Um algoritmo para a geração da linha de centro de geometrias tubulares com ramificações genéricas

Márcio Ricardo Pivello
Ignacio Larrabide
Raúl A. Feijóo

Apresentamos neste trabalho um algoritmo para o cálculo aproximado da linha de centro de geometrias tubulares, tridimensionais, de forma arbitrária. O algoritmo baseia-se somente na conectividade da malha e consiste em identificar frentes de avanço, formadas por arestas dos elementos que definam aproximadamente uma seção transversal da geometria, e calcular seu centróide. Avançando-se esta seção ao longo da geometria, através de todas as suas possíveis ramificações, obtém-se um conjunto de centróides que, devidamente conectados, constituem um modelo de elementos unidimensionais que representam aproximadamente a linha de centro da geometria original. Este algoritmo será utilizado na geração de um modelo unidimensional detalhado do sistema arterial humano, a ser empregado na modelagem 3D–1D acoplada do sistema cardiovascular.

Categories and Subject Descriptors: []:

General Terms:

Additional Key Words and Phrases:

1. INTRODUÇÃO

A modelagem computacional do sistema cardiovascular humano (SCVH) tem grande importância para o entendimento do escoamento sangüíneo, tanto no que diz respeito aos fenômenos físicos envolvidos quanto ao desenvolvimento de algumas patologias, como aneurismas e estenoses, por exemplo. Os modelos mais realistas consistem em acoplar uma representação 3D detalhada da região de maior interesse com um modelo 1D da árvore arterial, que representa o coração e as maiores artérias do corpo [Urquiza et al. 2006; Larrabide and Feijóo 2007]. Estes modelos 1D, por si só, já fornecem informações importantes sobre o escoamento médio do sistema, tais como vazão e fluxo sangüíneo e pressão arterial. Matematicamente, trata-se de uma simplificação das equações de Navier – Stokes para domínios deformáveis, onde as artérias não representadas geometricamente são modeladas como terminais de Windkessel. Por se tratar de modelos 1D, o modelo geométrico não precisa representar fielmente a forma do sistema arterial real, como pode ser visto na figura 1, que mostra o modelo geométrico da árvore arterial proposto por Avolio [Avolio 1980], que consiste de somente 128 segmentos arteriais. Por outro lado, se os modelos unidimensionais fossem mais detalhados, poderiam ser empregados em outras áreas, como por exemplo o planejamento de cirurgias. Nestes casos, a região que sofrerá a intervenção cirúrgica sofre uma modificação no leito arterial para desviar a circulação sangüínea. Devido à complexidade do sistema arterial, há diversas opções para executar o desvio, também chamado de *bypass*, e um modelo 1D detalhado do sistema arterial permitiria simular várias configurações possíveis. Com base no resultado das simulações, o cirurgião poderia escolher então a opção mais adequada ao paciente.

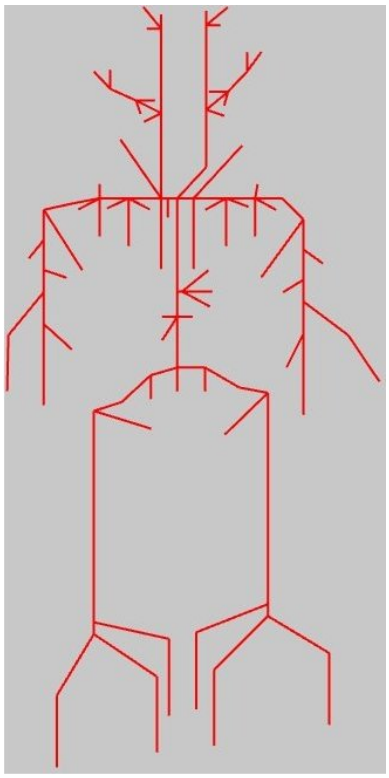

Fig. 1. Modelo unidimensional de Avolio para o sistema arterial humano.

Neste sentido, apresentamos neste trabalho um algoritmo para a geração de um modelo geométrico unidimensional do sistema arterial, a partir da linha de centro de um modelo 3D detalhado do mesmo, a fim de empregá-lo em simulações 1D e 3D–1D acopladas do SCVH.

O cálculo da linha de centro de uma geometria 3D, entretanto, não é um problema trivial, e sua solução depende de vários fatores, tais como o formato e o tamanho dos dados de entrada. No caso de imagens médicas, normalmente obtidas via ressonância magnética (MRI) ou tomografia computadorizada (CT), a linha de centro pode ser calculada ainda na fase de segmentação das imagens. Tal abordagem é discutida em detalhes em [Frangi 2001; Krissian et al. 2004].

Em [Krissian et al. 2004] 3 algoritmos para extração de linhas de centro de vasos sangüíneos são comparados, dos quais destacamos dois:

—O **Cálculo da Linha de Centro via Invariante Topológica** baseia-se na Transformada de Distância do Objeto [Borgefors 1984], aplicando-se o divergente sobre o gradiente da Transformada de Distância, e eliminando pontos do domínio segundo valores decrescentes do divergente.

—A **Detecção da Linha de Centro via Análise Multiescalas** utiliza a matriz Hessiana para descrever características e a orientação da geometria em nível local.

Valores baixos no gradiente dos auto-valores expressam a variação da intensidade na direção dos auto-vetores associados.

Frangi [Frangi 2001] propõe um método que opera nas imagens 3D diretamente, ao invés de utilizar as projeções 2D, usando um modelo 3D deformável, que consiste no acoplamento da linha de centro do vaso com uma representação da sua superfície. Tanto a linha de centro quanto a superfície são representados via B-Splines.

Entretanto, a representação geométrica do sistema arterial completo requer alto nível de detalhes, e sua segmentação demandaria grande esforço não somente computacional, mas também em horas de trabalho. Isso, é claro, sem mencionar a dificuldade em se obter dados de MRI ou CT de um único paciente suficientes para construir uma imagem do sistema arterial completo. Por isso, será utilizado neste trabalho um modelo previamente segmentado, representado por uma malha de triângulos, adquirido da empresa **21st Century Solutions** [21st Century Solutions]. O modelo representa detalhadamente os principais vasos do sistema cardiovascular, tanto artérias como veias, além do próprio coração - embora somente as artérias sejam de interesse neste caso em particular.

Note que, a partir da triangulação da superfície, seria possível utilizar algoritmos baseados em geodésicas, como [Verroust and Lazarus 2000]. Isto porém, exigiria maior esforço computacional que o algoritmo proposto, que trabalha somente com busca sobre a conectividade. De fato, o único cálculo efetuado durante todo o processo é o das coordenadas do centróide, o que não exerce grande influência no custo computacional do algoritmo.

O restante deste trabalho está organizado da seguinte forma: Na seção 2 será apresentada a formalização do problema, com a elaboração de um algoritmo simples, para o cálculo da linha de centro de uma geometria cilíndrica sem ramificações. Este algoritmo será generalizado na seção 3, para uma geometria genérica, com ramificações arbitrárias e malha irregular. Os aspectos computacionais da implementação do algoritmo serão abordados na seção 4. Os resultados serão apresentados na seção 5, e as conclusões acerca do algoritmo, bem como as futuras direções do trabalho, serão apresentadas na seção 6.

2. O PROBLEMA BÁSICO

Nesta seção apresentaremos o algoritmo para uma geometria simples, sem ramificações e com malha regular, a partir do qual as etapas básicas serão detalhadas. Em etapas posteriores este será generalizado para malhas irregulares, com geometrias genéricas.

O algoritmo é inicializado fornecendo-se a malha de triângulos e uma seção de entrada, definida pelos pontos localizados em uma das extremidades do cilindro. Aqui, uma seção é definida como um conjunto de pontos que, quando conectados, constituem um contorno fechado. O primeiro passo é calcular o centróide desta seção, avançando-se então sobre o domínio, buscando-se a seção vizinha através da conectividade da malha. Uma vez identificada a nova seção, os pontos adjacentes são conectados e o processo é repetido até que todo o domínio seja percorrido. Para uma geometria sem ramificações no domínio este procedimento é descrito pelo algoritmo 1, **CenterLineStraightTube**.

ALGORITMO 1. ***: CenterLineStraightTube***

(*1*) *Dados:*
Malha M
Seção inicial S_0
(*2*) *Calcular* C_0*, Centróide de* S_0
(*3*) *Encontrar a nova seção* S_1*, adjacente a* S_0
(*4*) *Calcular* C_1*, Centróide de* S_1
(*5*) *Criar elemento* E_0*, formado pelos nós* C_0 *e* C_1
(*6*) S_1 *é a última seção?*
Sim : fim
Não: $S_0 = S_1$ *e volta ao passo 3.*

A figura 2 mostra a malha tridimensional do cilindro acima e a malha 1D associada abaixo, gerada pela aplicação do algoritmo 1. Neste caso, com a geometria regular e uma malha uniforme, as seções identificadas são transversais, situação improvável em geometrias complexas.

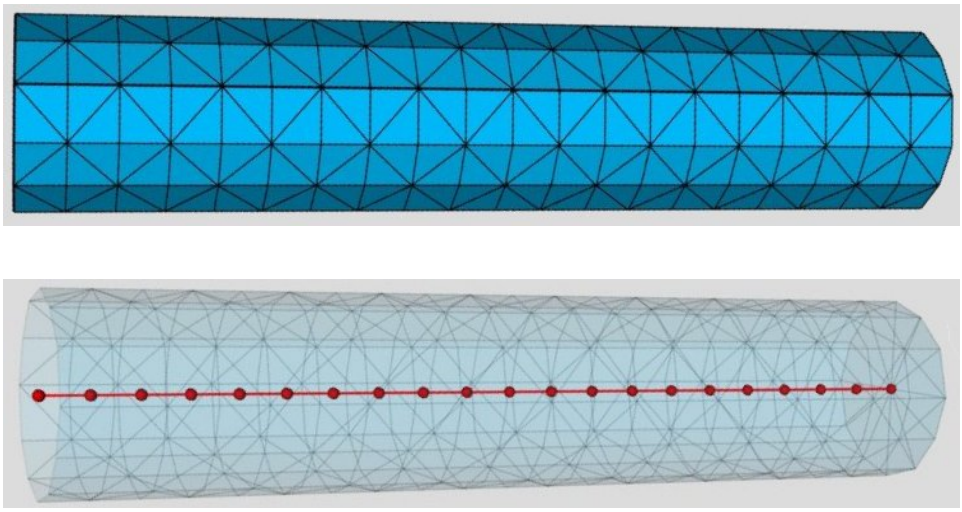

Fig. 2. Malha 3D de um cilindro de seção circular (esquerda) e a respectiva malha 1D (direita)

Resta ainda definir o algoritmo para a identificação da seção adjacente à seção atual, que permitá avançar sobre a geometria. Primeiramente, cria-se um conjunto com os elementos que possuem os nós que formam esta seção. Depois, as arestas destes elementos são separadas em dois grupos: um com as arestas que possuem nós neste conjunto e outro com aquelas que não possuem. As arestas do segundo grupo formam a seção adjacente. Para garantir que a busca ocorra sempre na direção correta, os elementos que possuem arestas que formam seções já percorridas são agrupados em um conjunto e removidos do espaço de busca. O algoritmo 2, chamado de **NextSection**, formaliza este procedimento.

ALGORITMO 2. ***: NextSection***

(*1*) *Dados:*
  Malha M
  S_0: Conjunto dos nós que definem a seção de contorno
  E_0: Conjunto dos elementos já percorridos
(*2*) *Criar E_1, conjunto dos elementos que possuem nós em S_0 mas não pertencem a E_0*
(*3*) *Criar A_1, conjunto das arestas dos elementos de E_1*
(*4*) *Criar A_2, subconjunto de A_1 formado pelas arestas que não possuem nós em S_0*
(*5*) *Atualizar E_0 com os elementos que possuem as arestas que formam a seção A_2*
(*6*) *Retornar A_2, conjunto das arestas que formam a nova seção de contorno.*

Ao utilizarmos como ponto de partida os *elementos* e não as *arestas* que possuem nós no conjunto de entrada, a busca é de certa forma otimizada, uma vez que o número de elementos é consideravelmente menor que o de arestas. Assim, a identificação das arestas da próxima seção é feita em um espaço de busca menor, pois somente as arestas associadas aos elementos já identificados são verificadas.

3. O ALGORITMO GERAL

Considere o caso ilustrado na figura 3. Se o algoritmo **CenterLineStraightTube** for aplicado a este domínio sem nenhuma modificação, ele percorrerá as seções após a ramificação como sendo uma única. Logo, o centróide calculado estará localizado entre as duas seções, e não representará a linha de centro da geometria, mas sim a linha de centro média entre os dois segmentos criados pela ramificação. Por outro lado, se aplicarmos o algoritmo a cada uma das seções separadamente, o resultado obtido será o esperado. O algoritmo 1 deve então ser modificado, incluindo-se não somente um teste para a identificação de ramificações, mas um *loop* que garanta que os segmentos também sejam percorridos. A introdução destas modificações leva à formulação do algoritmo 3, **CenterLine**:

ALGORITMO 3. ***: CenterLine***

(*1*) *Dados:*
  Malha M
  Seção de contorno inicial S_0
(*2*) *Calcular C_0, Centróide de S_0*
(*3*) *Encontrar a nova seção S_1, adjacente a S_0*
(*4*) *Analisar S_1, em busca de ramificações*
(*5*) *S_1 tem ramificações?*
  Não:
    Calcular C_1, Centróide de S_1
    Criar elemento E_0, formado pelos nós C_0 e C_1
    Se S_1 é a última seção, fim.
    Senão, $S_0 = S_1$ e volta ao passo 2.
  Sim:
    Repetir os passos 2 a 5 até que todos os segmentos sejam percorridos.

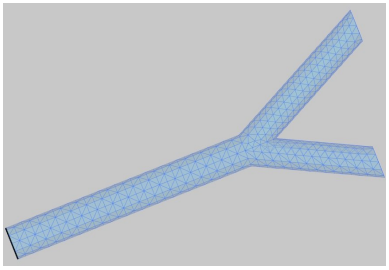

Fig. 3. Malha 3D de um cilindro de seção circular com ramificação.

O critério para a identificação de ramificações é simples: cada elemento da seção é formado por dois nós, e cada nó é compartilhado por dois elementos. Em uma seção sem ramificações é fácil verificar que, percorrendo-se a matriz de conectividade da seção e buscando-se o elemento adjacente seguindo-se sempre a mesma direção, chega-se ao primeiro elemento depois de se ter visitado todos os elementos da matriz, completando-se assim um contorno fechado. Se houver alguma ramificação na seção (e.g., se na seção sob consideração a artéria estiver se bifurcando), o elemento de partida é encontrado novamente antes de se visitar toda a matriz. Logo, estes elementos definem uma ramificação. Eliminando-se do espaço de busca os elementos já visitados e recomeçando-se o procedimento com os elementos restantes, todas as ramificações da seção serão identificadas. Este procedimento está formalizado no algoritmo 4, **FindBranch**.

ALGORITMO 4. ***: FindBranch***

(1) *Dados:*
  A_0:Conjunto de arestas fornecido como dado de entrada
  S_0:Conjunto de arestas que definem uma seção (inicialmente vazio)
  S:Conjunto das seções identificadas em A_0 (inicialmente vazio)
(2) *a_0 = primeira aresta não percorrida de A_0*
(3) *Retirar a_0 de A_0 e inserí-la em S_0*
(4) *Buscar a_1, vizinha de a_0 em A_0*
(5) *Retirar a_1 de A_0 e inserí-la em S_0*
(6) *Buscar a_{next}, vizinha de a_1 em A_0*
(7) *a_{next} é adjacente a a_0?*
  Não:
    Retirar a_{next} de A_0 e inserí-la em S_0
    $a_1 = a_{next}$ e voltar ao passo 5
  Sim:
    Retirar a_{next} de A_0 e inserí-la em S_0
    Inserir S_0 em S
    iniciar nova seção ($S_0 = \oslash$)
    voltar ao passo 1
(8) *Repetir os passos 1 - 7 até que todas as arestas tenham sido percorridas.*
(9) *Retornar S, conjunto das seções identificadas.*

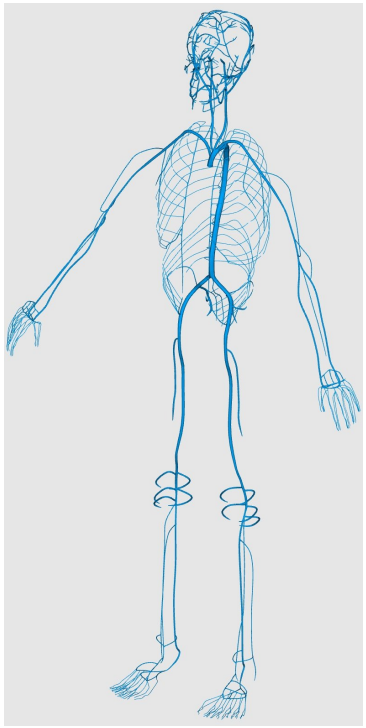

Fig. 4. Malha 3D representando os principais vasos do sistema cardiovascular humano.

Portanto, o algoritmo completo para o cálculo da linha de centro de uma geometria 3D genérica, é obtido simplesmente pela incorporação dos algoritmos **FindBranch** e **NextSection** ao algoritmo 5, **CenterLine**:

ALGORITMO 5. ***: CenterLineGeneral***

(1) *Dados:*
  Malha M
  Seção de contorno inicial S_0
(2) *Calcular* C_0, *Centróide de* S_0
(3) $S_1 =$***NextSection***(S_0)
(4) $vS =$***FindBranch***(S_1)
(5) vS *tem ramificações?*
  Não:
    $S1 = vS_0$
    Calcular C_1, *Centróide de* S_1
    Criar elemento E_0, *formado pelos nós* C_0 *e* C_1
    Se S_1 *é a última seção, fim.*
    Senão, $S_0 = S_1$ *e volta ao passo 3.*
  Sim:
    Para cada ramificação vS_i, $S_0 = vS_i$ *e volta ao passo 2*

Um aspecto importante do algoritmo diz respeito à ordem em que as seções que definem ramificações são percorridas, o que pode ser feito de duas maneiras:

(1) Percorrer a árvore recursivamente, em profundidade: Uma vez identificada uma ramificação, o algoritmo é aplicado recursivamente sobre ela - e, por conseqüência, às ramificações de suas ramificações - até que toda a árvore tenha sido percorrida.

(2) Percorrer a árvore em largura, ou horizontalmente. Nesta caso, a recursividade não é empregada. Quando uma ramificação é encontrada, as seções que a definem são armazenadas em uma lista de espera, e fornecidas como entrada ao algoritmo **FindBranch** na ordem em que foram inseridas, ou seja: primeiro a entrar, primeiro a sair. Dessa forma, define-se um segmento e suas respectivas ramificações, antes que as ramificações subjacentes sejam percorridas.

A segunda opção foi escolhida para este trabalho, devido à presença de ramificações *bypass*, presentes em conjuntos arteriais no cérebro, braços e pernas, e que poderiam levar a um encerramento prematuro do algoritmo se a árvore arterial fosse percorrida em profundidade ao invés de largura. A influência desse tipo de ramificação será estudado em detalhes durante a análise dos resultados obtidos.

4. IMPLEMENTAÇÃO COMPUTACIONAL

A implementação computacional foi feita em C++, seguindo-se o padrão ANSI-ISO. Dessa forma, tem-se um código portável e independente do sistema operacional. Foram utilizados os contêineres padrão da biblioteca STL [Stroustroup 2000] bem como seus algoritmos, sempre que necessário. Esta escolha, além de reduzir o tempo de implementação, confere ao código mais portabilidade e certamente ganho de performance.

A elaboração das classes foi feita seguindo-se a estrutura já definida no ambiente de desenvolvimento integrado HeMoLab [Larrabide and Feijóo 2007], que divide a árvore arterial em segmentos, elementos, nós e terminais. A classe `Mesh` armazena os dados da malha original, lida de um arquivo no formato `TriSurf`. À medida em que os centróides são calculados, são armazenados em objetos da classe `Pipe1D`, que representa elementos 1D lineares. Os elementos localizados entre duas ramificações consecutivas são armazenados em objetos da classe `Segment`, que representa os segmentos arteriais. Os segmentos, por sua vez, são armazenados em objetos da classe gabarito `Tree`, que representa uma árvore binária generalizada, ou árvore n-ária. As seções são representadas como `arrays` contendo os índices dos pontos que as compõem.

5. RESULTADOS

Os resultados obtidos pela aplicação do algoritmo `CenterLineGeneral` à arvore arterial serão apresentados nesta seção. A deformação da geometria 1D em algumas regiões também será discutida.

Para a geração da malha 1D foi usada a geometria representada na figura 4. A malha tem 459051 elementos e 229479 nós. A entrada da artéria aorta foi utilizada como seção inicial. A malha 1D obtida tem 21421 elementos, 742 segmentos arteriais e 22743 nós, e está ilustrada na figura 5. Foi utilizado um computador intel

Xeon com sistema operacional Linux, kernel 2.6.17. O código computacional foi compilado com compilador intel versão 9.0. O tempo de execução com opção de compilação `-fast` foi de aproximadamente 1 minuto.

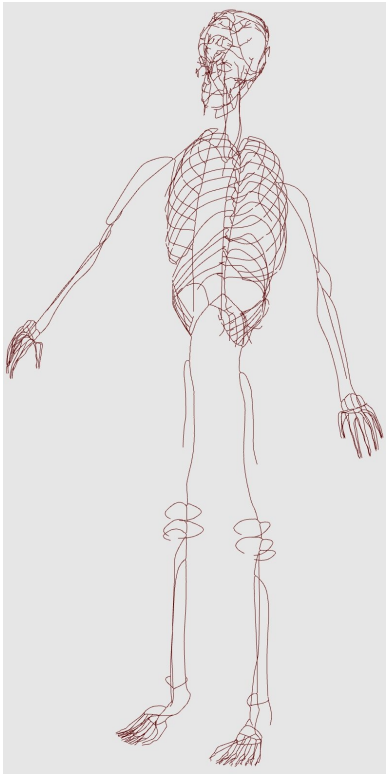

Fig. 5. Malha 1D do sistema arterial

A figura 6 mostra o arco aórtico, artéria braquial direita e aorta descendente. Na comparação da sobreposição das malhas 3D e 1D na região é possível notar a distorção na conexão entre as ramificações. Detalhes desta distorção na artéria braquial direita são mostrados na figura 7.

Isto ocorre porque, atualmente, os segmentos que representam as ramificações são conectados simplesmente ligando-se o primeiro ponto de um com o último ponto de outro. Devido à topologia da geometria, normalmente o último centróide do segmento pai localiza-se no vértice da bifurcação, e dependendo do refinamento da malha na região, bem como do ângulo entre os dois segmentos, pode ocorrer uma distorção excessiva da malha 1D no local. A figura 6 mostra essa configuração no caso das ramificações da aorta que dão origem às artérias carótidas comuns e à artéria braquial direita. Já a figura 8 mostra detalhes de alguns vasos localizados na cabeça, onde a diferença de diâmetro entre os vasos é menor, e as ramificações ocorrem também em menor ângulo. Note que, neste caso, a deformação na malha 1D é consideravelmente menor.

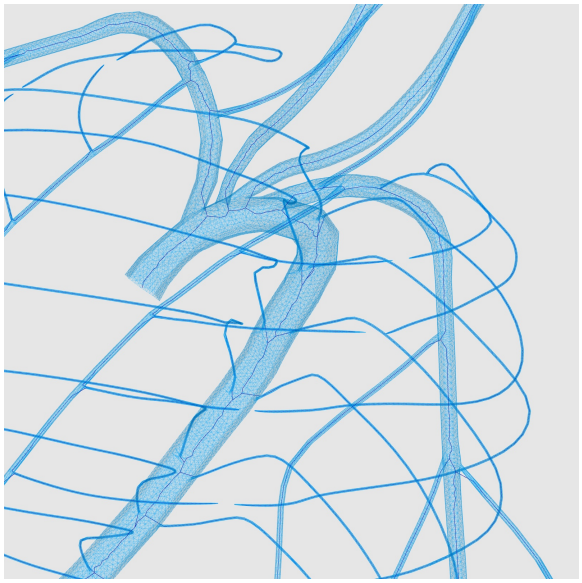

Fig. 6. Detalhe das malhas 3D e 1D da região do arco aórtico e aorta descendente.

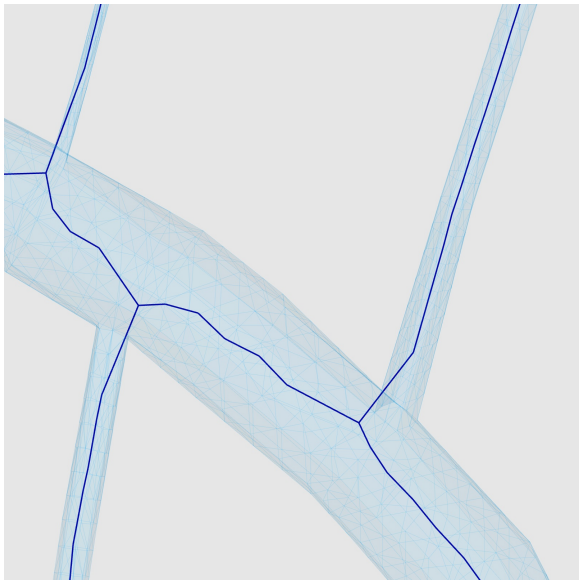

Fig. 7. Detalhe das malhas 3D e 1D no início da artéria braquial direita.

Embora esta deformação na malha não prejudique o atual modelo 1D do sistema arterial, que utiliza apenas grandezas escalares, a solução deste problema será foco da próxima etapa deste trabalho, pois poderia afetar futuras formulações que levem em conta o ângulo das ramificações na árvore arterial.

Com relação às ramificações e junções do tipo *bypass*, o algoritmo mostrou-se

Fig. 8. Detalhes de artérias localizadas na cabeça, onde a deformação nas ramificações é menor

bastante robusto. A figura 9 mostra a sobreposição das malhas 3D e 1D no braço direito, na região das conexões entre as artérias ulnar, radial e braquial. A descontinuidade da malha em alguns nós deve-se apenas ao critério de parada do algoritmo, e não representa uma falha ao percorrer tais regiões. O algoritmo **NextSection** apenas identifica a seção adjacente à atual, com base na existência ou não de elementos que tenham nós localizados na seção atual, e que ainda não tenham sido visitados. Quando uma ramificação em *bypass* é encontrada, ele não a percorre novamente porque seus elementos já foram visitados quando a ramificação anterior foi identificada. Assim, nenhuma seção adjacente é identificada, o que significa o fim do segmento. A resolução deste problema não é complexa, mas exige algumas mudanças significativas na estrutura de dados e no algoritmo. A primeira delas seria armazenar a árvore arterial em uma estrutura de grafos em vez de uma árvore binária generalizada. Isto seria necessário porque segmentos pertencentes a ramificações com *bypass* têm na verdade dois segmentos pais, e tal conexão não pode ser representada em uma estrutura de árvore convencional. A segunda modificação seria no algoritmo **CenterLineGeneral**. Uma vez percorrida toda a árvore arterial, um passo adicional seria implementado, realizando um teste simples. O algoritmo **NextSection** seria aplicado novamente a cada extremidade de segmento, tendo como espaço de busca os elementos que possuem nós na última seção de todos os outros segmentos da árvore. Caso fosse encontrada uma conexão, a relação de parentesco daquele segmento seria modificada na estrutura de grafos, identificando-o como um *bypass*.

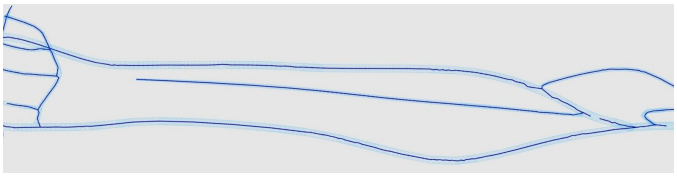

Fig. 9. Detalhe das malhas 3D e 1D da região do braço direito.

6. CONCLUSÕES E PRÓXIMAS ETAPAS

—O algoritmo proposto obteve sucesso no cálculo da linha de centro de geometrias complexas, tendo sido capaz de percorrer um modelo 3D do sistema arterial humano fornecendo como resultado uma representação 1D equivalente ao modelo original.

—A conexão entre segmentos adjacentes deve ser melhorada

—O modelo 1D não tem qualquer tipo de parametrização. A representação atual é feita apenas pelos elementos obtidos através do cálculo do centróide.

As próximas etapas deste trabalho incluem as seguintes tarefas:

—Desenvolvimento de um algoritmo para a minimização da distorção das conexões entre segmentos

—Representação dos segmentos do modelo 1D via splines

—Inclusão de importantes conjuntos arteriais no modelo, tais como polígono de Willis, artérias mesentéricas e renais, entre outros.

REFERENCES

21st Century Solutions. www.anatomium.com.

Avolio, A. 1980. Multi–branched model of the human arterial system. *Med. Biol. Engrg. Comp. 18*, 709–718.

Borgefors, G. 1984. Distance transformations in arbitrary dimensions. *CVGIP 27*, 321–345.

Frangi, A. F. 2001. Three-dimensional model-based analysis of vascular and cardiac images. Ph.D. thesis, University Medical Center, Utretch, The Netherlands.

Krissian, K., Kikinis, R., and Westin, C. 2004. Three-dimensional model-based analysis of vascular and cardiac images. Tech. rep. september. Technical Report. Laboratory of Mathematics in Imaging, Harvard Medical School.

Larrabide, I. and Feijóo, R. 2007. Hemolab - hemodynamics modeling laboratory, an application for modelling the human cardiovascular system. Submitted to CGI 2007. May 30th – June 2nd.

Stroustroup, B. 2000. *A Linguagem de Programação C++*. Ed. Bookman. 3a. edição.

Urquiza, S., Blanco, P., Vénere, M., and Feijóo, R. 2006. Multidimensional modeling for the carotid blood flow. *Comput. Meth. Appl. Mech. Engrg. 195*, 4002–4017.

Verroust, A. and Lazarus, F. 2000. Extracting skeletal curves from 3d scatterd data. *The Visual Computer 16*, 15–25.