



Universidade Federal de Alagoas

Programa de Pós-Graduação em Matemática

DISSERTAÇÃO DE MESTRADO

**Registro Automático de Superfícies Usando
Spin-Image**

Thales Miranda de Almeida Vieira

Rio São Francisco



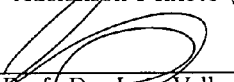
Registro Automático de Superfícies usando Spin-Images

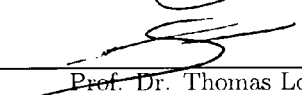
Thales Miranda de Almcida Vicira

Dissertação de Mestrado na área de concentração de Computação Gráfica submetida em 6 de Fevereiro de 2007 à Banca Examinadora, designada pelo Colegiado do Programa de Pós-Graduação em Matemática da Universidade Federal de Alagoas, como parte dos requisitos necessários à obtenção do grau de mestre em Matemática.

Banca Examinadora:


Prof. Dr. Adelailson Peixoto (orientador)


Prof. Dr. Luiz Velho


Prof. Dr. Thomas Lewiner

Agradecimentos

A minha família.

Ao meu orientador, Adailson Peixoto, pela dedicação, oportunidade e apoio fundamentais para a realização deste trabalho.

Aos professores Luiz Velho e Thomas Lewiner, pela disponibilidade e atenção concedidas e suas grandes colaborações neste trabalho.

Ao professor Vinicius Mello, por suas importantes sugestões e críticas a este trabalho.

Ao Departamento de Matemática da Universidade Federal de Alagoas, pelo apoio concedido desde a Iniciação Científica.

À FAPEAL, pelo apoio financeiro.

Resumo

Este trabalho descreve um método baseado em três etapas para reconstrução de modelos a partir de malhas capturadas de scanners 3D. Malhas obtidas a partir de diferentes pontos de visão de um scanner têm sua representação em sistemas de coordenadas local. Portanto, para a reconstrução final do modelo, é necessário realizar um alinhamento dessas malhas, ou registro. O algoritmo mais famoso para realizar registro de nuvens de pontos é o algoritmo ICP. Porém, um dos requisitos desse algoritmo é uma estimativa inicial do alinhamento das malhas, que muitas vezes é feita manualmente. Para automatizar esse processo, este trabalho utiliza descritores *spin-image* para identificar regiões de sobreposição entre as malhas e estimar seus alinhamentos. Após este registro inicial, o alinhamento é refinado através do algoritmo ICP, e finalmente o modelo é reconstruído usando uma técnica chamada VRIP.

Palavras-chave: registro de superfícies, spin-images, reconstrução 3d, malhas, range-images.

Abstract

This work describes a method based on three stages for reconstructing a model from a given set of scanned meshes obtained from 3D scanners. Meshes scanned from different scanner's view points have their representation in local coordinate systems. Therefore, for final model reconstruction, an alignment of the meshes is required. The most popular algorithm for cloud data registration is the ICP algorithm. However, ICP requires an initial estimate of mesh alignment, which is, many times, done manually. To automate this process, this work uses a surface representation called *spin-images* to identify overlap areas between the meshes and to estimate their alignment. After this initial registration, the alignment is refined by the ICP algorithm, and finally the model is reconstructed using a method called VRIP.

Keywords: surface registration, spin-images, 3d reconstruction, meshes, range-images.

Sumário

1	Introdução	1
1.1	Registro de Superfícies	1
1.2	Objetivos	3
1.3	Contribuições	3
1.4	Aplicações	4
1.5	Estrutura do Trabalho	4
2	Descrição Geral do Método	5
2.1	Introdução	5
2.2	Alinhamento Inicial	5
2.3	Refinamento do Registro	7
2.4	Reconstrução da Malha Final	8
3	Alinhamento Automático Inicial	10
3.1	Introdução	10
3.2	Descritores Spin-Images	10
3.2.1	Spin-maps	11
3.2.2	Spin-images	12
3.2.3	Parâmetros das Spin-images	13
3.2.4	Textured Spin-Images	14
3.2.5	Comparação de Spin-Images	15
3.3	Seleção de Pontos	16
3.3.1	Seleção Aleatória de Pontos	17
3.3.2	Seleção por Curvatura Local	17
3.4	Geração de Correspondências	21
3.5	Filtragens de Correspondências	22
3.5.1	Filtragem Usando Medidas de Similaridade	22
3.5.2	Filtragem Usando Consistência Geométrica	23
3.5.3	Agrupamento	23
3.6	Cálculo da Transformação: Algoritmo de Horn	24
3.7	Validação	26
4	Refinamento do Registro usando ICP	28
4.1	Introdução	28
4.2	Visão Geral	28
4.3	Seleção de Pontos	29
4.4	Geração de Correspondências	30
4.4.1	Kd-trees	30

4.5	Atribuição de Pesos às Correspondências	31
4.6	Rejeição de Correspondências	32
4.7	Métricas de Erro e Minimização	33
5	Reconstrução da Malha Final	35
5.1	Introdução	35
5.2	Integração Volumétrica	35
5.3	Marching Cubes	38
6	Resultados e Discussão	40
6.1	Introdução	40
6.2	Resultados	40
6.2.1	Criação de Spin-images	40
6.2.2	Filtragem de Correspondências	43
6.2.3	Algoritmo ICP	44
6.2.4	Cálculo de Curvatura e Resolução das Malhas	44
6.3	Exemplos	45
7	Conclusão e Trabalhos Futuros	50

Lista de Figuras

1.1	Registro de superfícies.	1
1.2	Malhas geradas a partir de range images capturadas de diferentes pontos de visão de um objeto.	2
2.1	Procedimento para registro automático de superfícies	6
2.2	Primeira etapa: alinhamento inicial de um par de malhas.	7
2.3	Segunda etapa: refinamento do alinhamento entre duas malhas usando ICP.	8
2.4	Terceira etapa: Construção da malha final usando VRIP.	9
3.1	Base criada pelo ponto orientado (p, n) e coordenadas (α, β) de um ponto x	11
3.2	<i>Spin-map</i> com sua respectiva <i>Spin-image</i>	12
3.3	Efeito de diferentes <i>bin sizes</i> em <i>spin-images</i>	14
3.4	<i>Spin-map</i> e <i>spin-image</i> convencional e com textura.	15
3.5	Malha com pontos de alta medida de curvatura $\kappa_1^2 + \kappa_2^2$ destacados.	18
3.6	Região de Voronoi de uma vizinhança estrelada.	19
3.7	Exemplo de histograma utilizado para detecção de <i>outliers</i>	22
3.8	Agrupamento e validação.	27
4.1	Processo de criação de uma <i>kd-tree</i> bidimensional.	31
4.2	Rejeição de pares contendo pontos de fronteira.	33
5.1	Cálculo da função de distância com sinal $d_i(v)$	36
5.2	Efeito da função de distância $D(v)$ em duas curvas alinhadas com pequeno erro.	37
6.1	Pontos correspondentes e suas <i>spin-images</i>	41
6.2	Maior espaçamento possível entre duas visões de um objeto contendo as regiões de alta curvatura.	42
6.3	Registro inconsistente devido à simetria.	44
6.4	Malhas em diferentes resoluções com pontos de maior curvatura destacados.	45
6.5	Registro de malhas com diferentes resoluções.	46
6.6	Malhas alinhadas do coelho.	47
6.7	Modelo reconstruído do coelho.	47
6.8	Malhas alinhadas da cabeça.	48
6.9	Modelo reconstruído da cabeça.	48
6.10	Malhas alinhadas do cavalo.	49
6.11	Modelo reconstruído do cavalo.	49
7.1	Modelo de um vaso simétrico.	51

Capítulo 1

Introdução

1.1 Registro de Superfícies

O problema de registro de superfícies consiste em obter as transformações espaciais que realizem o melhor alinhamento entre um conjunto de malhas que representam visões de um mesmo objeto (Figura 1.1).

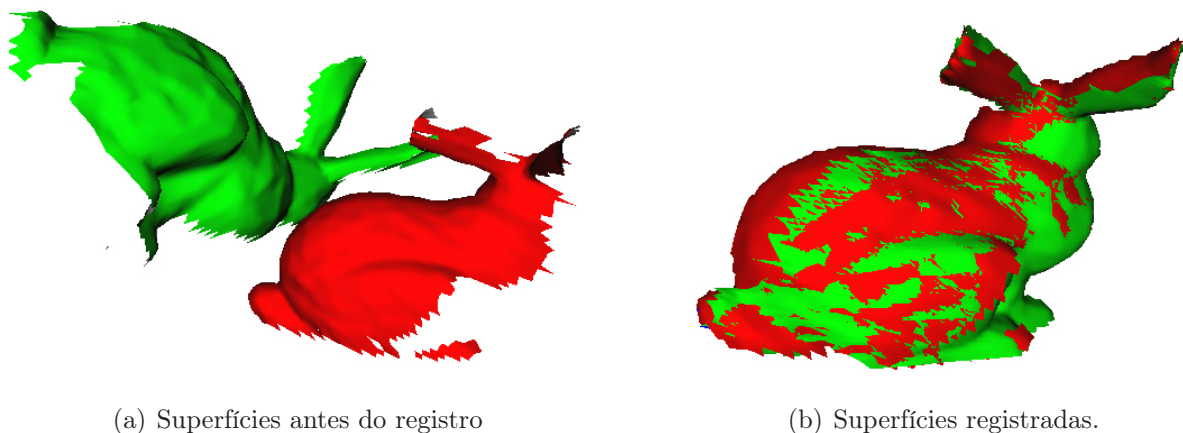


Figura 1.1: Registro de superfícies.

Atualmente, a modelagem de objetos tridimensionais a partir de objetos reais pode ser feita utilizando-se scanners 3D. Esses equipamentos são capazes de capturar pontos tridimensionais a partir de um determinado ponto de visão, utilizando o conceito de *range images*, como em (Levoy et al. 2000) e (Neugebauer 1997).

As *range images* pertencem a uma classe especial de imagens digitais, onde cada pixel da imagem expressa a distância entre o ponto de visão e um ponto da cena. A partir dessas imagens, é possível gerar malhas através de triangulações dos pixels.

Porém, é impossível realizar uma varredura completa de um objeto qualquer a partir de uma única visão, devido a limitações topológicas e geométricas, como obstruções, por exemplo. Isso justifica a necessidade de captura de várias *range images* para a reconstrução

de um objeto (Figura 1.2). Essas range images vão gerar malhas m_i cujos pontos são descritos em sistemas de coordenadas locais S_i . Portanto, essas malhas precisam ser registradas, ou alinhadas, de acordo com um sistema de coordenadas global.

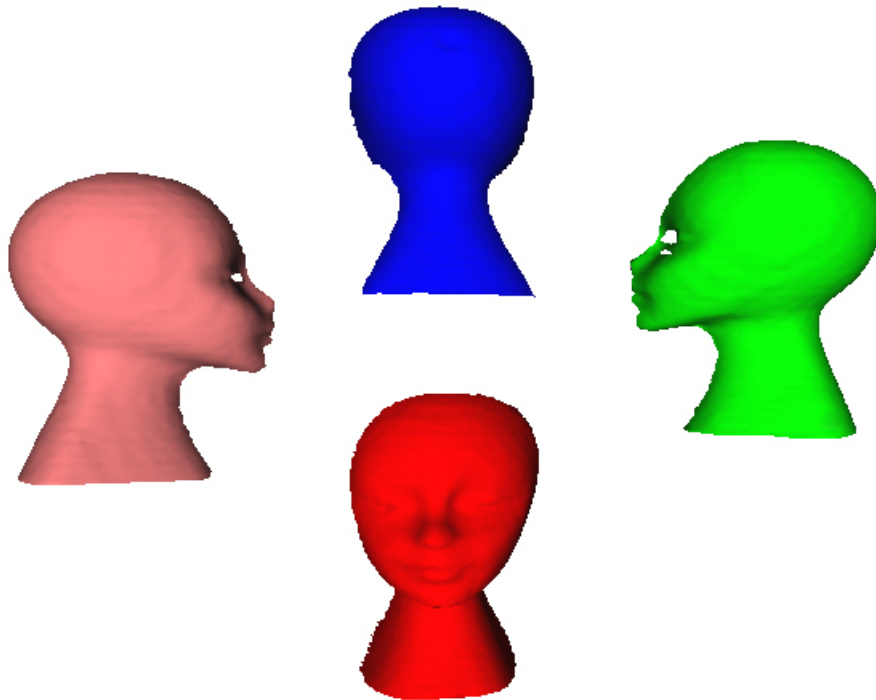


Figura 1.2: Malhas geradas a partir de range images capturadas de diferentes pontos de visão de um objeto.

Do ponto de vista matemático, dadas duas malhas m_A e m_B , onde o sistema de coordenadas de m_A é fixado como sistema de coordenadas global, o registro de superfícies consiste em encontrar o melhor movimento rígido T , composto por uma rotação e uma translação, que, aplicado a m_B , otimize sua posição no sistema de coordenadas global, ou o sistema de coordenadas de m_A .

Atualmente, uma das abordagens mais utilizadas para realizar o registro entre duas malhas parcialmente sobrepostas envolve a análise de similaridades geométricas e de cor (textura) das regiões de sobreposição.

Um dos algoritmos mais utilizados no processo de registro de superfícies é o ICP (*Iterative Closest Point*), desenvolvido por Besl & McKay (1992). Através de um processo iterativo, o algoritmo original ICP cria pares $P_i = \{x_{A_i}, x_{B_i}\}$ de pontos correspondentes nas 2 malhas, usando como critério a proximidade dos pontos, e encontra uma transformação que minimiza o erro de alinhamento entre as 2 malhas. Essa

transformação pode ser calculada minimizando-se a função de erro dada por

$$f(R, t) = \frac{1}{N} \sum_{i=1}^N \|x_{A_i} - (Rx_{B_i} + t)\|^2, \quad (1.1)$$

onde N é a quantidade de pares, e R e t representam a matriz de rotação e o vetor de translação. A transformação encontrada é aplicada as pontos da malha m_B e o processo é repetido até que as transformações encontradas sejam desprezíveis. Este algoritmo será explorado mais detalhadamente no capítulo 4.

Porém, devido a problemas de mínimo local, uma das exigências do ICP é que as malhas tenham uma estimativa inicial do alinhamento, e que apenas a região de sobreposição seja levada em conta. Isso descarta a possibilidade de alinhamento automático usando apenas o ICP.

1.2 Objetivos

Dado um conjunto de malhas que representem um determinado objeto, este trabalho apresenta uma estratégia para alinhar automaticamente todos os pares de malhas que possuam regiões de sobreposição, e finalmente gerar um modelo final. Inicialmente, é utilizada uma estratégia baseada em descritores de superfície *Spin-Images* (Johnson 1997) e *Textured Spin-Images* (Brusco et al. 2005) para geração de correspondências entre pontos das malhas, e conseqüentemente uma estimativa inicial do alinhamento entre malhas com regiões de sobreposição considerável. Em seguida, o algoritmo ICP é executado para refinar o alinhamento. Finalmente, o modelo final é reconstruído usando um método volumétrico desenvolvido por Curless & Levoy (1996) chamado VRIP.

1.3 Contribuições

Este trabalho apresenta uma solução para realizar todo o processo de registro de superfícies e reconstrução de modelos tridimensionais que integra três módulos independentes. A principal contribuição deste trabalho é que a integração dos módulos permite o registro e reconstrução das malhas de forma completamente automática.

O primeiro módulo é responsável pelo alinhamento automático inicial e foi implementado neste trabalho utilizando descritores *spin-images*. Neste trabalho, são apresentadas algumas idéias para melhorar este procedimento, como, por exemplo, utilizar funções das curvaturas principais para selecionar pontos

A segunda etapa realiza o refinamento do alinhamento e também foi implementada neste trabalho, utilizando uma variante do algoritmo ICP.

Na última etapa, os resultados foram obtidos utilizando-se um pacote chamado MeshMerge, desenvolvido por Rocchini et al. (2004).

Finalmente, este trabalho apresenta uma análise de todo o procedimento realizada a partir de alguns experimentos.

1.4 Aplicações

O processo de registro de superfícies é um problema de alto interesse para diversas áreas da ciência, como a robótica, as engenharias, a medicina, a geologia, a modelagem geométrica e a visão computacional.

Os objetos alinhados podem ser utilizados, por exemplo, para localização de modelos, reconhecimento de objetos 3D, rastreamento em tempo real e, finalmente, reconstrução de objetos tridimensionais a partir de malhas obtidas de scanners tridimensionais ou outros equipamentos.

1.5 Estrutura do Trabalho

Este trabalho está estruturado da seguinte forma:

Capítulo 2. Apresenta uma visão geral da estratégia adotada neste trabalho.

Capítulo 3. Contém definições envolvendo os descritores spin-images e seus parâmetros, descreve o procedimento para geração das correspondências entre pontos, incluindo suas filtragens, além do cálculo do alinhamento inicial estimado e validações.

Capítulo 4. Descreve cada etapa do algoritmo ICP original e algumas de suas variantes mais relevantes, dando ênfase à implementação realizada.

Capítulo 5. Descreve o método para reconstrução de modelos a partir de malhas pré-alinhadas, que é implementado no VRIP.

Capítulo 6. Exibe resultados obtidos neste trabalho e discute alguns pontos relevantes do problema.

Capítulo 7. Apresenta as conclusões sobre o trabalho desenvolvido e idéias para trabalhos futuros.

Capítulo 2

Descrição Geral do Método

2.1 Introdução

Este capítulo apresenta uma visão geral do procedimento adotado para registro automático de superfícies e reconstrução de objetos tridimensionais. Esse método tem como dados de entrada um conjunto de malhas que representam diferentes visões de um objeto. Essas malhas podem ser obtidas, por exemplo, a partir de range images capturadas por scanners 3D. Opcionalmente, além da informação de geometria, pode haver informação de textura associada às malhas. Podemos dividir o procedimento em três etapas:

1. Na primeira etapa, as malhas são testadas duas a duas para verificar a existência de áreas de sobreposição e estimar um alinhamento inicial.
2. A segunda etapa recebe as malhas pré-alinhadas e executa o algoritmo ICP para refinar os alinhamentos, minimizando a métrica de erro dada pela Equação 3.23.
3. A terceira etapa aplica um método para reconstrução de uma única malha que represente o objeto, a partir das malhas alinhadas pelo ICP. A malha reconstruída é o resultado final do procedimento.

A Figura 2.1 ilustra o processo de registro automático de superfícies adotado neste trabalho.

2.2 Alinhamento Inicial

Dado um conjunto de malhas $M = \{m_i\}$, onde cada malha é representada em um sistema de coordenadas local, é necessário determinar os pares de malhas que possuem regiões de sobreposição, ou mais especificamente, determinar pares de pontos localizados em diferentes malhas que representem um mesmo ponto no objeto real. Esse procedimento é

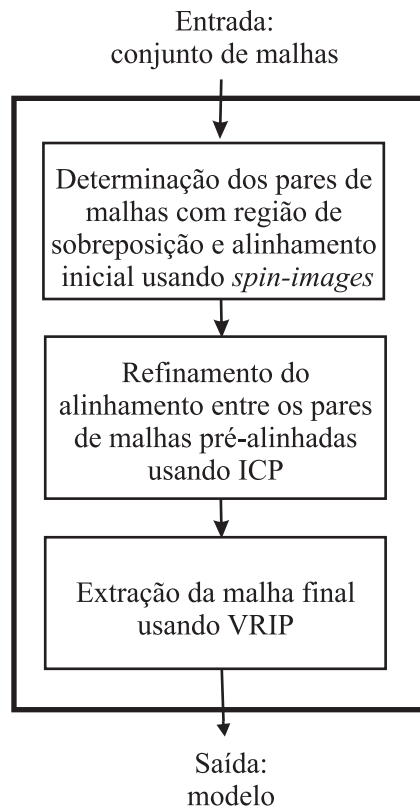


Figura 2.1: Procedimento para registro automático de superfícies

realizado comparando-se as malhas duas a duas, de modo que no final desta etapa, seja possível identificar os pares de malhas com regiões de sobreposição e estimativas de suas transformações de alinhamento.

Neste trabalho, essas correspondências são realizadas utilizando-se descritores *spin-images*, que são imagens bidimensionais criadas por pontos orientados da superfície que fornecem uma descrição local da geometria da superfície e são invariantes por movimentos rígidos. Com esse propósito, foi utilizada também uma extensão das *spin-images*, desenvolvida recentemente, chamada *textured spin-images* (Brusco et al. 2005). Esses descritores consideram, além da geometria, informações de textura dos objetos. O capítulo 6 apresenta uma breve análise comparativa entre os 2 métodos.

A representação de pontos por imagens bidimensionais abre espaço para a utilização de técnicas de comparação de imagens. Pontos correspondentes devem ter imagens parecidas.

A análise bruta de similaridade entre imagens é um algoritmo quadrático, pois envolve a comparação de cada imagem de uma malha com todas as imagens de uma segunda malha. Portanto, é conveniente a escolha de pontos específicos das malhas para geração de *spin-images*. Neste trabalho, a curvatura local da superfície foi o parâmetro utilizado para seleção dos pontos geradores de *spin-images*.

Em seguida, os pares de pontos com *spin-images* similares passam por filtragens, e finalmente são agrupados para o cálculo de uma transformação que melhor aproxime esses pares de pontos. Esse cálculo é feito pelo Algoritmo de Horn (1987).

A transformação encontrada é aplicada à malha, e finalmente é realizada uma validação final para avaliar a qualidade da transformação. Caso a transformação não seja aceita, novos grupos são criados e testados até que se encontre uma transformação consistente ou não seja possível realizar novos agrupamentos. Neste último caso, o alinhamento entre as duas malhas não é realizado, por exemplo, devido a não existência de regiões de sobreposição entre as malhas

A Figura 2.2 exibe um diagrama ilustrando o procedimento acima para cada par de malhas. Esta etapa será descrita em detalhes no capítulo 3.

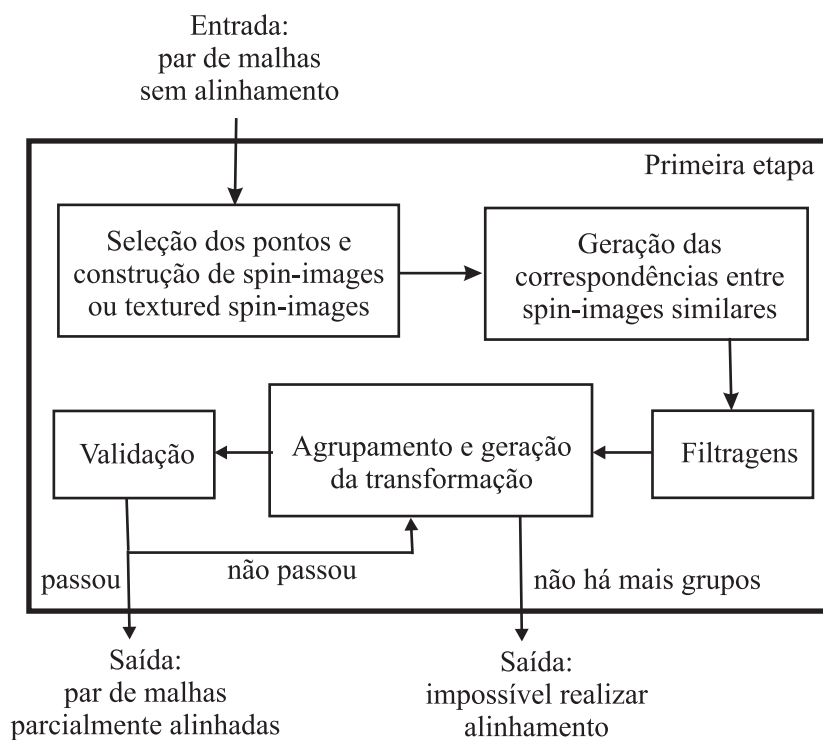


Figura 2.2: Primeira etapa: alinhamento inicial de um par de malhas.

2.3 Refinamento do Registro

Uma vez que as malhas estão pré-alinhadas, é possível detectar as regiões de sobreposição usando distância entre ponto e superfície. Pontos da primeira malha que estejam próximos à segunda malha pertencem à região de sobreposição. Com esses requisitos validados, é possível aplicar o algoritmo ICP para refinar os registros entre as malhas.

Neste trabalho, o algoritmo ICP original foi implementado com a utilização de uma *kd-tree* (Bentley 1975), para acelerar o processo de seleção de pontos mais próximos.

Além disso, foi utilizado um critério de rejeição de pares de acordo com um *threshold* de distância, para eliminar pares fora da região de sobreposição, e a métrica de erro dada pela Equação 3.23.

A Figura 2.3 exibe um diagrama ilustrando os passos do algoritmo ICP, que é o tema principal do capítulo 4.

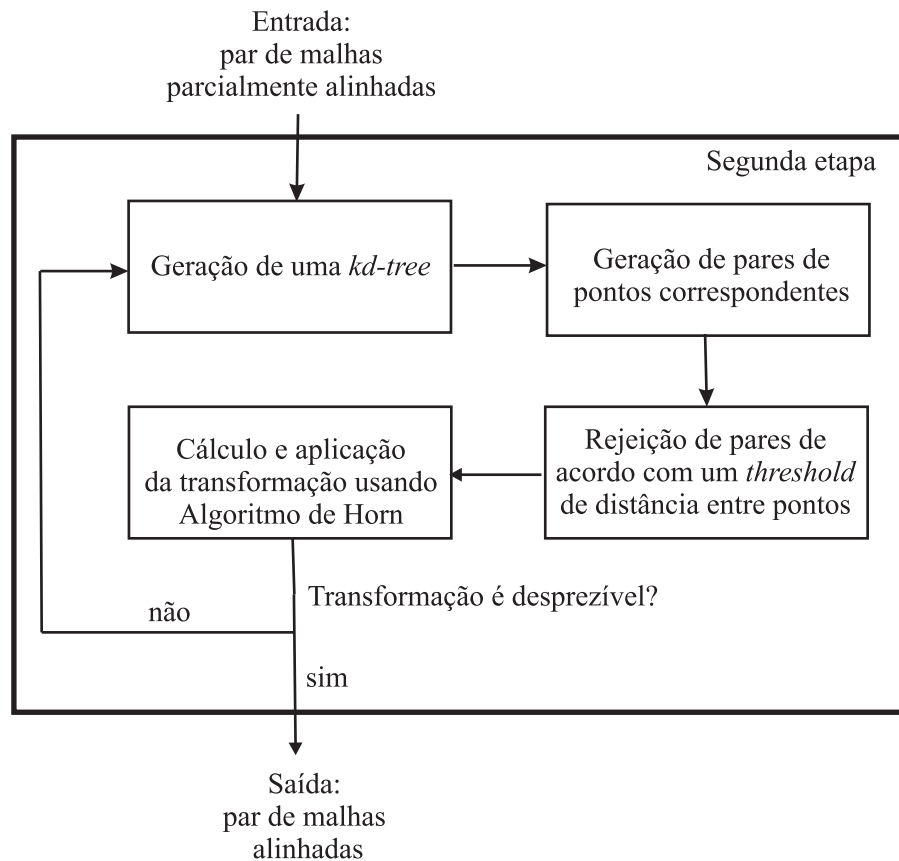


Figura 2.3: Segunda etapa: refinamento do alinhamento entre duas malhas usando ICP.

2.4 Reconstrução da Malha Final

A etapa final do procedimento é a reconstrução de uma única malha a partir das malhas alinhadas pelo ICP. Dados os pares de malhas alinhados e suas respectivas transformações, é possível, através de composição de transformações, representar todas as malhas num sistema de coordenadas global, desde que o grafo de alinhamento seja conexo.

Uma técnica chamada VRIP é capaz de realizar a reconstrução da malha final. Dado um conjunto de malhas representadas num sistema de coordenadas global, esta técnica define um volume e uma função de distância em cada vértice do volume. Finalmente, a malha resultante é extraída aplicando-se o algoritmo Marching Cubes (Lorenson & Cline 1987) com uma tabela para resolver ambiguidades definida por Montani et al. (1994). Esse processo é exibido na Figura 2.4.

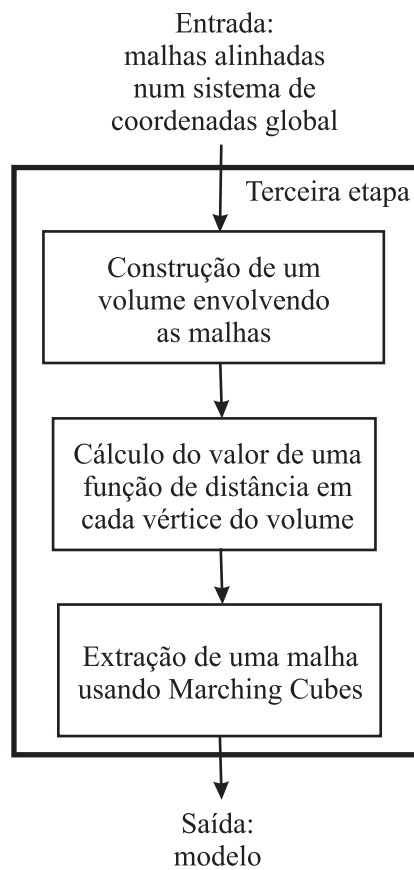


Figura 2.4: Terceira etapa: Construção da malha final usando VRIP.

O capítulo 5 descreve os passos da técnica implementada no VRIP.

Capítulo 3

Alinhamento Automático Inicial

3.1 Introdução

Este capítulo define os descritores *spin-images* e seus parâmetros, descreve os passos para detecção de regiões de sobreposição entre duas malhas, e conseqüentemente suas respectivas transformações espaciais para o alinhamento inicial das superfícies.

3.2 Descritores Spin-Images

Os descritores *spin-images* (Johnson 1997) são imagens bidimensionais que descrevem propriedades globais da geometria de um objeto a partir de bases locais criadas em pontos orientados da superfície. Dentre suas principais características, destaca-se a invariância por movimentos rígidos, ou seja, as *spin-images* de uma superfície não variam se nela forem aplicadas transformações de rotação e translação.

Essa característica implica que, dadas duas malhas extraídas de *range images* criadas a partir de diferentes visões de um mesmo objeto, suas *spin-images* serão iguais em pontos correspondentes, ou seja, pontos que representem um mesmo ponto do objeto real.

Portanto, considerando o problema de registro de superfícies, as *spin-images* se apresentam como boas candidatas para realizar a descrição de pontos das superfícies. Desse modo, a geração de correspondências entre pontos tridimensionais pertencentes a malhas extraídas de diferentes visões de um objeto pode ser feita através da comparação de imagens bidimensionais, pois pontos correspondentes devem ter *spin-images* semelhantes.

Outros problemas comuns no processo de registro de superfícies, como a presença de oclusões, buracos e ruídos podem ser parcialmente resolvidos com o uso das *spin-images*, e serão discutidos neste capítulo.

As subseções seguintes descrevem os passos para geração das *spin-images* e discutem seus principais parâmetros. Além disso, os descritores *textured spin-images* serão definidos como uma extensão das *spin-images*.

3.2.1 Spin-maps

Um ponto orientado numa superfície é um ponto associado a uma direção. Para a criação de *spin-images*, associa-se aos pontos da superfície o seu vetor normal. Este vetor pode ser calculado, por exemplo, como o vetor normal a um plano de mínimos quadrados que melhor se encaixa nos vértices vizinhos ao ponto.

Cada ponto orientado p com vetor normal n define uma sistema de coordenadas local, usando o plano P , tangente à superfície em p , e a reta L , que passa por p e é paralela a n . A base desse sistema de coordenadas, (α, β) , é definida pela distância perpendicular à reta L , e pela distância perpendicular com sinal até o plano P , respectivamente (Figura 3.1). Isso define um sistema de coordenadas cilíndricas, onde não é possível obter a coordenada de ângulo polar.

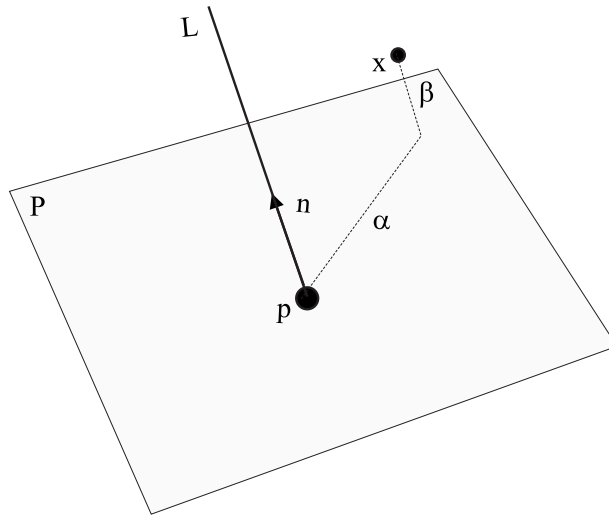


Figura 3.1: Base criada pelo ponto orientado (p, n) e coordenadas (α, β) de um ponto x .

Transformando pontos da superfície em pontos no sistema de coordenadas definido acima, teremos uma aplicação denominada *spin-map*. Essa aplicação projeta pontos da superfície em pontos bidimensionais, dado um ponto orientado $O = (p, n)$. Portanto, um *spin-map* é uma aplicação $S_O: \mathbb{R}^3 \rightarrow \mathbb{R}^2$, que mapeia

$$S_O(x) \rightarrow (\alpha, \beta) = \left(\sqrt{\|x - p\|^2 - (n \cdot (x - p))^2}, n \cdot (x - p) \right). \quad (3.1)$$

A imagem da aplicação *spin-map* será utilizada para a criação da *spin-image*. Desde já, é interessante ressaltar que, como o sistema de coordenadas utilizado depende de uma base atrelada a um ponto orientado da superfície, esta aplicação é invariante por movimentos rígidos da superfície.

3.2.2 Spin-images

A imagem de uma aplicação *spin-map* é um conjunto de pontos bidimensionais, projetados a partir de pontos da superfície, que descreve a geometria do objeto.

Porém, conjuntos desse tipo não são modelos ideais para realização de comparações, devido a variações de amostragem, visão e ruído, por exemplo, o que não garante a exata posição entre vértices em duas malhas obtidas de diferentes visões. Por outro lado, a geometria global da superfície será a mesma.

Para extrair as propriedades globais desses conjuntos desconsiderando os efeitos de variações locais em posições de vértices, é realizada uma discretização da imagem da aplicação *spin-map*.

Nessa discretização, o espaço bidimensional é dividido em pequenas caixas, ou *bins*. Observando este espaço como uma imagem, onde cada pixel é determinado por um *bin*, determina-se o tom de cinza de cada pixel de acordo com a quantidade de pontos contidos em seu *bin*. A esta imagem, daremos o nome de *spin-image*. A Figura 3.2 exhibe o conjunto imagem de uma aplicação *spin-map* e sua respectiva *spin-image*.

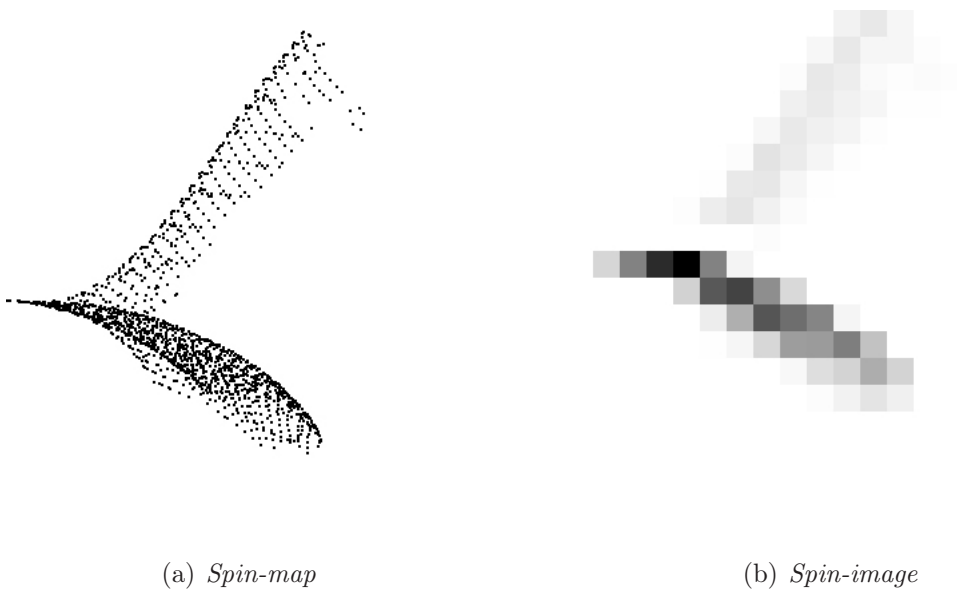


Figura 3.2: *Spin-map* com sua respectiva *Spin-image*.

Para suavizar o processo de acúmulo de pontos nos *bins*, deixando a imagem menos sensível à posição exata dos pontos, é interessante utilizar uma interpolação bilinear para distribuir valores pelos *bins* vizinhos, com peso determinado pela posição relativa do ponto no *bin*.

3.2.3 Parâmetros das Spin-imagens

Alguns parâmetros são fundamentais para o sucesso na geração de correspondências entre pontos utilizando *spin-images*. Destacam-se o tamanho dos *bins*, ou *bin size*, a largura da imagem, ou *image width*, e o ângulo suporte.

Tamanho dos *bins*

O tamanho dos *bins*, ou das caixas, é fundamental para a criação de uma *spin-image* com uma boa descrição da superfície. *Bins* grandes podem tornar a descrição da superfície não muito clara. *Bins* pequenos podem não ser suficientes para anular os efeitos das posições individuais dos pontos.

A maneira mais natural de medir o tamanho dos *bins* é como um múltiplo da resolução da malha, que é definida como a mediana dos comprimentos de todas as arestas da malha.

A Figura 3.3 exibe três *spin-images* obtidas de um mesmo ponto orientado com diferentes tamanhos de *bins*. Na Figura 3.3(a), é possível ver claramente que o tamanho pequeno dos *bins* não resolve o problema da posição individual dos pontos. Na Figura 3.3(b), *bins* grandes prejudicam a descrição da superfície. Finalmente, a Figura 3.3(c) é um exemplo de descrição adequada da superfície por *spin-images*.

Largura da imagem

A largura da imagem, ou *image width*, determina a quantidade de *bins*, ou pixels da imagem, na horizontal e na vertical. Uma vez fixado o tamanho dos *bins*, quanto maior a largura da imagem, maior a região da superfície que será descrita pela *spin-image*.

Denomina-se distância suporte o produto do tamanho dos *bins* pela largura da imagem. Essa distância indica o quão distante os pontos da superfície podem estar do ponto orientado para fazerem parte da *spin-image*. Portanto, este é outro parâmetro relevante para alcançar uma descrição adequada da superfície.

Larguras muito altas, ou distâncias suporte muito altas, podem gerar *spin-images* diferentes quando geradas de pontos correspondentes de diferentes visões, pois cada visão possui suas próprias regiões de oclusão. Por outro lado, larguras muito baixas, ou distâncias suporte muito baixas, descrevem regiões pequenas da superfície, que podem, inclusive, ser semelhantes a regiões diferentes da mesma superfície, o que geraria falsas correspondências.

Ângulo suporte

O ângulo suporte indica o ângulo máximo entre a normal do ponto orientado da base da *spin-image* e as normais dos pontos da superfície. Pontos com ângulos maiores serão desconsiderados na criação das *spin-images*.

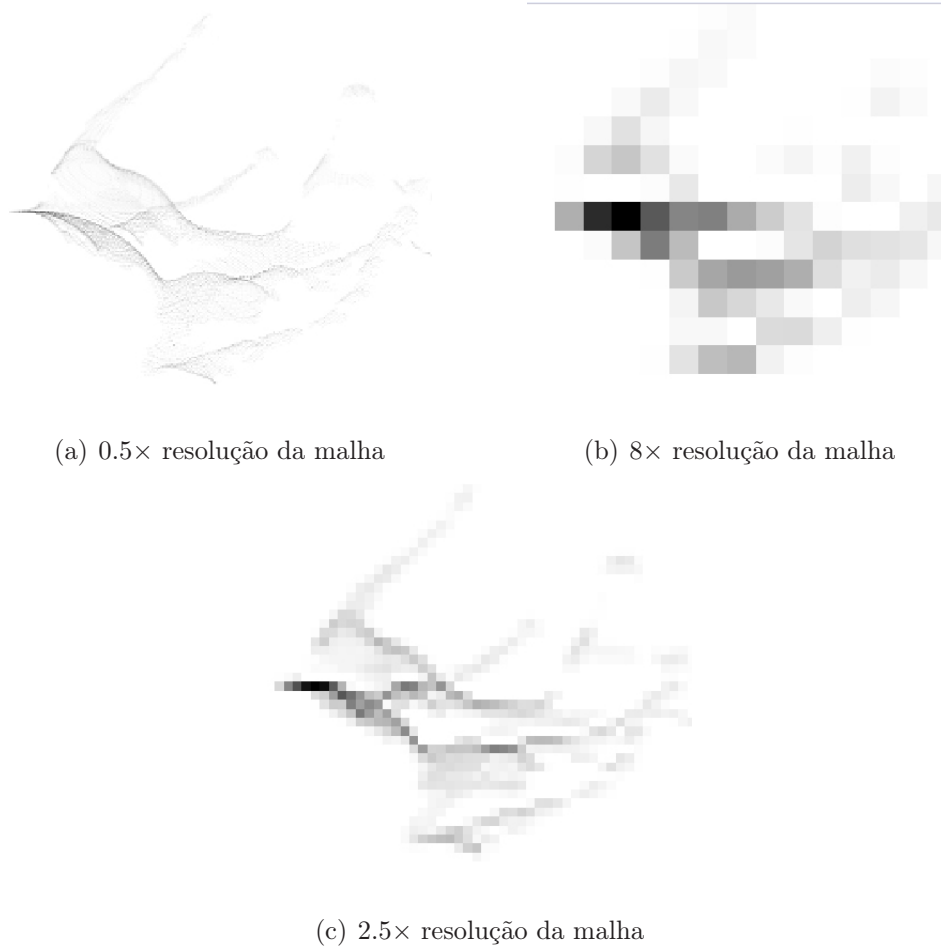


Figura 3.3: Efeito de diferentes *bin sizes* em *spin-images*.

A importância deste parâmetro é semelhante à importância da distância suporte para excluir pontos de regiões distantes da base.

3.2.4 Textured Spin-Images

As *Textured Spin-Images*, ou TSI, são extensões das *spin-images* que consideram informação de textura na formação das imagens. Sua formulação é parecida com a original.

Seja x um ponto da superfície e $I(x)$ a luminância associada aos valores (R, G, B) da textura da superfície em x . É necessário quantizar a luminância $I(x)$ em L intervalos de tons de cinza l_i .

Um *spin-map* com textura é uma extensão da Equação 3.1, definido pela aplicação

$$TSI_O(x) \rightarrow (\alpha, \beta, l_i) = \left(\sqrt{\|x - p\|^2 - (n \cdot (x - p))^2}, n \cdot (x - p), l_i \right). \quad (3.2)$$

A partir do *spin-map* com textura, é possível criar uma *spin-image* com textura. Porém, neste caso os *bins* não vão acumular a quantidade de pontos que caem em cada um, mas seus tons de cinza quantizados l_i .

Além disso, fazendo $L = 1$, existirá apenas um intervalo $[0, 1]$ de tons de cinza, o que caracteriza as *spin-images* convencionais. Desse modo, as *spin-images* podem ser consideradas um caso particular das *spin-images* com textura.

A Figura 3.4 exibe um *spin-map* e suas *spin-images* convencional e com textura, com $L = 4$.

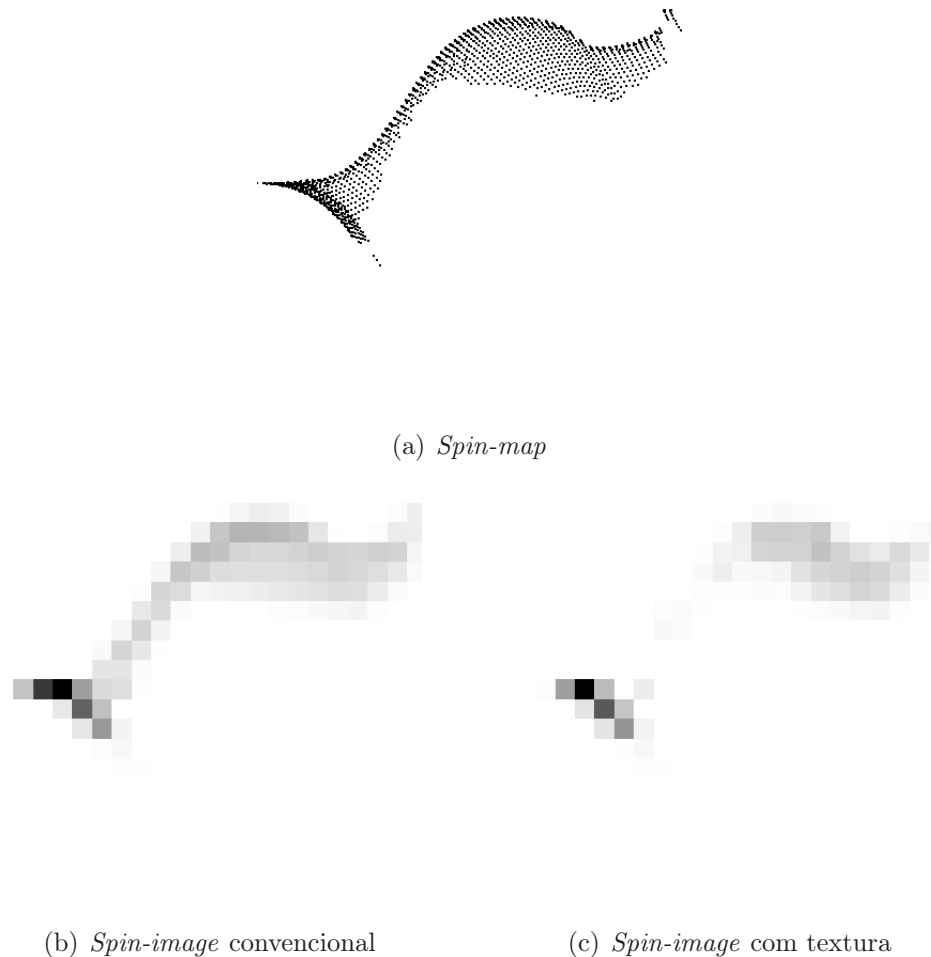


Figura 3.4: *Spin-map* e *spin-image* convencional e com textura.

3.2.5 Comparação de Spin-Images

Naturalmente, imagens criadas por pontos correspondentes pertencentes a diferentes visões não serão exatamente iguais. É necessário aplicar técnicas para medir a similaridade entre imagens, e criar correspondências entre os pares de pontos que apresentem imagens mais parecidas.

Neste trabalho, as técnicas de comparação de imagens utilizadas foram aquelas adotadas por Johnson (1997).

Define-se o coeficiente de correlação linear entre duas imagens P e Q pela função

$$R(P, Q) = \frac{N \sum p_i q_i - \sum p_i \sum q_i}{\sqrt{(N \sum p_i^2 - (\sum p_i)^2) (N \sum q_i^2 - (\sum q_i)^2)}}, \quad (3.3)$$

onde p_i e q_i são os tons de cinza dos pixels de P e Q , e N é a quantidade de pixels da imagem. Este coeficiente deve variar entre -1 , quando as imagens são totalmente diferentes, e 1 , quando as imagens forem idênticas. Portanto, valores mais altos de R indicam imagens mais parecidas.

Porém, nem sempre se espera que *spin-images* de pontos correspondentes sejam similares por toda a imagem. Isso ocorre, por exemplo, por que uma *spin-image* criada a partir de uma determinada visão pode não conter dados de uma região que uma outra visão pode ter, ou seja, espera-se que as visões tenham oclusões.

Essas oclusões ocorrem normalmente em regiões inteiras das superfícies, e implicam que regiões das *spin-images* não contenham esses dados. Seguindo esse raciocínio, uma maneira de resolver parcialmente esse problema é desconsiderar *bins* que não contenham dados em alguma das imagens P ou Q no cálculo do coeficiente de correlação linear.

Porém, utilizando-se puramente o coeficiente de correlação linear, alguns casos apresentarão resultados não desejados. Por exemplo, duas imagens com pequena região de sobreposição podem ter um coeficiente alto c_1 , se apenas estão região de sobreposição for parecida, enquanto duas imagens com grande região de sobreposição podem ter um coeficiente menor que c_1 , se forem um pouco menos parecidas em sua grande região de sobreposição. Faz-se necessário, então, o uso de uma outra técnica que leve em conta a quantidade de pixels na região de sobreposição.

Neste caso, uma técnica que resolve bem este problema é a medida de similaridade, que considera, além do coeficiente de correlação linear, sua variância, para medir sua confiabilidade. Define-se a medida de similaridade pela função

$$C(P, Q) = (\operatorname{atanh}(R(P, Q)))^2 - \lambda \left(\frac{1}{N - 3} \right), \quad (3.4)$$

onde N é a quantidade de pixels na região de sobreposição e λ é um peso associado à variância.

Para estimar λ , é necessário, para cada *spin-image* criada, medir a quantidade de *bins* que possuem valor não nulo. Define-se λ como a metade da mediana desses valores.

3.3 Seleção de Pontos

O processo de criação e comparação de *spin-images* tem alto custo computacional e de armazenamento na memória. Seja m a quantidade de pontos da primeira malha, e n a quantidade de pontos da segunda malha. Para a geração de correspondências entre

pontos dessas malhas, é necessário criar e armazenar m *spin-images* de uma malha e n *spin-images* da segunda malha, e comparar todas as *spin-images* de uma malha com todas as *spin-images* da outra malha. Esse procedimento resulta num algoritmo de ordem quadrática.

Portanto, criar *spin-images* para todos os pontos das malhas torna-se um processo inviável, tanto computacionalmente, quanto em termos de armazenamento na memória. Uma maneira de resolver esse problema é efetuar uma seleção de pontos nas malhas para criação e comparação de *spin-images*.

Neste trabalho, foram implementadas duas técnicas para seleção de pontos nas malhas. A primeira é a seleção aleatória de pontos em ambas as malhas. A segunda é a seleção de acordo com a curvatura local da superfície.

As subseções seguintes descrevem essas duas técnicas.

3.3.1 Seleção Aleatória de Pontos

A seleção aleatória é um procedimento simples onde uma determinada proporção de pontos das malhas é selecionado aleatoriamente para criação de *spin-images*.

Porém, esse método não garante que os pontos selecionados em uma malha serão iguais aos pontos selecionados na segunda malha. Isso pode tornar impossível o registro das malhas. Uma variante para resolver esse problema é selecionar todos os pontos de uma malha, e selecionar aleatoriamente uma pequena proporção na segunda malha. Ainda assim, nada garante que os pontos selecionados aleatoriamente na segunda malha façam parte de uma região do objeto que não pertença à primeira malha.

3.3.2 Seleção por Curvatura Local

A seleção por curvatura local da superfície é um processo mais elaborado que envolve cálculos de curvatura em todos os pontos das malhas. Uma proporção dos pontos com maior curvatura é selecionado em ambas as malhas para a criação de *spin-images*.

Neste trabalho, foram analisadas a curvatura gaussiana $\kappa_1\kappa_2$, a curvatura média $\frac{\kappa_1+\kappa_2}{2}$, a curvatura principal máxima κ_1 e a soma dos quadrados das curvaturas principais $\kappa_1^2 + \kappa_2^2$. A medida que apresentou melhores resultados na geração das correspondências foi a soma dos quadrados das curvaturas principais. A Figura 3.5 exhibe uma malha com seus 15% pontos de maior medida $\kappa_1^2 + \kappa_2^2$ destacados.

Seguem abaixo alguns conceitos preliminares da Geometria Diferencial e a técnica utilizadas para cálculo das curvaturas, baseada em (Meyer et al. 2003).

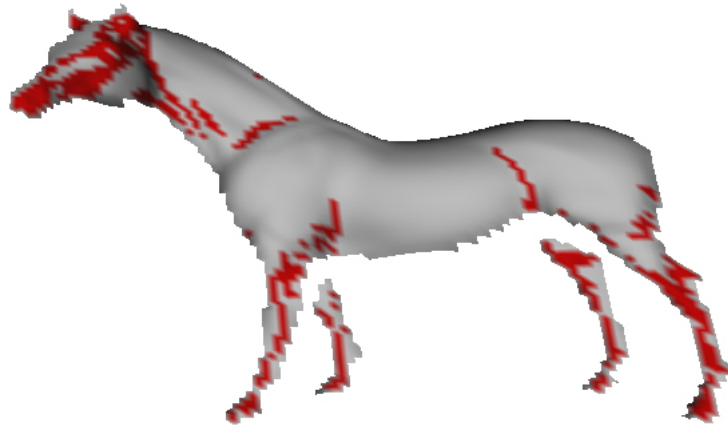


Figura 3.5: Malha com pontos de alta medida de curvatura $\kappa_1^2 + \kappa_2^2$ destacados.

Conceitos de Geometria Diferencial

Seja S uma superfície diferenciável bidimensional em \mathbb{R}^3 . Para cada ponto $p \in S$, é possível calcular seu plano tangente, ortogonal a seu vetor normal n .

Para cada direção unitária e_θ do plano tangente, define-se a curvatura normal $\kappa^N(\theta)$ como a curvatura da curva que é a intersecção da superfície com o plano contendo n e e_θ .

As curvaturas principais κ_1 e κ_2 serão, respectivamente, o maior e o menor valor das curvaturas normais, considerando-se todas as direções e_θ em p .

A curvatura média em p é definida como a média das curvaturas normais de p , de acordo com a equação

$$\kappa_H = \frac{1}{2\pi} \int_0^{2\pi} \kappa^N(\theta) d\theta. \quad (3.5)$$

A partir dessas definições, é possível formular uma famosa definição da curvatura média, dada por

$$\kappa_H = \frac{\kappa_1 + \kappa_2}{2}. \quad (3.6)$$

Finalmente, a curvatura Gaussiana é definida pela equação

$$\kappa_G = \kappa_1 \kappa_2. \quad (3.7)$$

A técnica utilizada neste trabalho, faz uso de uma importante relação entre minimização da área da superfície e a curvatura média, dada pela equação

$$2\kappa_H n = \lim_{\text{diam}(A) \rightarrow 0} \frac{\nabla \mathcal{A}}{\mathcal{A}}, \quad (3.8)$$

onde \mathcal{A} é um pequeno elemento de área orientada.

O operador $K(p) = 2\kappa_H(p)n(p)$ é chamado de operador de Laplace-Beltrami. Uma vez conhecido o vetor $K(p)$, o cálculo do vetor normal $n(p)$ e da curvatura $\kappa_H(p)$ são imediatos.

Descrições mais detalhadas dos conceitos acima são encontradas em (Carmo 2005).

Médias espaciais

As definições acima devem ser adaptadas para superfícies discretas, ou malhas. Nesta técnica, essa adaptação é feita utilizando médias espaciais. Neste caso, as propriedades de um ponto serão analisadas de acordo com um espaço contido em sua vizinhança imediata, ou vizinhança estrelada. Com essa idéia, a curvatura Gaussiana discreta $\hat{\kappa}_G$, por exemplo, é definida pela equação

$$\hat{\kappa}_G = \frac{1}{\mathcal{A}} \iint_{\mathcal{A}} \kappa_G d\mathcal{A}, \quad (3.9)$$

onde \mathcal{A} é uma determinada área contida na vizinhança estrelada do ponto.

As áreas utilizadas para o cálculo das médias espaciais são regiões de Voronoi. Considere o conjunto de faces triangulares, vértices e arestas que formam a vizinhança estrelada de um ponto. Os polígonos que determinam os elementos de área são formados pelo ponto central da estrela, os pontos médios das arestas que conectam cada vértice com o ponto central, e os circuncentros dos triângulos. A união desses polígonos é chamada região de Voronoi, ou $\mathcal{A}_{Voronoi}$, como ilustrado na Figura 3.6.

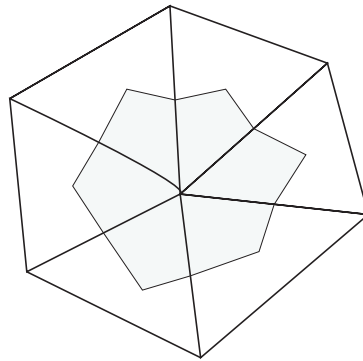


Figura 3.6: Região de Voronoi de uma vizinhança estrelada.

O cálculo da área de uma região de Voronoi é feita triângulo a triângulo. Se os triângulos da vizinhança estrelada não são obtusos, é possível utilizar propriedades geométricas simples para chegar à equação

$$\mathcal{A}_{Voronoi} = \frac{1}{8} \sum_{x_j \in N_1(i)} (\cot \alpha_{ij} + \cot \beta_{ij}) \|x_i - x_j\|^2. \quad (3.10)$$

Caso existam triângulos obtusos, a área deve ser calculada triângulo a triângulo. Os triângulos não obtusos terão área igual à área de Voronoi neste triângulo. Se um triângulo

obtusos tiverem ângulo obtuso no ponto central da estrela, a área será metade da área do triângulo. Senão, sua área será um quarto da área do triângulo. Essa nova área será chamada \mathcal{A}_{Mixed} .

Curvatura Média Discreta

Para a obtenção das normais da superfície e curvatura média, o operador Laplace-Beltrami será calculado. Inicialmente, o Laplaciano da superfície em relação aos parâmetros u e v é calculado, usando a discretização da superfície como espaço paramétrico conforme. Desse modo, é possível obter a relação

$$\iint_{\mathcal{A}_M} K(x) dA = - \iint_{\mathcal{A}_M} \Delta_{u,v} x du dv. \quad (3.11)$$

Usando o teorema de Gauss, chega-se à formulação final

$$\iint_{\mathcal{A}_M} K(x) dA = \frac{1}{2} \sum_{x_j \in N_1(i)} (\cot \alpha_{ij} + \cot \beta_{ij})(x_i - x_j), \quad (3.12)$$

onde α_{ij} e β_{ij} são os ângulos opostos à aresta determinada pelos pontos (x_i, x_j) e $N_1(i)$ é o conjunto de vértices na vizinhança estrelada do vértice x_i .

Usando a região \mathcal{A}_{Mixed} e a Equação 3.12, chega-se ao operador

$$K(x_i) = \frac{1}{2\mathcal{A}_{Mixed}} \sum_{x_j \in N_1(i)} (\cot \alpha_{ij} + \cot \beta_{ij})(x_i - x_j). \quad (3.13)$$

A partir deste operador, a curvatura média κ_H é facilmente obtida extraindo-se metade da norma de K .

Curvatura Gaussiana Discreta

A seguinte equação é consequência do Teorema de Gauss-Bonnet:

$$\iint_{\mathcal{A}_M} \kappa_G dA = 2\pi - \sum_j \epsilon_j, \quad (3.14)$$

onde ϵ_j são os ângulos externos da fronteira da região. A partir dessa equação, é possível chegar à forma

$$\iint_{\mathcal{A}_M} \kappa_G dA = 2\pi - \sum_{j=1}^n \theta_j, \quad (3.15)$$

onde θ_j é o ângulo do triângulo j no vértice x_i e n é a quantidade de triângulos.

Finalmente, o operador de curvatura Gaussiana discreto é dado por

$$\kappa_G(x_i) = \frac{1}{A_{Mixed}} \left(2\pi - \sum_{j=1}^n \theta_j \right). \quad (3.16)$$

Curvaturas Principais Discretas

Tendo conhecimento das curvaturas média e Gaussiana, o cálculo das curvaturas principais é direto, dado pelas equações

$$\kappa_1(x_i) = \kappa_H(x_i) + \sqrt{\Delta(x_i)} \quad (3.17)$$

$$\kappa_2(x_i) = \kappa_H(x_i) - \sqrt{\Delta(x_i)}, \quad (3.18)$$

onde $\Delta(x_i) = \kappa_H^2(x_i) - \kappa_G(x_i)$.

3.4 Geração de Correspondências

Após a criação das *spin-images* de pontos selecionados, o próximo passo é efetuar as comparações entre as *spin-images*. Para cada ponto p associado a uma *spin-image*, a maneira mais simples de encontrar seu ponto correspondente seria encontrar o ponto cuja imagem tenha a maior medida de similaridade com sua imagem, ou seja, a imagem mais parecida. Porém, nem sempre imagens parecidas implicam pontos correspondentes, devido à simetria de objetos, por exemplo.

A técnica utilizada neste trabalho para encontrar correspondências baseia-se na análise de um histograma das medidas de similaridade (Johnson 1997). Bons candidatos a correspondências se encontram no topo do histograma. Portanto, é necessário definir um *threshold* para separar esses bons candidatos, ou *outliers* do histograma.

Estatisticamente, distribuições bem comportadas, como é o caso das medidas de similaridade das *spin-images*, podem ter seus *outliers* identificados pelo estudo dos *quartis*. Seja f_s a diferença entre o maior quartil e o menor quartil. Os *outliers* extremos encontram-se $3f_s$ acima do maior quartil (Devore 1999). Este valor será justamente o *threshold* utilizado para identificar *outliers*.

A Figura 3.7 exhibe um exemplo de histograma, destacando-se a mediana m , o menor e maior quartil q_1 e q_2 , respectivamente, o valor de $f_s = q_2 - q_1$, e o *threshold* de distância $q_2 + 3f_s$. Os valores destacados em tom mais escuro são os *outliers*.

Uma vez identificados, os *outliers* serão dados como pontos correspondentes de p . A próxima seção trata da filtragem aplicada a essas correspondências para eliminar falsos pares de pontos.

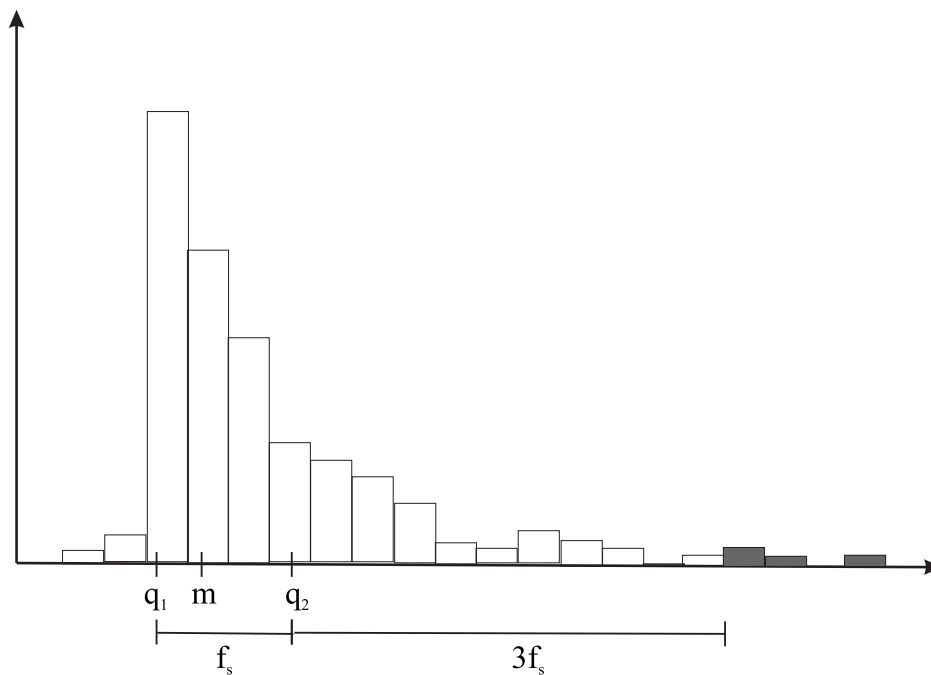


Figura 3.7: Exemplo de histograma utilizado para detecção de *outliers*

3.5 Filtragens de Correspondências

Já foi visto que pontos correspondentes devem ter *spin-images* parecidas. Porém, a recíproca nem sempre é verdadeira. Duas razões frequentes para que um ponto encontre mais de um ponto correspondente pela análise das medidas de similaridade são: a simetria do modelo e o fato de pontos próximos criarem *spin-images* parecidas. Além disso, o ponto pode pertencer a uma região fora da região de sobreposição entre as malhas. Neste caso, não existem pontos correspondentes na outra malha, o que contribui para a criação de falsas correspondências.

Portanto, é necessário realizar filtragens que garantam a consistência das correspondências. Neste trabalho são realizadas duas filtragens. A primeira baseia-se numa análise estatística das medidas de similaridade das correspondências. A segunda realiza um teste de consistência geométrica entre os pontos correspondentes. As correspondências que passam por essas duas filtragens são agrupadas para o cálculo de uma transformação que melhor alinhe os pontos correspondidos. As subseções seguintes descrevem essas filtragens e o critério utilizado para agrupar correspondências.

3.5.1 Filtragem Usando Medidas de Similaridade

Seja M o conjunto das medidas de similaridade das imagens das correspondências, e m o maior valor desse conjunto. Esta filtragem elimina todas as correspondências que apresentem medida de similaridade abaixo de uma certa fração de m . Neste trabalho, as

correspondências com medida de similaridade abaixo da metade de m foram descartadas nesta primeira filtragem.

As correspondências envolvendo pontos contidos em regiões onde não há sobreposição entre as superfícies devem ser descartados nesta filtragem, pois como não há correspondências consistentes envolvendo tais pontos, suas medidas de similaridade devem ser baixas, provavelmente abaixo da metade de m .

3.5.2 Filtragem Usando Consistência Geométrica

Após a primeira filtragem, as correspondências devem passar por uma filtragem de consistência geométrica.

Duas correspondências $C_1 = [s_1, m_1]$ e $C_2 = [s_2, m_2]$ serão consideradas geometricamente consistentes se $|\|s_1 - s_2\| - \|m_1 - m_2\|| < \varepsilon$, para ε suficientemente pequeno, onde s_1 e m_1 são pontos correspondentes das malhas, assim como s_2 e m_2 . Em termos de coordenadas *spin-map*, isso é equivalente a analisar as funções

$$d_{gc}(C_1, C_2) = \frac{\|S_{m_2}(m_1) - S_{s_2}(s_1)\|}{(\|S_{m_2}(m_1)\| + \|S_{s_2}(s_1)\|) / 2}, \quad (3.19)$$

$$D_{gc}(C_1, C_2) = \max(d_{gc}(C_1, C_2), d_{gc}(C_2, C_1)), \quad (3.20)$$

onde S é a aplicação *spin-map* definida pela Equação 3.1 e a função *max* retorna o maior de seus parâmetros. O quociente da Equação 3.19 normaliza a diferença das distâncias de acordo com a média das normas das coordenadas *spin-map*. Se D_{gc} é pequeno, então as correspondências são geometricamente consistentes.

Para realizar a filtragem em várias correspondências, é necessário criar um procedimento que teste cada correspondência C com todas as outras, e verificar se uma quantidade razoável de correspondências é consistente com C . Em caso afirmativo, C passa pela filtragem. Em caso negativo, C é removida e o teste continua.

Finalmente, é necessário definir dois parâmetros. O primeiro é chamado *threshold* de consistência geométrica (T_{gc}). Se $D_{gc}(C_1, C_2) < T_{gc}$, então C_1 e C_2 são considerados geometricamente consistentes.

O segundo parâmetro, $prop_{GC}$, determina a proporção mínima de correspondências que devem ser geometricamente consistentes com C para que C passe na filtragem.

3.5.3 Agrupamento

Após as filtrações, as correspondências apresentam um bom grau de consistência. O próximo passo a ser executado é o agrupamento de algumas correspondências utilizando critérios que garantam o cálculo de uma boa transformação. Cada grupo criado com pelo menos duas correspondências deve ser capaz de gerar uma transformação.

Nesta etapa, serão utilizados dois critérios: a consistência geométrica, através da função d_{gc} , e a distância entre as correspondências. Quanto mais espalhadas as correspondências estiverem, maiores as chances de se obter uma boa transformação, minimizando o efeito de ruídos. Esses dois requisitos são combinados para gerar as funções

$$w_{gc}(C_1, C_2) = \frac{d_{gc}(C_1, C_2)}{1 - e^{-((\|S_{m_2}(m_1)\| + \|S_{s_2}(s_1)\|)/(2\gamma))}} \quad (3.21)$$

$$W_{gc}(C_1, C_2) = \max(w_{gc}(C_1, C_2), w_{gc}(C_2, C_1)) \quad (3.22)$$

Se C_1 e C_2 forem geometricamente consistentes e distantes, W_{gc} será pequeno. A variável γ deve ser definida em k vezes a resolução da malha. Esse valor estimula o agrupamento de correspondências distantes pelo menos k vezes a resolução da malha. Neste trabalho, foi utilizado $k = 4$.

Seja n a quantidade de correspondências. Será possível realizar n agrupamentos, um para cada correspondência. Segue abaixo o procedimento para geração do grupo de uma correspondência C .

Seja G_i o grupo gerado por C_i e T_{gc} o *threshold* de consistência geométrica definida anteriormente. Este grupo deve ser inicializado com a correspondência C_i , ou seja, $G_i = \{C_i\}$. Para cada correspondência C_j , se $W_{gc}(C_i, C_j) < T_{gc}$, adiciona-se C_j ao grupo G_i .

Cada grupo vai gerar uma transformação, que ao ser aplicada à malha deve passar por uma validação final. Uma vez encontrada uma transformação válida, o processo pára, e esta transformação será o alinhamento inicial resultante. A seção seguinte descreve o Algoritmo de Horn, responsável pelo cálculo de uma transformação dado um grupo de correspondências.

3.6 Cálculo da Transformação: Algoritmo de Horn

Dado um grupo de correspondências, formadas por pontos em dois sistemas de coordenadas locais, a saber, os sistemas de coordenadas das duas malhas, é necessário encontrar o movimento rígido que minimize a distância entre os pares de pontos. Uma solução para esse problema usando mínimos quadrados foi desenvolvida por Horn (1987), e será brevemente descrita nesta seção.

Neste método, as rotações são representadas por quatérnios, que podem ser vistos como vetores no R^4 , e as translações são representadas por vetores do \mathbb{R}^3 . O vetor de registro de uma superfície será denotado por um vetor $\vec{q} = [\vec{q}_R | \vec{q}_T]^t$, onde \vec{q}_R é um quatérnio representando a rotação, e \vec{q}_T é um vetor do \mathbb{R}^3 representando a translação.

O objetivo desse método é encontrar a transformação \vec{q} que minimize uma função de mínimos quadrados dada pela equação

$$f(\vec{q}) = \frac{1}{n} \sum_{i=1}^n \|p_{2,i} - R(\vec{q}_R)\vec{p}_{1,i} - \vec{q}_T\|^2, \quad (3.23)$$

onde n é a quantidade de correspondências, $p_{k,i}$ são pontos da malha k , e $R(\vec{q}_R)$ é a matriz de rotação associada ao quatérnio \vec{q}_R . Essa matriz é dada pela fórmula

$$R(\vec{q}_R) = R(q_0, q_1, q_2, q_3) = \begin{pmatrix} q_0^2 + q_1^2 - q_2^2 - q_3^2 & 2(q_1q_2 - q_0q_3) & 2(q_1q_3 + q_0q_2) \\ 2(q_1q_2 + q_0q_3) & q_0^2 + q_2^2 - q_1^2 - q_3^2 & 2(q_2q_3 - q_0q_1) \\ 2(q_1q_3 - q_0q_2) & 2(q_2q_3 + q_0q_1) & q_0^2 + q_3^2 - q_1^2 - q_2^2 \end{pmatrix}.$$

Este método encontra a melhor rotação através da análise dos autovetores de uma matriz simétrica Q 4×4 .

Dado um grupo de correspondências de pontos das malhas m_1 e m_2 , é necessário, inicialmente, calcular os centróides c_1 e c_2 dos pontos das malhas m_1 e m_2 , respectivamente, de acordo com a fórmula

$$c_k = \frac{1}{n} \sum_{i=1}^n p_{k,i}, \quad (3.24)$$

onde $p_{k,i}$ é um ponto da malha k .

Em seguida, são definidos novos pontos $p'_{k,i}$, de acordo com a equação

$$p'_{k,i} = p_{k,i} - c_k. \quad (3.25)$$

A matriz Q utiliza nove produtos S_{ij} , dados pela equação

$$S_{mn} = \sum_{i=1}^n p'_{1,i}(m)p'_{2,i}(n), \quad (3.26)$$

onde $p'_{1,i}(m)$ e $p'_{2,i}(n)$ são coordenadas dos pontos, com $1 < m < 3$ e $1 < n < 3$.

Finalmente, Q é dada por

$$Q = \begin{pmatrix} (S_{11} + S_{22} + S_{33}) & S_{23} - S_{32} & S_{31} - S_{13} & S_{12} - S_{21} \\ S_{23} - S_{32} & (S_{11} - S_{22} - S_{33}) & S_{12} + S_{21} & S_{31} + S_{13} \\ S_{31} - S_{13} & S_{12} + S_{21} & (-S_{11} + S_{22} - S_{33}) & S_{23} + S_{32} \\ S_{12} - S_{21} & S_{31} + S_{13} & S_{23} + S_{32} & (-S_{11} - S_{22} + S_{33}) \end{pmatrix}.$$

O autovetor unitário correspondente ao maior autovalor dessa matriz será a melhor rotação \vec{q}_R . A melhor translação \vec{q}_T será dada por

$$\vec{q}_T = c_2 - R(\vec{q}_R)c_1. \quad (3.27)$$

A transformação \vec{q} pode então ser aplicada a todos os pontos da malha m_1 . A validação desta transformação utiliza os pontos transformados da malha m_1 e os pontos da malha m_2 . Este processo é descrito na seção seguinte. Maior rigor matemático do Algoritmo de Horn pode ser encontrado em (Horn 1987).

3.7 Validação

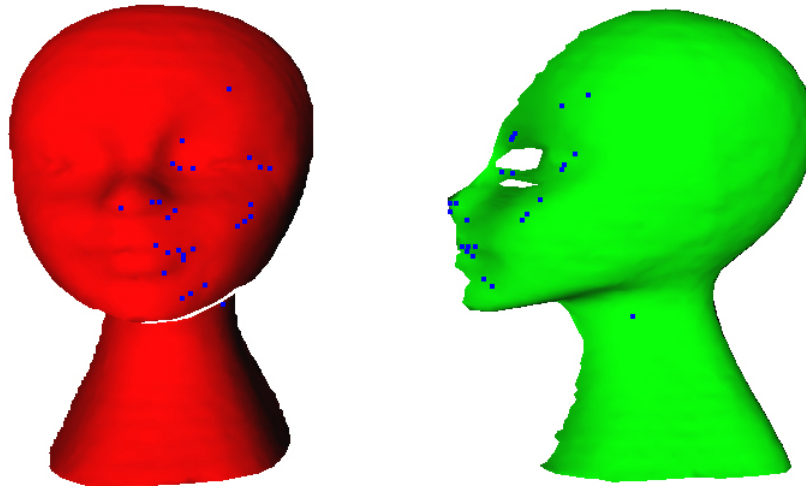
A etapa final do alinhamento automático inicial é a validação da transformação. Uma vez encontrada uma transformação que passe pelo teste de validação, o algoritmo pára e inicia-se a etapa de refinamento do registro.

Uma transformação será considerada válida se a região de sobreposição entre as duas superfícies após aplicar a transformação for maior que uma certa razão da área das superfícies. Um ponto p pertencente a uma malha m_1 estará na região de sobreposição das malhas m_1 e m_2 se existir $q \in m_2$ tal que $\|p - q\| < D_v$, onde D_v é um *threshold* de distância que deve ser definido entre 1 e 2 vezes a resolução da malha.

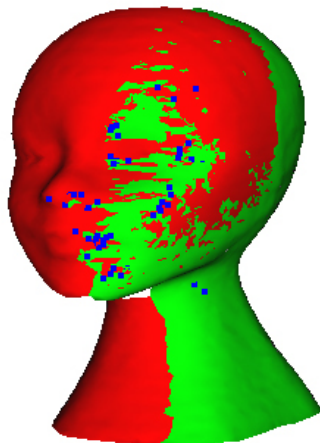
Como o algoritmo de Horn minimiza a distância entre as correspondências, estas devem ser automaticamente consideradas pertencentes à região de sobreposição. Em seguida, novas correspondências pertencentes à região de sobreposição são criadas da seguinte maneira: para cada correspondência (p_1, p_2) , com $p_1 \in m_1$, selecionam-se todos os vizinhos $p_{1,i}$ de p_1 em m_1 conectados por arestas. Seja q o ponto mais próximo de $p_{1,i}$ em m_2 . Se $\|p_{1,i} - q\| < D_v$, então a correspondência $(p_{1,i}, q)$ é criada. Esse processo é repetido para todos os vizinhos de p_1 , e para cada correspondência, até que não seja possível criar novas correspondências. Se a quantidade de correspondências finais for maior que uma certa razão da quantidade de pontos das malhas, então a transformação é validada. Caso contrário, um novo agrupamento é feito e uma nova transformação é calculada.

A Figura 3.8 ilustra o processo de validação. A Figura 3.8(a) exhibe duas malhas e seus pontos correspondentes agrupados. Após o cálculo da transformação usando o algoritmo de Horn, as malhas são alinhadas, como mostra a Figura 3.8(b). Finalmente, a Figura 3.8(c) exhibe a região de sobreposição detectada, em tom mais escuro. Boas transformações alcançam regiões de sobreposição com, no mínimo, 30% do total de pontos das malhas.

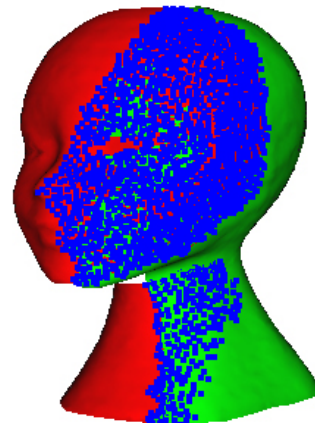
Uma vez obtida uma transformação válida, um pequeno ajuste na transformação é calculado. Utilizando agora o conjunto de correspondências final, o algoritmo de Horn é novamente aplicado. Como a quantidade de pares de pontos agora é maior, será possível



(a) Grupo de correspondências entre duas malhas.



(b) Alinhamento inicial das malhas após aplicação do Algoritmo de Horn usando um grupo de correspondências.



(c) Região de sobreposição detectada para validação da transformação

Figura 3.8: Agrupamento e validação.

obter uma transformação mais precisa. Finalmente, esta nova transformação é aplicada e inicia-se a etapa de refinamento, tema do próximo capítulo.

Capítulo 4

Refinamento do Registro usando ICP

4.1 Introdução

Este capítulo descreve o algoritmo ICP e algumas de suas variantes. Este algoritmo é responsável pelo registro final entre duas superfícies previamente alinhadas de acordo com o método descrito no capítulo anterior.

4.2 Visão Geral

O algoritmo ICP (*Iterative Closest Point*) (Besl & McKay 1992) é a técnica mais utilizada atualmente para realizar registro de superfícies. Dadas duas malhas pré-alinhadas, o ICP executa um processo iterativo para refinar a transformação inicial até convergir para um mínimo local. Se o alinhamento inicial for bom, o ICP converge corretamente. Ao final deste processo, as duas malhas com alinhamento estimado terão um alinhamento mais preciso.

Seja $M_1 = \{\vec{p}_i\}$ o conjunto de pontos da malha m_1 e $M_2 = \{\vec{q}_i\}$ o conjunto de pontos da malha m_2 . O algoritmo ICP original segue os passos abaixo, para cada iteração k :

1. Para cada ponto \vec{p}_i , calcule seu ponto mais próximo $\vec{x}_i \in M_2$;
2. Encontre a transformação \vec{q}_k que minimiza a distância entre os pares de pontos (\vec{p}_i, \vec{x}_i) usando o algoritmo de Horn e encontre o erro d_k dado pela Equação 3.23;
3. Aplique a transformação \vec{q}_k aos pontos de M_1 ;
4. Se $d_k - d_{k-1} < \tau$, onde τ é um *threshold* pequeno, o processo pára. Senão, volte para o primeiro passo.

A transformação resultante será a composição das transformações \vec{q}_i , ou seja, $\vec{q} = \vec{q}_k \circ \vec{q}_{k-1} \circ \dots \circ \vec{q}_1$.

Nos últimos anos, muitas variantes desse algoritmo foram desenvolvidas, afetando todos os estágios do procedimento. Segundo Rusinkiewicz & Levoy (2001), o algoritmo ICP pode ser dividido em seis estágios:

1. Seleção de pontos em uma ou nas duas malhas;
2. Geração de correspondências entre os pontos selecionados;
3. Atribuição de pesos nos pares de pontos;
4. Rejeição de pares de acordo com algum critério;
5. Determinação de uma métrica de erro;
6. Minimização da métrica de erro.

As seções seguintes definem os estágios do algoritmo, descrevendo algumas de suas variantes, inclusive a solução implementada neste trabalho. As análises comparativas das diversas variantes foram realizadas por Rusinkiewicz & Levoy (2001).

4.3 Seleção de Pontos

O primeiro estágio do algoritmo ICP utiliza algum critério para seleção dos pontos que serão considerados na geração de correspondências. As estratégias mais utilizadas são:

- Seleção de todos os pontos;
- Amostragem uniforme dos pontos (Turk & Levoy 1994);
- Amostragem aleatória dos pontos a cada iteração (Masuda et al. 1996);
- Amostragem uniforme no espaço das normais (Rusinkiewicz & Levoy 2001).

Além disso, em paralelo a todas as estratégias acima existe a opção de realizar a amostragem em apenas uma das malhas. Neste caso, deve se considerar todos os pontos de uma malha e aplicar alguma das técnicas acima à outra malha.

Fazendo-se uma análise da convergência em função do número de iterações realizadas, todas as estratégias apresentam resultados semelhantes em grande parte dos modelos (Rusinkiewicz & Levoy 2001). Este trabalho adota a estratégia de seleção de todos os pontos, tendo em vista a simplicidade do método.

4.4 Geração de Correspondências

A geração dos pares de pontos é a etapa mais importante do algoritmo ICP. Além de ser geralmente a etapa que requer maior custo computacional, a geração de correspondências consistentes é determinante para obtenção de transformações precisas.

Para cada ponto $p \in M_1$ selecionado em uma malha, os critérios mais utilizados para determinar seu ponto correspondente são:

- Encontrar seu ponto mais próximo em m_2 (Besl & Mckay 1992);
- Encontrar a intersecção do raio que sai do ponto p , na direção de sua normal, com a malha m_2 (Chen & Medioni 1992);
- Projetar o ponto p na malha m_2 do ponto de vista do scanner gerador da range image de m_2 (Blais & Levine 1995);

Um critério opcional que pode ser utilizado nos métodos acima é a restrição apenas a pares que tenham ângulo entre suas normais limitado, por exemplo, a 45° . Segundo Rusinkiewicz & Levoy (2001), apesar de apresentar performance inferior em alguns tipos de geometria, os algoritmos de ponto mais próximo são robustos em grande parte das geometrias. Por essa razão, o algoritmo de ponto mais próximo com restrição de ângulo de normais limitado foi a técnica implementada neste trabalho.

Nesta técnica, para cada ponto $p_i \in M_1$, calcula-se o ponto mais próximo $x_i \in M_2$. Supondo que uma malha tenha n pontos e a segunda malha tenha m pontos, este procedimento tem complexidade $O(nm)$. Para acelerar esse processo, Simon (1996) propôs a utilização de uma *kd-tree* para busca dos pontos mais próximos. Desse modo, a busca pelo ponto mais próximo se restringe a uma busca numa árvore binária, reduzindo a complexidade do algoritmo para $O(n \log m)$.

A subseção seguinte descreve os passos para criação e busca em uma *kd-tree*.

4.4.1 Kd-trees

Uma *kd-tree* é uma árvore binária que generaliza o método de biseção para k dimensões. Dado um ponto $p \in M_1$, o problema de busca do ponto mais próximo de p em M_2 pode ser resolvido com uma *kd-tree* tridimensional. O processo de criação da *kd-tree* segue abaixo.

Inicialmente, é necessário dividir o espaço com um plano paralelo ao plano yz que passe por um ponto q de M_2 de modo que a quantidade de pontos em cada subespaço seja igual. O subespaço abaixo do plano será o filho à esquerda, e o subespaço acima do plano será o filho à direita. O ponto q pode ser facilmente identificado como a mediana das coordenadas x dos pontos. Esse procedimento continua nos filhos deste nó, dividindo-se o espaço sucessivamente pelos planos yz , xz e xy , até que não haja mais que um ponto em

cada subespaço. Estes pontos estarão associados às folhas da árvore. Os nós das árvores devem incluir duas informações fundamentais. A primeira é o ponto q que divide seu espaço. O segundo é um código t informando se o plano divisor é paralelo ao plano yz , xz ou xy . Esse processo de criação tem complexidade $O(n \log n)$. A Figura 4.1 ilustra essa idéia no caso bidimensional.

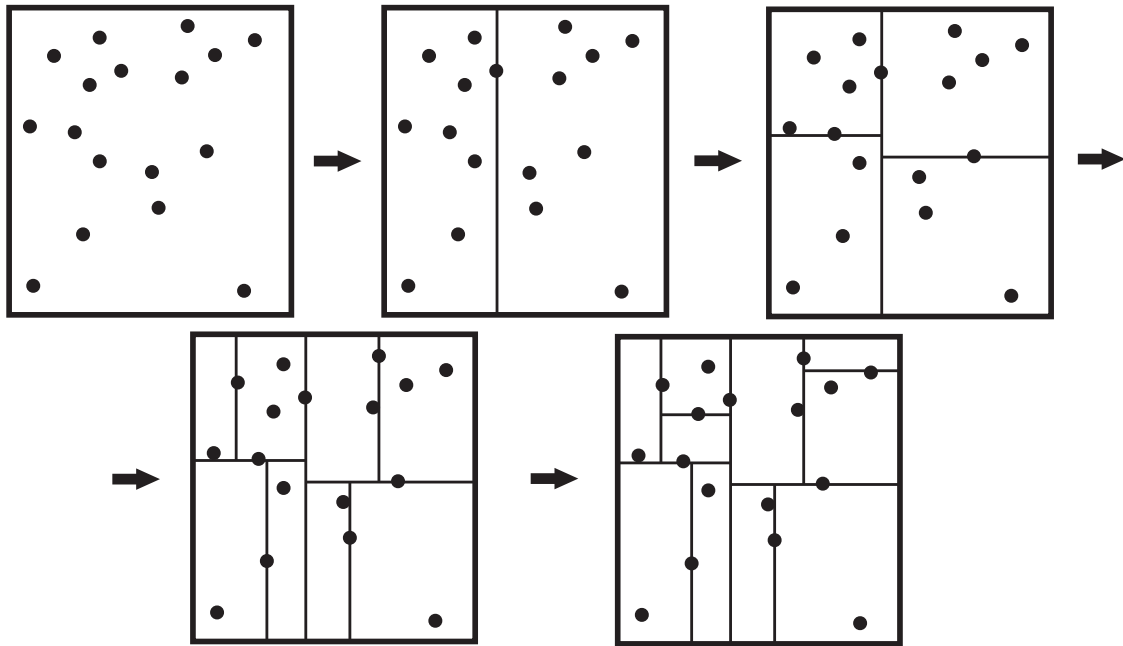


Figura 4.1: Processo de criação de uma *kd-tree* bidimensional.

O processo de busca é um algoritmo recursivo. Inicialmente, devem se definir um *threshold* D_{max} que será a distância máxima aceita para que um ponto em M_2 seja candidato a ponto mais próximo. Este valor deve estar entre 1 e 2 vezes a resolução da malha. Ao término do processo, haverá uma pequena lista de candidatos a ponto mais próximo. Finalmente, os pontos dessa lista são testados um a um para identificação do ponto mais próximo.

Dado um ponto $p \in M_1$, a busca começa pela raiz da árvore. Se $\|p - q\| < D_{max}$, q é adicionado na lista de candidatos. Se a coordenada indicada pelo parâmetro t for menor que a mesma coordenada de q , a busca segue pelo filho à esquerda. Senão, a busca segue pelo filho à direita, até chegar a uma folha da árvore, quando o algoritmo retorna a lista de candidatos.

4.5 Atribuição de Pesos às Correspondências

Opcionalmente, é possível atribuir um peso a cada correspondência criada, de acordo com algum critério que meça a confiabilidade do par de pontos. Duas estratégias interessantes

para estimar e atribuir pesos às correspondências consideram a distância entre o par de pontos e o ângulo de suas normais.

Na primeira estratégia, pares de pontos com distâncias altas terão pesos baixos. Seja $Dist_{max}$ a maior distância dentre os pares de pontos. Segundo Godin et al. (1994), os pesos podem ser estimados de acordo com a equação

$$W = 1 - \frac{Dist(p_1, p_2)}{Dist_{max}}, \quad (4.1)$$

onde p_1 e p_2 são pontos de cada correspondência.

A segunda estratégia analisa o ângulo entre as normais dos pares de pontos, de acordo com o produto escalar dado pela equação

$$W = n_1 \cdot n_2, \quad (4.2)$$

onde n_1 e n_2 são os vetores normais de cada par de pontos.

Essas estratégias apresentam, de modo geral, performance ligeiramente superior ao uso de pesos constantes nos pares de pontos (Rusinkiewicz & Levoy 2001). Desse modo, neste trabalho foi adotada a estratégia de utilizar pesos constantes.

4.6 Rejeição de Correspondências

Esta etapa é fundamental para a rejeição de pares de pontos fora da região de sobreposição das malhas. De modo geral, as estratégias mais utilizadas para rejeição de pares de pontos utilizam como critério a distância entre os pontos. Seguem abaixo algumas das estratégias mais utilizadas.

- Rejeição das correspondências cuja distância seja maior que uma distância determinada pelo usuário;
- Rejeição dos $n\%$ pares mais distantes (Pulli (1999) sugere 10%);
- Rejeição das correspondências cuja distância seja maior que um múltiplo do desvio padrão das distâncias (Masuda et al. (1996) sugere 2.5 vezes o desvio padrão).

Seguindo a idéia de detecção de áreas de sobreposição descrita no capítulo anterior, este trabalho utiliza um *threshold* de distância determinado como um múltiplo da resolução da malha. Bons resultados são alcançados definindo a distância máxima entre 1 e 2 vezes a resolução da malha.

Outra idéia interessante para rejeição de pares consiste em descartar todas as correspondências contendo pontos na fronteira das malhas. Este procedimento rejeita pares que contém pontos da fronteira de uma malha e pontos da segunda malha localizados

nas proximidades da fronteira da primeira malha, porém fora da região de sobreposição. A Figura 4.2 ilustra essa idéia. Esta estratégia também foi implementada neste trabalho.

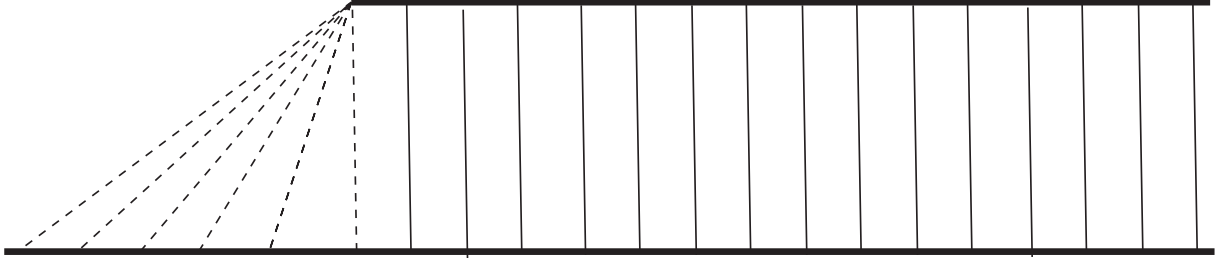


Figura 4.2: Rejeição de pares contendo pontos de fronteira.

4.7 Métricas de Erro e Minimização

Uma vez determinadas as correspondências consistentes com seus determinados pesos, é necessário definir uma métrica para estimar o erro de alinhamento entre esses pares de pontos. Finalmente, calcula-se a transformação que minimize a métrica de erro.

As duas métricas mais utilizadas para estimar erro de alinhamento são a média da soma dos quadrados das distâncias entre os pares de pontos e a média da soma dos quadrados das distâncias entre ponto e plano.

Seja p_i um ponto da malha m_1 e x_i seu ponto correspondente da malha m_2 . A média da soma dos quadrados das distâncias é definida pela função

$$f(q) = \frac{1}{n} \sum_{i=1}^n \|x_i - R(q_R)p_i - q_T\|^2. \quad (4.3)$$

O método mais utilizado para encontrar a transformação que minimiza essa aplicação é o algoritmo de Horn, descrito no capítulo anterior. Outras soluções para essa minimização são a decomposição de valor singular (Arun et al. 1987), matrizes ortonormais (Horn et al. 1988) e quatérnios duais (Walker et al. 1991). A precisão numérica e estabilidade desses métodos é semelhante, segundo Eggert et al. (1997).

A segunda métrica utiliza a distância entre ponto e plano para estimar o erro, de acordo com a função dada por

$$f(\vec{q}) = \frac{1}{n} \sum_{i=1}^n \|x'_i - R(\vec{q}_R)\vec{p}_i - \vec{q}_T\|^2, \quad (4.4)$$

onde $x'_i = r |\min_{r \in S_i} \|r - R(\vec{q}_R)\vec{p}_i - \vec{q}_T\|$ e S_i é o plano que contém o ponto x_i e tem normal igual à normal de x_i .

Neste caso, não existem soluções diretas. Os métodos de solução mais utilizados são o método não linear Levenberg-Marquardt (Levenberg 1944) ou através de uma linearização do problema, fazendo $\sin \theta \sim \theta$ e $\cos \theta \sim 1$, considerando θ pequeno.

Neste trabalho, foi implementada a métrica da média da soma dos quadrados das distâncias entre os pares de pontos com solução dada pelo algoritmo de Horn.

Após a transformação ser encontrada e aplicada à malha, o algoritmo ICP retorna à etapa de geração de correspondências. Como a transformação alterou a posição dos pontos, diferentes correspondências podem ser geradas, resultando em uma transformação diferente. Após algumas iterações, as transformações tendem a ser cada vez menores, fazendo com que os novos pares de pontos não sejam diferentes dos anteriores. Neste caso, a transformação se aproxima da identidade e o algoritmo pára.

Com o alinhamento final determinado pela transformação resultante do algoritmo ICP, as malhas são alinhadas num sistema de coordenadas global. A etapa final consiste em extrair um único modelo a partir de várias malhas alinhadas. O próximo capítulo descreve uma técnica para extração de malhas em volumes.

Capítulo 5

Reconstrução da Malha Final

5.1 Introdução

Este capítulo descreve uma técnica desenvolvida por Curless & Levoy (1996) para reconstrução de modelos a partir de malhas alinhadas chamada VRIP.

Esta técnica pode ser dividida em duas etapas. A primeira etapa realiza uma integração volumétrica das malhas, criando um volume contendo as malhas alinhadas e definindo uma função de distância. A segunda etapa executa um método muito conhecido para extração de malhas a partir de volumes, chamado *Marching Cubes* (Lorenson & Cline 1987). As seções seguintes descrevem essas etapas.

5.2 Integração Volumétrica

Dado um conjunto $M = \{m_1, m_2, \dots, m_n\}$ de malhas alinhadas em um sistema de coordenadas global com seus respectivos pontos de visão p_i do scanner, é necessário definir um volume contendo todas as malhas. A resolução desse volume deve ser coerente com as resoluções e dimensões das malhas. Resoluções de volume muito altas são inúteis, pois o nível de detalhe do modelo final depende da resolução das malhas. Resoluções de volume baixas causam perda nas informações das malhas.

Tendo em mãos o volume com boa resolução e contendo todas as malhas, o próximo passo é definir neste volume uma função implícita de distância com sinal $D: V \rightarrow \mathbb{R}$, onde V é o conjunto de vértices do volume. Essa distância será uma estimativa da distância do vértice à malha final. Se, por exemplo, $D(v) = 0$ para algum vértice v , então a malha deve passar por este vértice. Dado o volume com sua respectiva função de distância bem definida, o algoritmo *Marching Cubes* realiza a extração da malha final.

A função D será construída a partir da combinação de n funções de distância com sinal $d_1(v), d_2(v), \dots, d_n(v)$, e funções de peso $w_1(v), w_2(v), \dots, w_n(v)$.

Cada função $d_i(v)$ está associada a uma malha m_i e representa, em cada vértice v , a distância com sinal de v à malha. Seja p_i o ponto de visão do scanner em relação à malha m_i , e $r_{i,v}(t) = p_i + t \cdot (v - p_i)$ o raio que sai do ponto de visão em direção ao vértice v . O próximo passo será calcular a interseção $q_{i,v}$ de $r_{i,v}$ com m_i . Finalmente, a função de distância com sinal $d_i(v)$ fica bem definida da seguinte maneira:

$$d_i(v) = \|q_{i,v} - p_i\| - \|v - p_i\|. \quad (5.1)$$

A Figura 5.1 ilustra o cálculo de $d_i(v)$.

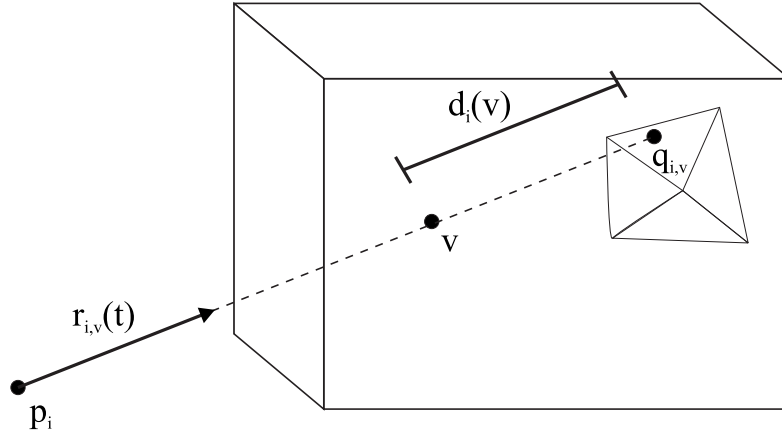


Figura 5.1: Cálculo da função de distância com sinal $d_i(v)$.

Opcionalmente, é possível atribuir pesos aos vértices do volume, de acordo, por exemplo, com a incerteza do triângulo da malha que contenha a interseção $q_{i,v}$. Tendo conhecimento dos vetores normais aos vértices deste triângulo, é possível, através de uma interpolação linear, obter o vetor normal em $q_{i,v}$. Um critério bastante utilizado para definir o peso de um vértice v do volume é o produto escalar do vetor normal em $q_{i,v}$ e o vetor direção da visão do scanner. Ângulos menores implicam maior confiabilidade no processo de captura, devido à iluminação mais adequada.

Os pesos $w_i(v)$ são associados às funções de distância $d_i(v)$, e finalmente a função $D(v)$ será calculada como uma média das funções de distância $d_i(v)$ ponderada pelos pesos $w_i(v)$. Essa função é dada por

$$D(v) = \frac{\sum_{i=1}^n w_i(v) d_i(v)}{\sum_{i=1}^n w_i(v)}. \quad (5.2)$$

Além disso, o peso final $W(v)$ é dado por

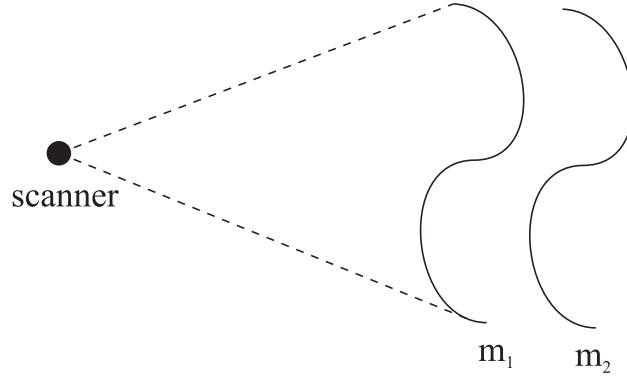
$$W(v) = \sum_{i=1}^n w_i(v). \quad (5.3)$$

Uma maneira incremental de realizar os cálculos de $D(v)$ e $W(v)$ é através das funções

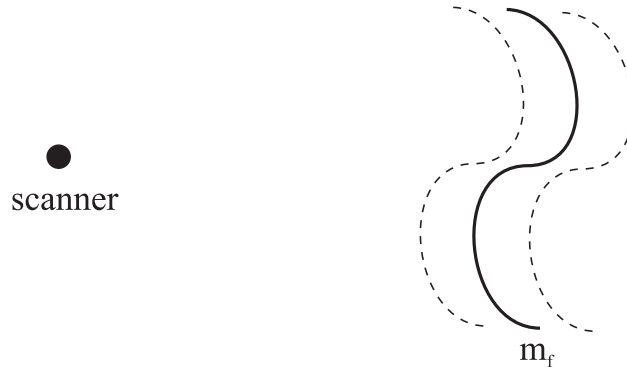
$$D_{i+1}(v) = \frac{W_i(v)D_i(v) + w_{i+1}(v)d_{i+1}(v)}{W_i(v) + w_{i+1}(v)} \quad (5.4)$$

$$W_{i+1}(v) = W_i(v) + w_{i+1}(v). \quad (5.5)$$

Uma malha resultante da reconstrução a partir de duas malhas alinhadas e com mesmo ponto de visão do scanner deve ter sua região de sobreposição definida como a média das regiões de sobreposição das duas malhas, de acordo com a Equação 5.2. Portanto, essa técnica elimina os pequenos erros de alinhamento ainda existentes. Um exemplo bidimensional é exibido na Figura 5.2, onde duas curvas m_1 e m_2 alinhadas com pequeno erro são exibidas na Figura 5.2(a) e a curva resultante m_f é exibida na Figura 5.2(b).



(a) Curvas m_1 e m_2 alinhadas com pequeno erro.



(b) Curva resultante m_f .

Figura 5.2: Efeito da função de distância $D(v)$ em duas curvas alinhadas com pequeno erro.

Um último critério fundamental para o sucesso do método é a definição de um *threshold*, a partir do qual as distâncias serão desprezadas, fixando o peso do vértice como nulo. Considere o caso em que o modelo apresente duas malhas m_1 e m_2 que intersectem, respectivamente, as retas $r_{1,v}$ e $r_{2,v}$, para um mesmo vértice v do volume. Além disso, essas duas malhas representam lados opostos do objeto, sem região e sobreposição. Sob essas

condições, o cálculo da função $D(v)$ vai retornar uma média inconsistente das duas malhas, considerando-as como duas malhas sobrepostas. Existindo tal *threshold*, as malhas opostas não iriam interferir entre si. Além disso, os custos computacionais ficariam restritos às vizinhanças das malhas.

O valor desse *threshold* deve ser grande o suficiente para que as "rampas" de distância nas vizinhanças das malhas fiquem bem definidas, e pequeno o suficiente para evitar a influência de malhas opostas.

Como a função de distância apresenta valores positivos e negativos, este *threshold* deve ser convertido em dois valores D_{min} e D_{max} , negativo e positivo, respectivamente, que determina o intervalo onde as distâncias serão consideradas, na vizinhança das malhas. O valor utilizado em (Curless & Levoy 1996) é de metade do intervalo máximo de incerteza.

Detalhes para implementação eficiente desta técnica são encontrados no seu artigo original (Curless & Levoy 1996).

5.3 Marching Cubes

Nesta seção o algoritmo Marching Cubes original (Lorensen & Cline 1987) é descrito. É importante destacar que diversas variantes desta técnica existem atualmente. A implementação do VRIP se baseia numa variante que utiliza tabelas para resolver algumas das ambiguidades.

O Marching Cubes é um algoritmo bastante popular para extrair isosuperfícies de dados volumétricos. Seja D uma função escalar implícita definida no volume, como a função de distância definida na seção anterior. A superfície que satisfaz $D(v) = \alpha$, com α constante, é chamada isosuperfície definida por α . Dada uma função D definida nos vértices de um volume e um valor α , a extração de sua isosuperfície pode ser aproximada por uma isosuperfície linear por partes, composta de um conjunto de triângulos.

O algoritmo padrão realiza uma varredura voxel por voxel, verificando quais voxels a superfície intersecta. Caso haja interseção, novos triângulos da superfície são criados. Esse processo envolve a verificação dos valores da função de distância em cada vértice. Neste caso específico, serão consideradas a função de distância D com valor $\alpha = 0$. Portanto, se houver variação de sinal da função de distância nos vértices do voxel, a isosuperfície intersecta o voxel.

A geração dos triângulos é composta de duas etapas. A primeira etapa determina o padrão topológico de intersecção através de uma tabela de 256 casos, que pode ser resumida a 15 casos, desconsiderando-se simetrias de rotação e reflexão. A segunda etapa determina a geometria de cada vértice dos triângulos da isosuperfície, usando interpolações lineares nas arestas do voxel. Finalmente, a isosuperfície será formada pelo conjunto de

triângulos e vértices extraídos dos voxels do volume. Mais detalhes podem ser encontrados em (Lorensen & Cline 1987) e (Newman & Yi 2006).