

Leonardo Kandler Bittencourt

Δ^+ -Connection

Reconstrução 3D de Objetos com Seções Planares Variáveis

Joinville / SC

Novembro de 2009

Leonardo Kandler Bittencourt

Δ^+ -Connection

Reconstrução 3D de Objetos com Seções Planares Variáveis

Orientador: Prof. Marcelo da Silva Hounsell, PhD

UDESC – Universidade do Estado de Santa Catarina
Centro de Ciências Tecnológicas
Departamento de Ciência da Computação

Joinville / SC

Novembro de 2009

Monografia sob título “ Δ^+ -Connection: Reconstrução 3D de Objetos com Seções Planares Variáveis”, defendida por Leonardo Kandler Bittencourt, aprovada em Julho de 2009, em Joinville, Santa Catarina, pela banca examinadora constituída pelos professores:

Prof. Marcelo Hounsell da Silva, PhD
Departamento de Ciência da Computação – UDESC
Orientador

Prof. Alexandre Gonçalves Silva, MSc
Departamento de Ciência da Computação – UDESC

Prof. Roberto Silvio Ubertino Rosso Junior, PhD
Departamento de Ciência da Computação – UDESC

*Aos meus pais Cide e Salete
e às minhas irmãs Márcia e Marlize,
dos quais sempre recebi muito apoio,
incentivo e carinho.*

Agradecimentos

Ao meu orientador, por seu apoio e incentivo em todo momento.

Aos alunos das turmas de Modelagem Geométrica e ao bolsista Joel Cesar Cabral Junior pela cooperação no desenvolvimento do trabalho.

Aos meus caros amigos, que me apoiaram durante todo o curso.

“Lutar sempre, cair às vezes, desistir jamais!”

Autor desconhecido

Resumo

A reconstrução tridimensional através de seções planares está dividida em três etapas: geração da malha, correspondência e bifurcação. Tem como principal aplicação a área médica, nas modelagens de estruturas fisiológicas, e a geologia, com a reconstrução de terrenos através de curvas de níveis.

Este trabalho teve como objetivo melhorar o algoritmo Δ -Connection original no que diz respeito às etapas de geração da malha e correspondência. Para tal, foi proposta uma divisão da geração de malha em cinco sub-etapas menores independentes entre si, as quais permitiram que diversas técnicas sejam implementadas, combinadas e testadas. Esta divisão não afetou os algoritmos existentes, que se encaixam facilmente no modelo genérico proposto. Na etapa de correspondência, não foi possível dividir em etapas menores como realizado na geração da malha, porém propôs-se uma nova heurística que, além do alinhamento dos centróides, considera o tamanho do contorno como parte solução. Esta nova heurística apresentou melhorias em modelos com problemas de correspondência e não afetou os resultados visuais de outros modelos. Também, através de testes, definiu-se uma métrica que pode determinar a qualidade visual de um objeto, ou seja, uma correlação entre a análise quantitativa e qualitativa.

Sendo assim, implementou-se um conjunto de soluções para geração da malha e correspondência, estabelecendo métricas de avaliação de automática do modelo gerado.

Abstract

The three-dimensional reconstruction by planar sections is divided in three steps: tiling, correspondence and branching. Its main application is the medical area, modeling of physiological structures, and geology, with the reconstruction of land by contour lines.

This work objectiveed improve the original Δ -Connection algorithm in the tiling and correspondence steps. For this, it was proposed a tiling split in five smaller independent steps, which allow that many techniques are implemented, combined and tested. This split don't affected the exiting algorithms, that fit easily in the generic model proposed. In the correspondence step wasn't possible split in smaller steps like tiling, but proposed to a new heuristic that, beyond the alignment of the centroids, considers the size of the countour as solution's part. This new heuristic showed improvements on models with correspondence problems and it didn't affected the others models view results. Also, through the tests, defined a measure that can determinate the object visual quality, in other words, a correlation between a quantitative and qualitative analysis.

Thus, there were implemented a set of solutions for tiling and correspondence, establishing metrics to measure automatically the model generated.

Sumário

1 Introdução.....	15
2 Fundamentação Teórica.....	19
3 Trabalhos Relacionados.....	29
3.1 Correspondência (<i>Correspondence</i>).....	29
3.1.1 Correspondência pelo Mínimo Custo.....	29
3.1.2 Correspondência por Sobreposição.....	30
3.1.3 Correspondência por Distância no Domínio dos Inteiros.....	31
3.1.4 Correspondência por Alinhamento dos Centróides.....	33
3.2 Geração de Malha (<i>Tiling</i>).....	34
3.2.1 Distribuição Proporcional Local.....	35
3.2.2 Menor <i>Span</i>	37
3.2.3 Balanceamento pela Distância.....	38
3.3 Métricas.....	40
4 Projeto de Solução.....	42
4.1 Visão Geral da Geração da Malha.....	42
4.1.1 Soluções para Equivalência das Curvas.....	45
4.1.2 Soluções para Orientação das Curvas.....	46
4.1.3 Soluções para Torção das Curvas.....	47
4.1.4 Soluções para Balanceamento dos Pontos.....	48
4.1.5 Soluções para Conexão dos Pontos.....	52

4.2 Visão Geral da Correspondência.....	53
4.2.1 Mudança de Significado.....	53
4.2.2 Correspondência pela Geometria do Contorno.....	54
4.2.3 Melhorando a Heurística de Correspondência do Alinhamento dos Centróides.....	55
4.3 Solução para Tamanho do Contorno.....	56
A)Perímetro.....	56
B)Área pela Caixa Envoltória (<i>Bounding Box</i>).....	57
C)Área pela Circunferência Envoltória (<i>Bounding Circle</i>).....	57
D)Área Real	57
4.4 Arquitetura do <i>Software</i>	58
4.5 Novas Funcionalidades.....	62
5 Implementação.....	64
5.1 Interfaces.....	64
5.2 Arquitetura Final do <i>Software</i>	66
5.3 Implementação das Métricas.....	68
6 Resultados.....	70
6.1 Dados da Geração de Malha.....	70
6.2 Dados de Correspondência.....	76
6.3 Discussão.....	81
6.4 Site.....	82
7 Conclusão.....	83
7.1 Trabalhos Futuros.....	84
Referências.....	86
APÊNDICE A – Funções.....	89
APÊNDICE B – Comparativo e Resultados.....	90

Lista de Figuras

Figura 1.1. Reconstrução do coração. Fonte: Anzolin et al (2008).....	16
Figura 2.1. Diferentes formas de correspondência. Fonte: Zamansky (2007).....	21
Figura 2.2. Geração da Malha. Fonte: Chen e Shen (2008).....	22
Figura 2.3. Linking triangular e quadrangular.....	22
Figura 2.4. Diferentes métricas de geração de malha. Fonte: Meyers (1994).....	23
Figura 2.5. Bifurcação. Fonte: Gattass e Peixoto (2000).....	24
Figura 2.6. Diferentes formas de bifurcação. Fonte: Bajaj et al (1996).....	24
Figura 2.7. Segmento de reta no R^3	26
Figura 2.8. Regra da mão direita. Fonte: Vainsencher (2005).....	26
Figura 2.9. Projeção de um ponto em um segmento de reta.....	28
Figura 3.1. Contornos iniciais e sua representação em um grafo.....	30
Figura 3.2. Contornos que se conectação e o grafo de definição da correspondência.....	31
Figura 3.3. Definição da correspondência. Fonte: Anzolin (2006).....	32
Figura 3.4. Objeto reconstruído sem bifurcação. Fonte: Gatass e Peixoto (2000)	33
Figura 3.5. Volumes cônicos entre os planos. Fonte: Nonato, (1998).....	33
Figura 3.6. Diferentes valores para β . Fonte: Vargas (2001).....	34
Figura 3.7. Distâncias entre os centróides.....	35
Figura 3.8. Contornos com números de pontos diferentes. Fonte: Li, Ma e Tan (2006).....	36
Figura 3.9. Linearização de um contorno.....	37
Figura 3.10. Contornos linearizados e com correspondência inicial dos pontos.....	37
Figura 3.11. Criação do valor proporcional.....	38

Figura 3.12. Geração da malha pela menor aresta.....	39
Figura 3.13. Balanceamento pelas distâncias. Fonte: Anzolin et al (2008).....	40
Figura 3.14. Definição dos pontos iniciais.....	40
Figura 3.15. Triangulação para geração de malha. Fonte: Anzolin et al (2008).....	41
Figura 3.16. Tamanho dos spans entre dois contornos.....	42
Figura 4.1. Sub-etapa da geração de malha: equivalência das curvas.....	46
Figura 4.2 Diferentes formas de obter o centróide de um contorno.....	48
Figura 4.3. Dois contornos com pontos no mesmo sentido.....	48
Figura 4.4. União de conjuntos.....	51
Figura 4.5. Criação dos novos pontos e conexão entre os contornos.....	51
Figura 4.6. Projeção dos novos pontos.....	52
Figura 4.7. Distribuição cega de pontos.....	54
Figura 4.8. Formas de conexão dos pontos em padrões.....	55
Figura 4.9. Variação do Δ	57
Figura 4.10. Definição dos pontos extremos da ordenada e acissa.....	59
Figura 4.11. Definição dos pontos extremos do contorno.....	60
Figura 4.12. Diagrama de classes do Δ - <i>Connection</i>	62
Figura 4.13. Diagrama de classes do Δ^+ - <i>Connection</i>	63
Figura 4.14. Interface de parametrização do processo de geração de malha.....	65
Figura 4.15. <i>Linking</i> sem e com translação.....	66
Figura 5.1. Interface de parametrização da geração de malha.....	67
Figura 5.2. Interface de parametrização da correspondência.....	68
Figura 5.3. Nova interface de correspondência.....	69
Figura 5.4. Algoritmo de correspondência Δ^+ - <i>Connection</i>	70
Figura 5.5. Arquitetura UML do Δ^+ - <i>Connection</i>	73
Figura 6.1. Comparativo visual da veia.....	75

Figura 6.2. Variação de spans em contornos de tamanho constante.....	76
Figura 6.3. Comparativo visual do Fêmur.....	77
Figura 6.4. Comparativo visual do Pulmão.....	78
Figura 6.5. Comparativo visual do coração.....	79
Figura 6.6. Objetos gerados com a nova técnica de correspondência.....	80
Figura 6.7. Reconstrução do coração com diferentes parametrizações.....	81
Figura 6.8. <i>Site</i> da pesquisa.....	82

1 Introdução

Vários campos da medicina utilizam técnicas de computação gráfica para modelar estruturas fisiológicas tridimensionais. O uso dos sistemas computacionais no estudo e análise de estruturas anatômicas têm crescido nas últimas décadas. As informações destas estruturas anatômicas são obtidas a partir de contornos em uma sequência de seções planares paralelas ou fatias bidimensionais paralelas, na qual é aplicada uma técnica de interpolação que resultará em um objeto tridimensional. Essas fatias são extraídas de equipamentos de tomografia computadorizada (CT), imagem de ressonância magnética (MRI) e imagem de ultrasonografia [Lawrence, 2005].

Outra aplicação pode ser encontrada na geologia com a reconstrução tridimensional de terrenos, no qual os modelos são reconstruídos a partir de uma sequência de superfícies com contornos que representam as curvas de níveis. Com a imagem reconstruída torna-se mais fácil uma análise detalhada do terreno, a fim de identificar novas jazidas de petróleo ou de minérios, por exemplo [Gattass e Peixoto, 2000], [Tavares et al, 2003].

Várias técnicas de interpolação foram propostas nos últimos anos, Sidlesky (2006), Zamansky (2007), Barrequet (2007), Anzolin, Hounsell e Silva (2008), porém não existe apenas uma única técnica que realize a reconstrução resultando sempre em um objeto fielmente igual à realidade. As técnicas baseadas em métodos heurísticos trazem flexibilidade e eficiência, apresentando uma solução rápida e aproximada [Gattass e Peixoto, 2000]. Através de combinações de regras pré-estabelecidas, cada qual com um conjunto de axiomas, a interpolação ocorre em três passos, determinando a geometria e topologia da superfície [Gattas e Peixoto, 2000], [Anzolin, Hounsell e Silva, 2008]:

- Correspondência (*correspondence*): ocorre o estabelecimento das conexões entre os contornos de fatias adjacentes;
- Geração da Malha (*tiling*): geração de uma malha para conectar contornos adjacentes;
- Bifurcações (*branching*) : como são resolvidas as mudanças topológicas entre uma fatia e outra (relação m-n).

O Δ -Connection proposto por Anzolin, Hounsell e Silva (2008) é uma técnica de interpolação tridimensional que utiliza regras heurísticas para interpolação de contornos poligonais com mudança topológica através de seções planares. O algoritmo apresenta ênfase nas etapas de correspondência e geração de malhas focando na rapidez e simplicidade dos cálculos. Os contornos são poligonais e têm a mesma orientação, tratando curvas convexas e côncavas não auto intersectantes e curvas que não contêm contornos dentro de contornos. A visualização final apresenta a estrutura do objeto como um todo, não gerando suavização de arestas [Anzolin, Hounsell e Silva, 2008].

A solução apresentada por Anzolin, Hounsell e Silva (2008) tem aplicação preferencial na visualização de canais, onde o modelo como um todo é o mais importante. Nestes tipos de objetos apresentou soluções satisfatórias na interpolação dos objetos. Porém, em objetos que possuam regiões com muitos contornos próximos, por exemplo a Figura 1.1, a reconstrução final não foi a desejada.

Na Figura 1.1 pode ser vista a interpolação do coração, onde a região superior do conjunto possui curvas menores e mais próximas entre si em relação ao resto. Os contornos que se conectarão baseiam-se no parâmetro Δ , onde caso a distância entre os dois contornos adjacentes for menor que o valor de Δ , serão conectados, mas se a distância for maior, não se conectarão. Em (a) foi definido um valor pequeno para o parâmetro Δ , resultando em um objeto, no qual várias fatias não se conectam. Em (b) foi definido um Δ maior, porém pequenos contornos que não deviam se conectar se conectaram, gerando um objeto indesejável, conforme pode-se visualizar na parte superior do objeto, onde contornos de um canal se conectaram a contornos de outro canal que não deveriam, deformando o objeto.

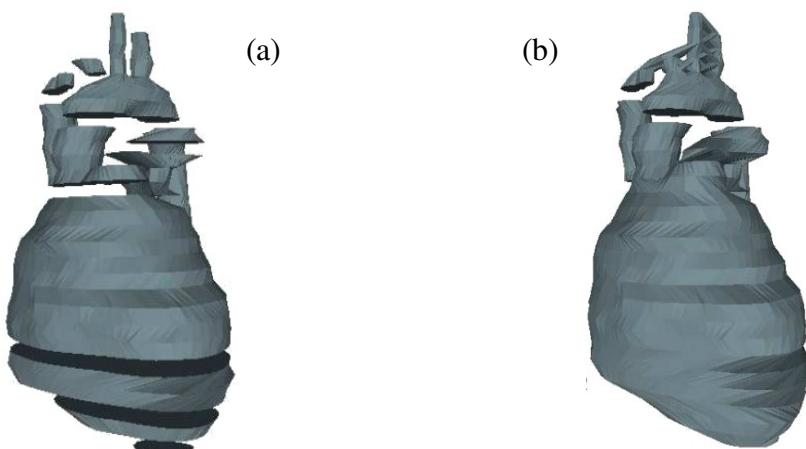


Figura 1.1. Reconstrução do coração. Fonte: Anzolin et al (2008).

Soluções para um maior número de casos podem ser encontradas através de uma combinação de variadas regras heurísticas nas etapas da interpolação. Com um maior número de heurísticas para cada etapa e com a possibilidade de escolha de qual técnica utilizar nas fases da interpolação se tem um maior número de soluções para diferentes formas de objetos, aumentando a aplicabilidade e flexibilidade do algoritmo.

1.1 Objetivos

O objetivo geral deste trabalho de conclusão de curso (TCC) é melhorar o algoritmo Δ -Connection na tarefa de interpolação tridimensional a partir de fatias planares aumentando sua flexibilidade.

Os objetivos específicos são os seguintes:

- Implementar e avaliar o desempenho de um conjunto de alternativas para o cálculo de geração de malha e correspondência no algoritmo Δ -Connection;
- Estabelecer métricas para avaliar automaticamente a qualidade do modelo gerado;
- Desenvolver uma solução para o caso onde haja uma variação muito grande dos tamanhos dos perfis no modelo, ou seja, objetos em que haja variação muito grande do diâmetro dos contornos.

Ao final deste TCC espera-se:

- o melhor entendimento do problema da interpolação de contornos poligonais e reconstrução tridimensional;
- um conjunto de soluções para cada etapa da reconstrução de forma a poder combiná-las;
- uma implementação das soluções acima na forma de *software* parametrizado, e;
- um *site* com os principais aspectos do projeto, o *software* e alguns exemplos de objetos gerados pelos algoritmos.

1.2 Estrutura do Trabalho

Este trabalho de conclusão de curso está organizado da seguinte forma:

O capítulo 2 apresenta inicialmente conceitos das etapas da reconstrução tridimensional. Posteriormente são descritos alguns fundamentos e funções necessárias em relação às técnicas escolhidas para a modelagem dos objetos;

No capítulo 3, são apresentados os trabalhos correlatos de geração da malha, correspondência e métricas.

O capítulo 4 apresenta o projeto de solução proposto para a etapa de geração de malha e correspondência.

O capítulo 5 mostra a nova arquitetura do *software*, interface, diagrama de classe, modificações realizadas e detalha as soluções algorítmicas propostas por este trabalho.

O capítulo 6 apresenta os resultados e discussões do desenvolvimento.

No capítulo 7 são apresentadas as conclusões e propostas alguns trabalhos futuros.

2 Fundamentação Teórica

A modelagem geométrica é uma área da computação gráfica que estuda mecanismos de criação de modelos geométricos através de uma síntese de técnicas de variadas áreas como cálculo vetorial, geometria analítica e descritiva, teoria de conjuntos, análise numérica e matemática matricial, entre outras. Esta combinação de técnicas matemáticas tratam da síntese, manipulação e topologia dos objetos gráficos computacionais [Shirley, 2005]. Os objetos reconstruídos devem ser armazenados de forma a atenderem requisitos de modelagem geométrica [Foley, 1996].

2.1 Decomposição do Problema

Decompor o problema traz algumas vantagens. A decomposição permite uma abordagem modular; o método utilizado para resolver um dos sub-problemas pode ser alterado sem dependência com nenhum método usado para solucionar outro sub-problema, ou seja, há total independência entre a solução dos sub-problemas.

Alguns dos sub-problemas são difíceis de se resolver utilizando somente a computação. Se alguns dos resultados não forem aceitáveis, a abordagem modular permite a intervenção do usuário para corrigir esta etapa antes de prosseguir para resolver o próximo sub-problema [Meyers, 1994]. Adicionalmente, pode-se ter mais uma vantagem no fato da independência acima mencionada, pois ela permite que hajam variadas combinações de soluções entre os sub-problemas, aumentando a aplicabilidade e a acertividade do algoritmo.

Gattas e Peixoto (2000), Barrequet (2007), Anzolin, Hounsell e Silva (2008) e Chen e Shen (2008) dividem a interpolação em três fases: correspondência, bifurcação e geração de malha. Por exemplo, se para cada um dos três sub-problemas haja três soluções algorítmicas, tem-se uma combinação ($3 \times 3 \times 3$) que resulta em 27 diferentes possíveis soluções para reconstrução do objeto.

2.2 O Problema da Correspondência

O problema da correspondência, também conhecido como *correspondence*, surge quando um objeto está representado por mais de um contorno em pelo menos um dos planos. É necessário determinar qual(is) contorno(s) se conectarão com qual(is) contorno(s) do plano adjacente [Meyers, 1994].

Anzolin, Hounsell e Silva (2008) definem o problema como:

Dado um conjunto i de curvas C_{ij} , onde $i = 1, \dots, n$ e n é o número de diferentes curvas no plano X_j , identificar todos subconjuntos de curvas no plano X_k que correspondem (que devem se conectadas) a outras no plano $X_{(k+1)}$.

A Figura 2.1 apresenta cinco soluções de correspondência que podem ser geradas a partir dos dois planos iniciais ilustrados no canto superior esquerdo (a). Em (b) nenhum contorno se conecta, enquanto em (c) e (f) tem-se duas formas de conectar todos os contornos, em (d) apenas há uma conexão, enquanto em (e) dois contornos se conectam a apenas um no plano adjacente.

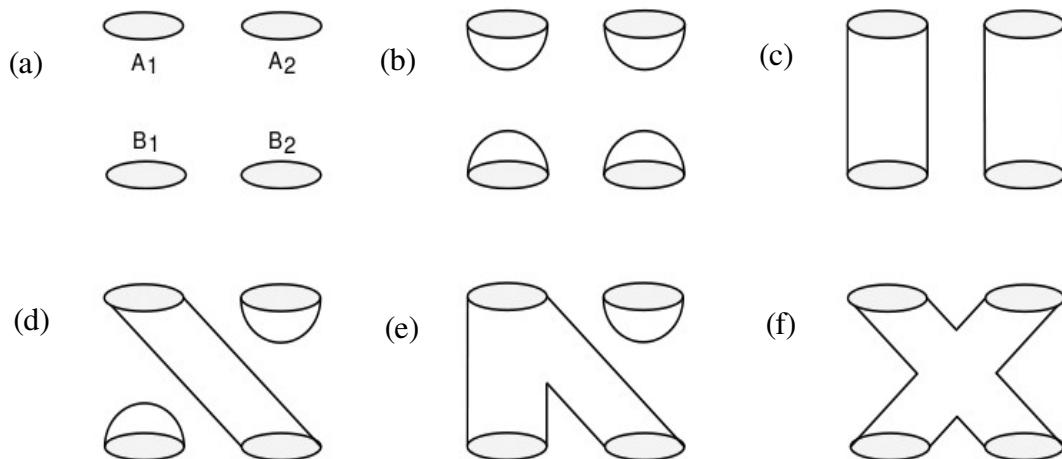


Figura 2.1. Diferentes formas de realizar correspondência. Fonte: Zamansky (2007).

O problema da correspondência só se torna trivial quando a relação entre as curvas nos planos é de 1 para 1 ($1 \leftrightarrow 1$), ou seja, um contorno em cada plano, onde os dois contornos se conectam (Figura 2.1 (c) e (d)). No caso de um dos planos conter um contorno e o outro mais de um ($1 \leftrightarrow n$) o problema da correspondência já deve levar em consideração algum critério para a conexão e o mesmo acontece quando são tratadas mais de um contorno nos dois planos ($m \leftrightarrow n$), como foi ilustrado na Figura 2.1 (e) e (f).

Segundo Gattass e Peixoto (2000), a decisão da correspondência pode ser tomada com base nas informações das distâncias entre os contornos. De um modo geral, se os contornos se encontram

distantes entre si, não há conexão entre eles, do contrário é possível estabelecer a conexão destes contornos.

2.3 O Problema da Geração de Malha

Após definido, na correspondência, quais contornos dos planos adjacentes se conectarão, é necessário gerar a superfície entre os contornos. A essa etapa dá-se o nome de geração de malha ou também, *tiling*.

Na geração da malha, define-se quais pontos de um contorno serão conectados a quais pontos do contorno correspondente para que uma malha seja gerada [Gattas e Peixoto, 2000]. Ou seja, com os contornos representados na forma poligonal o processo de geração de malha consiste na conexão de todos os vértices dos contornos fechados, conforme Figura 2.2.

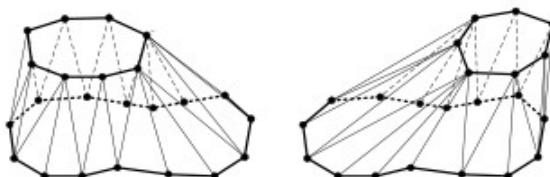


Figura 2.2. Geração da Malha. Fonte: Chen e Shen (2008).

Na geração de malha as arestas que conectam pontos vizinhos no mesmo contorno são denominados segmentos de contorno, enquanto arestas que conectam pontos de um contorno com outro em planos adjacentes são denominados *spans* [Meyers, 1994][Fuchs, 1977].

Para criar a malha, cada face da superfície final é composta por três (triangular) ou quatro (quadrangular) vértices conectados, sendo um ou dois destes vértices pertencentes a uma curva e um ou dois deles a outra. A Figura 2.3 (a) representa o processo de triangulação entre duas curvas, uma representada pelos pontos Q e outra pelos pontos P , e uma das faces formada pelos pontos P_j , Q_i e P_{j+1} da superfície que será gerada e (b) mostra a malha quadrangular com uma das faces formada pelos pontos M_1, N_1, N, M_2 .

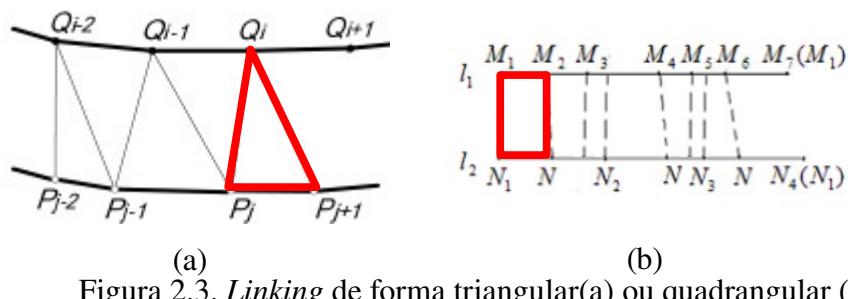


Figura 2.3. *Linking* de forma triangular(a) ou quadrangular (b).

O resultado final da malha pode variar dependendo da técnica aplicada. Na Figura 2.4 (a) tem-se os contornos que serão conectados e em (b) e (c) duas diferentes malhas conforme diferentes técnicas aplicadas.

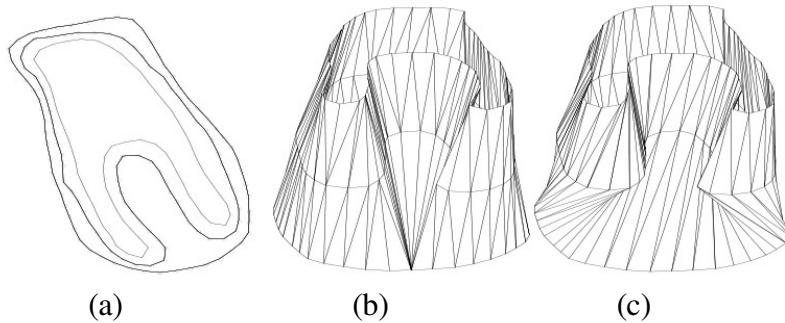


Figura 2.4. Em (a) contornos que devem se conectar. (b) e (c) diferentes métricas de geração de malha. Fonte: Meyers (1994).

É difícil obter uma conexão aceitável quando os contornos correspondentes têm formas diferentes [Gitlin, O'Rourke e Subramanian, 1996]. Nestes casos, a técnica aplicada talvez não tenha informação suficiente para gerar uma solução aceitável. Por este motivo, talvez seja necessário a intervenção do usuário na reconstrução [Meyers, 1994].

2.4 Problema da Bifurcação

O problema de bifurcação (*branching*) ocorre quando um contorno de um plano possui mais de uma curva correspondente no plano adjacente, ou seja, há uma relação $(1 \leftrightarrow n)$ ou $(m \leftrightarrow n)$ entre contornos adjacentes. Se um contorno possui m correspondentes, ocorrem $(m-1)$ bifurcações entre estes contornos. Nesse caso, há mudança topológica entre os dois planos, gerando uma sela conforme Figura 2.5 [Gattass e Peixoto, 2000].

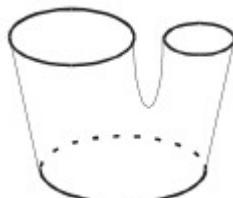


Figura 2.5. Bifurcação. Fonte: Gattass e Peixoto (2000).

A bifurcação representa os pontos de sela da superfície gerada, o que torna a geração da

malha algo delicado, já que definir o local onde será a intersecção não tem solução trivial. Porém, há várias soluções para resolver o problema da bifurcação. A Figura 2.6 apresenta quatro soluções para os contornos que se conectarão (a). Em (b) a bifurcação ocorre no ponto médio entre os planos, o que tende a ser mais próximo do real. Em (c) a divisão ocorre no plano que contém apenas um contorno, não gerando mudança topológica, enquanto em (d) a bifurcação acontece logo no início do intervalo dos planos, gerando um objeto com maior volume. Finalmente em (e) não é realizada a bifurcação, pois foi levado em conta a relação $1 \leftrightarrow 1$, o que fez que as duas curvas do plano superior fossem englobadas em um contorno.

Alguns algoritmos (Levin (1986), Floater e Westgaard (1996) e, Cong e Parvin (2001)) trazem soluções para resolver o problema do bifurcação através de funções implícitas ou reconstrução volumétrica. Porém, Anzolin, Hounsell e Silva (2008) apenas a consideram como o resultado final da intersecção da malha gerada onde há mudança topológica, não tratando especificamente esta etapa.

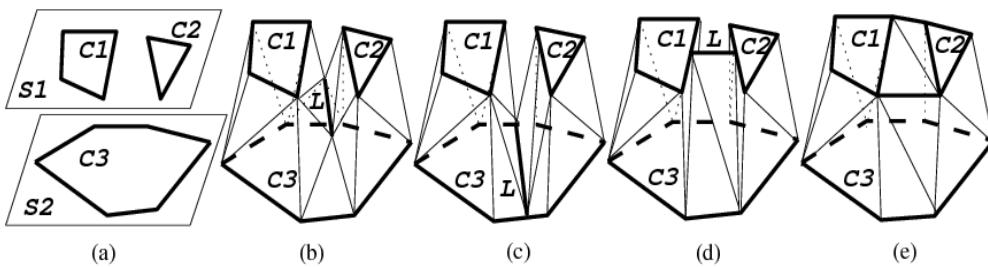


Figura 2.6. (a) Bifurcação de contornos adjacentes, (b) – (e) outras diferentes formas. Fonte:

Bajaj et al (1996).

2.5 Δ-Connection

O algoritmo Δ -Connection de Anzolin, Hounsell e Silva (2008), base deste trabalho, traz a solução para cada fase da interpolação da seguinte forma:

- Correspondência (*correspondence*): primeiramente são calculados os centróides de todos os contornos utilizando a média aritmética dos pontos extremos do eixo x e y (item 4.1.1). Feito isso, são calculadas as distâncias entre os centróides dos contornos a cada par de superfícies adjacentes. Neste momento, também é definido o maior e menor valor entre estas distâncias. Por fim, as distâncias são comparadas com um parâmetro Δ fornecido, na qual aplica-se a heurística de interpolação:
 - se Δ for menor que a distância mínima, nenhum contorno é interpolado;
 - se Δ for maior ou igual à distância máxima, todos os contornos são interpolados;

- se Δ estiver no intervalo entre a distância mínima e máxima, verifica-se também:
 - se a distância entre duas curvas for menor ou igual a Δ , então essas curvas são conectadas;
 - ou se a distância for maior que Δ , então não ocorrerá conexão dessas duas curvas.
- Geração da malha (*tiling*): após a definição de quais contornos se conectarão pela correspondência, um retângulo mínimo é envolvido em cada contorno, então um vértice do retângulo é eleito como base e o ponto de cada curva mais próximo a este vértice é considerado como inicial. Posteriormente os contornos são reordenados de forma que os pontos iniciais fiquem próximos no espaço e é realizado o balanceamento dos pontos de forma que os contornos tenham o mesmo número de pontos, para isso cada curva passará a ter seus pontos originais mais o número de pontos da curva correspondente. Neste passo, são identificados os maiores segmentos entre dois pontos consecutivos e cada novo ponto é adicionado na metade do segmento de maior distância, proporcionando assim um balanceamento de distâncias entre todos os pontos do contorno. Após esses procedimento a conexão entre os pontos dos contornos correspondentes é realizada.
- Bifurcação (*branching*): o tratamento da bifurcação é o resultado direto da geração de malha. Não é decidido o local em que ocorrerá a bifurcação, sendo este resultado da malha gerada entre as curvas, sem nenhum cálculo específico. Desta forma há um ganho computacional, porém a visualização é inadequada quanto feita internamente.

2.6 Equação Paramétrica da Reta

As equações paramétricas são formas de representar as retas, entre outras figuras geométricas, através de um ou mais parâmetros, ou seja, uma variável irá fazer a ligação de duas equações que pertencem a uma mesma reta [Venturi, 1949].

Dados um ponto $A(x, y, z)$ e um segmento de reta AB, conforme Figura 2.7, a coleção dos pontos que se expressam na forma

$$P(t) = A + t(B - A), \quad t \in \mathbb{R} \quad \text{Equação (2.1)}$$

é uma reta em \mathbb{R}^3 . Esta é a equação vetorial paramétrica do segmento de reta AB no \mathbb{R}^3 (t é chamado parâmetro)[Vainsencher , 2005].

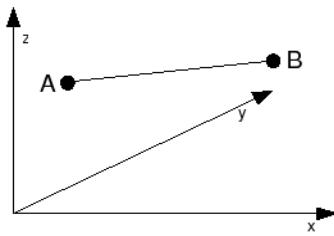


Figura 2.7. Um segmento de reta no \mathbb{R}^3 .

2.7 Produto Vetorial

Em matemática, o produto vetorial é uma operação binária sobre vetores no espaço. Pode ser denominado também como produto externo. Seu principal uso baseia-se no fato que o resultado de um produto vetorial é sempre perpendicular a ambos vetores originais [Venturi, 1949].

Dados dois vetores \mathbf{u} e \mathbf{v} pertencentes ao espaço \mathbb{R}^3 , define-se o produto vetorial $\mathbf{u} \times \mathbf{v}$ como sendo um terceiro vetor, onde a direção é perpendicular ao plano dos vetores \mathbf{u} e \mathbf{v} .

O sentido do vetor $\mathbf{w} = \mathbf{u} \times \mathbf{v}$ é dado pela regra da mão direita, no qual dispondendo os dedos médio e indicador da mão direita, apontando no mesmo sentido dos vetores \mathbf{u} e \mathbf{v} , o dedo polegar apontará o sentido do vetor \mathbf{w} [Vainsencher , 2005], conforme Figura 2.8.

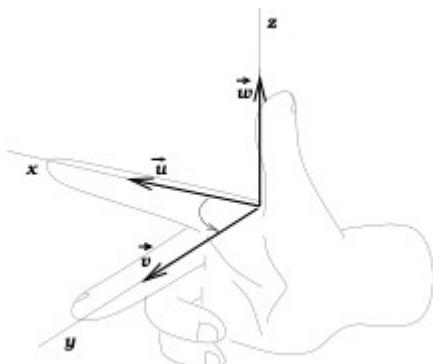


Figura 2.8. Regra da mão direita. Fonte: Vainsencher (2005).

Para encontrar o valor de $\mathbf{u} \times \mathbf{v}$ utiliza-se a equação:

$$\mathbf{u} \times \mathbf{v} = (y_u.z_v - y_v.z_u, z_u.x_v - z_v.x_u, x_u.y_v - x_v.y_u) \quad \text{Equação (2.2)}$$

Tal expressão pode ser escrita numa forma mais mnemônica, através do determinante:

$$\mathbf{u} \times \mathbf{v} = \begin{vmatrix} \mathbf{i} & \mathbf{j} & \mathbf{k} \\ x_u & y_u & z_u \\ x_v & y_v & z_v \end{vmatrix}$$

2.8 Produto Escalar

O produto interno de um vetor \mathbf{u} por um vetor \mathbf{v} é definido por $u \cdot v \cos \theta$, onde: \mathbf{u} e \mathbf{v} são os módulos dos vetores e θ é o menor ângulo entre eles.

Considerando os vetores na sua forma de componentes, encontra-se uma outra maneira de calcular o produto interno aplicando diretamente a definição acima da seguinte forma:

$$\mathbf{u} \cdot \mathbf{v} = (iu_x, ju_y, ku_z)(iv_x, jv_y, kv_z)$$

$$\mathbf{u} \cdot \mathbf{v} = iu_x iv_x + iu_x jv_y + iu_x kv_z + ju_y iv_x + ju_y jv_y + ju_y kv_z + ku_z iv_x + ku_z jv_y + ku_z kv_z$$

$$\text{como } i \cdot i = j \cdot j = k \cdot k = 1 \cdot 1 \cdot \cos 0^\circ = 1 \text{ e,}$$

$$i \cdot j = i \cdot k = j \cdot k = 1 \cdot 1 \cdot \cos 0^\circ = 0, \text{ tem-se:}$$

$$\mathbf{u} \cdot \mathbf{v} = u_x v_x + u_y v_y + u_z v_z \quad \text{Equação (2.3)}$$

onde então o produto interno será simplesmente a soma das parcelas provenientes dos produtos das coordenadas na mesma direção [Venturi, 1949].

2.8.1 Ponto Projetado em um Segmento de Reta 3D

Para determinar (projetar) um ponto $P' = (X_0, Y_0, Z_0)$ e um segmento de reta formado por dois pontos (A e B) no R^3 , conforme Figura 2.7, utiliza-se a equação paramétrica da reta (Eq.2.1). Sabendo-se que o produto escalar entre dois vetores é proporcional ao cosseno do ângulo entre eles e que a projeção gera um ângulo reto, então conclui-se:

$AB = B - A$ e como $P'P = P - P'$ substituindo na Eq.2.1 obtém-se:

$$P' = P - (t(B-A) + A)$$

$$P' = (P - A) - t(B-A).$$

Sendo

$$AB = \begin{vmatrix} B_x - A_x \\ B_y - A_y \\ B_z - A_z \end{vmatrix}$$

$$P'P = \begin{vmatrix} (P_x - A_x) - t(B_x - A_x) \\ (P_y - A_y) - t(B_y - A_y) \\ (P_z - A_z) - t(B_z - A_z) \end{vmatrix}$$

e sabendo que $\mathbf{AB} \times \mathbf{PP} = |\mathbf{AB}| |\mathbf{P'P}| \cos 90^\circ = 0$ (produto escalar), então realizando o produto escalar através da Eq.2.3 e isolando a variável t tem-se:

$$t = \frac{(B_x - A_x)(P_x - A_x) + (B_y - A_y)(P_y - A_y) + (B_z - A_z)(P_z - A_z)}{(B_x - A_x)^2 + (B_y - A_y)^2 + (B_z - A_z)^2} \quad \text{Equação (2.4)}$$

No qual tem-se valor $0 \leq t \leq 1$ se o ponto é projetado no segmento ou qualquer outro valor se não é. Substituindo t na equação paramétrica $(P - A) = (t(B-A))$, obtém-se o novo ponto [Mortenson, 2006].

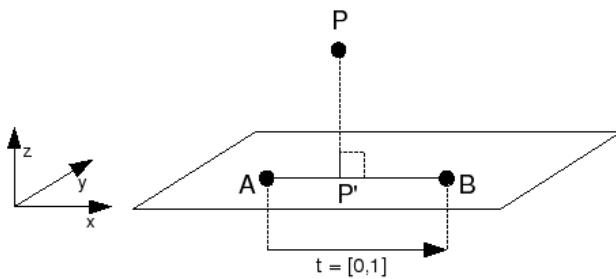


Figura 2.9. Projeção de um ponto em um segmento de reta.

2.9 Conceitos Estatísticos

Para demonstrar informações mais precisas acerca de uma análise de dados, diferentes formas devem ser consideradas. Só assim garante-se que elas estarão representando a realidade, ou algo muito próximo. A seguir serão apresentados alguns desses modelos estatísticos.

Média Aritmética

A média aritmética do conjunto x_1, x_2, \dots, x_n é representada por \bar{x} e calculada por:

$$\bar{x} = (x_1 + x_2 + \dots + x_n) / n = (\sum x_i) / n \quad [\text{Milone. 2003}, \text{[Soares, 2005]}]$$

Mediana

A mediana de um conjunto de valores, anotada por m_e , é definida como sendo o valor que

separa o conjunto ordenado em dois subconjuntos do mesmo tamanho. Assim se “n” (número de elementos) é ímpar a mediana é o valor central do conjunto. Caso contrário, a mediana é a média dos valores centrais do conjunto [Milone. 2003], [Soares, 2005]. Tem-se:

$$m_e = x_{(n+1)/2} \text{ se } "n" \text{ é ímpar e}$$

$$m_e = [x_{(n/2)} + x_{(n/2)+1}] / 2 \text{ se } "n" \text{ é par .}$$

Moda

A moda de um conjunto de valores, anotada por m_o e é definida como sendo o valor (ou os valores) do conjunto que mais se repete. A moda ao contrário da mediana e da média pode não ser única [Milone. 2003], [Soares, 2005].

Desvio Padrão

O desvio padrão define-se como a raiz quadrada da variância (medida de dispersão). O desvio padrão é representado por “s” [Milone. 2003], [Soares, 2005], sendo:

$$s = \sqrt{\sum \frac{(x_i - \bar{x})^2}{n}} = \sqrt{\sum \frac{x_i^2}{n} - \bar{x}^2}$$

O desvio padrão é uma medida padronizada dos desvios de cada valor individual até o valor da média aritmética.

3 Trabalhos Relacionados

A seguir são discutidas as soluções adotadas por vários autores quanto ao problema da geração de malha (*tiling*) e correspondência (*correspondence*), mesmo que suas aplicações se destinem a solução mais completa de reconstrução 3D. Os trabalhos serão apresentados em ordem cronológica, divididos entre as duas etapas.

3.1 Correspondência (*Correspondence*)

Seguem nos próximos subitens, os trabalhos correlatos que utilizam heurísticas para a solução do problema da correspondência.

3.1.1 Correspondência pelo Mínimo Custo

Skinner (1991) resolveu o problema da correspondência considerando o menor custo entre os contornos adjacentes, representados em um grafo. Cada nó do grafo equivale a um contorno e todas as conexões de cada nó representam as possíveis correspondências dos contornos no plano adjacente. Conforme demonstra a Figura 3.1, em (a) estão dois planos adjacentes com seus contornos e em (b) todas as possíveis correspondências de cada contorno.

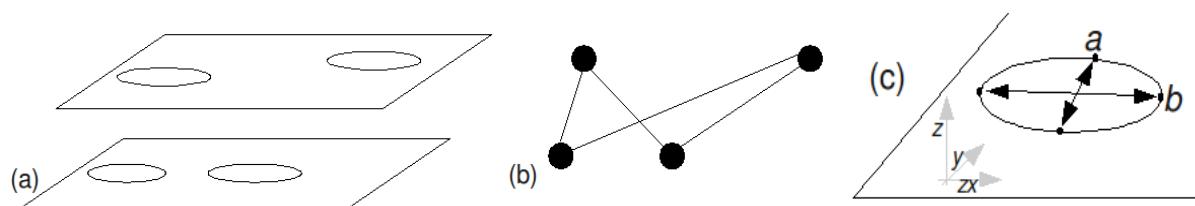


Figura 3.1. Contornos iniciais e sua representação em um grafo.

Para definir custos para cada aresta do grafo, é calculada a elipse que melhor se ajusta a cada par de contorno adjacente. Para obter a elipse, Skinner (1991) considera x , y e z como as coordenadas do centróide do contorno e a e b como o maior e menor comprimentos dos eixos da elipse que envolve o contorno, aplicados no \mathbb{R}^5 (para simplificar, pode-se assumir que os contornos estão em planos paralelos ao plano xy , ou seja, desconsidera-se o plano z). A Figura 3.2 (c) apresenta os obter os valores a e b de um contorno formado por quatro pontos, por exemplo. O custo de uma aresta no grafo é o valor da distância Euclidiana no espaço. Então, sendo $C1$ e $C2$ contornos adjacentes, o custo da aresta entre os dois é:

$$\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2 + (a_2 - a_1)^2 + (b_2 - b_1)^2} \quad \text{Equação 3.1}$$

Após realizado os cálculos para todas possíveis arestas do grafo, é definido como conexão, sempre o primeiro ordinal dos custos de cada nó, ou seja, para cada nó do grafo o menor custo com os outros nós do plano adjacente (possíveis conexões).

Ao final, a árvore de menor distância do grafo representa a solução para o problema da correspondência, conforme Figura 3.2, onde em (a) tem-se os contornos que se correspondem através das setas e em (b) o grafo que definiu a conexão, que no caso, os menores custos de cada nó geraram uma bifurcação.

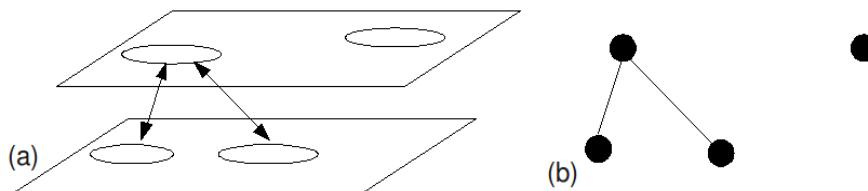


Figura 3.2. Contornos que se conectam e o grafo de definição da correspondência.

O método apresenta bons resultados em objetos ramificados porém, sua principal limitação é a inflexibilidade para resolver diferentes tipos de topologias [Meyers, 1994].

3.1.2 Correspondência por Sobreposição

No trabalho proposto por Barequet e Sharir (1996) são tratados contornos com mudanças topológicas, isto é, mais de um contorno em um ou nos dois planos ($1 \rightarrow n$ ou $m \rightarrow n$). Para

realizar a decisão da correspondência é realizada a projeção no R^2 de cada par de planos adjacentes. Os contornos que se sobrepõem, são conectados, seguindo a seguinte heurística: caso os contornos se sobreponham serão conectados, do contrário não. A Figura 3.3 demonstra os contornos em dois planos adjacentes (a), a projeção no R^2 e a intersecção resultante que define a correspondência (b) e a correspondência resultante (c).

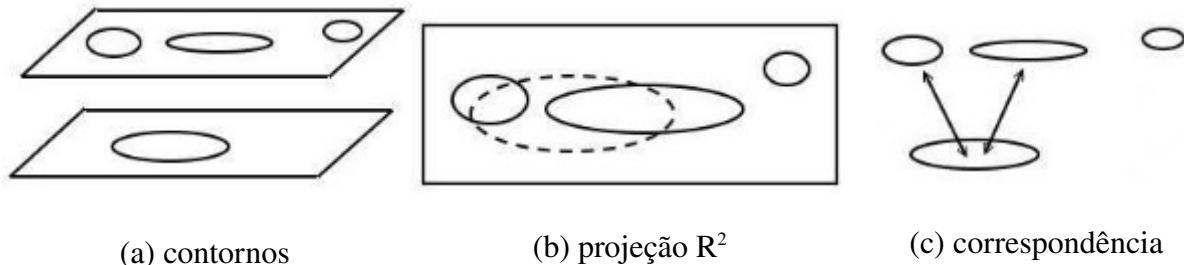


Figura 3.3. Definição da correspondência. Fonte: Anzolin (2006).

A Figura 3.4 ilustra os contornos iniciais (a), a projeção dos planos (b) e o resultado final da reconstrução, onde no plano superior ocorre mudança topológica.

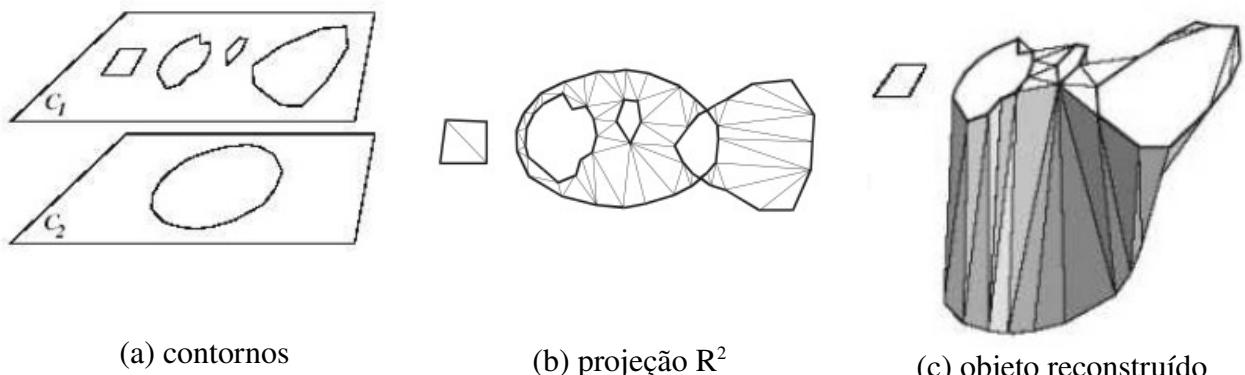


Figura 3.4. Objeto reconstruído sem bifurcação. Fonte: Gatass e Peixoto (2000)

No trabalho de Barequet e Sharir (1996) a heurística só é utilizada pra definir a correspondência dos contornos e o tratamento de bifurcações não é considerado, sendo um dos poucos trabalhos que utilizam heurísticas e não consideram uma das etapas de reconstrução 3D.

3.1.3 Correspondência por Distância no Domínio dos Inteiros

O β -*Connection* é uma técnica de reconstrução tridimensional baseada na triangulação de Delaunay e nas idéias de Nonato [Nonato, 1998]. Desta forma, a estratégia de reconstrução proposta

é volumétrica e baseada em contornos. Essa técnica oferece uma família de modelos que podem ser obtidos a partir de um mesmo conjunto de seções planares, tornando possível múltiplas opções para resolver o problema de correspondência. Para resolver este problema o algoritmo efetua um cálculo de distância entre contornos e recebe um parâmetro β do usuário pra decidir quais se conectarão. Este parâmetro é um valor inteiro positivo que expressa uma distância entre os contornos para a conexão. A heurística definida é a seguinte: Se a distância entre dois contornos quaisquer for menor que o valor do parâmetro β definido pelo usuário, então estes contornos serão conectados. Isto é, para um β definido pelo usuário, todos contornos que estejam a uma distância menor ou igual a β ficarão conectados.

A contribuição do β -*Connection* é que a distância é calculada pela avaliação do número de volumes com aparência cônica de tamanhos uniformizados, que são necessários para promover a conexão. Esses volumes, quando sobrepostos sequencialmente, descrevem a forma como será construída a bifurcação, uma vez que não se repetem no espaço entre planos. A Figura 3.5 demonstra os volumes cônicos calculados entre os planos, os quais servirão de base para a conexão dos contornos, conforme detalhado a seguir.

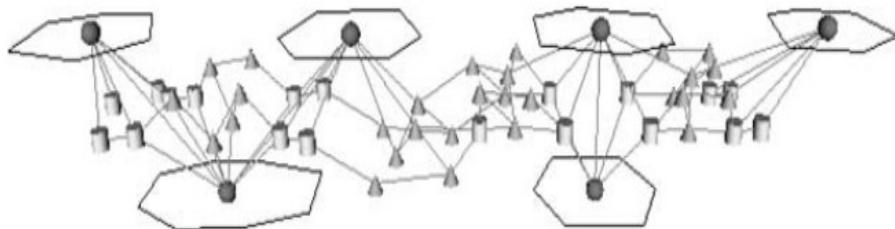


Figura 3.5. Volumes cônicos entre os planos. Fonte: Nonato, (1998).

A Figura 3.6 apresenta diferentes soluções de conexão resultantes da estratégia proposta por Vargas (2001). A partir dos contornos situados em planos paralelos (a), na primeira solução (b) o parâmetro β definido é menor que todas as distâncias entre os contornos de dois planos consecutivos, não ocorrendo nenhuma conexão, na segunda solução (c) o valor do parâmetro β vale 1, então todas os contornos de dois planos consecutivos com distância entre eles menor ou igual a 1 são conectados, e na última ilustração (d) é atribuído o valor 3 para β sendo conectados os contornos adjacentes com distâncias inferior ao parâmetro.

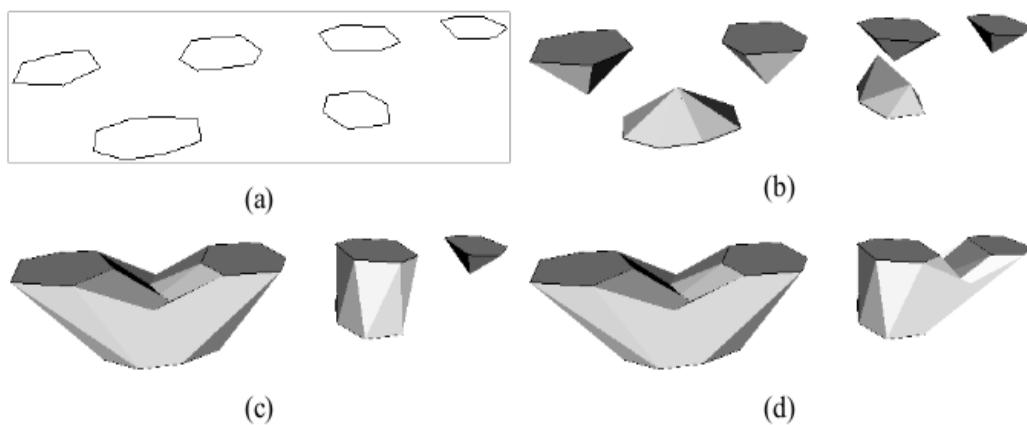


Figura 3.6. Diferentes valores para β . Fonte: Vargas (2001).

Segundo Vargas (2001), a técnica de reconstrução β -*Connection* oferece maior flexibilidade na escolha das componentes conexas, pois partindo de um mesmo conjunto de seções planares pode-se chegar a diferentes formas de objetos, o que é difícil através de outros algoritmos da literatura, resolvendo os problemas de ramificação e geração de malha de forma satisfatória.

3.1.4 Correspondência por Alinhamento dos Centróides

No algoritmo de Anzolin, Hounsell e Silva (2008), a correspondência é realizada em três etapas. Primeiramente é calculado o centróide de todos os contornos do modelo através da média aritmética dos seus pontos extremos no R^2 . Posteriormente, são calculadas as distâncias entre todos os centróides dos contornos de dois planos adjacentes. Neste passo Anzolin, Hounsell e Silva (2008) optaram por desconsiderar uma dimensão e realizar o cálculo com duas dimensões, também desprezando a raiz quadrada, ao qual trouxe otimização do processo. Na Figura 3.7, os centróides estão representados pelos pontos e as distâncias entre eles pelas setas com valores de $d2D[0,0]$, $d2D[1,1]$, $d2D[1,0]$, $d2D[0,1]$, os contornos pontilhados estão em um plano e os contínuos em outro adjacente. Finalizado esse passo, é informado ao usuário o valor mínimo e máximo das distâncias anteriormente calculadas.

Com base no valor mínimo e máximo o algoritmo recebe um parâmetro Δ , definido pelo usuário, que é comparado com cada distância calculada, afim de determinar se dois contornos serão conectados ou não. A heurística definida é a seguinte:

- Se Δ for menor que a distância mínima, nenhum contorno é interpolado;
- Se Δ for maior ou igual que a distância máxima, todos os contornos são interpolados;
- Se Δ estiver no intervalo entre a distância mínima e máxima, é realizada a seguinte regra para cada uma das distâncias:
 - Se a distância entre os dois contornos for menor ou igual a Δ , então esses contornos são conectados;
 - Se a distância for maior que Δ , então não ocorrerá conexão dessas dois contornos.

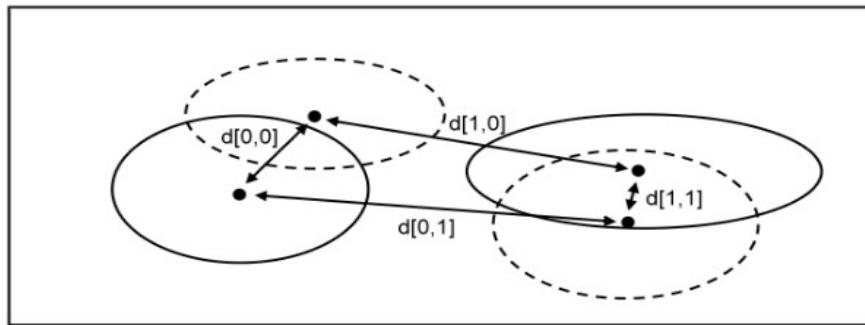


Figura 3.7. Distâncias entre os centróides.

Todas as soluções adotam, de forma explícita ou implícita, uma condição que reflete formas alternativas de avaliar a distância dos contornos nos planos adjacentes. Esta condição fica cada vez mais explícita e flexível se, considerar as soluções na sequência indicada a seguir:

1. Sobreposição dos Contornos;
2. Mínimo Custo;
3. Distância no Domínio dos Inteiros
4. Alinhamento dos Centróides.

Um dos algoritmos já incorpora aspectos de forma de contorno (2) e outros até a bifurcação (1 e 3), mas nenhum destes, entretanto, leva em consideração um indicativo de tamanho relativo dos contornos.

3.2 Geração de Malha (*Tiling*)

Os itens a seguir tratarão de técnicas para realizar a geração de malha entre os contornos que se correspondem, ou seja, que se conectarão.

3.2.1 Distribuição Proporcional Local

Li, Ma e Tan (2006) propõem que, para gerar a malha entre dois contornos adjacentes o número de pontos dos contornos devem ser iguais, de forma que se tenha correspondência um-para-um. Para isso, quando necessário, são adicionados no contorno de menor número de pontos a diferença de pontos entre os contornos. Ou seja, assumido que o maior número de pontos é M e o menor número é N, N passará a ter $N+(M-N)$ pontos.

A geração de malha segundo Li, Ma e Tan (2006) ocorre da seguinte forma:

Os dois pontos mais próximos no \mathbb{R}^3 entre os contornos adjacentes são definidos como iniciais. Na Figura 3.8 definimos M_1 e N_1 , indicados pelos círculos, como os pontos mais próximos. Então lineariza-se o perímetro dos dois polígonos de forma a criar duas retas (l_1 e l_2), que devem ter mesma direção. A linearização é calculada através da distância entre os pontos, consequentemente os valores de M, por exemplo, iniciando em M_1 (sempre com valor zero) serão $M_2 = d2D(M_1, M_2)$, $M_3 = d2D(M_1, M_2) + d2D(M_2, M_3)$, $M_4 = d2D(M_1, M_2) + d2D(M_2, M_3) + d2D(M_3, M_4)$, assim sucessivamente, conforme mostra a Figura 3.9. O mesmo ocorre com os pontos de N.

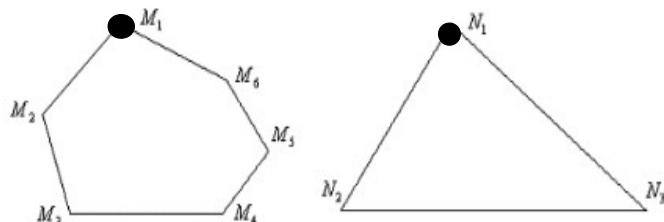


Figura 3.8. Dois contornos com números de pontos diferentes ($M=6$ e $N=3$). Fonte: Li, Ma e Tan (2006).

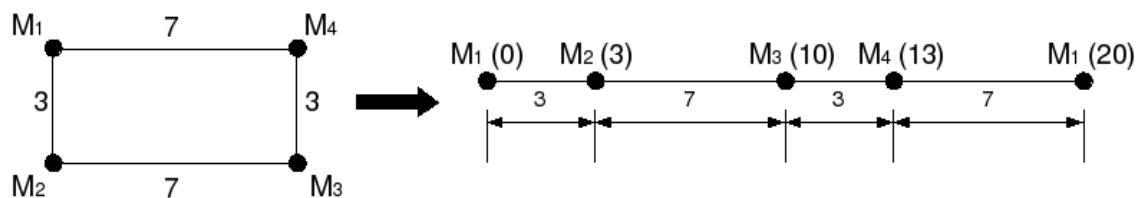


Figura 3.9. Linearização de um contorno.

Após isso, o valor de cada ponto é dividido pelo somatório do valor de todos os pontos, por

exemplo, $M_1/(\sum M_{1-M})$, $M_2/(\sum M_{1-M})$, $M_3/(\sum M_{1-M})$, $M_4/(\sum M_{1-M})$ de forma a obter uma sequência de valores denominada por T, o mesmo é feito em N obtendo uma sequência S, sendo $T = \{t_2, \dots, t_M\}$ e $S = \{s_2, \dots, s_N\}$. Então, a partir de cada valor de menor número de pontos (S) é procurado o valor mais próximo na sequência T, conforme Figura 3.10. Note que a correspondência inicial entre os pontos de M e N é dada pela maior proximidade e não pela exatidão. Por exemplo, M_3 poderia ter valor $t=0,33$ e estaria suficientemente perto de N_2 com $s=0,36$, fazendo com que ocorra um pequeno acúmulo de erro no cálculo da proporcionalidade.

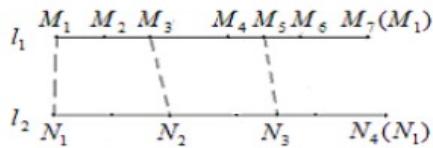


Figura 3.10. Contornos linearizados e com correspondência inicial dos pontos.

Os pontos que restaram em T serão criados proporcionalmente em S. Para determinar o novo ponto N é tomada a equação paramétrica do segmento de reta em questão em M e reproduzida proporcionalmente a mesma relação de distância (paramétrica) no contorno N, daí o nome de Distância Proporcional Local. A Figura 3.11 exemplifica este passo mostrando o que ocorre com cada valor restante. No caso M_2 restou para ser criado proporcionalmente em I_2 , sendo assim, faz-se a relação x está para 0.2 assim como 0.6 está para 0.5, tendo como resultado $x = 0.24$, novo valor que será inserido na sequência S.

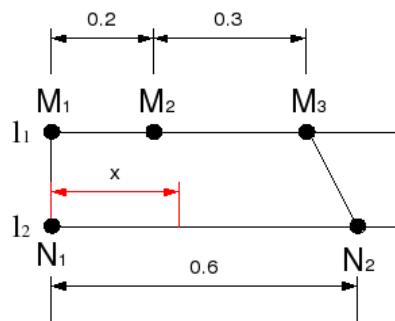


Figura 3.11. Criação do valor proporcional M_2 de I_1 em I_2 .

Com a sequência S com seus novos pontos, utiliza-se a Eq.3.2 para obter todos os pontos N

tridimensionalmente, formando a nova curva N balanceada.

$$M_i = (1-t_i)M_1 + t_i M_7 \text{ e } N_j = (1 - s_j)N_1 + s_j N_4 \text{ onde:} \quad \text{Equação (3.2)}$$

$$t_i \in (0, 1), s_j \in (0, 1), i = 2, \dots, M, j = 2, \dots, N$$

Ao final todos M – N pontos no contorno M podem ser combinados com os novos pontos no contorno N, de forma que se tenha correspondência um-para-um. Após a geração de pontos são feitas conexões repetitivas gerando triângulos de maneira trivial. Para aplicação deste algoritmo, Li, Ma e Tan (2006) pressupõem que os contornos tenham certa concentricidade e que os contornos estejam na mesma direção, pois do contrário será gerado um objeto com muitas torções na malha.

A principal vantagem que o algoritmo traz é o balanceamento de pontos, no qual os novos pontos são distribuídos proporcionalmente à curva M em N, o que faz que regiões de um contorno com muita concentração de pontos possuam correspondência de pontos proporcionalmente semelhantes no respectivo contorno da superfície adjacente. Porém em contornos com muitos pontos terá um grande custo computacional devido à linearização, às comparações a fim de encontrar os pontos iniciais que se relacionam e na geração do novo contorno tridimensionalmente.

3.2.2 Menor Span

Chen, Chen, Chiang e Hsieh (2007) realizam a conexão triangular entre os pontos mais próximos dos contornos. Para tal, define M com pontos $M_1, M_2, M_3, \dots, M_m$ para o contorno inferior e N com pontos $N_1, N_2, N_3, \dots, N_n$ para o superior. Os vértices são indexados seguindo o sentido horário observando os contornos adjacentes pelo topo.

São atribuídos como os pontos iniciais (M_i, N_i) os dois pontos que possuírem a menor distância tridimensional (d3D) entre os pontos dos contornos adjacentes, ou seja, a menor distância no \mathbb{R}^3 . M_iN_i é considerada um *span* do primeiro triângulo que será reconstruído. O triângulo será obtido adicionando mais um *span* que poderá ser N_{j+1} ou M_{i+1} . Este *span* (N_{j+1} ou M_{i+1}) é definida conforme critério da Eq.3.3, onde a estratégia para o próximo vértice é baseada no menor *span*. O triângulo $M_iN_jN_{j+1}$ é formado se $d3D(M_i, N_{j+1}) < d3D(M_{i+1}, N_j)$, do contrário será $M_iN_jN_{i+1}$, conforme Figura 3.12. O procedimento continua até todos os pontos da curva N sejam conectados à M .

$$\text{MIN}(\text{d3D}(M_i, N_{j+1}), \text{d3D}(M_{i+1}, N_j))$$

Equação (3.3)

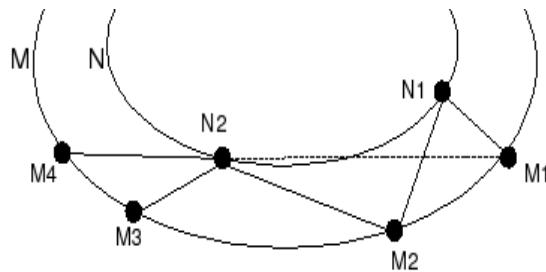


Figura 3.12. Geração da malha pela menor aresta.

Ao final, o algoritimo de Chen, Chen, Chiang e Hsieig (2007) não gera nenhum ponto adicional, porém pelo fato de calcular a distância entre pontos tridimensionalmente tem um gasto computacional considerável. Também caso o número de pontos dos contornos sejam muito diferentes, um ponto do contorno de menor número conectará muitos pontos do outro contorno, podendo assim gerar *spans* extensos e consequentemente uma imagem distorcida.

3.2.3 Balanceamento pela Distância

Para geração de malha, Anzolin, Hounsell e Silva (2008) definiram que para cada par de contornos a serem conectados devem ser analisados os números de pontos que cada uma possui. Supondo que se tenha o contorno M que será interpolado com o contorno adjacente N, cada curva passará a ter seus pontos mais o número de pontos da curva que se conectarão, ou seja, a soma dos dois conjuntos de pontos ($M + N$), tendo em vista que apesar de aumentar a estrutura de dados do objeto que será gerado, quanto mais pontos forem distribuídos na curva, maior a suavidade e perfeição da malha. A geração da malha conterá $2(M + N)$ pontos.

A atribuição dos novos pontos na curva não é feita de forma arbitrária, a cada dois pontos consecutivos é analisada a distância entre eles e cada novo ponto é adicionado entre os pontos que tiverem a maior distância entre si, proporcionando assim um balanceamento dos pontos do contornos pelas distâncias entre todos os pontos de cada contorno, o que evita que se obtenham pontos concentrados em uma parte específica da curva. A Figura 3.13 mostra em (a) duas curvas que serão interpoladas, uma com quatro e outra com seis pontos. Depois da etapa de atribuição de pontos (b) cada uma contém dez pontos distribuídos de acordo com a técnica apresentada.

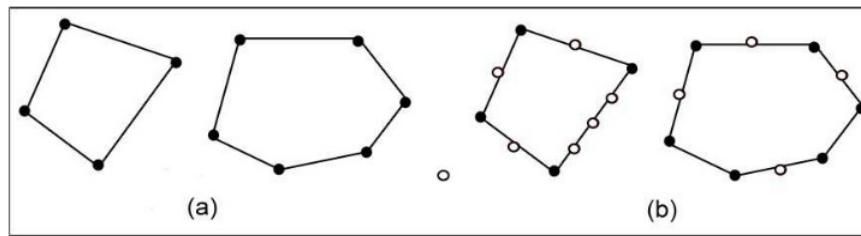


Figura 3.13. Balanceamento pelas distâncias. Fonte: Anzolin et al (2008).

Para definir os pontos iniciais é calculado o *bounding box* em cada contorno e eleito um vértice do *bounding box*. Os pontos que estiverem mais próximos no \mathbb{R}^2 ao vértice do retângulo serão os pontos iniciais da reordenação e geração da malha, conforme mostra Figura 3.14.

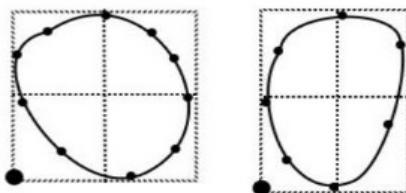


Figura 3.14. Definição dos pontos iniciais.

Após as etapas de definição do ponto inicial e balanceamento de pontos, basta ser feita uma triangulação entre os vértices dos dois contornos. Para a triangulação é gerada uma face (triângulo) para cada dois pontos consecutivos do primeiro contorno e um do segundo, como ilustrado na Figura 3.15 onde temos as faces “P1Q1P2” e “Q1P2Q2”, respectivamente.

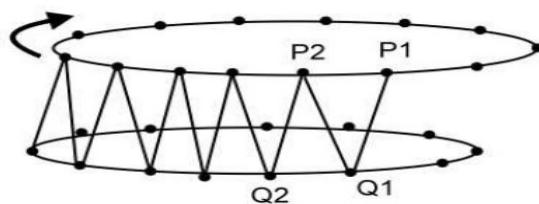


Figura 3.15. Triangulação para geração de malha. Fonte: Anzolin et al (2008).

Anzolin, Hounsell e Silva (2008) trazem uma solução completa para as etapas de geração da malha e correspondência. Devido ao fato de criar duas vezes a soma de pontos dos contornos que se conectarão tende a gerar faces suaves, porém, tem um custo computacional considerável devida a criação dos pontos e geração das faces.

3.3 Métricas

Um problema chave na reconstrução tridimensional é avaliar qual a qualidade do objeto final gerado, conforme a técnica utilizada. Pelo fato de não se ter apenas uma única solução na interpolação, segundo Barrequet, Shapiro e Tan (2000) a mensuração da qualidade do objeto final é um tanto subjetiva e intuitiva.

Mesmo assim, há duas formas de avaliação: uma definida pelo critério utilizado no algoritmo [Sloan e Painter, 1987] e outra por características do objeto final gerado.

Diferentes critérios algorítmicos têm sido utilizados para distinguir a qualidade das superfícies, segue um breve resumo de algumas dos mais importantes:

a) Menor Aresta: Cristiansen e Sederberg (1978) descreveram um método que escolhe o menor *span* entre duas possíveis durante a construção da malha. Embora rápido, o método falha em alguns casos comuns. Eles incorporaram a normalização do tamanho e posição na tentativa de evitar que em alguns casos o algoritmo falhasse. Já Kaneda et al (1987) modificaram a proposta de Cristiansen e Sederberg (1978) na tentativa de evitar alguns problemas encontrados em contornos que tivessem significantes diferenças topológicas.

Kaneda et al (1987) e Chen, Chen e Chiang (2007) introduziram o *triangle criteria* que seleciona a menor aresta possível de um triângulo para cada *span* compartilhada entre dois triângulos, conforme Figura 3.5;

b) Área: Fuchs et al (1977) utilizou a Área Mínima para este propósito. Segundo Meyers (1994) é, provavelmente, a melhor e mais utilizada métrica entre as existentes. É fácil de computá-la e frequentemente produz bons resultados;

c) Volume: Keppel (1975) utilizou como critério o Volume Máximo. A maior contribuição do trabalho de Keppel (1975) foi demonstrar que poderia ser calculada a contribuição de um simples *tile* (superfície unitária formada pela geração de malha entre dois contornos) em uma função objetiva relacionada ao volume total de um objeto.

Já para avaliar a qualidade de um objeto reconstruído, o aspecto visual seria uma forma interessante porém, pelo fato de ser um critério subjetivo, não pode ser realizado computacionalmente; mas através de características do objeto como: tempo de reconstrução, número de pontos gerados e tamanho dos *spans*, pode-se definir critérios que indicam a qualidade do objeto.

Para tamanho das arestas deve-se atentar que somente a média aritmética dos *spans* pode não indicar a realidade pelo fato que pode-se ter alguns *spans* muito longos e outros muito curtos, gerando uma média aritmética aceitável, porém um objeto provavelmente deformado. Por exemplo, conforme Figura 3.16, há *spans* extensos e outros curtos, fazendo com que a média aritmética mostre que o valor médio de cada aresta é aceitável, o que não corresponde à realidade.

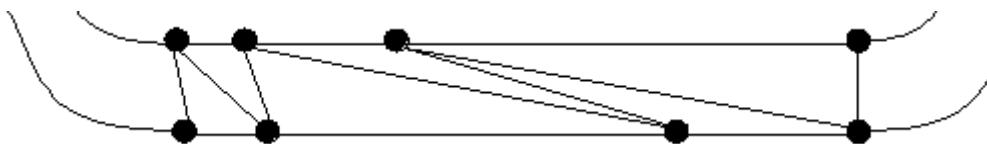


Figura 3.16. Tamanho dos *spans* entre dois contornos.

Vale ressaltar que os objetos gerados com *spans* maiores do que os necessários estão associados a objetos que dão o aspecto de estarem retorcidos em determinadas seções de interpolação.

Uma vez que o tamanho da *span* é uma métrica que aparece tanto relacionada ao aspecto visual quanto à qualidade do algoritmo em si, então se vê que avaliar criteriosamente esta informação pode ser uma métrica única e valiosa para o processo de reconstrução. Para melhorar esta informação, além da média aritmética dos *spans*, também deve ser considerado a maior e menor *span*, mediana e desvio padrão. Com estas informações adicionais pode-se obter um resultado mais próximo da realidade.

4 Projeto de Solução

Este capítulo descreve a subdivisão de etapas identificadas na geração da malha, bem como um conjunto de soluções para cada uma. Também apresenta diferentes soluções para a etapa da correspondência.

4.1 Visão Geral da Geração da Malha

Através de uma análise da etapa de geração da malha na reconstrução tridimensional percebeu-se que esta pode ser subdividida em 5 principais sub-etapas: equivalência das curvas (*curve offsetting*), orientação (*curve orientation*), entrelaçamento (*curve twisting*), balanceamento (*point balancing*) e conexão (*point linking*).

Baseando-se no exposto, criou-se a Tabela 4.1 detalhando todas as etapas e sub-etapas da geração da malha, bem como os algoritmos que se inserem em cada item para enfatizar tanto a subdivisão interna da etapa quanto os números de soluções que existem para elas, oriundas tanto da literatura quanto propostas por esta pesquisa.

A Tabela 4.1 tem objetivo de permitir uma visão geral de todo o processo ao mesmo tempo que, segue de referência para facilitar o entendimento das propostas que são apresentadas ao longo do tempo. Os itens da Tabela 4.1 que possuem a anotação (-) são os algoritmos cuja expectativa é de ter menor custo computacional, enquanto os com anotação (+) são os que cuja expectativa são os de ter maior custo. Os itens que possuem (*) são propostas ou extensões apresentadas por esta pesquisa e os itens com (#) são soluções já existentes no algoritmo Δ -Connection. Estas expectativas foram baseadas nos cálculos utilizados nos algoritmos.

Equivalência dos Curvas <i>(Curve Offsetting)</i>	Com (*)	Centróide	a) <i>Bounding Box</i> (#)	
			b) <i>Bounding Circle</i> (*) c) $\sum p$ (somatório dos pontos) (-)(#) d) Centro de gravidade (+)	
Orientação das Curvas <i>(Curve Orientation)</i>	Sem (#)			
	Com (+) (*) Sem (-) (#)			
Torção da Curvas <i>(Curve Twisting)</i>	Ponto Inicial <i>(First Point)</i>	a) Mais próximos a um ponto auxiliar (-) (#) b) Mais próximos no R^2 (*) c) Mais próximos no R^3 (+)		
	Reordenação <i>(Reordering)</i>	a) Força bruta (movendo ponto a ponto) (+) b) Indireta (#) c) Função (-) (*)		
Balanceamento dos Pontos <i>(Point Balancing)</i>	Com (*)	Por índice	a) Distribuição Cega (*) b) Distribuição Proporcional Local (+) c) Distribuição Pela Distância (#) d) Distribuição Enxuta (-) (*)	
		No espaço	a) Distribuição Exata (-) (*) b) Distribuição Proximal (+) (*)	
	Sem (#)			
Conexão dos Pontos <i>(Point Linking)</i>	Padrão	a) Triangular (-) (#) b) Retangular (*) c) Múltipla (*)(+)		
	Desbalanceadas	R^2 (-) (*)		
		R^3 (+)		

Tabela 4.1. Subdivisão da etapa de geração de malha.

Equivalência das Curvas

A equivalência das curvas trata da traslação de um dos contornos de forma que fique concêntrico, ou equivalente ao contorno correspondente na superfície adjacente, a fim de evitar distorções caso os contornos estejam distantes. Daí o nome da etapa. Esta etapa é optativa, pode-se tê-la ou não. Caso opte-se por ser incluída o objeto tende a ter menos torção na malha, o que resulta

em um melhor aspecto visual. Para tal, conforme Figura 4.1 dadas duas curvas que serão conectadas (a), faz-se a translação de uma das curva de forma que fiquem concêntricas (b), então os pontos serão conectados (c) e por fim é realizada novamente a translação para que a primeira curva retorne à posição original (d).

Para que as curvas estejam concêntricas deve-se utilizar o centróide de cada uma como parâmetro para a translação.

Orientação das Curvas

Nesta etapa é avaliada a orientação do sentido dos pontos das curvas. As curvas devem estar todas orientadas ou sentido horário ou anti-horário. Não pode haver uma alternância entre os dois sentidos no mesmo objeto, pois isto gerará torção na malha.

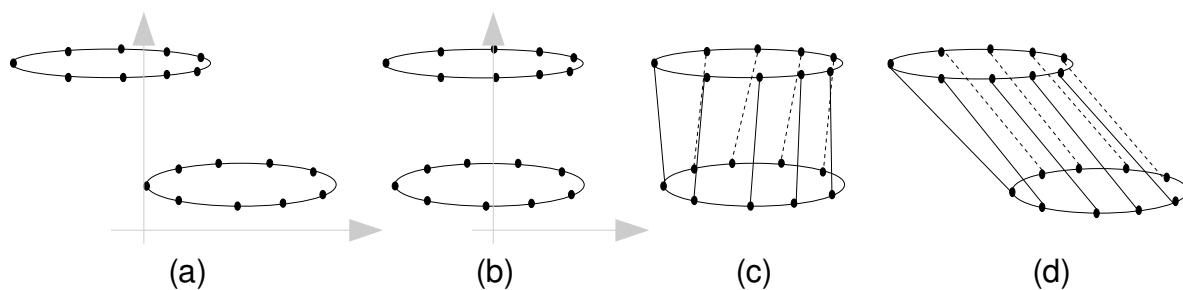


Figura 4.1. Sub-etapa da geração de malha: equivalência das curvas.

Torção das Curvas

Esta etapa trata da reordenação dos pontos do contorno de forma que os pontos iniciais nos dois contornos sejam o primeiro ponto e está dividida em duas fases. Primeiramente, define-se quais serão os pontos iniciais de cada contorno. Após isto os contornos são reordenados em função dos pontos iniciais.

Balanceamento dos Pontos

Esta etapa pode ocorrer ou não, mas da mesma forma que a equivalência das curvas, caso seja inserida no algoritmo tende a resultar em um objeto com melhor aspecto visual do que sem. Trata da forma com que o número de pontos dos contornos serão balanceados, ou seja, se serão

alterados de posição ou se novos pontos serão adicionados. Caso sejam adicionados novos pontos, como eles estarão distribuídos na curva. O balanceamento traz suavidade ao objeto.

Conexão dos Pontos

Esta sub-etapa traz a forma com que os pontos de dois contornos que devem serem interpoladas serão conectados, ou seja, como os pontos da curva k se conectarão com os pontos da curva $k + 1$.

4.1.1 Soluções para Equivalência das Curvas

O reposicionamento de contornos é dependente do cálculo do centróide. O centróide é o ponto no interior de uma forma geométrica que define o seu centro geométrico [Figueiredo e Carvalho, 1991]. Para calcular o centróide de uma curva pode-se utilizar diferentes técnicas como seu centro de gravidade, média aritmética dos pontos, *bounding box* e *bounding circle*.

Para calcular o centro gravitacional de um polígono deve-se dividí-lo em triângulos e realizar uma relação de entre os centros de gravidade desses triângulos. Essa divisão em sub-polígonos é realizada pois, o cálculo do centro de gravidade de um triângulo é trivial. O centro gravitacional de um triângulo é obtido através da média aritmética de todos os seus vértices. Pelo fato de serem gerados triângulos e calculados vários centróides e suas relações, tem-se um grande custo computacional [Figueiredo e Carvalho, 1991].

O *bounding box* de um contorno é o paralelepípedo mínimo que contém todo conjunto de pontos. Bastam dois pontos para a definir, pois as diferenças de valores nas coordenadas x, y e z estabelecem o valor dos lados da *bounding box*. Como menos informações de entrada são necessárias, menor será o número de cálculo, sendo assim, há um grande ganho computacional [Figueiredo e Carvalho, 1991].

Já o *bounding circle* é o círculo mínimo que contém todos os pontos. Consiste em determinar os dois pontos mais distantes do contorno e calcular o ponto médio entre estes, que será o centróide.

A solução pela média de todos os vértices da curva é uma solução de baixo custo computacional, porém pode ter o resultado afetado por características do contorno, por exemplo, a

concentração de vértices em uma região do contorno causa um deslocamento indesejável no centróide em direção a essa região.

Na Figura 4.2 (a) o ponto C representa o cálculo do centróide pelo centro gravitacional e o ponto M pela média de todos os vértices. Em (b) o centróide foi obtido através da técnica de *bounding box* e em (c) através da *bounding circle*.

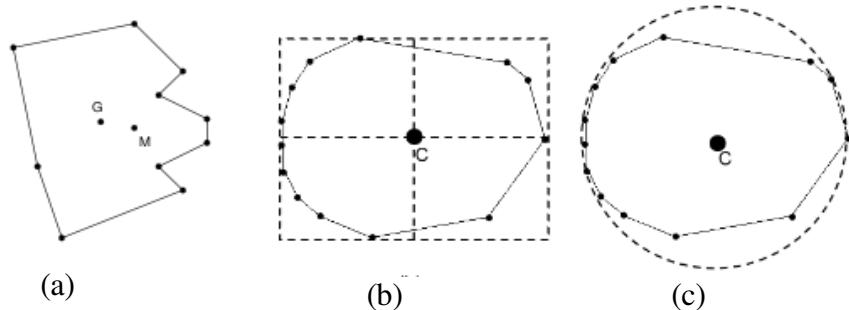


Figura 4.2 Diferentes formas de obter o centróide de um contorno.

4.1.2 Soluções para Orientação das Curvas

Baseando-se no produto vetorial, item 2.7, pode-se verificar se um contorno está orientado no mesmo sentido do contorno adjacente.

Conforme Figura 4.3, dados dois contornos P e Q para verificar se os pontos estão no mesmo sentido faz-se:

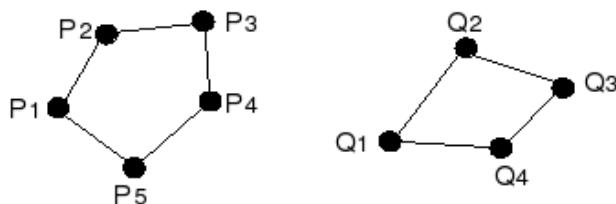


Figura 4.3. Dois contornos com pontos no mesmo sentido.

$$u_1 = (P_1P_2 \times P_2P_3) / (|P_1P_2| \times |P_2P_3|) \text{ para contorno 1 e,}$$

$$u_2 = (Q_1Q_2 \times Q_2Q_3) / (|Q_1Q_2| \times |Q_2Q_3|) \text{ para contorno 2.}$$

Com os vetores resultantes calcula-se:

$$(u_1 \times u_2) / (|u_1| \times |u_2|) = \cos \theta$$

Se o valor resultante for igual a 1, os vetores são colineares e estão no mesmo sentido, do contrário, caso o resultado for -1, os vetores estão em sentidos opostos.

Desta forma, caso algum dos contornos esteja divergindo do contorno adjacente pode-se reordená-lo para que se tenha os contornos de todos os planos no mesmo sentido, evitando torções na malha gerada. Este passo, se inserido no momento da reconstrução trará maior custo computacional devido a exigência de processamento, mas garantirá que todas as curvas estão orientadas no mesmo sentido.

4.1.3 Soluções para Torção das Curvas

Há várias formas de se definir os pontos iniciais das curvas entre duas curvas. Mais comumente utilizados são os pontos mais próximos bi ou tridimensionalmente [Chen, Chen, Chiang e Hsieh, 2007],[Li, Ma e Tan , 2006] ou ainda, gerando um ponto auxiliar que elegerá como pontos iniciais, o ponto mais próximo a ele [Anzolin, Hounsell e Silva, 2008]. Percebe-se também que é possível fazer a busca da proximidade com os pontos projetados em um mesmo plano, ou seja, através do cálculo da d2D.

Após identificar quais seriam os pontos de referência dos contornos, é útil reordenar os pontos de forma que estes passem a ser os primeiros. Para reordenação das curvas as técnicas identificadas foram: força bruta, forma indireta ou por função são formas de reordenação, onde:

a) forma indireta: realizada por Anzolin, Hounsell e Silva (2008), onde os pontos iniciais também são reordenados de forma que fiquem no começo da curva, porém para tal são utilizados dois vetores auxiliares V1 e V2 para dividir o vetor principal, no qual V1 armazena os pontos que estão entre o índice do ponto inicial até o índice final do vetor e V2 que armazena do primeiro índice até o índice do ponto inicial definido. Ao final, são concatenados V1+V2 resultando no vetor reordenado. Nesta técnica também tem-se um pré-processamento, ou seja custo computacional, porém tem-se ganho temporal durante a execução.

b) força bruta: move-se o ponto unitariamente reordenando o vetor de pontos a cada passo até que todos os pontos estejam na posição que precisam estar. Esta opção traz um pré-processamento devido a reordenação de todas as curvas inicialmente, porém tem ganho

de desempenho durante o tempo de execução.

c) função: ao invés de alterar a posição do ponto no respectivo vetor, pode-se criar uma função de reposicionamento do índice, onde seja repassada a posição do ponto que se quer, as coordenadas x, y e z do ponto referente serão retornadas. Nesta técnica, não existe um pré-processamento, sendo assim tem-se ganho computacional, porém o tempo de execução será maior, pois o índice efetivo será calculado durante o processo. A vantagem desta técnica só seria percebida no caso específico que houvesse alta carga de pré-processamento e baixa de processamento propriamente dito. Entranto, este não parece ser um caso usual.

4.1.4 Soluções para Balanceamento dos Pontos

Além das técnicas apresentadas no capítulo 3, trabalhos relacionados, foram discutidas e criadas variações de algumas técnicas. Estas variações têm por finalidade colaborar em algumas das métricas que serão apresentadas na próxima seção.

A etapa de balanceamento dos pontos é executada para garantir que as duas curvas possuam o mesmo número de pontos antes de serem interpoladas, pois dependendo da torção e conexão dos pontos, poderá ser alcançada uma melhor relação entre os pontos, fazendo com que gere-se menos *spans* inclinados e então, a superfície tende a ter menos (ou nenhuma) torção. Isto facilita a triangulação e permite que a malha gerada seja mais suave, evitando inclusive intersecções entre as faces. Caso haja diferença de pontos entre os contornos, será necessário que tenha mais de um *span* para o mesmo ponto, gerando *spans* longos e inclinados e consequentemente um efeito de torção da superfície.

Os algoritmos de balanceamento dos pontos estão divididos em dois tipos: no espaço ou por índice. Os algoritmos de balanceamento por índice utilizam como base o índice do vetor que armazena os pontos da curva para inserção de novos pontos ou modificação de algum existente. Para algoritmos de balanceamento utilizam como base os pontos no R^2 ou R^3 , estes estarão inseridos no tipo de balanceamento no espaço.

a) Distribuição Proporcional Exata

Uma variação de Li, Ma e Tan (2006) que foi percebida e pode trazer contribuições seria ao

em vez de somente criar a diferença de pontos dos contornos em N, fazer com que os pontos de N também sejam replicados em M. Desta forma o erro que ia se acumulando na seleção dos pontos nos segmentos das curvas não ocorrerá mais e a proporcionalidade será exata.

Para isso, depois de obtidas as duas sequências T e S, é realizada a união entre seus valores, de forma a criar apenas uma sequência R para as duas curvas. Caso T e S possuam registros com valores iguais somente um permanecerá na sequência final, conforme Figura 4.4.

$$\begin{aligned} T &= \{0, 0.22, 0.33, 0.55, 0.66, 0.77\} \\ S &= \{0, 0.27, 0.63\} \\ \hline R &= \{0, 0.22, 0.27, 0.33, 0.55, 0.63, 0.66, 0.77\} \end{aligned}$$

Figura 4.4. União das sequências T e S.

Neste caso deverá ser aplicada a Eq.3.2 não só para N, mas para M também. Ou seja, serão realizados mais cálculos.

A correspondência um-para-um será mantida e a mesma distribuição proporcional dos pontos ocorrerá em M e N, conforme Figura 4.5.

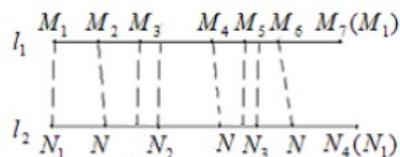


Figura 4.5. Criação dos novos pontos e conexão entre os contornos.

O objeto final terá um maior número de pontos gerados ($2(M+N)$), consequentemente maior processamento e um objeto com maior suavidade na superfície gerada. Da forma como proposto, são tomados intervalos de pontos e entre estes, criados pontos respeitando o posicionamento relativo em M e reproduzindo-o em N (e vice-versa). Isto melhora a distribuição proporcional de pontos de um contorno em relação a outro, sem calcular as distâncias no espaço Euclidiano.

b) Distribuição Proximal

Outra proposta baseada em Li, Ma e Tan (2006) é ao em vez de gerar os novos pontos (na quantidade da $DIF(M, N)$) pela equação paramétrica em busca da proporcionalidade e, portanto de

pontos próximos, o ponto é gerado através da projeção do ponto de M em um segmento de reta em N no R^3 (item 2.6), ou sejam, essa projeção garante que o ponto gerado é (são) o(s) ponto(s) faltante(s) mais próximo(s) (ou proximal, daí o nome escolhido) do seu equivalente no outro contorno.

Esta alternativa foi idealizada pela observação do fato de que, mesmo garantindo a proporcionalidade, o algoritmo original pode gerar pontos distantes entre si. Esta projeção sempre resultará em um ângulo de 90 graus com o segmento de reta pelo fato de ser a menor distância. Caso a projeção não coincidir com algum dos segmentos do contorno adjacente, ou seja, sua projeção não resulte em um ângulo de 90 graus em N, pode-se utilizar Distribuição Proporcional Local proposta inicialmente por Li, Ma e Tan (2006).

Na Figura 4.6, M tem 7 pontos e N tem 4 pontos. Serão criadas a diferença de M e N pontos, ou seja, 3 pontos. Os quadrados representam dois novos pontos criados através da projeção do ponto em M no segmento de reta em N. O triângulo representada um ponto criado pela Eq.3.3 devido este ponto não ter projeção em nenhum segmento de N, conforme as linhas tracejadas.

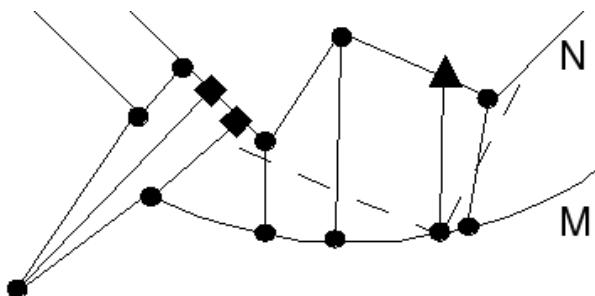


Figura 4.6. Projeção dos novos pontos.

O número de pontos gerados nesta técnica são equivalentes ao proposto por Li, Ma e Tan (2006), porém tem-se uma forma diferente de balancear os pontos, na qual pelo fato do ponto ser projetado não gerará pontos distantes, consequentemente menor tendência de deformação do objeto.

c) Distribuição Enxuta

Analizando a correspondência dos pontos proposta por Anzolin, Hounsell e Silva (2008), notou-se que o cálculo pode ser otimizado diminuindo o número de pontos gerados em cada contorno. Anzolin, Hounsell e Silva (2008) propõem que o número de pontos de M seja adicionado

em N e o número de pontos de N adicionado em N , resultando em um total de $2(M + N)$ pontos. Se utilizarmos a equação módulo de DIF (M, N) para encontrar os pontos que serão adicionados em N ao final teremos apenas $2(\text{MAX}(M,N))$ de pontos nos contornos. Ou seja, o número de pontos gerados e de faces serão menores, consequentemente tem-se um ganho computacional.

d) Distribuição Cega

Considerando M o contorno de maior número de pontos e N de menor, a distribuição cega ocorre inserindo n pontos em N , no qual n é resultado da Eq.4.1.

Esta estratégia objetiva distribuir um conjunto de pontos entre outros pontos, sem levar em consideração nenhum outro fator (de distância, por exemplo, daí o nome “cega”). A distribuição entretanto divide os pontos extras de maneira equitativa pelo outro contorno.

$$N \times (\text{Div}(M,N) - 1) + \text{Mod}(M,N) \quad \text{Equação (4.1)}$$

A distribuição é realizada em duas etapas, a primeira etapa identifica uma quantidade de pontos extras igual a ser gerada em todos os segmentos e a segunda etapa garante igualdade de pontos entre as duas curvas. Então, primeiramente serão criados em cada segmento de dois em dois pontos, no sentido horário, o número de pontos do resultado inteiro da divisão de M por N decrementado de um ($\text{Div}(M,N) - 1$). Quando um contorno tem um número proporcional interno, esta etapa é suficiente. Entretanto, é possível sobrar um resto na divisão que deve então ser distribuído pelos “novos” segmentos de pontos. Após será obtido o resto da divisão de M por N ($\text{Mod}(M,N)$) e esse número de pontos serão distribuídos nos segmentos.

Conforme Figura 4.7, M possui 8 pontos e N possui 3 pontos, considera-se M_1 e N_1 os pontos iniciais, aplicando o algoritmo tem-se $\text{Div}(8,3)-1 = 1$, no qual é representado por pontos circulares. O resto da divisão é 2 e estes pontos estão representados por quadrados.

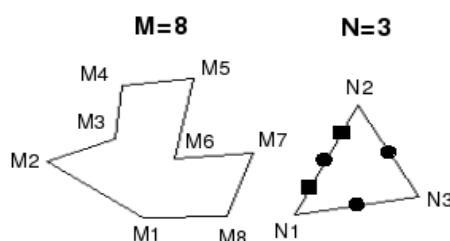


Figura 4.7. Distribuição cega de pontos.

Esta solução distribui os pontos de forma a diminuir o tamanho dos maiores segmentos do

contorno.

4.1.5 Soluções para Conexão dos Pontos

Há duas situações a serem consideradas na conexão dos pontos: primeira técnica básica é criar *spans* (arestas entre os contornos) com o menor tamanho possível. Este cálculo de tamanho pode ser realizado utilizando a distância no R^2 ou no R^3 . Esta estratégia funciona para quando o contorno está balancedo ou não. Entretanto, quando há balanceamento, podem ser usadas algumas outras estratégias, pois pode-se forçar (prever) padrões para faces criadas.

Para encontrar a distância no R^2 utiliza-se a A.1 (Apêndice A, Equação 1), enquanto para a distância no R^3 entre dois pontos utiliza-se a A.2 (Apêndice A, Equação 2).

Pelo fato da A.2 utilizar uma dimensão a mais no cálculo de cada distância, tende a ser mais dispendiosa computacionalmente.

Caso não seja considerada a distância, pode-se utilizar conforme Figura 4.8 :

- triangular (a), onde cada ponto se conecta a outros dois pontos sequenciais da curva adjacente, gerando faces triangulares. Por exemplo, M2N1M1 é um triângulo gerado na conexão dos contornos na Figura 4.8 (a). Isto se o número de pontos dos contornos forem iguais;
- múltipla (b), onde cada ponto da curva de menor número de pontos será vértice de $DIV(M,N)$ faces triangulares. Caso a divisão não seja exata, ou seja, $MOD(M,N) \neq 0$, o valor de $MOD(M,N)$ faces deverá ser distribuída unitariamente a partir do primeiro vértice, gerando mais um triângulo nos $MOD(M,N)$ primeiros vértices. Em (b) linhas contínuas estão os triângulos gerados pelo $DIV(5,3) = 1$ e, em linhas tracejadas os $MOD(5,3) = 2$ distribuídos unitariamente por vértice. Neste caso o número de pontos das curvas devem ser diferentes.
- ou retangular (c), onde cada ponto se conecta a apenas um ponto da curva adjacente, gerando faces retângulares como, por exemplo, M1N1NM2. Isto se o número de pontos das curvas forem iguais.

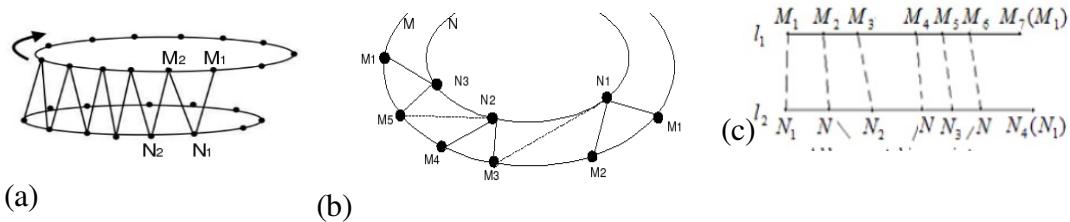


Figura 4.8. Formas de conexão dos pontos em padrões.

4.2 Visão Geral da Correspondência

Analizando os algoritmos da literatura que tratam a correspondência com regras heurísticas, percebeu-se que diferentemente da geração de malha, esta etapa não possibilita a fragmentação em vários pequenos passos. Desta forma, foi considerado que o processo de correspondência na reconstrução ocorre em apenas uma única etapa.

4.2.1 Mudança de Significado

Anteriormente o Δ significava distância e era orientado pelos seus valores mínimo e máximo, os quais serviam de base para o usuário entrar com o parâmetro, pois do contrário, haveria dificuldade para indicar o Δ desejado. Com isto, necessitava-se de um cálculo prévio para obter estes dois valores. Notou-se então, que pode ser considerado um valor percentual equivalente como entrada para variar entre os limites existentes, de forma que o usuário não necessite dos valores previamente e não seja necessário um pré-processamento para o cálculo dos mesmos. Sendo assim, formulou-se a equação de ajuste de significado:

$$\Delta\text{Percent} = (\Delta\text{max} - \Delta\text{min}) * \text{deltaPercent} + \Delta\text{min} \quad \text{Equação (4.2)}$$

onde, $\text{deltaPercent} \in [0,100]$ e é definido pelo usuário;

Δmin é a distância máxima entre todos os centróides e;

Δmin é a menor distância entre os centróides.

Desta forma mantém-se o mesmo processo de funcionamento do algoritmo, porém traz-se uma facilidade na inserção do parâmetro Δ através de um número percentual ao invés de um decimal.

4.2.2 Correspondência pela Geometria do Contorno

Baseando-se na solução de Anzolin, Hounsell e Silva (2008), percebeu-se que além da distância dos centróides dos contornos, também poderia ser considerada a relação entre as áreas ou comprimentos de cada contorno, pois contornos próximos com formas semelhantes tendem a se conectar [Meyers, 1994], melhorando assim a correspondência. Então, propôs-se conectar dois contornos considerando o parâmetro de atração baseado na distância, através da projeção de um contorno sobre outro, ponderado por uma medida de tamanho dos mesmos.

Com base nisto, para obter os valores Δ de cada relação de contornos adjacentes, foi proposta a equação:

$$\Delta\text{Percent} = \frac{d2D(C1, C2)}{f(S(C1), S(C2))} \quad \text{Equação (4.3)}$$

onde:

$S(C1)$ é um indicativo de tamanho do contorno 1;

$S(C2)$ do contorno 2 e;

$f(S(C1), S(C2))$ é uma função que define as relações entre os dois ($S(C1)$ e $S(C2)$).

Desta forma, a distância será compensada pela relação das áreas entre os dois, fazendo com que dois contornos de tamanhos grandes em relação a outros contornos existentes no modelo, que tenham centróides distantes, possam se conectar.

Para tal, há diferentes maneiras de calcular o valor de $S(x)$, onde foram identificadas: perímetro, área real, área do *bounding box* ou área do *bounding circle*. Da mesma forma, identificou-se que poderia utilizar a soma ou multiplicação como $f(x)$, pois estas operações tem como função misturar os valores de dois tamanhos de contornos.

Caso o valor de $f(x)$ seja igual a um, tem-se novamente o algoritmo inicial do Δ -Connection. Na seção 4.3 far-se-á uma descrição detalhada de cada técnica identificada para área e perímetro do contorno e a discussão sobre os resultados desta abordagem se encontrará na seção 5.1.

Para o caso do coração onde há problema de correspondência, dois fatores podem estar influenciando o funcionamento do Δ : a relação do tamanho dos contornos (em relação a todo modelo ou em relação aos pares de planos adjacentes) e o valor da distância em relação ao espectro

dos contornos adjacentes, o qual foi desconsiderado no Δ -Connection original quando utilizou d2D entre centróides.

4.2.3 Melhorando a Heurística de Correspondência do Alinhamento dos Centróides

Em se tratando de tamanho dos contornos, uma medida que pode ser utilizada é sua área ou perímetro. Pode-se sugerir que o valor limiar de conexão deva ser mais estreito para áreas ou tamanhos menores e mais abrangente para os maiores. Seria interessante então, que o Δ variasse em torno do valor definido pelo usuário e em uma folga com amplitude controlada (R , do inglês *range*), desde que esta não extrapole os limites existentes, conforme Figura 4.9, no qual estaria-se realizando uma correção adaptativa do tamanho dos contornos.



Figura 4.9. Variação do Δ .

Este ajuste pode ser obtido com a seguinte equação:

$$\Delta_{adj} = \Delta_{Percent} + R * \left[\frac{S(C1) * S(C2) - S_{min}^2}{S_{max}^2 - S_{min}^2} \right] \quad \text{Equação (4.4)}$$

onde, $R \in [0,100]$ e é definido pelo usuário;

$S(C1)$: é o tamanho do contorno C1;

$S(C2)$: é o tamanho do contorno C2;

S_{max} é o tamanho do maior contorno e;

S_{min} é o tamanho do menor.

Note que a proposta da Eq. 4.4 atende à Eq. 4.3, mas inclui uma variável extra de controle. A Eq. 4.4 foi concebida tendo-se em mente o fato de que as áreas de ambos os contornos precisaram ser consideradas (levando a um termo de área ao quadrado), mas que o ajuste decorrente não

poderia extrapolar os limites relacionados às áreas de maior e menor valores, que supostamente seriam S_{max}^2 e S_{min}^2 . Estes seriam os piores casos possíveis (extremos).

O fato de R ser multiplicado pela relação das áreas, também o torna um valor percentual deste intervalo, no qual com seu valor igual a zero tem-se a forma proposta inicialmente no Δ -Connection. Também para considerar a distância entre planos deve-se ter a opção de realizar o cálculo da distância no R^3 (d3D). Para o cálculo da área ou perímetro pode ser utilizado as técnicas da seção 4.3. Mesmo com estes melhoramentos a lógica do Δ -Connection permanece inalterada.

4.3 Solução para Tamanho do Contorno

Os itens a seguir descrevem as formas propostas pela presente pesquisa para o cálculo da função $S(x)$.

A) Perímetro

Para encontrar o perímetro de um contorno é realizada a soma das distâncias no R^2 (d2D) entre todos os pontos que estão conectados. Neste caso, pode-se também considerar a possibilidade do perímetro do *bounding box* e *circle*. São propostas apenas as distâncias entre pontos no R^2 , pois neste caso calcula-se cada contorno isoladamente, não tendo relação com nenhum de outro plano.

B) Área pela Caixa Envoltória (*Bounding Box*)

Para encontrar a área por *bounding box* (retângulo mínimo que envolve o contorno) de um contorno são percorridos todos os seus pontos, identificando os pontos extremos no eixo da ordenada e da abcissa. Obtém-se a distância no R^2 dos dois pontos extremos da ordenada e a distância dos pontos extremos da abcissa, multiplicam-se os valores obtidos de forma a se ter o valor da área do retângulo mínimo externo do contorno. A Figura 4.10 ilustra os pontos identificados, no caso com quadrado os pontos da ordenada e círculos os da abcissa. Para se obter a área *bounding box* multiplica-se o valor da distância entre os pontos extremos da ordenada pelos da abcissa.

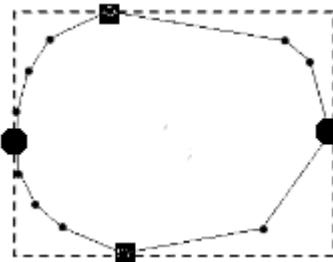


Figura 4.10. Definição dos pontos extremos da ordenada e acissa.

C) Área pela Circunferência Envoltória (*Bounding Circle*)

Para encontrar a área de um contorno por *bounding circle* (circunferência mínima que envolve o contorno), percorre-se todos os pontos a fim de encontrar os mais distantes no \mathbb{R}^2 . Após identificados, a distância entre eles é o diâmetro da circunferência. Como o valor da metade do diâmetro é o raio e a área da circunferência é o quadrado do raio multiplicado pelo valor de *Pi*, então aplicando a equação obtém-se a área *bounding circle*. Na Figura 4.11 estão identificados os pontos com maior distância do contorno, ou seja, o diâmetro da circunferência.

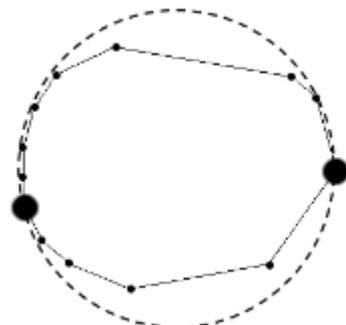


Figura 4.11. Definição dos pontos extremos do contorno.

D) Área Real

Para calcular a área real de um contorno (polígono irregular) deve-se dividí-lo em polígonos menores de fórmula de área conhecida e realizar a soma de todas as áreas obtidas, assim tendo a área real do contorno [Figueiredo e Carvalho, 1991]. Essa divisão em sub-polígonos é realizada pois, o cálculo da área de polígonos simples é trivial. Pelo fato de ser necessário gerar variados polígonos e, posteriormente calculados suas áreas, tem-se um grande custo computacional. Ainda quando este polígono é convexo, a subdivisão é mais custosa em termos computacionais devido necessidade de uma série de testes adicionais.

4.4 Arquitetura do Software

O presente trabalho tomou como base o trabalho Anzolin (2006), cuja arquitetura de software do algoritmo Δ -Connection sofreu alterações. Métodos e atributos existentes foram modificados e outros foram adicionados a fim de modularizar o algoritmo indo ao encontro do paradigma de orientação a objetos e para se ter a granularidade a ponto de poder inserir mais opções nas sub-etapas da reconstrução tridimensional.

A Figura 4.12 apresenta o diagrama de classes do Δ -Connection e a Figura 4.13 a nova arquitetura proposta no Δ^+ -Connection. A maior alteração na estrutura foi com relação a manter a alta coesão e baixo acoplamento entre as classes (premissas do paradigma de orientação a objetos), e por isso foram criados novos pacotes e arquivos específicos, dividindo-os entre contextos de:

- Interface, onde qualquer ação, manipulação ou característica de tela deve estar inserida (pacote com.udesc.larva.delta.ui);
- Manipulação de arquivos, na qual qualquer implementação necessária para manipular arquivos, por exemplo, ler XML ou gerar VRML devem estar neste contexto (pacote com.udesc.larva.delta.files);
- Algoritmos para interpolação, que inclui os algoritmos criados para resolver os sub-problemas. Neste contexto também são englobados cálculos auxiliares utilizados nos algoritmos como: distância entre pontos 2D ou 3D, centróides, reordenamento, geometria, topologia, entre outros (pacote com.udesc.larva.delta.interpolation).

Os dois sub-problemas foram divididos em métodos específicos, a fim de ir ao encontro das vantagens colocadas na seção 2.1.

Adicionalmente a estas alterações foram percebidas repetições desnecessárias de alguns cálculos (código duplicado, laços desnecessários e cálculo repetido dos centróides), além de formas cálculos onerosos computacionalmente que poderiam ser otimizados.

Apesar da reestruturação aplicada, o algoritmo Δ^+ -Connection mantém todas as funcionalidades do Δ -Connection, em especial a estratégia de decisão heurística da conexão dos contornos utilizando o Δ . A aplicação permite a edição de contornos a serem interpolados e toda informação relativa ao conjunto de curvas que foram editadas e salvas em um arquivo XML servem de entrada para o algoritmo de reconstrução, o qual realiza a interpolação e escreve um arquivo VRML com as informações geométricas do objeto gerado.

Este objeto pode ser visualizado no *browser* após o processo, onde é chamado pelo programa (caso esteja previamente configurado). O arquivo XML utilizado pelo algoritmo de interpolação não precisa necessariamente ser editado pela aplicação, podendo receber dados de fontes reais (obtidos via tomografia computadorizada, ressonância magnética ou ultrassonografia) editados em qualquer outro lugar e adaptados ao modelo XML. A seguir será apresentada a nova interface adicionada na aplicação para a entrada dos parâmetros utilizados durante a interpolação das curvas ne execução do algoritmo.

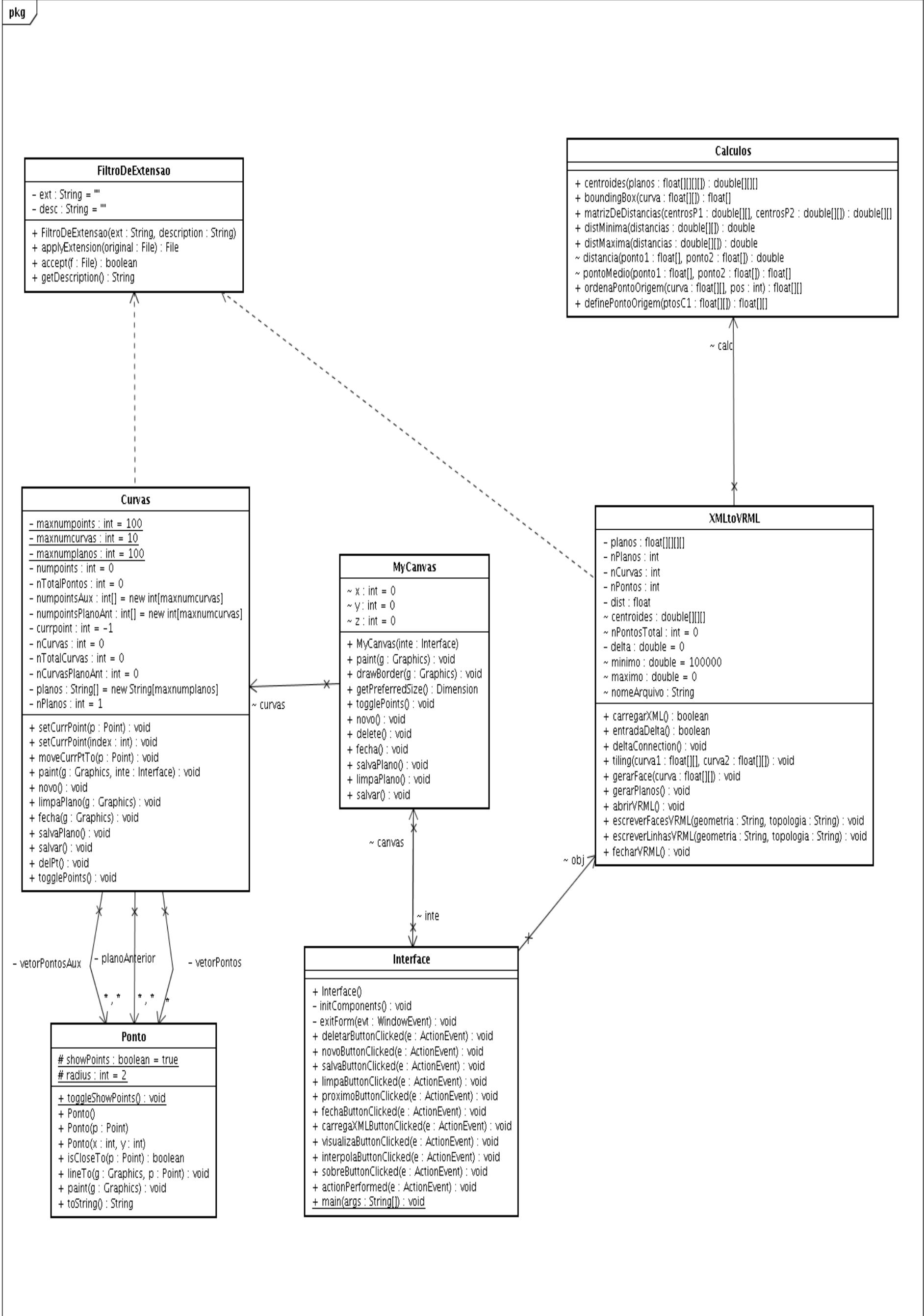


Figura 4.12. Diagrama de classes do Δ-Connection.

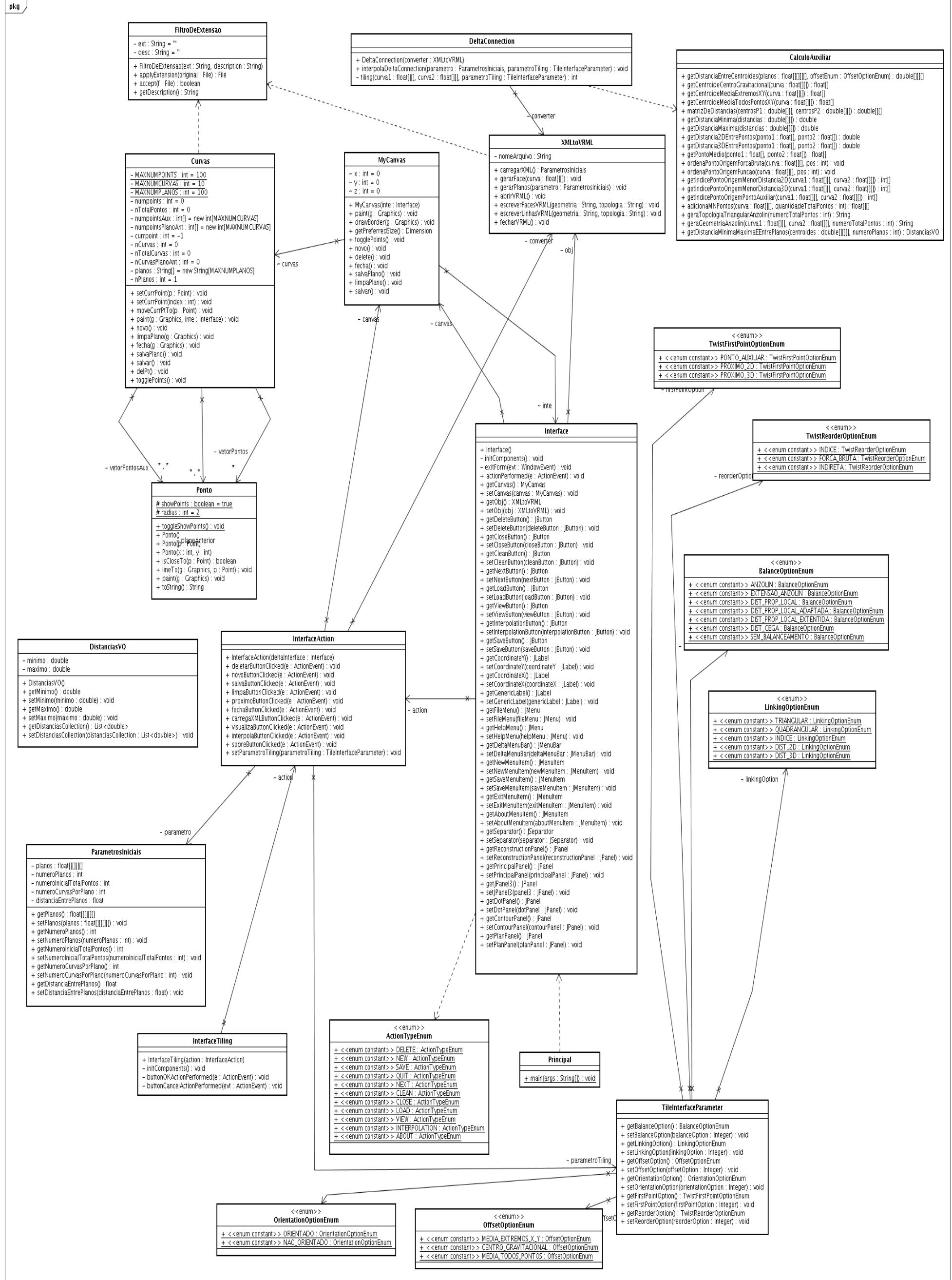


Figura 4.13. Diagrama de classes do Δ^+ -Connection.

4.5 Novas Funcionalidades

Nesta nova estrutura, além da modificação do código existente, foi adicionada uma interface antes do processo de interpolação ser executado, a qual oferece ao usuário a opção de parametrizar cada sub-etapa do processo de geração de malha, conforme Figura 4.14.

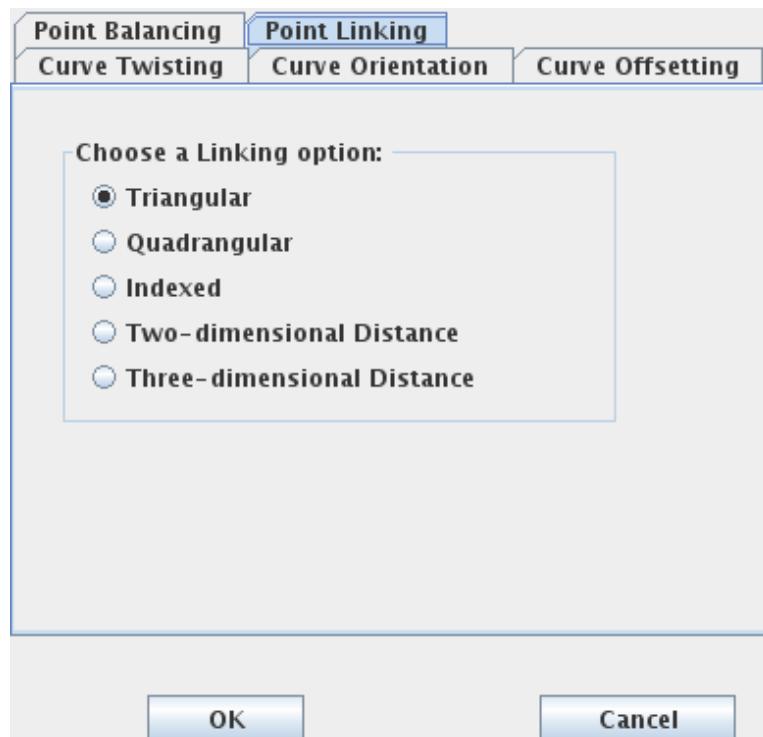


Figura 4.14. Interface de parametrização do processo de geração de malha.

Quanto às funcionalidades da nova interface têm-se:

- Aba *Point Balancing*: permite a escolha de qual algoritmo de balanceamento de pontos utilizar na interpolação;
- Aba *Point Linking*: permite a escolha de qual algoritmo de conexão dos pontos utilizar na interpolação;
- Aba *Curve Offsetting*: permite escolher um algoritmo de equivalência das curvas ou deixar sem nenhuma técnica de equivalência nesta sub-etapa.
- Aba *Curve Orientation*: nesta opção o usuário garante que todos os contornos estão orientados no mesmo sentido ou não. Caso não estejam, o algoritmo verificará cada contorno e as orientará no mesmo sentido.

- *Aba Curve Twisting*: nesta aba deverá ser parametrizada qual técnica utilizar para definir os pontos iniciais do contorno e, após isto, qual será a forma de reordenação das curvas.

Porém algumas técnicas não podem serem combinadas entre si. As identificadas são:

- Só se pode utilizar *linking* nas opções distância bi e tridimensional caso alguma técnica de equivalência seja escolhida. Conforme Figura 4.15 (a), caso os pontos mais próximos (bi ou tridimensional) sejam conectados sem translação a imagem gerada será distorcida. Transladando os contornos para que fiquem concêntricas e conectando os pontos (b), ao retornar à posição original (c) não haverá distorção.

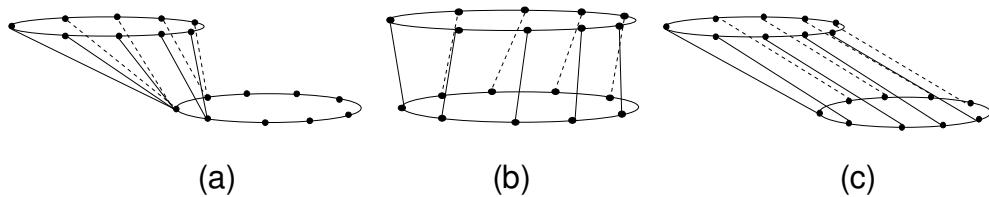


Figura 4.15. *Linking* sem (a) e com (c) a translação.

- Somente é viável utilizar a opção *linking* triangular e retangular se o número de pontos dos contornos forem iguais.

5 Implementação

Neste capítulo será apresentada a implementação do aplicativo Δ^+ -Connection, as novas interfaces criadas, estrutura e organização das classes e, os novos recursos.

5.1 Interfaces

Além da interface padrão existente no Δ -Connection foram adicionadas duas interfaces para a parametrização das etapas de geração de malha e correspondência. Estas interfaces foram desenvolvidas no idioma inglês objetivando padronizar e facilitar a alteração ou desenvolvimento de novas técnicas por um maior número de pesquisadores.

Na interface de parametrização da geração de malha houve a adequação do nome da opção de conexão dos pontos (*linking*) de *Indexed* para *Multiple*, o qual traz mais proximidade com a principal característica da técnica, gerar múltiplas conexões triangulares. A Figura 5.1 ilustra a interface final de parametrização da geração de malha.

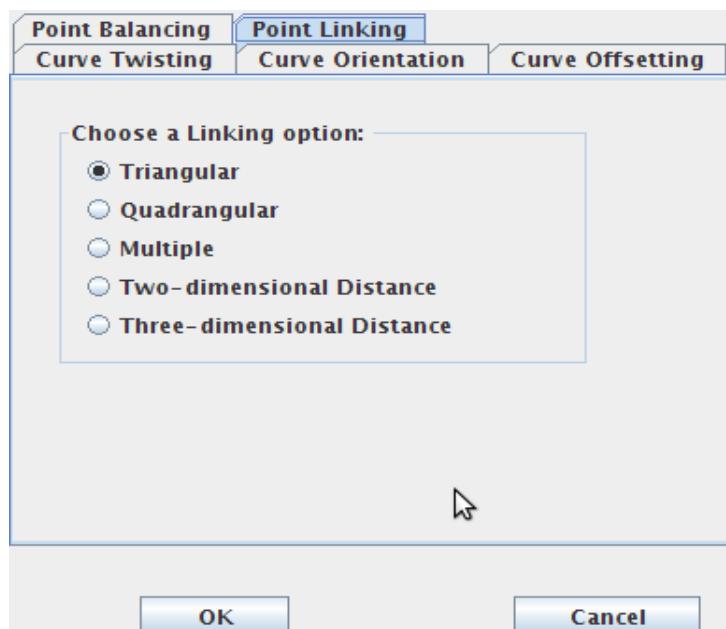


Figura 5.1. Interface de parametrização da geração de malha.

Para cada objeto a ser reconstruído, se forem consideradas todas as possibilidades de combinações das técnicas nas sub-etapas da geração de malha no Δ^+ -Connection, ter-se-ão 1440 diferentes soluções.

Inicialmente para a interface de parametrização da correspondência, visando a solução da seção 4.2.2 foi desenvolvida a interface conforme Figura 5.2, onde do lado esquerdo foi disponibilizado a parametrização do cálculo da área e do lado direito as opções para a função de relação $f(x)$. As opções disponíveis eram:

- *Default*: é a opção existente do Δ -Connection para definição da correspondência;
- *Perimeter*: efetua o cálculo do perímetro do contorno;
- *Bounding Box Area*: calcula a área *bounding box* do contorno;
- *Bounding Circle Area*: calcula a área *bounding circle* do contorno;
- + : operação de adição para $f(x)$;
- * : operação de multiplicação para $f(x)$.

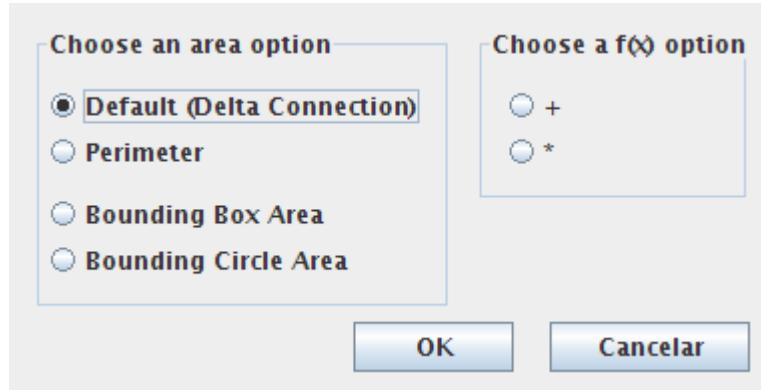


Figura 5.2. Interface de parametrização da correspondência.

Após gerar inúmeras vezes objetos com diferentes parametrizações, não chegou-se a um resultado visual final aceitável. Sendo assim, analisou-se novamente esta solução proposta e percebeu-se que a Eq. 4.3 não solucionava o problema, pois não se tinha efetivamente uma relação que variasse conforme o tamanho dos contornos. Com isso, adaptou-se a interface para a solução proposta na seção 4.2.2, tendo como resultado final a tela conforme Figura 5.3, onde:

- *Default*: é a opção existente do Δ -Connection para definição da correspondência;
- *Perimeter*: efetua o cálculo do perímetro do contorno;
- *Bounding Box Area*: calcula a área *bounding box* do contorno;
- *Bounding Circle Area*: calcula a área *bounding circle* do contorno;

- 2D: cálculo da distância entre centróides no R^2 ;
- 3D: cálculo da distância entre centróides no R^3 ;
- Delta (%): valor do parâmetro Δ , porém agora em valor percentual;
- R(%): valor do parâmetro de variação conforme a geometria dos contornos, caso seu valor seja zero, tem-se correspondência inicial do Δ -Connection.

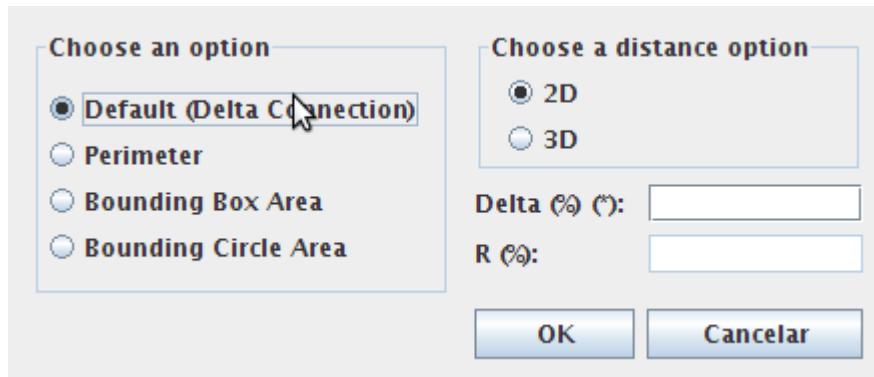


Figura 5.3. Nova interface de correspondência.

Com esta implementação foram gerados objetos com melhoramentos em alguns casos e sem impactar na qualidade de todos os objetos gerados. Estes resultados serão apresentados no próximo capítulo.

5.2 Arquitetura Final do Software

A partir do algoritmo existente foi proposta uma nova organização das classes e adição de novos recursos conforme descrito na seção 4.1. Porém, no desenvolvimento foi necessária a inserção de novas classes a fim de atender o proposto e também a implementação da correspondência.

Dentre estas alterações estão:

- CalculoMetrica: classe para realizar o cálculo dos *spans*, número final de pontos e, inserí-los em um arquivo de saída;
- InterfaceCorrespondence: classe que especifica as características e opções disponíveis na tela de parametrização da correspondência;
- CorrespondenceAreaTypeEnum: enumeração dos tipos de tamanho do contorno a utilizar no cálculo da correspondência;

- CorrespondenceInterfaceParameter: classe que une os parâmetros da interface.
- CalculoAuxiliar: foram adicionados novos métodos para realizar cálculos da área de *bounding box*, *bounding circle* e perímetro de um contorno, métodos para a função f(x) da correspondência.

Para a última versão de solução de correspondência (seção 4.2.2), foi implementado o código conforme Figura 5.4.

```

if(!parametroCorrespondence.getCorrespondenceAreaType().equals(CorrespondenceAreaTypeEnum.DEFAULT_DELTA)){
    double numerator = areasCalc[i][a]*areasCalc[i+1][b] - Math.pow(areas.getMinimo(), 2);
    double divisor = Math.pow(areas.getMaximo(), 2) - Math.pow(areas.getMinimo(), 2);
    double areaRelation = numerator/divisor;
    deltaCalculated = deltaDefault + range*areaRelation;
} else{
    deltaCalculated = deltaDefault;
}
if(distancias[a][b]<= deltaCalculated){ //interpola curva a e b
    int numeroPontosTiling = this.tiling(parametro.getPlanos()[i][a], parametro.getPlanos()[i+1][b], parametroTiling);
    this.numeroPontosGerados += (numeroPontosTiling -(parametro.getPlanos()[i][a].length + parametro.getPlanos()[i+1][b]));
} else{ //apenas gera uma face para cada curva
    this.converter.gerarFace(parametro.getPlanos()[i][a]);
    this.converter.gerarFace(parametro.getPlanos()[i+1][b]);
}
}

```

Figura 5.4. Algoritmo de correspondência Δ^+ -Connection.

A primeira etapa do algoritmo verifica se a opção escolhida é diferente do Δ -Connection padrão, para que não seja necessário calcular a variação de tamanhos com nesta opção. Isto foi inserido para facilitar a parametrização do usuário que, do contrário deveria inserir valor zero no parâmetro R , e também para evitar processamento desnecessário. Caso seja outra opção, faz-se o cálculo da variação, conforme Eq.4.2. Posteriormente é verificado se este novo valor de Δ é maior ou igual à distância percentual (já convertida para o seu valor decimal) inserida pelo usuário, caso for, os contornos se conectarão, do contrário não.

Sendo assim, a arquitetura final do aplicativo ficou com a configuração UML do diagrama de classes conforme Figura 5.5.

Após as alterações efetuadas o algoritmo mantiveram-se todas as características e requisitos conforme anteriormente o Δ -Connection:

- Arquivo de representação das curvas em XML;
- Algoritmo parâmetro de controle de flexibilidade;
- Algoritmo simétrico, ou seja, a reconstrução de cima para baixo é igual à reconstrução de baixo para cima;
- A decisão do local de bifurcação não é tratada, sendo consequência da geração de malha;

- Aplicação preferencial na visualização de canais.

5.3 Implementação das Métricas

Para obter os dados das métricas propostas na seção 3.6, foi necessário calcular a quantidade final de *spans* (arestas que conectam contornos em planos adjacentes) do objeto e o tamanho de cada um. Para tal, foi implementada uma classe denominada `CalculoMetrica.java` com dois principais métodos:

- `metricaSpan`: é invocado a cada par de planos interpolados e através da topologia e geometria recebida por parâmetro, calcula o tamanho de cada *span* no \mathbb{R}^3 ;
- `adicionarSpan`: método utilizado internamente na ação anterior para adicionar em um arquivo de saída cada tamanho do *span* calculado e a quantidade total de *spans* gerados.

Também foi calculada a quantidade total de *spans* e segmentos de contornos (arestas de um contorno no mesmo plano), bem como seus comprimentos através dos métodos:

- `metricaSpanSegmento`: calcula o perímetro de cada face gerada na malha;
- `adicionarSpanSegmento`: adiciona o valor do perímetro calculado no item acima e adiciona em um arquivo de saída. Também realiza a contagem da quantidade de *spans* e segmentos de contorno.

De posse da população total de valores para cada técnica/objeto, foi montada uma planilha eletrônica com o software OpenOffice SpreadSheet 3.0.1, que extraia as métricas anteriormente descritas na seção 2.9, conforme suas fórmulas nativas:

- AVERAGE – média;
- MEDIAN – mediana;
- MIN – valor mínimo;
- MAX – valor máximo;
- STDEVPA – desvio padrão da população total.

A métrica de tempo foi calculada em cada etapa e suas respectivas sub-etapa do processo de reconstrução na classe `DeltaConnection.java`. Ao final do processo, tanto os tempos parciais quanto o total são informados ao usuário.

Unindo todos estes dados, obteve-se as tabelas do apêndice B.2, B.3, B.4, B.5 e B.6 a fim de facilitar a análise e comparação dos mesmos.

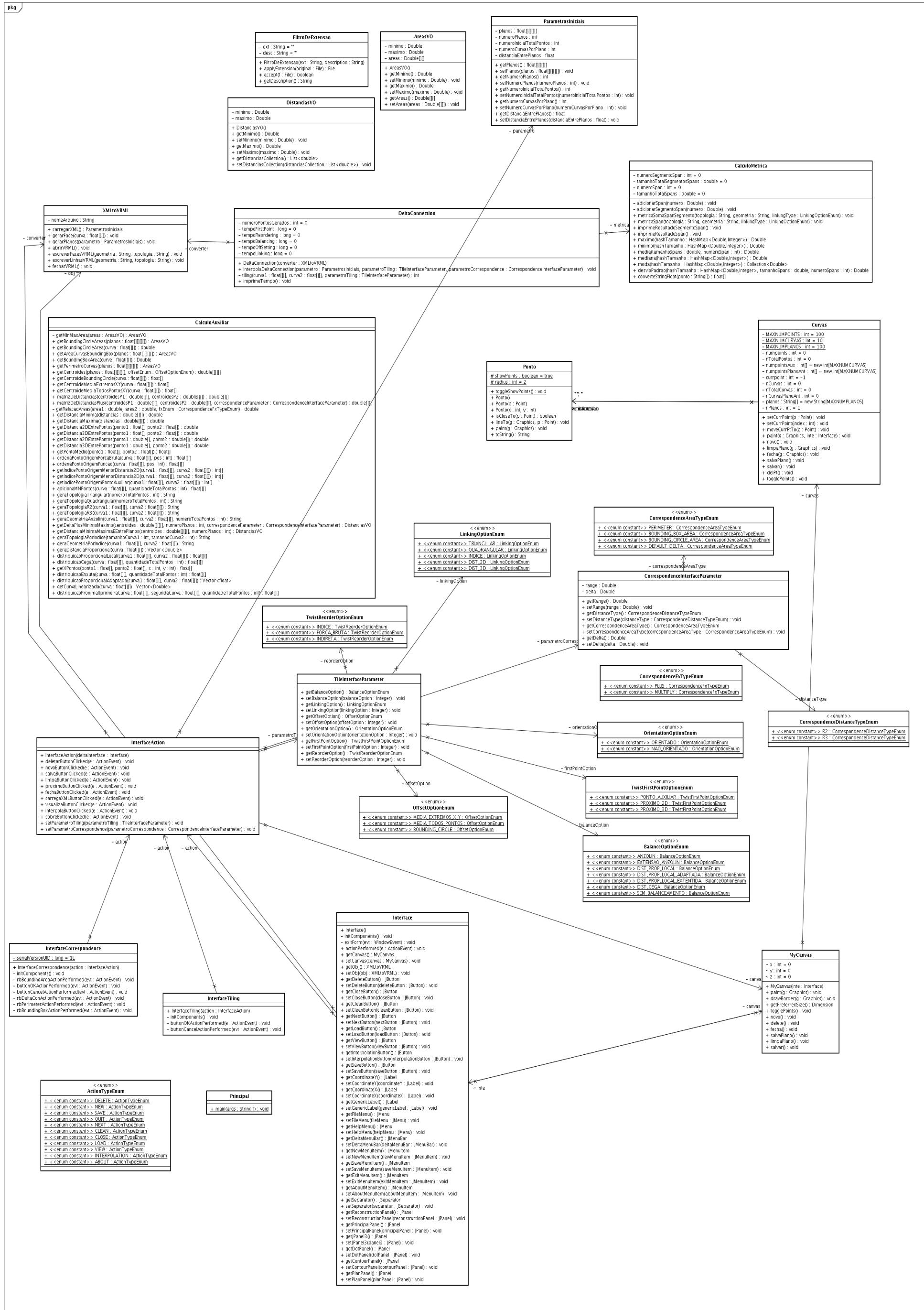


Figura 5.5. Arquitetura UML do Δ^+ -Connection.

6 Resultados

Neste capítulo são apresentados os resultados obtidos com a implementação do Δ^+ -Connection em suas etapas de geração de malha e correspondência, bem como um comparativo com as soluções existentes. Os dados utilizados como entrada do algoritmo foram cedidos por autores da área. Tratam-se de dados reais relativos às partes do corpo humano obtidos por meio de aparelhos de medição.

6.1 Dados da Geração de Malha

Nesta seção serão apresentados os dados relativos à geração de malha.

A) Geração Mais Eficiente

Para se ter uma referência para comparação visual, foram gerados objetos para cada solução da literatura implementada no Δ^+ -Connection, que no caso inclui Anzolin, Hounsell e Silva (2008), Chen et al (2007) e Li, Ma e Tan (2006). Estes serão doravante identificados por AHS, CCCH e LT, respectivamente por questões de economia de espaço. Também foram realizadas inúmeras reconstruções com diferentes combinações de parametrização da geração da malha a fim de encontrar uma boa solução visual. Para cada objeto, foram extraídos os valores das métricas, conforme já descrito na seção anterior. Estes dados foram gerados em um computador com processador AMD Turion X2 2.0 Mhz, com 4GB de memória RAM e sistema operacional Linux Ubuntu 9.04 32bits.

A seguir será realizada a comparação entre cada par de objetos de melhor qualidade visual, sendo um o melhor resultado de parametrização do Δ^+ -Connection (Delta4+) e outro a melhor solução visual dos algoritmos existentes. As características (número de pontos, planos e contornos),

os objetos gerados com as diferentes técnicas e os valores de suas métricas estão disponíveis no apêndice B. A tabela B.1 descreve a parametrização completa utilizada para cada algoritmo.

A solução Delta4+ foi obtida através de testes de combinações partindo da configuração de AHS, inicialmente alterando as técnicas de balanceamento, o que gerou alternativas denominadas Delta1+. A melhor alternativa visual dentre as Delta1+ foram realizadas variações com as técnicas de primeiro ponto (gerando as Delta2+), depois consideradas todas as opções de conexão dos pontos (gerando as Delta3+) e por último variações das técnicas de equivalência, obtendo-se então uma combinação de melhor aspecto visual (Delta4+).

B) Aplicação das Métricas

No primeiro exemplo da reconstrução de uma veia a partir de um conjunto de contornos com 1285 pontos distribuídos entre 65 curvas em 19 planos, conforme a Figura 6.1, identificou-se que AHS (a) e (b) Delta4+ apresentaram as melhores soluções visuais.

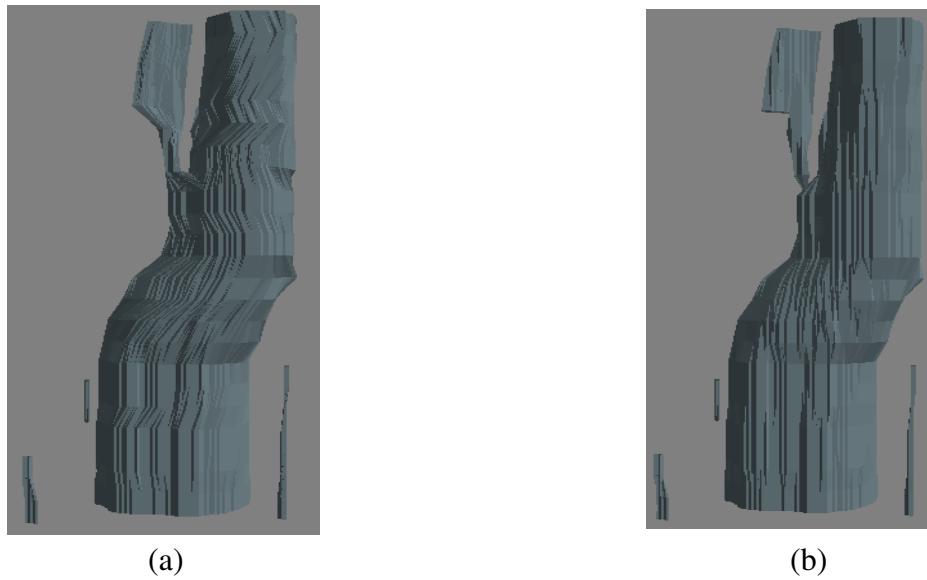


Figura 6.1. Comparativo visual da veia.

Observando os detalhes de cada imagem, pode-se notar que a solução Delta4+ apresenta uma malha mais suave, ou seja, com menos torções. Estas análises visuais foram efetuadas com modelos sendo renderizados da forma facetada (*flat shading*), pois se fossem visualizadas com renderizadores suavizados (*gouraud shading* ou *phong shading*) o resultado visual seria mascarado pela técnica de suavização visual. Comparando as métricas dos dois algoritmos, conforme Tabela 6.1, observa-se que apesar de gerarem quase o mesmo número de pontos finais, AHS o fez em um menor tempo, mas Delta4+ obteve a menor média e desvio padrão, ou seja, seus *spans* são menores

e variam menos em função da mediana. Porém, Delta4+ apresentou uma falha no canal do canto esquerdo superior.

Os itens marcados em negrito indicam o menor valor na métrica entre todos os algoritmos.

Algoritmo	Tempo (ms)	No. Pontos Iniciais	No. de Pontos Finais	No. Spans	Spans				
					Média	Mediana	Mínimo	Máximo	Desvio Padrão
AHS	870	1285	6691	21624	10.991992	10.198039	10.000000	45.642805	2.986153
LMT	598	1285	1545	5666	11.536869	10.198039	10.000000	48.741784	4.332388
CCCH	455	1285	1285	10812	12.029230	10.488088	10.000000	51.429563	3.758807
Delta4+	1508	1285	5731	19704	10.259139	10.049876	10.000000	18.867962	0.659701

Tabela 6.1. Resultado das métricas da veia.

Adicionalmente foram gerados dados considerando, além dos *spans*, os tamanhos dos segmentos dos contornos, ou seja, o perímetro de cada face. Analisando esses dados, obteve-se a mesma configuração de valores mínimos que quando somente considerado *span*, conforme pode ser comprovado na Tabela 6.2, que demonstra as métricas incluindo tanto tamanho de segmentos quanto de *spans* do objeto veia. Essa mesma sequência de valores mínimos se dá, pois o tamanho dos contornos não varia, ou seja, seus segmentos podem variar, mas o tamanho total (somatório de todos segmentos) sempre será o mesmo. Conforme ilustra a Figura 6.2, apesar de variar o número de *spans* e consequentemente o número de segmentos de contorno (que basicamente identifica o perímetro do contorno) tem-se invariavelmente o mesmo valor para ambos os casos.



Figura 6.2. Variação de *spans* em contornos de tamanho constante.

Sendo assim, desconsiderou-se utilizar esta forma para análise e manteve-se somente *span*, pois além de resultar na mesma avaliação, significa um processamento inferior e mais sensível às variações das técnicas de geração de malha.

Algoritmo	Tempo (ms)	No. Pontos Iniciais	No. de Pontos Finais	No. Spans e Segmentos	Veia (BloodVein)				
					Spans e Segmentos				
					Média	Mediana	Mínimo	Máximo	Desvio Padrão
AHS	870	1285	6691	32436	22.641663	21.256960	20.010153	93.476464	5.996490
LMT	598	1285	1545	11332	23.013941	22.828427	10.163136	101.998324	9.642954
CCCH	455	1285	1285	16218	24.294483	22.669299	11.000000	106.927088	8.653944
Delta4+	1508	1285	5731	29556	20.272964	20.755142	10.000031	44.867962	3.366304

Tabela 6.2. Resultado das métricas da veia.

No segundo exemplo, ilustrado na Figura 6.3, apresenta-se a reconstrução de um fêmur a partir de um conjunto de contornos com 1285 pontos distribuídos entre 65 curvas em 30 planos. Em (a) é apresentado o resultado do algoritmo de AHS e (b) a combinação denominada Delta4+.

Da mesma forma que o objeto anterior, estes dois algoritmos apresentaram melhor qualidade visual e também mesma sequência de valores mínimos na tabela de métrica, conforme pode-se ver em negrito na Tabela 6.3. Foram os dois objetos que apresentaram menor média e menor desvio padrão. Ou seja, novamente os melhores resultados qualitativos foram para Delta4+, AHS, LMT e CCCH (nesta ordem).

No terceiro exemplo, conforme a Figura 6.4, a reconstrução de um pulmão a partir de um conjunto de contornos com 1285 pontos distribuídos entre 65 curvas em 19 planos, tem-se novamente os mesmos dois algoritmos como melhor qualidade visual e também a mesma sequência de métricas conforme Tabela 6.4. A imagem (a) é resultado do algoritmo de AHS e (b) a combinação denominada Delta4+.

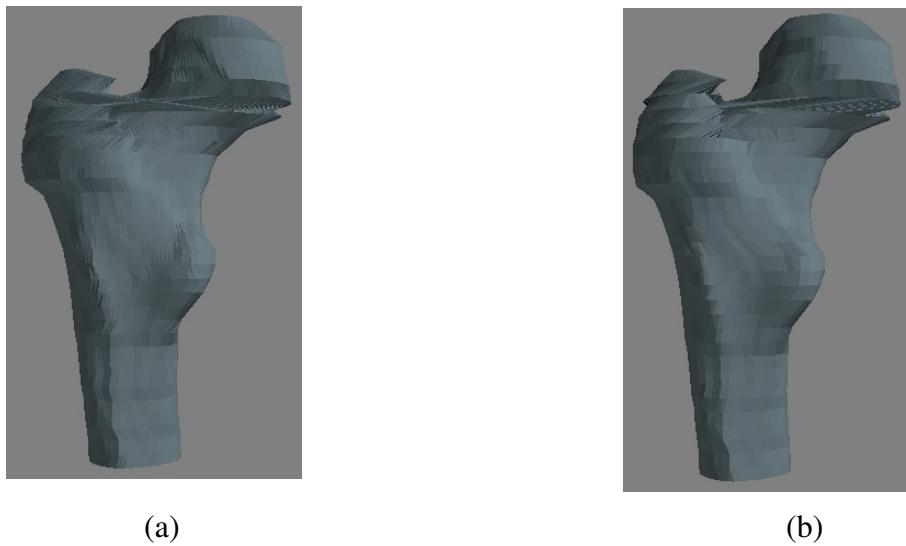


Figura 6.3. Comparativo visual do Fêmur.

Algoritmo	Tempo (ms)	No. Pontos Iniciais	No. de Pontos Finais	No. Spans	Spans				
					Média	Mediana	Mínimo	Máximo	Desvio Padrão
AHS	496	1285	4666	13524	0.160734	0.109915	0.100000	1.335871	0.155264
LMT	239	1285	1492	3588	0.178202	0.109736	0.100000	1.547527	0.205474
CCCH	238	1285	1285	6226	0.189203	0.124599	0.100000	1.197815	0.158996
Delta4+	864	1285	4594	13380	0.134915	0.104518	0.100000	1.409906	0.132335

Tabela 6.3. Resultado das métricas do fêmur.

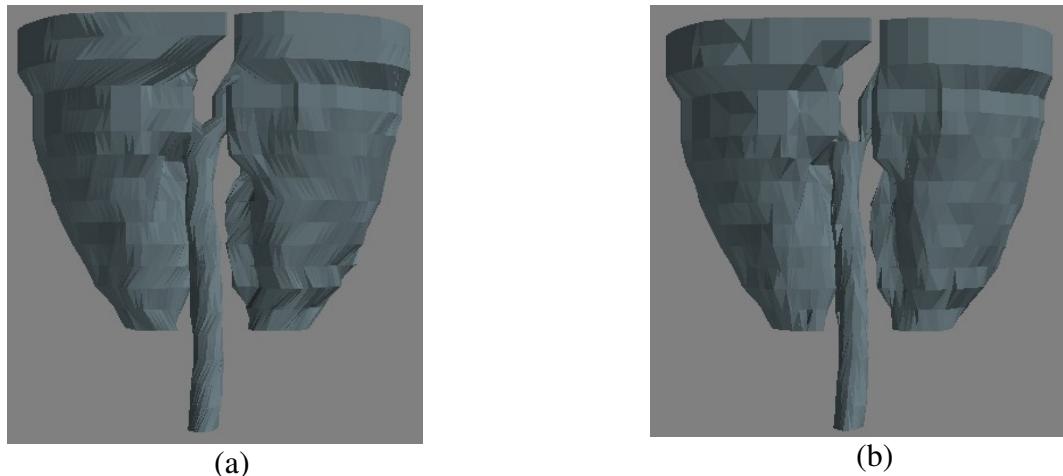


Figura 6.4. Comparativo visual do Pulmão.

Algoritmo	Tempo (ms)	No. Pontos Iniciais	No. de Pontos Finais	No. Spans	Spans				
					Média	Mediana	Mínimo	Máximo	Desvio Padrão
AHS	242	1285	3936	10604	120.476414	110.797112	100.000000	318.632233	27.714761
LMT	162	1285	1474	2840	129.822049	111.266060	100.000000	432.152751	45.302336
CCCH	87	1285	1285	5302	182.402979	142.305877	100.000000	810.977188	108.034076
Delta4+	473	1285	3704	10140	110.860347	105.147308	100.000000	394.405882	18.900305

Tabela 6.4. Resultado das métricas do pulmão.

Novamente a mesma sequência ocorre no coração como ilustrado a Figura 6.5 e Tabela 6.5, que apresenta a reconstrução de um coração a partir de um conjunto de contornos com 1285 pontos distribuídos entre 65 curvas em 30 planos. O fato do coração apresentar problemas visuais será descrito no próximo capítulo, pois trata-se de correspondência e não de geração de malha.

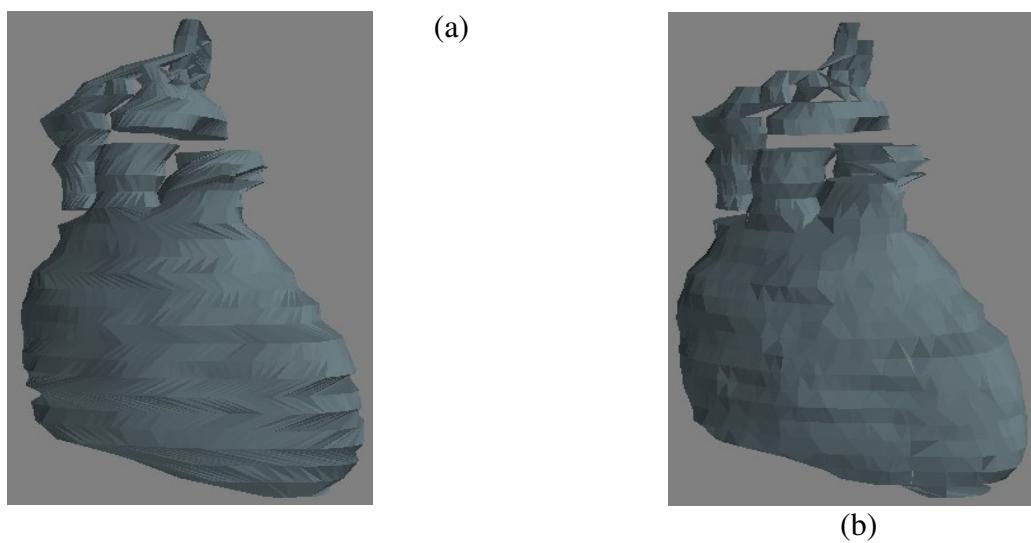


Figura 6.5. Comparativo visual do coração.

Algoritmo	Tempo (ms)	No. Pontos Iniciais	No. de Pontos Finais	No. Spans	Spans				
					Média	Mediana	Mínimo	Máximo	Desvio Padrão
AHS	241	1285	3962	10708	0.191834	0.144522	0.100000	0.698353	0.107979
LMT	78	1285	1534	2926	0.214455	0.150147	0.100000	1.304034	0.171052
CCCH	77	1285	1285	5354	0.223874	0.167227	0.100000	1.135772	0.145252
Delta4+	368	1285	3820	10424	0.150993	0.115729	0.100000	0.701374	0.089234

Tabela 6.5. Resultado das métricas do coração.

Para definir a qualidade visual dos objetos, utilizou-se uma população de cinco pessoas que ordenaram os resultados das técnicas de cada objeto (Apêndice B.7) qualitativamente conforme sua avaliação. O critério de avaliação foi a torção da malha e, todos ordenaram conforme detalhado nos casos acima.

Sendo assim, pode-se perceber que em todos os casos foram as mesmas técnicas que apresentaram os melhores resultados. Baseando-se nisso e analisando as tabelas de métricas percebe-se que esses foram os que tiveram menores médias e desvios padrões, ou seja, seus *spans* foram os menores e variaram pouco em relação à mediana.

C) Expectativa do Custo de Processamento

Na Tabela 4.1 da seção 4.1 foram apontadas as expectativas dos algoritmos com menor e maior custo de processamento, ou seja, tempo. Como dentre os algoritmos utilizados para análise nem todas opções são utilizadas, sendo que ainda alguns não possuem alguma(s) sub-etapa(s), não pôde-se confirmar todas as expectativas. Porém, estruturou-se a Tabela 6.6 comparando os tempos de cada sub-etapa dos algoritmos de CCCH, AHS e Delta4+, que utilizam duas, três e quatro sub-etapas, respectivamente, da geração da malha, além de AHS e Delta4+ terem apresentado os melhores resultados visuais e CCCH consistentemente o menor tempo geral. Os objetos utilizados foram o fêmur e o pulmão, já que o primeiro possui o menor número de planos, 19 e o segundo o maior, 34, dentre os objetos utilizados.

Técnica/Sub-etapa	Fêmur					Pulmão				
	Equiv.	Orien.	Torção	Balanc.	Conex.	Equiv.	Orien.	Torção	Balanc.	Conex.
CCCH(ms)	0	0	98	0	140	0	0	30	0	57
	0.00%	0.00%	41.18%	0.00%	58.82%	0.00%	0.00%	34.48%	0.00%	65.52%
AHS (ms)	0	0	20	243	233	0	0	15	100	127
	0.00%	0.00%	4.03%	48.99%	46.98%	0.00%	0.00%	6.20%	41.32%	26.85%
Delta4+ (ms)	3	0	97	322	442	6	0	42	182	243
	0.35%	0.00%	11.23%	37.27%	51.16%	1.27%	0.00%	8.88%	38.48%	51.37%

Tabela 6.6. Comparativo de tempos nas sub-etapas de geração de malha.

Dentre estes três algoritmos pode-se afirmar que a sub-etapa de torção de AHS teve um tempo menor do que CCCH e Delta4+, da mesma forma que o balanceamento foi o menor em Delta4+ e a conexão em AHS, ou seja, em relação às expectativas explicitadas anteriormente, estas estão coerentes. Pode-se notar que a configuração de sub-etapas menos custosas se mantém uniformes para os dois objetos, ou seja, independente do número de planos essa expectativa é a mesma. Porém, não é possível confirmar que estas foram as menos custosas, devido não serem englobadas todas as possíveis soluções para análise.

Conclui-se observando que, a medida que se utilizam algoritmos que tratam separadamente cada sub-etapa com heurísticas específicas, o resultado qualitativo é melhor apesar de que o quantitativo seja menor. Como pode-se comprovar, o algoritmo de CCCH foi o menos custoso em tempo com todos os objetos, porém sua qualidade visual foi consideravelmente inferior em relação aos outros algoritmos.

6.2 Dados de Correspondência

Inicialmente foram gerados dados utilizando a solução proposta na seção 4.2.2, tendo como objetivo melhorar a correspondência proposta por AHS utilizando o coração como caso de teste que foi o caso que apresentou graves problemas de conexão. Os testes iniciais foram feitos com a estratégia da Eq. 4.3. Porém, além de não se encontrar uma parametrização com interpolação final aceitável para este objeto, a implementação afetou os resultados visuais dos outros objetos utilizados deformando-os.

Desta forma, foi desenvolvida e utilizada a solução proposta na seção 4.2.3 que, ao contrário da anterior, gerou objetos semelhantes aos anteriormente existentes, além de mostrar avanços na reconstrução do coração, que será descrito nesta seção.

A Figura 6.6 apresenta o resultado final da reconstrução da veia (a), fêmur (b) e pulmão (b), bem como as parametrizações de correspondência (Δ e R) utilizados em cada um. É possível parametrizar de forma que se tenha o Δ -Connection original (Δ , $R=0$, 2D), porém nestes objetos foram utilizadas parametrizações com o novo algoritmo. Apesar de utilizar-se de uma nova técnica, o resultado visual foi muito semelhante ao existente.

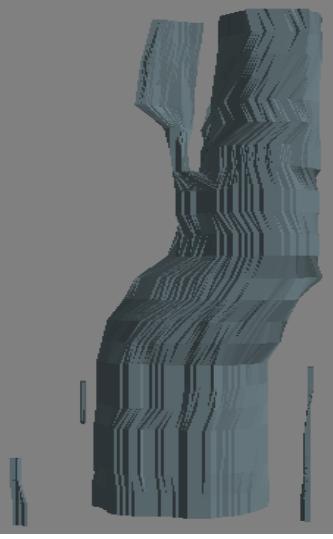
		 (c)
Correspondência: <i>B. Circle</i> Distância: 3D Δ : 5 R : 1000 Geração de Malha: Anzolin	Correspondência: <i>B. Box</i> Distância: 2D Δ : 3 R : 400 Geração de Malha: Anzolin	Correspondência: <i>B. Box</i> Distância: 2D Δ : 2 R : 600 Geração de Malha: Anzolin

Figura 6.6. Objetos gerados com a nova técnica de correspondência.

Para a reconstrução do coração foram estruturados itens sequenciais de testes, a fim de ter um guia para a solução através da análise de cada um. Sendo assim, definiu-se:

1. Encontrar o valor percentual de Δ equivalente ao valor decimal da solução de AHS;
2. Testar d3D como parâmetro distância entre os centróides, ao invés de d2D;
3. Encontrar uma correção adaptativa de tamanho com d2D, ou seja, valores para Δ e r que reconstruissem o coração;
4. Encontrar uma correção adaptativa de tamanho com d3D, ou seja, valores para Δ e r que reconstruissem o coração.

No item 1, foram utilizados os valores mínimo e máximo calculados pelo algoritmo do Δ -Connection, além do parâmetro de entrada inserido pelo usuário para compor a regra de proporcionalidade. Sendo $\Delta_{\text{mínimo}} = 0$, $\Delta_{\text{máximo}} = 1.925$ e $\Delta = 0.1$, fez-se:

$$\Delta \text{ percent} = \frac{\Delta}{\Delta \text{ máximo}} * 100$$

obtendo como resultado $\Delta \text{ percent} = 5.2\%$.

Com este valor percentual, o algoritmo resultou no objeto conforme Figura 6.7 (a), onde teve o mesmo resultado do Δ -Connection, ou seja, comprovando a semelhança apesar de utilizar valor percentual.

Com o item 2, objetivava-se obter alguma melhora no objeto considerando também a distância dos contornos entre os planos, ou seja, a distância real dos centróides dos contornos adjacentes no espaço R^3 e não a distância em um plano comum projetado. Conforme pode-se observar na Figura 6.7 (a), não obteve-se o resultado esperado, gerando um objeto semelhante quando utilizado o Δ -Connection. Sendo assim, percebeu-se que seria necessário inserir o parâmetro de folga do tamanho dos contornos, conforme previsto no item 3.

Então no item 3, utilizou-se a parametrização da folga de relação dos tamanhos dos contornos. Várias tentativas foram realizadas, ao passo que os resultados visuais foram melhorando à medida que diminuia-se o percentual do Δ e aumentava-se o da relação dos tamanhos (R), de forma a obter a parametrização e objeto final conforme Figura 6.7 (b). Comparando os objetos (a) e (b), pode-se notar que em (b) houve a conexão dos contornos no meio do objeto e nos contornos menores (parte superior) não ocorreu o entrelaçamento da malha (visível na parte superior da Figura 6.7 (a)), de forma que resultou em um objeto mais próximo à realidade.

No item 4, apenas considerou-se novamente a distância entre planos, ou seja, d3D entre os centróides, porém não houve nenhuma melhoria visual, conforme pode ser visualizado na Figura 6.7 (c), equivalendo ao objeto gerado no item 3, mas com maior custo computacional por estar considerado uma dimensão a mais.

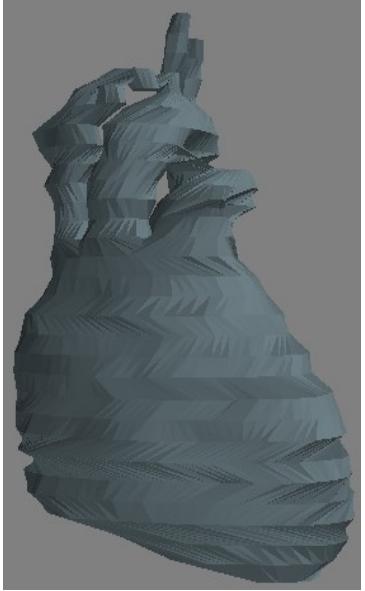
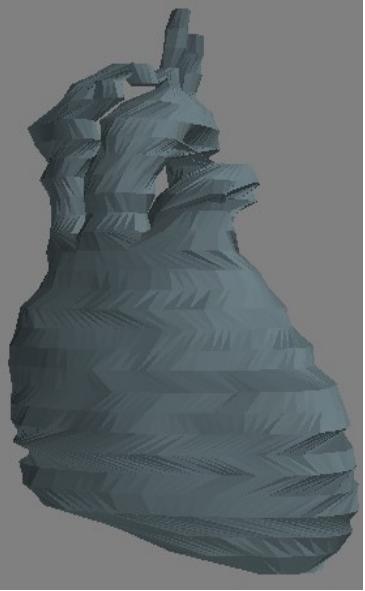
		
(a)	(b)	(c)
Correspondência: Default Distância: 3D $\Delta\%$: 5.2 R : Nenhum Geração de Malha: AHS	Correspondência: <i>B. Box</i> Distância: 2D $\Delta\%$: 2 R : 1600 Geração de Malha: AHS	Correspondência: <i>B. Box</i> Distância: 3D $\Delta\%$: 2 R : 1600 Geração de Malha: AHS

Figura 6.7. Reconstrução do coração com diferentes parametrizações.

Para uma análise mais detalhada das melhores situações obtidas até então, montou-se a Tabela 6.7 para comparação do objeto coração gerado pelo Δ -Connection e Δ^+ -Connection, em duas visões diferentes, no plano +Z e -Z.

Comparando os objetos da Tabela 6.7 no plano +Z, pode-se notar que da parte inferior até o início das bifurcações não há diferença visual, onde aí sim, no Δ^+ -Connection os contornos medianos se conectaram e no Δ -Connection não. Já nos canais menores da parte superior, com o Δ -Connection os contornos estão se conectando, o que faz com que haja entrelaçamento e distorção da malha do objeto. No Δ^+ -Connection há uma conexão uniforme destes canais superiores melhorando o aspecto visual do objeto.

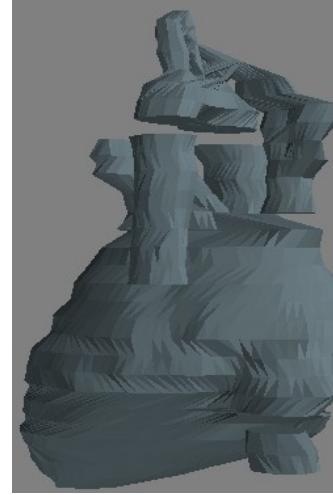
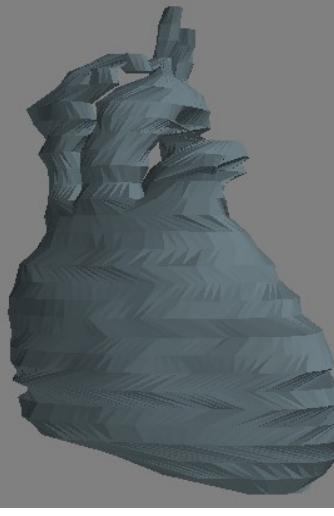
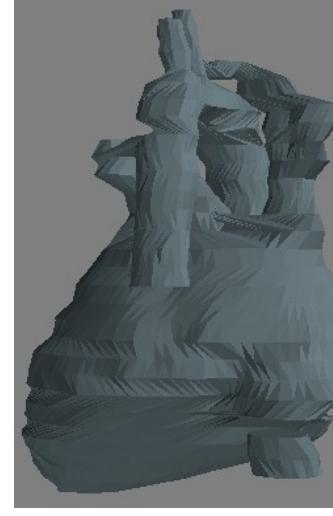
Algoritmo	Plano +Z	Plano -Z	Parâmetros
Δ (Original)			Corresp.: <i>Default</i> Distância: 2D $\Delta\%$: 5 R : 0 Ger. Malha: AHS
Δ^+ (Plus)			Corresp.: <i>B. Box</i> Distância: 2D $\Delta\%$: 2 R : 1600 Ger. Malha: AHS

Tabela 6.7. Comparação do coração entre Δ -Connection e Δ^+ -Connection.

Com relação ao plano -Z, o objeto do Δ -Connection original apresentou os mesmos problemas do plano +Z, onde não conectou os contornos medianos e conectou os menores (entrelaçamento). Da mesma forma Δ^+ -Connection apresentou melhorias na conexão, excetuando a bifurcação na parte inferior, onde não houve a conexão.

Analizando os parâmetros utilizados no Δ^+ -Connection ($\Delta=2$ e $R=1600$) pode-se entender que neste caso, como houve diminuição do valor do Δ , teve-se menor rigor para o alinhamento inicial, porém houve uma significativa variação (folga) em função da dos tamanho dos contornos. O fato de R ter um valor percentual grande pode ser justificado pela relação das áreas não indicarem uma heurística com um significado palpável, ou seja, provavelmente deve-se ter uma relação entre áreas mais adequada que permita um valor de R menor (preferivelmente entre zero e cem porcento)

com um significado mais explícito.

6.3 Discussão

As tabelas com o resultado das métricas dos objetos no apêndice B apresentam em negrito os menores valores de cada métrica dentre os algoritmos aplicados na reconstrução. É relevante ressaltar que a aplicação das métricas foi inserida em um momento diferente do cálculo do tempo de reconstrução, pois como demandam processamento, iriam influenciar no resultado final.

Analizando cada objeto no Apêndice B pôde-se perceber que todos apresentaram a mesma sequência de qualidade visual (métrica qualitativa), sendo sempre Delta+4, Delta+1, AHS, LMT e CCH, respectivamente. Ao analisar as tabelas referente à cada objeto e técnica (B2, B3, B4 e B5), pôde-se perceber que a sequência da qualidade visual foi exatamente a mesma das menores médias dos *spans*, ou seja, quanto menor a média com menores variações (desvio padrão), melhor a qualidade visual do objeto.

Sendo assim, constatou-se empiricamente que os objetos com menores médias dos *spans* apresentaram um resultado visual de maior qualidade. Isto porque quanto menores forem os *spans* (média) e menor for a sua variação em relação à mediana (desvio padrão), o objeto terá menos torção, apresentando um melhor aspecto visual, como pode ser observado nas imagens. Então se estabelece uma relação entre uma métrica quantitativa e uma métrica qualitativa, ou seja, quanto menor a média e o desvio padrão, mais qualidade visual terá o objeto gerado.

A métrica de tempo também é significativa nesta análise, pois todos os modelos utilizados tem o mesmo número de pontos nos arquivos que descrevem os contornos, mas com diferentes quantidades de planos e contornos. Portanto, partem da mesma magnitude de dados apenas variando sua disposição. Porém, não pode ser considerada isoladamente, mas sim como parte das métricas já que as técnicas com menores tempos apresentaram os piores resultados visuais.

Com relação à correspondência, pôde-se constatar que a geometria (área ou perímetro) do contorno foi uma característica relevante a ser considerada visando uma solução eficaz, pois traz uma relação mais estreita entre os contornos, principalmente entre os de tamanhos consideravelmente diferentes. Utilizando esta informação adicional, pôde-se gerar o objeto do coração mais próximo à realidade do que havia anteriormente, onde contornos que não se conectavam, mas deveriam, efetivamente conectaram-se e vice-versa.

De qualquer forma, pode haver uma relação ainda mais eficaz do que a proposta, pois esta não gerou uma heurística com significado palpável, o que pode justificar o fato do parâmetro R ter resultado em um valor alto e ainda ocorreu falta de conexão de alguns contornos (que deveriam se conectar).

6.4 Site

Todos os dados referentes aos resultados obtidos, bem como a aplicação desenvolvida e o código fonte de algoritmo estarão disponíveis em um *site*, semelhante ao ilustrado pela Figura 6.8, que foi desenvolvido durante o trabalho e estará disponível no site do grupo LARVA.



Δ⁺ - Connection

Reconstrução 3D de Objetos com Seções Planares Variáveis

GPLV Group of Research in Virtual Reconstruction

Projeto: Neste trabalho, um conjunto de soluções algorítmicas da etapa de geração de malha e correspondência da reconstrução tridimensional são apresentados. Para resolver este problema com aplicação na área médica e na representação de terrenos, conceitos e técnicas da modelagem geométrica foram estudadas. Uma reestruturação da arquitetura e otimizações algorítmicas da aplicação de reconstrução tridimensional Δ-Connection são apresentadas e foram implementadas.

Resultados:
Para visualizar os resultados (wrl) é necessário instalar um [plugin](#) e selecionar a opção "fit" do browser que será instalado.

Descrição dos Objetos				
Veia (Blood Velse)	Coração (Heart)	Fêmur (Hip)	Pulmão (Lung)	
xml	xml	xml	xml	

Objetos Reconstruídos				
Algoritmo/Objeto	Veia (Blood Velse)	Coração (Heart)	Fêmur (Hip)	Pulmão (Lung)
Anzolin	wrl	wrl	wrl	wrl
Chen	wrl	wrl	wrl	wrl
Li, Ma, Tan	wrl	wrl	wrl	wrl
Delta4+	wrl	wrl	wrl	wrl

Downloads:

Monografia	Implementação	Código Fonte (SVN)
----------------------------	-------------------------------	------------------------------------

Copyright © Universidade do Estado de Santa Catarina - Todos os direitos reservados à Bittencourt & Hounsell (2009).
Última atualização em: 30/10/2009.


Figura 6.8. Site da pesquisa.

7 Conclusão

A estratégia de solução heurística para a reconstrução tridimensional enfatiza a divisão do problema em três grande etapas que foram apresentadas e discutidas neste trabalho. Esta divisão possibilita a criação de várias soluções para cada etapa, sendo que algumas delas foram detalhadas neste texto. Esta mesma ideia, a tentativa de divisão do problema em etapas menores, foi então aplicada às etapas de geração de malha e correspondência. Também definiu métricas de avaliação dos objetos gerados, a fim de identificar uma relação qualitativa-quantitativa.

Com relação à geração de malha a pesquisa demonstrou que todas as soluções identificadas podem ser descritas como uma combinação de etapas menores (as quais foram devidamente identificadas ao longo do texto) em um total de cinco: alinhamento, orientação e torção dos contornos, balanceamento e conexão dos pontos. Os algoritmos existentes se encaixaram neste modelo genérico proposto sem sofrer alterações no seu resultado final. Além disso, pôde-se experimentar combinações diferentes, de forma a obter soluções alternativas para melhor atender às métricas. Pôde-se comprovar também que o uso do maior número de heurísticas específicas para cada sub-etapa leva a um resultado qualitativo melhor, daí a importância de se ter conseguido identificar esta subdivisão.

Os testes indicaram que com a média dos *spans* é possível determinar a qualidade visual dos objetos reconstruídos, sendo assim, estabeleceu-se uma métrica que demonstrou boa correlação entre as análises quantitativas e qualitativas. Os testes foram realizados com dados gerados pelo editor de contornos e também com dados reais relativos às imagens médicas cedidas por autores da área. Entretanto, esses testes precisam ser aplicados de forma sistemática e exaustiva em um maior número de objetos, a fim de se confirmar de maneira mais confiável esta métrica.

No que diz respeito à correspondência, foi identificado que todas as soluções, de forma implícita ou explícita, avaliam a distância entre os planos adjacentes, desconsiderando um indicativo de tamanho relativos aos contornos, porém concluiu-se que estes tamanhos são características relevantes para soluções eficazes. Na etapa da correspondência não foi possível subdividir em processos menores, porém não impediu que uma nova proposta pudesse ser concebida e que a

mesma fosse mais eficaz do que as existentes. Sendo assim, definiu-se uma nova heurística baseada do alinhamento dos centróides, adaptativo ao tamanho dos contornos que trouxe melhorias visuais ao caso de teste utilizado, o coração, de forma a resultar em um objeto mais próximo à realidade dos que haviam anteriormente. Esta nova heurística, além de melhorar o modelo do coração que anteriormente apresentava problemas de correspondência, não comprometeu o resultado visual dos outros modelos.

O aplicativo Δ^+ -Connection além de incorporar novas soluções para geração da malha e correspondência, serve como uma plataforma de testes que permite:

- Experimentar combinações diferentes das soluções existentes (em um total de 1440 para geração de malha e 8 para correspondência);
- Implementar novas heurísticas (independentes) para cada sub-etapa, possibilitada pela atual modularização;
- Comparar as novas soluções propostas com as existentes em um mesmo ambiente.

Sendo assim, implementou-se e avaliou-se o desempenho de um conjunto de alternativas para o cálculo de geração de malha e correspondência, estabelecendo métricas para avaliar automaticamente a qualidade do modelo gerado. Também obteve-se uma solução diferente e melhor que as existentes, para o caso onde haja uma variação muito grande dos tamanhos dos perfis no modelo, como o objeto do coração. Os resultados apresentados permitiram a validação da proposta mostrando que o presente trabalho alcançou o objetivo de melhorar o Δ -Connection original, por isso, a solução foi denominada Δ^+ -Connection.

7.1 Trabalhos Futuros

A seguir estão algumas sugestões de trabalhos futuros para acrescentar ao algoritmo de interpolação:

- No atual algoritmo algumas técnicas de balanceamento adicionam novos pontos nos pares de contornos que se conectarão. Porém, nos casos onde haja bifurcação, como a conexão ocorre aos pares, os pontos criados em cada n conexões. Então pode-se fornecer a opção para que sejam gerados, nos contornos com correspondência 1-m ou m-n, apenas a diferença de pontos com o contorno de maior número de pontos. Ou seja, quando conectar ambos C1 e C2 com C3, por exemplo, apenas gerar a diferença de pontos de cada um com o Max(C1,

C_2, C_3), resultando em três contornos de mesmo número de pontos ($\text{Max}(C_1, C_2, C_3)$). Desta forma, diminui-se o número de pontos gerados e consequentemente o número de faces e tempo, também deixa a malha mais uniforme para a curva base, unificando a forma de sua subdivisão para os n contornos conectados;

- Definir, implementar e disponibilizar outras relações entre o tamanho e/ou formato dos contornos para a correspondência, de forma que se tenham heurísticas palpáveis e ainda mais eficientes.
- Pode-se desenvolver um conversor de modelos do formato “vtk” para a estrutura XML proposta por AHS, devido o fato de encontrar-se mais facilmente diferentes modelos neste formato (“vtk”). Desta forma, permite-se ampliar os testes para aumentar a confiabilidade desejada.

Referências

- AGOSTON, M. K. Computer Graphics and Geometric Modeling: Implementation & Algorithms . Springer. Cupertino, CA, USA, 2005.
- ANZOLIN, G. R., HOUNSELL, M. S., SILVA, A. G. Delta-Connection: A Solution for 3D Object Reconstruction. **INFOCOMP Journal of Computer Science**, 7(2-Junho):65-73. ISSN: 1807-4545. 2008.
- ANZOLIN, G. R. Interpolação de Curvas com Mudança Topológica. Trabalho de conclusão do curso de Bacharelado em Ciência da Computação. Universidade do Estado de Santa Catarina. 2006.
- BARREQUET, G., SHAPIRO, D., TAL, A. **Multilevel sensitive reconstruction polyhedral surfaces from parallel sections**, The Visual Computer, 16 (2000), 116--133.
- BAREQUET, G.; SHARIR, M. **Piecewise-linear interpolation between polygonal slices**. Computer Methods in Applied Mechanics and Engineering, v. 63, n. 2, p. 251–272, 1996.
- BARREQUET, G. VAXMAN, A. **Nonlinear Interpolation between Slices**. Association for Computing Machinery. Junho, 2007.
- BAJAJ, C. L. COYLE, E. J.; LIN, K.-N. **Surface and 3D Triangular Meshes from Planar Cross Sections**. Department of Computer Science, Purdue University 1996.
- CHEN, X., SHEN, J. **A Tetrahedral Mesh Generation Algorithm from Medical Images**. International Symposium on Information Science and Engineering. 2008.
- CHEN, Y., CHEN, Y., CHIANG, A. e HSIEH, K. **A reliable surface reconstruction system in biomedicine**. Elsevier Ireland Ltd <www.intl.elsevierhealth.com/journals/cmpb>. Taiwan. Janeiro, 2007.
- CONG, G.; PARVIN, B. **Robust and Efficient Surface Reconstruction from Contours**. Information and Computing Sciences Division, Lawrence Berkeley National Laboratory. University of California. Berkeley, California. 2001.
- CRISTIANSEN, H. N. e SERDERBERG, T. W. **Conversion of Complex Counter Line Definition into Polygonalement Mosaics**. Computer Graphics, volume 12, páginas 187-192. Agosto, 1978.
- FIGUEIREDO, L. H.; CARVALHO, P. C. P. Introdução à Geometria Computacional. [S.l.]: Rio de

Janeiro: Instituto de Matemática Pura e Aplicada, 18º Colóquio Brasileiro de Matemática, 1991. 108 p. Disponível vem em <<http://www.inf.ufpr.br/andre/geom/geom.html>>.

Acessado em 31 mai. 2009.

FLOATER, M. e WESTGAARD, G. Smooth Surface Reconstruction from Cross-Sections Using Implicit Methods. Rep., SINTEF. 1996.

FOLEY, R. N., PARFREY, P. S., HARNETT, J. D., KENT, G. M., MURRAY, D. C., BARRE, P. E. Computer Graphics: Principles and Practice. 2. ed. [S.l.]: Addison-Wesley Publishing Company, 1996. 1174 p.

FUCHS, H., KEDEM, Z.M., e USELTONM, S. Optimal surface reconstruction from planar contours. Communications of the ACM, 20(10):693–702, Outubro, 1977.

GATTASS, M.; PEIXOTO, A. Reconstrução de superfícies a partir de seções bidimensionais. Technical Report MCC 28/00, PUC-Rio, Julho 2000. Disponível em <http://www.visgraf.impa.br/Data/RefBib/PS_PDF/peixoto-gattass-jul00/SurfaceReconstruction.pdf>.

GITLIN, C., O'ROURKE, J., e SUBRAMANIAN, V. On Reconstructing Polyhedra from Parallel Slices. International Journal of Comput. Geometry & Applications, 6, 1, 103-122. 1996.

KANEDA, K., HARADA, K., NAKAMAE, E., YASUDA, M. e SATO, A. G. Resctruction and Semi-Transparent Display Method for Observing Inner Structure of an Object Consisting of Multiple Surfaces. The Visual Computer, 3:137-144, 1987.

KEPPEL, E. Aproximating Complex Surfaces by Triangulation of Contour Lines. IBM J. Res. Develop., 19:2-11, Janeiro, 1975.

LAWRENCE, M. P. Surface Reconstruction Using Variational Interpolation. M.Sc. Thesis, University of Saskatchewan Saskatoon, Novembro 2005. Disponível em <<http://www.collectionscanada.ca/obj/s4/f2/dsk3/SSU/TC-SSU-11232005163941.pdf>>

LEVIN, D. Multidimensional Reconstruction by Set-Valued Approximation. IMA Journal of Numerical Analysis, 6, 173-184. 1986.

LI, Z., MA, L. e TAN, W. Three-dimensional object reconstruction from contour lines. Association for Computing Machinery. Hong Kong. Junho, 2006.

MEYES, D. Reconstruction of Surfaces From Planar Contours. Doctor of Philosophy University of Washington. 1994.

MILONE, G. Estatística Geral E Aplicada. Editora Thomson. 1ª Edição. São Paulo, 2003. 498 pgs.

- MORTENSON, M. E. **Geometric Modeling**. Industrial Press Inc. 3^a Edição. New York, 2006. 505 pgs.
- NONATO, L. G. – “Reconstrução Volumétrica de Objetos definidos por seções Bidimensionais”, Tese de Doutorado, Departamento de Matemática, Pontifícia Universidade Católica, Rio de Janeiro, 1998.
- SHIRLEY, P. Fundamentals of Computer Graphics. A K Peters. 2^a Ed. Massachusetts. 2005.
- SIDLESKY, A., **Polygon reconstruction from line cross-sections**, M.Sc. Thesis, Dept. of Computer Science, Technion, Junho 2006.
- SKINNER, S. M. The Correspondence Problem: Reconstruction of Objects From Countors in Parallel Sections. Dissertação de Mestrado. Departamento de Engenharia e Ciência da Computação, Universidade de Washington, 1991.
- SLOAN, K. R., PAINTER, J. **From Countors to Surfaces: Testbed and Initial Results**. In *Proc. CHI+GI'87*, pages 115-120, Toronto, Canadá. Abril, 1987.
- SOARES, J. F. Introdução à Estatística. Editora Ltc. 2^a Edição. São Paulo, 2005. 340 pgs.
- TAVARES, G., SANTOS, R., LOPES, H., LEWINER, T., VIEIRA, A. W. **Topological reconstruction of Oil reservoirs from seismic surfaces**. International Association for Mathematical Geology. 2003.
- VAINSENCHER, I. Notas de geometria analítica elementar. Departamento de Matemática – UFMG. Belo Horizonte, 2005. Disponível em <<http://www.mat.ufmg.br/~israel/Ensino/G-an/g-an.pdf>>. Acessado em 15 mai. 2009.
- VARGAS, A.J.C - β -Connection: Uma Família de Objetos Tridimensionais Reconstruídos a Partir de Seções Planares. Dissertação de Mestrado. Universidade de São Paulo – São Carlos, 2001.
- VENTURI, J. J. Álgebra Vetorial e Geometria Analítica. 9^a Edição. Biblioteca Central UFPR. Curitiba, 1949. 242Pgs.
- ZAMANSKY, A. A Framework for Surface Reconstruction of Sparsely-Sampled Objects. Research Thesis Master of Science in Computer Sciences. Technion- Israel Institute of Technology. Julho, 2007.

APÊNDICE A – Funções

- **DIV(M,N)**: representa o resultado inteiro da divisão de M por N. Por exemplo, tendo $M = 8$ e $N = 2$, $\text{DIV}(8,2)$ é igual a 4 ou ainda, $M = 9$ e $N = 5$, $\text{DIV}(9, 5)$ tem como resultado final 1;
- **MOD(M,N)**: representa o resto da divisão de M por N. Por exemplo, tendo $M = 8$ e $N = 2$, $\text{MOD}(8,2)$ é igual a 0 ou ainda, $M = 9$ e $N = 5$, $\text{MOD}(9, 5)$ tem como resultado final 4;
- **DIF(M,N)**: denota-se a diferença entre M e N. Exemplificando, se $M = 5$ e $N = 4$, $\text{DIF}(5,4)$ resulta em 1;
- **MAX(M,N)**: resulta no maior valor entre os dois (M ou N). Sendo assim, se $M = 7$ e $N = 10$, $\text{MAX}(7,10)$ tem como valor final 10;
- **MIN(M,N)**: resulta no menor valor entre os dois (M ou N). Sendo assim, se $M = 7$ e $N = 10$, $\text{MIN}(7,10)$ tem como valor final 7;
- **Distância euclidiana entre pontos em 2D - d2D(M,N)**: para calcular a menor distância entre dois pontos bidimensionalmente utiliza-se a equação:

$$\mathbf{d2D} = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \quad (1).$$

Na qual o resultado será um valor decimal;

- **Distância euclidiana entre pontos em 3D – d3D(M,N)**: para calcular a menor distância entre dois pontos tridimensionalmente utiliza-se a equação:

$$\mathbf{d3D} = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2} \quad (2).$$

Ter-se-á como resultado final um valor decimal. Pelo fato de utilizar uma dimensão a mais que bidimensional, o cálculo é mais custoso.

APÊNDICE B – Comparativo e Resultados

B.1 - Parametrização dos Algoritmos

Parâmetro/Algoritmo		AHS	CCCH	LMT	Delta+1	Delta+2	Delta+3	Delta+4
Curve Twisting	First Point	Closed to Aux. Point	3D-Distance	3D-Distance	3D-Distance	3D-Distance	3D-Distance	3D-Distance
	Reordering	Index	Index	Index	Index	Index	Index	Index
Curve Orientation	No	No	No	No	No	No	No	No
Curve Offsetting	No	No	No	Average Extreme x and y	No	No	Average Extreme x and y	
Point Linking	Triangular	3D-Distance	Quadrangular	Quadrangular	3D-Distance	Quadrangular	3D-Distance	
Point Balancing	Distance Distr.	No Balancing	Local Proportional Distr.	Exatly Distr.	No Balancing	Local Proportional Distr.	Exatly Distr.	

B.2 – Métricas da Veia

Algoritmo	Tempo (ms)	No. Pontos Iniciais	No. de Pontos Finais	No. Spans	Veia (BloodVein)				
					Média	Mediana	Mínimo	Máximo	Desvio Padrão
AHS	870	1285	6691	21624	10.991450	10.198039	10.000000	45.642805	2.985413
LMT	598	1285	1545	5666	11.536869	10.198039	10.000000	48.741784	4.332388
CCCH	455	1285	1285	10812	12.029230	10.488088	10.000000	51.429563	3.758807
Delta1+	1379	1285	5731	9852	11.140271	10.124219	10.000000	48.580260	3.397523
Delta2+	1415	1285	5731	19704	11.228211	10.198039	10.000000	48.741769	3.433337
Delta3+	1249	1285	5731	9852	11.251780	10.138204	10.000000	48.580260	3.515362
Delta4+	1508	1285	5731	19704	10.259072	10.049876	10.000000	18.867962	0.659595

B.3 – Métricas do Coração

Algoritmo	Tempo (ms)	No. Pontos Iniciais	No. de Pontos Finais	No. Spans	Span				
					Média	Mediana	Mínimo	Máximo	Desvio Padrão
AHS	241	1285	3962	10708	0.192038	0.144682	0.100000	0.698353	0.108138
LMT	78	1285	1534	2926	0.214455	0.150147	0.100000	1.304034	0.171052
CCCH	77	1285	1285	5354	0.223874	0.167227	0.100000	1.135772	0.145252
Delta4+	368	1285	3820	10424	0.150982	0.115733	0.100000	0.701374	0.089206

B.4 – Métricas do Fêmur

Algoritmo	Tempo (ms)	No. Pontos Iniciais	No. de Pontos Finais	No. Spans	Span				
					Média	Mediana	Mínimo	Máximo	Desvio Padrão
AHS	496	1285	4666	13524	0.160681	0.109915	0.100000	1.335871	0.155199
LMT	239	1285	1492	3588	0.178202	0.109736	0.100000	1.547527	0.205474
CCCH	238	1285	1285	6226	0.189203	0.124599	0.100000	1.197815	0.158996
Delta4+	864	1285	4594	13380	0.134891	0.104511	0.100000	1.409906	0.132289

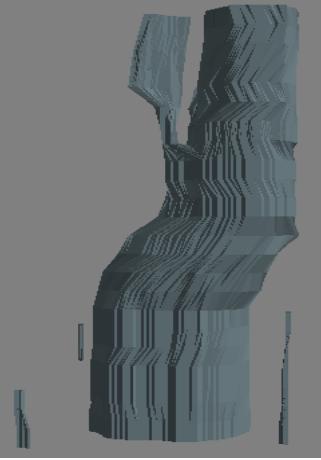
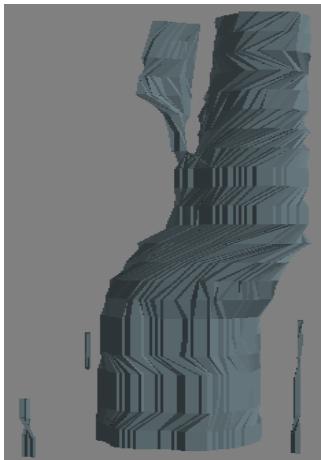
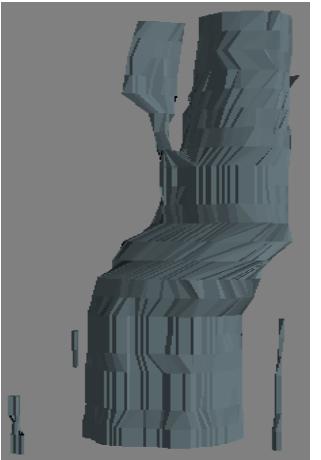
B.5 – Métricas do Pulmão

Algoritmo	Tempo (ms)	No. Pontos Iniciais	No. de Pontos Finais	No. Spans	Span				
					Média	Mediana	Mínimo	Máximo	Desvio Padrão
AHS	242	1285	3936	10604	120.897690	111.091179	100.000000	318.632233	28.113130
LMT	162	1285	1474	2840	129.822049	111.266060	100.000000	432.152751	45.302336
CCCH	87	1285	1285	5302	182.402979	142.305877	100.000000	810.977188	108.034076
Delta4+	473	1285	3704	10140	110.852218	105.135464	100.000000	394.405882	18.895067

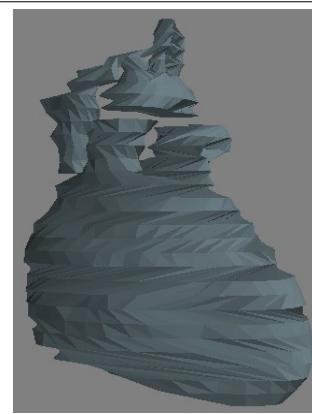
B.6 – Métricas da Veia: Segmentos de contorno e Spans

Algoritmo	Tempo (ms)	No. Pontos Iniciais	No. de Pontos Finais	No. Spans e Segmentos	Spans e Segmentos				
					Média	Mediana	Mínimo	Máximo	Desvio Padrão
AHS	870	1285	6691	32436	22.641663	21.256960	20.010153	93.476464	5.996490
LMT	598	1285	1545	11332	23.013941	22.828427	10.163136	101.998324	9.642954
CCCH	455	1285	1285	16218	24.294483	22.669299	11.000000	106.927088	8.653944
Delta1+	1379	1285	5731	19704	21.103474	21.669078	10.000043	101.686636	8.074481
Delta2+	1415	1285	5731	29556	23.175017	21.333267	20.000000	101.652875	6.919721
Delta3+	1249	1285	5731	19704	21.298615	21.693989	10.000043	101.686636	8.268876
Delta4+	1508	1285	5731	29556	20.272964	20.755142	10.000031	44.867962	3.366304

B.7 – Comparativo Visual dos Objetos

Veia (BloodVein)			
AHS	CCCH	LMT	Delta+4
			
$\Delta = 0.1$, Número de Pontos = 1285, Número de Contornos = 65, Número de Fatias = 24, Distância entre Fatias = 10			

Coração (Heart)



AHS

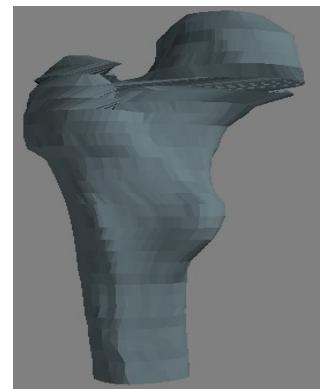
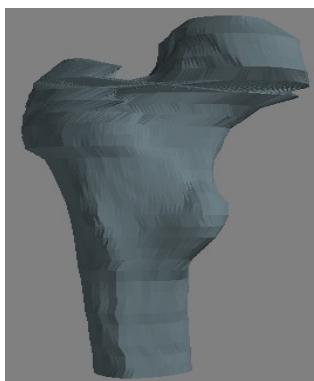
CCCH

LMT

Delta+4

$\Delta = 0.1$, Número de Pontos = 1285, Número de Contornos = 65, Número de Fatias = 30, Distância entre Fatias = 0.1

Fêmur (Hip)



AHS

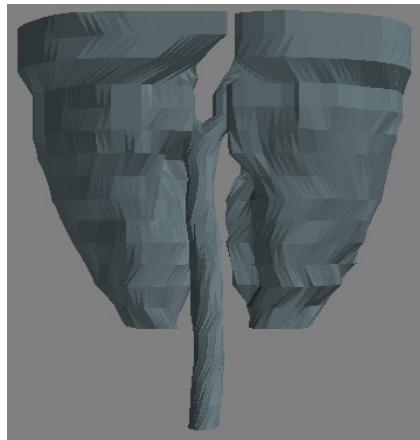
CCCH

LMT

Delta+4

$\Delta = 0.5$, Número de Pontos = 1285, Número de Contornos = 65, Número de Fatias = 34, Distância entre Fatias = 0.1

Pulmão (Lung)



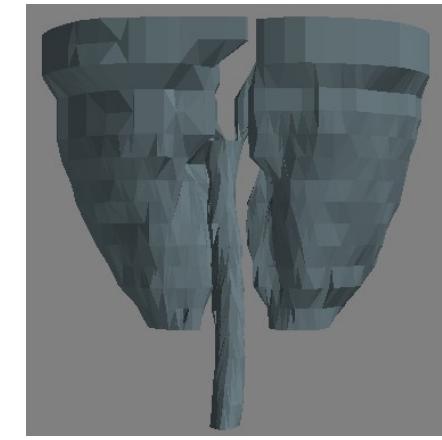
AHS



CCCH



LMT



Delta+4

$\Delta = 30000$, Número de Pontos = 1285, Número de Contornos = 65, Número de Fatias = 19, Distância entre Fatias = 100