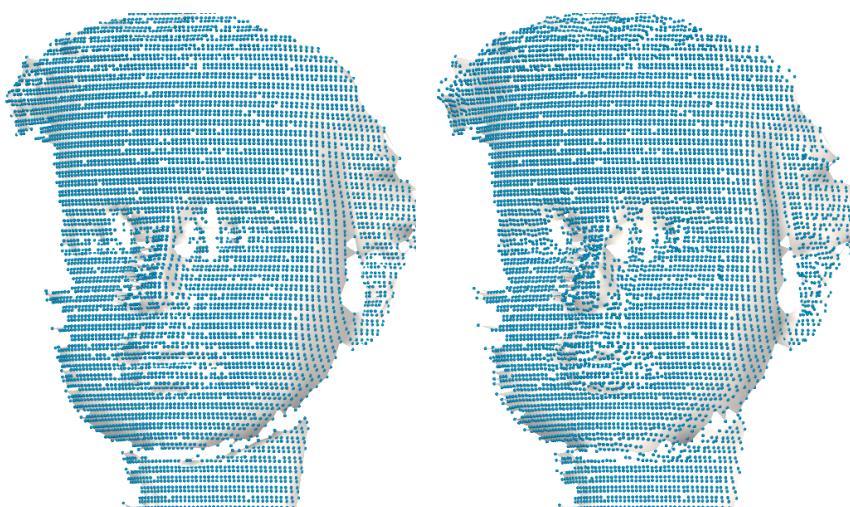


Figura 5.6: Suavização de um modelo real escaneado.

Capítulo 6

Conclusões

Neste capítulo são apresentadas as principais contribuições desta dissertação no contexto de suavização de superfícies representadas por nuvens de pontos, bem como algumas de suas limitações. Além disso, são apresentadas algumas ideias para trabalhos futuros no sentido de dar continuidade à abordagem proposta.

6.1 Contribuições

As principais contribuições deste trabalho de mestrado podem ser sucintamente sumarizadas tal como segue.

- Mostramos que a abordagem de dois passos (filtragem de normais/atualização dos pontos) para suavizar superfícies, comumente empregada no contexto de malhas, pode ser estendida para o contexto de nuvens de pontos. Nossa abordagem realiza um passo prévio para estimar as normais, e para os passos de estimativa e atualização de normais, utiliza e compara técnicas existentes.
- Baseado em métodos para atualizar vértices no contexto de malhas, propomos um método para atualizar os pontos que leva em conta a vizinhança de cada ponto. A influência de cada normal na vizinhança é dada por um peso

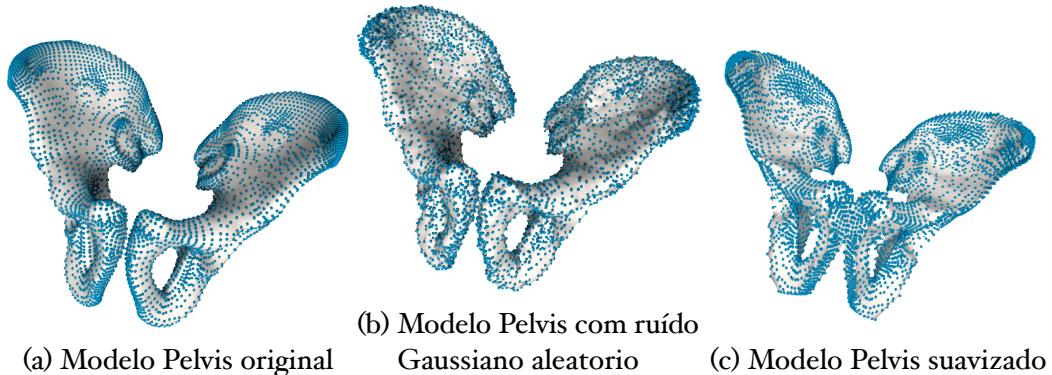


Figura 6.1: Um inconveniente com o ponto conjunto de superfícies é determinar a relação de vizinhança de pontos em superfícies com folhas próximas.

bilateral que leva em conta, além da distância entre os pontos, a distância entre as normais.

- Realizamos um conjunto de experimentos abrangente, variando o número de vizinhos e o número de iterações em cada passo do *pipeline*. Os experimentos mostram que os parâmetros utilizados podem influenciar consideravelmente a qualidade do processo de suavização.
- Finalmente, realizamos comparações com métodos do estado da arte mostrando que o esquema de filtragem de normais/atualização de pontos tem um desempenho superior a métodos existentes, sendo bastante competitivo e uma boa alternativa para suavização de superfícies baseadas em pontos.

6.2 Limitações

Como a maioria de métodos prévios de suavização de superfícies baseados em vizinhanças, nosso método não tem bom desempenho em presença de superfícies com folhas paralelas próximas, como pode ser visto na Figura 6.1.

Uma possível solução a esse problema é usar algum algoritmo para orientar

normais, como o proposto por Hoppe et al. (1992), após o passo de estimação de normais. Assim, no passo de filtragem de normais as normais seriam mais precisas e no passo de atualização dos pontos, normais com orientação oposta não influenciariam a nova posição dos pontos.

6.3 Trabalhos Futuros

Após a pesquisa, desenvolvimento e avaliação dessa técnica de suavização, algumas ideias foram propostas no sentido de consolidar a base para trabalhos futuros. Dentre elas destaca-se a utilização de um quadro teórico proposto por Mitra et al. (2004) para realizar um estudo mais profundo dos raios de vizinhança no passo de estimação de normais. Além disso, uma análise com diferentes distribuições de ruído é um estudo útil, principalmente fazendo uso de um *benchmark* como o proposto em Berger et al. (2013).

Referências Bibliográficas

- ALEXA, M.; BEHR, J.; COHEN-OR, D.; FLEISHMAN, S.; LEVIN, D.; SILVA, C. T. Computing and rendering point set surfaces. *IEEE Trans. Vis. Comput. Graph.*, v. 9, p. 3–15, 2003.
- AMENTA, N.; BERN, M. Surface reconstruction by voronoi filtering. In: *Proceedings of the fourteenth annual symposium on Computational geometry*, SCG '98, New York, NY, USA: ACM, 1998, p. 39–48 (SCG '98,).
- BELKIN, M.; SUN, J.; WANG, Y. Constructing laplace operator from point clouds in rd. In: *Proceedings of the twentieth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '09, Philadelphia, PA, USA: Society for Industrial and Applied Mathematics, 2009, p. 1031–1040 (SODA '09,).
- BENTLEY, J. L. Multidimensional binary search trees used for associative searching. *Commun. ACM*, v. 18, n. 9, p. 509–517, 1975.
- BERGER, M.; LEVINE, J. A.; NONATO, L. G.; TAUBIN, G.; SILVA, C. T. A benchmark for surface reconstruction. *ACM Trans. Graph.*, v. 32, n. 2, p. 20:1–20:17, 2013.
- BORRMANN, D.; ELSEBERG, J.; LINGEMANN, K.; NÜCHTER, A. The 3d hough transform for plane detection in point clouds: A review and a new accumulator design. *3D Research*, v. 2, n. 2, p. 1–13, 2011.
- BOULCH, A.; MARLET, R. Fast and robust normal estimation for point clouds with sharp features. *Computer Graphics Forum*, v. 31, n. 5, p. 1765–1774, 2012.

Referências Bibliográficas

- BUADES, A.; COLL, B.; MOREL, J. M. A non-local algorithm for image denoising. In: *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, 2005, p. 60–65 vol. 2.
- CLARENZ, U.; RUMPF, M.; TELEA, A. Finite elements on point based surfaces. In: *Proceedings of the First Eurographics conference on Point-Based Graphics*, SPBG’04, Aire-la-Ville, Switzerland, Switzerland: Eurographics Association, 2004a, p. 201–211 (SPBG’04,).
- CLARENZ, U.; RUMPF, M.; TELEA, A. Surface processing methods for point sets using finite elements. *Computers & Graphics*, v. 28, n. 6, p. 851 – 868, 2004b.
- DANIELS, J. I.; HA, L. K.; OCHOTTA, T.; SILVA, C. T. Robust smooth feature extraction from point clouds. In: *Proceedings of the IEEE International Conference on Shape Modeling and Applications 2007*, SMI ’07, Washington, DC, USA: IEEE Computer Society, 2007, p. 123–136 (SMI ’07,).
- DESBRUN, M.; MEYER, M.; SCHRÖDER, P.; BARR, A. H. Implicit fairing of irregular meshes using diffusion and curvature flow. In: *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, SIGGRAPH ’99, New York, NY, USA: ACM Press/Addison-Wesley Publishing Co., 1999, p. 317–324 (SIGGRAPH ’99,).
- FLEISHMAN, S.; COHEN-OR, D.; SILVA, C. T. Robust moving least-squares fitting with sharp features. *ACM Trans. Graph.*, v. 24, n. 3, p. 544–552, 2005.
- FLEISHMAN, S.; DRORI, I.; COHEN-OR, D. Bilateral mesh denoising. In: *ACM SIGGRAPH 2003 Papers*, SIGGRAPH ’03, New York, NY, USA: ACM, 2003, p. 950–953 (SIGGRAPH ’03,).
- GROSS, M.; PFISTER, H. *Point-based graphics*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2007.
- GUENNEBAUD, G.; GROSS, M. Algebraic point set surfaces. *ACM Trans. Graph.*, v. 26, n. 3, 2007.

- GUILLEMOT, T.; ALMANSA, A.; BOUBEKEUR, T. Non local point set surfaces. In: *3D Imaging, Modeling, Processing, Visualization and Transmission (3DIMPVT), 2012 Second International Conference on*, 2012, p. 324–331.
- HOPPE, H.; DEROSE, T.; DUCHAMP, T.; McDONALD, J.; STUETZLE, W. Surface reconstruction from unorganized points. *SIGGRAPH Comput. Graph.*, v. 26, n. 2, p. 71–78, 1992.
- HUANG, H.; LI, D.; ZHANG, H.; ASCHER, U.; COHEN-OR, D. Consolidation of unorganized point clouds for surface reconstruction. *ACM Trans. Graph.*, v. 28, n. 5, p. 176:1–176:7, 2009.
- JONES, T. R.; DURAND, F.; DESBRUN, M. Non-iterative, feature-preserving mesh smoothing. In: *ACM SIGGRAPH 2003 Papers*, SIGGRAPH '03, New York, NY, USA: ACM, 2003, p. 943–949 (SIGGRAPH '03,).
- KŁOSOWSKI, J.; HELD, M.; MITCHELL, J.; SOWIZRAL, H.; ZIKAN, K. Efficient collision detection using bounding volume hierarchies of k-dops. *Visualization and Computer Graphics, IEEE Transactions on*, v. 4, n. 1, p. 21–36, 1998.
- KORNPROBST, P.; TUMBLIN, J. *Bilateral filtering: Theory and applications*. Now Publishers Inc, 2009.
- LANGE, C.; POLTHIER, K. Anisotropic smoothing of point sets. *Comput. Aided Geom. Design*, v. 22, n. 7, p. 680–692, 2005.
- LEVIN, D. The approximation power of moving least-squares. *Math. Comput.*, v. 67, n. 224, p. 1517–1531, 1998.
- LEVOY, M.; WHITTED, T. *The use of points as a display primitive*. Relatório Técnico TR 85-022, University of North Carolina at Chapel Hill, 1985.
- LI, B.; SCHNABEL, R.; KLEIN, R.; CHENG, Z.; DANG, G.; JIN, S. Robust normal estimation for point clouds with sharp features. *Computers & Graphics*, v. 34, n. 2, p. 94 – 106, 2010.

Referências Bibliográficas

- LIAO, B.; XIAO, C.; JIN, L.; FU, H. Efficient feature-preserving local projection operator for geometry reconstruction. *Comput. Aided Design*, v. 45, p. 861–874, 2013.
- LIPMAN, Y.; COHEN-OR, D.; LEVIN, D. Data-dependent mls for faithful surface approximation. In: *Proceedings of the fifth Eurographics symposium on Geometry processing*, SGP '07, Aire-la-Ville, Switzerland, Switzerland: Eurographics Association, 2007a, p. 59–67 (SGP '07,).
- LIPMAN, Y.; COHEN-OR, D.; LEVIN, D.; TAL-EZER, H. Parameterization-free projection for geometry reconstruction. *ACM Trans. Graph.*, v. 26, n. 3, 2007b.
- MARIO BOTSCHE, LEIF KOBELT, M. P. P. A. B. L. *Polygon mesh processing*. Natick, MA, USA: AK Peters, 2010.
- MEAGHER, D. Geometric modeling using octree encoding. *Computer Graphics and Image Processing*, v. 19, n. 2, p. 129–147, 1982.
- MITRA, N. J.; NGUYEN, A. Estimating surface normals in noisy point cloud data. In: *Proceedings of the nineteenth annual symposium on Computational geometry*, SCG '03, New York, NY, USA: ACM, 2003, p. 322–328 (SCG '03,).
- MITRA, N. J.; NGUYEN, A.; GUIBAS, L. Estimating surface normals in noisy point cloud data. *International Journal of Computational Geometry & Applications*, v. 14, n. 04n05, p. 261–276, 2004.
- OHTAKE, Y.; BELYAEV, A.; BOGAEVSKI, I. Mesh regularization and adaptive smoothing. *Computer-Aided Design*, v. 33, n. 11, p. 789 – 800, 2001.
- ÖZTIRELI, A. C.; GUENNEBAUD, G.; GROSS, M. Feature preserving point set surfaces based on non-linear kernel regression. *Computer Graphics Forum*, v. 28, n. 2, p. 493–501, 2009.

- PAULY, M.; KEISER, R.; GROSS, M. Multi-scale feature extraction on point-sampled surfaces. In: *Computer graphics forum*, Wiley Online Library, 2003a, p. 281–289.
- PAULY, M.; KEISER, R.; KOBBELT, L. P.; GROSS, M. Shape modeling with point-sampled geometry. *ACM Trans. Graph.*, v. 22, n. 3, p. 641–650, 2003b.
- PAULY, M.; KOBBELT, L.; GROSS, M. *Multiresolution modeling of point-sampled geometry*. Relatório Técnico, ETH Zurich, 2002.
- PEARSON, K. On lines and planes of closest fit to systems of points in space. *Philosophical Magazine Series 6*, v. 2, n. 11, p. 559–572, 1901.
- PETRONETTO, F.; PAIVA, A.; HELOU, E. S.; STEWART, D. E.; NONATO, L. G. Mesh-free discrete laplace-beltrami operator. *Computer Graphics Forum*, v. 32, n. 6, p. 214–226, 2013.
- SCHALL, O.; BELYAEV, A.; SEIDEL, H.-P. Feature-preserving non-local denoising of static and time-varying range data. In: *Proceedings of the 2007 ACM symposium on Solid and physical modeling*, SPM '07, New York, NY, USA: ACM, 2007, p. 217–222 (SPM '07,). Disponível em: <<http://doi.acm.org/10.1145/1236246.1236277>>.
- SUN, X.; ROSIN, P. L.; MARTIN, R. R.; LANGBEIN, F. C. Fast and effective feature-preserving mesh denoising. *IEEE Trans. Vis. Comput. Graph.*, v. 13, n. 5, p. 925–938, 2007.
- TAUBIN, G. *Linear anisotropic mesh filtering*. Relatório Técnico, Technical Report RC22213, IBM Research Division TJ Watson Research Center, 2001.
- TOMASI, C.; MANDUCHI, R. Bilateral filtering for gray and color images. In: *Proceedings of the Sixth International Conference on Computer Vision*, ICCV '98, Washington, DC, USA: IEEE Computer Society, 1998, p. 839– (ICCV '98,).

Referências Bibliográficas

- WANG, J.; ZHANG, X.; YU, Z. A cascaded approach for feature-preserving surface mesh denoising. *Comput. Aided Design*, p. 597–610, 2012.
- WEBER, A.; FRIEDRICH, C. J. *Theory of the location of industries*. University of Chicago Press Chicago, 1962.
- YAGOU, H.; OHTAKE, Y.; BELYAEV, A. Mesh smoothing via mean and median filtering applied to face normals. In: *Geometric Modeling and Processing 2002. Proceedings*, 2002, p. 124–131.
- YAGOU, H.; OHTAKE, Y.; BELYAEV, A. Mesh denoising via iterative alpha-trimming and nonlinear diffusion of normals with automatic thresholding. In: *Computer Graphics International, 2003. Proceedings*, 2003, p. 28–33.
- ZHENG, Y.; FU, H.; AU, O.-C.; TAI, C.-L. Bilateral normal filtering for mesh denoising. *IEEE Trans. Vis. Comput. Graph.*, v. 17, n. 10, p. 1521–1530, 2011.