RECONSTRUÇÃO 3D DE OSSOS A PARTIR DE IMAGENS DE RAIO-X

Paulo César Gomes de Almeida[1], Francisco Assis da Silva[2], Mário Augusto Pazoti[2], Danillo Roberto Pereira[2], Almir Olivetto Artero[3]

[1]Discente da Faculdade de Informática da UNOESTE. [2]Docente da Faculdade de Informática da UNOESTE. [3]Docente da Faculdade de Ciências e Tecnologia da UNESP de Presidente Prudente

RESUMO

Este trabalho apresenta uma metodologia para a construção de um modelo tridimensional do osso inferior da perna (tíbia) de um ser humano, a partir de imagens de Raio-X. A metodologia adotada para se obter o objeto 3D é deformar o modelo até corresponder às imagens de Raios-X (frontal e laterais esquerda e direita) da tíbia. Foram utilizados os seguintes algoritmos para segmentação das imagens de Raio-X e modelagem tridimensional: *Canny*, *Border Following*, *Free Form Deformation* (FFD) e o algoritmo de deformação desenvolvido neste trabalho. O aplicativo de visualização definido e implementado mostra a imagem tridimensional do osso da tíbia resultante da deformação do objeto 3D. Os resultados dos experimentos demostram a possibilidade de representar o osso humano real obtido a partir de imagens de Raio-X em um modelo 3D, porém, uma grande dificuldade para essa atividade foi a quantidade de imagens disponíveis para o projeto.

Palavras-chave: reconstrução 3D; raio-X; modelagem tridimensional de ossos.

3D RECONSTRUCTION OF BONES FROM IMAGES OF X-RAY

ABSTRACT

This paper presents a method for building a three dimensional model of the lower leg bone (tibia) from a human, from X-ray images. The methodology adopted to obtain the 3D object is deforming the model until it corresponds to X-ray images (front and left and right sides) of the tibia. We used the following algorithms for X-ray images segmentation and three-dimensional modeling: Canny, Border Following, Free Form Deformation (FFD) and deformation algorithm developed in this work. The viewing application defined and implemented shows the three-dimensional image of the bone of the tibia resulting from deformation of the 3D object. The experimental results demonstrate the possibility of representing the real human bone obtained from X-ray images into a 3D model, however, a great difficulty for this activity was the number of images available for the project

Keywords: 3D reconstruction; X-ray; three dimensional modeling of bones.

1. INTRODUÇÃO

Os métodos de diagnóstico de doenças por imagem estão em constante renovação e incorporação, tudo em favor da saúde e do bem-estar da população. Com o aumento da expectativa de vida no país, percebe-se uma preocupação maior com a saúde. Ao saber que quanto mais precoce o diagnóstico, maiores as chances de cura dos pacientes, sendo assim de suma importância o setor de imagem para exames, detecção e prevenção de doenças. A Radiologia se tornou um ramo da medicina amplamente explorado, presente na grande maioria dos hospitais do país, com o uso de recursos de radiografias, ressonância magnética e tomografia computadorizada (SOUZA, 2012).

A técnica usando Raios-X é muito usada em hospitais, por ser de baixo custo e apresentar resultados rápidos, porém, formular diagnósticos somente a partir de Raio-X é complexo e tem a grande possibilidade de apresentar falhas, pois em uma imagem de Raio-X, as sombras de todos os objetos no caminho do feixe de Raio-X são sobrepostas. Assim, as sombras das estruturas normais podem mascarar ou inferir as sombras que indicam a doença, e a visualização do osso pode não ser nítida, não possibilitando uma visão realística com profundidade e nível de detalhes (BERTOLO, 2012).

Existe um grande interesse clínico, na ortopedia, por imagens tridimensionais (3D), para se entender, quantificar e prever a evolução e deformidades do osso, pois fornecem diagnósticos precisos para identificar patologias ou para planejamento cirúrgico. Atualmente, existem equipamentos para se obter imagens 3D usadas no auxílio em cirurgias de ortopedia, tais como tomografia computadorizada e ressonância magnética. Porém, os métodos tradicionais apresentam custo elevado, exposição a radiação, além de gerar uma sobrecarga organizacional, sendo necessário espaço físico adequado para a implantação de uma unidade de tomografia computadorizada (COSTA, 2002). Segundo Gunay, Shim e Shimada (2007), há um stress adicional para o paciente de terapia intensiva, pois necessita de deslocamento para as salas de exames. O autor também afirma a necessidade da criação de formas tridimensionais ósseas para visualização e planejamento pré-operatórios de cirurgias.

A maioria do acompanhamento assistido por computador para a realização de cirurgias ortopédicas usam modelos tridimensionais da anatomia do paciente. Estes modelos 3D aumentam a capacidade do cirurgião para visualizar formas anatômicas e diminui a invasão em procedimentos cirúrgicos, aumentando

assim, a precisão e segurança na cirurgia. As abordagens baseadas em tomografia computadorizada e ressonância magnética, usadas em muitos casos, possuem a desvantagem de apresentarem um custo elevado e induzem a radiação ao paciente, além de apresentarem um custo computacional elevado (RAJAMANI et al., 2006).

Visto a importância de imagens tridimensionais para auxiliar na medicina, este trabalho, propõe uma metodologia para a construção de um modelo tridimensional do osso inferior da perna (tíbia) de um ser humano a partir de imagens de Raio-X. Objetiva-se com isso propiciar uma visualização para servir de auxílio em diversas aplicações médicas, como no planejamento de cirurgias para médicos ortopedistas. A forma adotada para obter o modelo 3D, neste trabalho é, a partir de um modelo de um osso clinicamente normal, deformá-lo à anatomia do osso da tíbia de um ser humano representado em imagens de Raios-X (frontal e laterais esquerda e direita).

Propõe-se a implementação de uma metodologia para realizar a reconstrução de ossos 3D a partir de imagens de Raio-X, para isso são tratados os algoritmos, *Canny*, *Free Form Deformation*, *Contour Following* e o algoritmo de deformação desenvolvido neste trabalho.

As demais seções deste trabalho estão organizadas da seguinte maneira: na Seção 2 é apresentada uma alternativa de algoritmo para deformação de modelos tridimensionais e um trabalho que serviu de fonte de informação para a idealização e desenvolvimento deste trabalho; na Seção 3 é descrito suscintamente a metodologia aplicada neste trabalho; na Seção 4 é apresentada a construção do visualizador baseado no *framework* de desenvolvimento XNA; e por fim, na Seção 5 são apresentadas as conclusões e considerações finais do trabalho.

2. REVISÃO DA LITERATURA

2.1 Algoritmo para deformação aplicado a modelos tridimensionais

Na literatura, encontra-se o algoritmo *Free Form Deformation* (FFD) muito usado em vários trabalhos (SEDERBERG; PARRY, 1986). O FFD, segundo Parent (2002), cria uma grade de coordenadas com uma configuração padrão (um paralelepípedo) sobre um objeto. Para cada vértice do objeto, coordenadas relativas a esta grade local são determinadas, que registram o vértice na grade. A grade é manipulada, e com esta manipulação cada vértice é mapeado novamente dentro da grade modificada, a qual os realoca no espaço.

Inicialmente, o sistema de coordenadas local é definido por um conjunto não necessariamente ortogonal de três vetores (*S,T,U*). Para se encontrar esses três vetores no objeto, são realizadas as operações descritas em (1).

$$
\begin{aligned}
&Vetor\ \max = maior(Grade)\\
&Vetor\ \min = menor(Grade)\\
&Vetor\ P0 = Vetor(\ \min .x, \min .y, \min .x);\\
&Vetor\ S = Vetor(\ \max .x, \min .y, \min .z) - P0;\\
&Vetor\ T = Vetor(\ \min .x, \max .y, \min .z) - P0;\\
&Vetor\ U = Vetor(\ \min .x, \min .y, \max .z) - P0;
\end{aligned}
\qquad (1)
$$

Nas operações mostradas em (1), inicialmente são encontradas as coordenadas máximas e mínimas da grade gerada em volta da malha. Após isso, é criado um vetor com os valores mínimos da grade, o qual foi atribuído para a variável *P*0. Com o valor de *P*0 é gerado o vetor *S,* que corresponde ao valor máximo da grade no eixo *x* e os valores mínimos nos eixos *y* e *z*, subtraindo-se os valores do vetor *P*0. O vetor *T* é gerado com os valores mínimos no eixo *x* e *z* e o valor máximo no eixo *y* subtraindo-se os valores do vetor *P*0, e o vetor *U* possui os valores mínimos nos eixos *x* e *y* e o valor máximo no eixo *z*, subtraindo-se os valores de *P*0.

Com os vetores *S*, *T* e *U,* um vértice *P* qualquer da malha é registrado no sistema de coordenadas local determinando seus interpolantes trilineares, como mostrado em (2) (PARENT, 2002).

$$
\begin{aligned}
s &= (TxU).(P - P0)/((TxU).S)\\
t &= (UxS).(P - P0)/((UxS).T)\\
u &= (TxU).(P - P0)/((SxT).U)
\end{aligned}
\qquad (2)
$$

Na Figura 1 são mostrados os três vetores (*S*, *T*, *U*) com o ponto registrado e seus interpolantes.

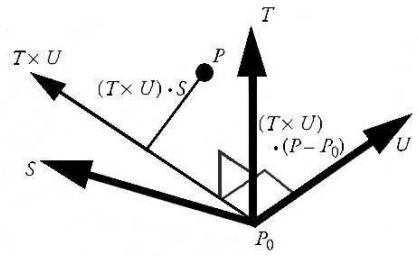


Figura 1. Vetores *S*, *T* e *U* com um ponto registrado no sistema de coordenadas.
Fonte: (PARENT, 2002).

Nas equações em (2), o produto de dois vetores gera um terceiro vetor que é ortogonal aos dois primeiros. O denominador normaliza o valor que está sendo computado. Dada as coordenadas locais (*s*, *t*, *u*) de um ponto e a grade de coordenadas locais não modificada, gerada pelos vetores *S*, *T* e *U*, uma posição de um ponto pode ser

reconstruída em espaço global, simplesmente movendo na direção dos eixos de coordenadas local. A escala que reconstitui o ponto no espaço global é mostrada em (3).

$$P = P0 + s.S + t.T + u.U \quad (3)$$

Para facilitar a modificação do sistema de coordenada local, uma grade de pontos de controle é criada no paralelepípedo definido pelos eixos *S*, *T* e *U*. Pode haver um número desigual de pontos nas três direções. Na Figura 2 são mostrados os eixos *S*, *T*, *U* com os pontos de controle adicionados (PARENT, 2002).

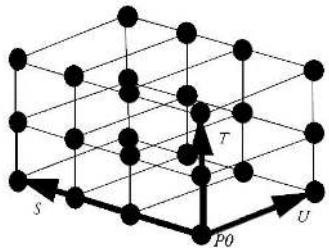


Figura 2. Pontos de controle no paralelepípedo criado com os eixos *S*, *T* e *U.*
Fonte: (PARENT, 2002).

Se existem n_S pontos na direção *S*, n_t pontos na direção *T* e n_u pontos na direção *U*, os pontos de controle são localizados de acordo com a Equação (4).

$$P_{ijk} = P0 + \frac{i}{l}.S + \frac{j}{m}.T + \frac{k}{n}.U \quad (4)$$

Na Equação (4), *i*, *j* e *k* representam a quantidade de pontos de controle que existem em cada eixo *S*, *T* e *U*, respectivamente, e *l*, *m* e *n* são contadores que vão do intervalo de *0* até a quantidade de pontos de controle em cada eixo. Esses pontos *i*, *j* e *k* estão igualmente espalhados pelo paralelepípedo gerado pelos eixos *S*, *T* e *U*. A deformação é gerada pela movimentação dos pontos de controle de sua posição inicial.

A função que realiza a deformação é uma função trivariada da interpolação de *Bézier*, a qual foi utilizada por tender a fazer uma curva no modelo das malhas, diminuindo, assim, a distorção gerada no modelo. A posição deformada do ponto p_{stu} é determinada pela utilização das coordenadas locais *s*, *t*, *u*, definidas pelas equações em (2), e pela função de interpolação de *Bézier* apresentada na Equação (5). Nessa equação, $P(s,t,u)$ representa a coordenada global do ponto deformado e P_{ijk} representa as coordenadas

globais dos pontos de controle (VARGAS, 2008).

$$P(s,t,u)=\sum_{i=0}^{l}\binom{l}{i}(1-s)^{l-i}s^{i}.\left(\sum_{j=0}^{m}\binom{m}{j}(1-t)^{m-j}t^{j}.\left(\sum_{k=0}^{n}\binom{n}{k}(1-u)^{n-k}u^{k}P_{ijk}\right)\right) \quad (5)$$

Em (5), *i*, *j* e *k* são contadores que começam em 0 e vão até *l*, *m* e *n,* que são as quantidades de pontos de controle que existem em cada vetor *S*, *T* e *U*.

Essa abordagem permite que o método se torne versátil e a deformação seja definida independente da representação de geometria, como mostra a Figura 3.

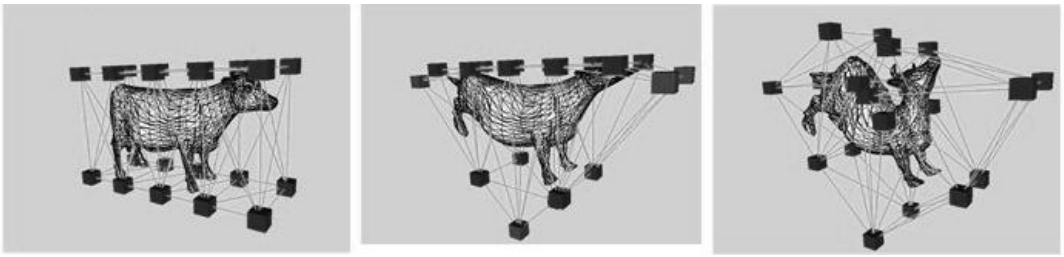


Figura 3. Exemplo de objeto deformado utilizando o método FFD.
Fonte: (HIROTA; MAHESHWARI; LIN, 1999).

2.2 Algoritmos usados para construção de modelos tridimensionais a partir de imagens de Raio-X.

O trabalho de Gunay, Shim e Shimada (2007) foi a fonte de informação para a idealização e desenvolvimento do trabalho apresentado neste artigo. Sua proposta é a geração da geometria tridimensional de um osso a partir de imagens de Raio-X, utilizando a técnica de deformação aplicada a um modelo 3D até que ele corresponda à imagem de Raio-X, e para isso usa os seguintes algoritmos:

- Algoritmo SQP (*Sequential Quadratic Programming*): usado para obter a diferença entre a entrada de imagens de Raio-X e do modelo 3D (GUNAY; SHIM; SHIMADA, 2007);
- *Marching Cube Algorithm*: usado para extrair dados volumétricos da superfície do osso (LORENSEN; CLINE, 1987);
- Métodos de *thresholding*: utilizado para correção do efeito *heel* em imagens radiográficas (OLIVETE; RODRIGUES; NASCIMENTO, 2004);
- Algoritmo FFD (Sederberg's *Free-form Deformation*): utilizado para deformar objetos 3D (SEDERBERG; PARRY, 1986).

3. METODOLOGIA APLICADA À DEFORMAÇÃO DE OSSOS 3D

A técnica adotada neste trabalho para obtenção de um objeto 3D do osso da tíbia de um ser humano baseia-se na utilização de um modelo 3D de osso clinicamente normal, o qual será deformado à anatomia óssea correspondente às imagens de Raios-X (faces frontal, laterais esquerda e direita) da tíbia.

Para tornar possível a técnica proposta foi necessário digitalizar e pré-processar imagens de Raios-X e manipular um modelo tridimensional do osso. A metodologia completa aplicada neste trabalho é mostrada na Figura 4.

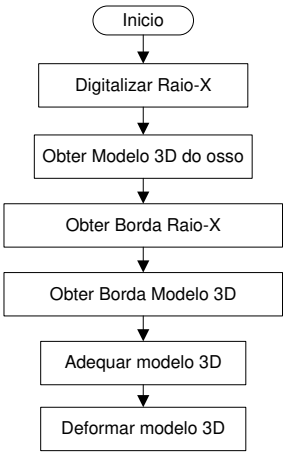


Figura 4. Metodologia aplicada neste trabalho.

3.1 Segmentação da anatomia do osso

Para a realização deste trabalho foram digitalizadas imagens de radiografias do osso da tíbia (do próprio autor), conforme o exemplo mostrado na Figura 5.



Figura 5. Exemplo de imagem de Raio-X digitalizada do osso da tíbia.

Após a digitalização das imagens de Raios-X utiliza-se o algoritmo de detecção de bordas, *Canny*, conforme apresentado na Figura 6.

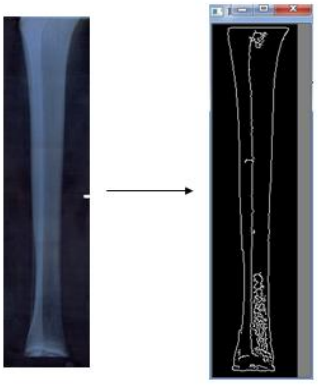


Figura 6. Resultado do algoritmo de extração de bordas *Canny*.

O detector de bordas *Canny* possui três objetivos básicos: baixa taxa de erro (todas as bordas devem ser encontradas), os pontos da borda devem ser bem localizados (as bordas localizadas devem ser próximas das bordas reais) e resposta única para os pontos de uma borda (o operador deve retornar apenas um ponto para cada ponto sobre a borda). Com a aplicação do método *Canny* foi possível representar essas ideias matematicamente e, numa primeira etapa, a imagem foi convolvida por uma função Gaussiana e, em seguida, foram determinadas a magnitude e a direção, conforme indicado pelas equações em (6):

$$
\begin{aligned}
g(x,y) &= f(x,y) * *G(x,y) \\
M(x,y) &= \sqrt{g_x^2 + g_y^2} \\
\alpha(x,y) &= \tan^{-1}\left(\frac{g_y}{g_x}\right) \\
g_i &= \frac{\partial g(x,y)}{\partial i}
\end{aligned}
\tag{6}
$$

Os valores de g_i podem ser obtidos a partir dos operadores de *Prewitt* ou de *Sobel*. $M(x,y)$ é uma imagem que contém a informação da magnitude em cada pixel e $\alpha(x,y)$ é uma imagem que contém a direção da normal à borda para cada pixel. A ideia consiste em verificar, para cada pixel p, se pelo menos um dos vizinhos de p possui a mesma direção que p, se sim, marca o pixel com $M(x,y)$ (as coordenadas de p), senão marca a imagem de saída com 0 (MARENGONI; STRINGHINI, 2009).

Finalmente utiliza-se um operador de corte para reduzir pontos de borda falsos. O

operador de corte, nesse caso, é baseado em histerese. O operador de corte baseado em histerese possui dois valores, um valor chamado de alto e outro chamado de baixo. São feitas as operações de corte usando esses valores e são obtidas duas imagens binárias, as quais são subtraídas uma da outra para obter a imagem final $(\mathrm{Img}_{baixa} - \mathrm{Img}_{alta})$.

Além das bordas há a necessidade de extração somente das bordas mais externas, das imagens de Raio-X, neste trabalho utiliza-se o algoritmo *Contour Following* ou "algoritmo do ceguinho". O algoritmo basicamente extrai o contorno externo de objetos, e é aplicado às imagens segmentadas. Na Figura 7 é mostrada a ideia básica do algoritmo *Contour Following*.

Segundo Costa e Cesar Junior (2001), o algoritmo *Contour Following* funciona da seguinte maneira: um ponto inicial pertencente ao contorno externo do objeto deve ser encontrado e selecionado, assumindo que os pixels do objeto são pretos e que os pixels do fundo são brancos, uma abordagem simples como passo inicial é a busca linha após linha (isto é, da esquerda para a direita, de cima para baixo, como na Figura 7 (a)), até que o primeiro pixel preto seja encontrado.

Depois de identificar o pixel inicial, o algoritmo percorre o objeto até o pixel inicial ser revisitado, indicando a conclusão da tarefa (Figura 7 (b)). Para obter o contorno parametrizado, as coordenadas dos pixels no caminho seguido ao longo do contorno externo devem ser armazenadas (COSTA; CESAR JUNIOR, 2001).

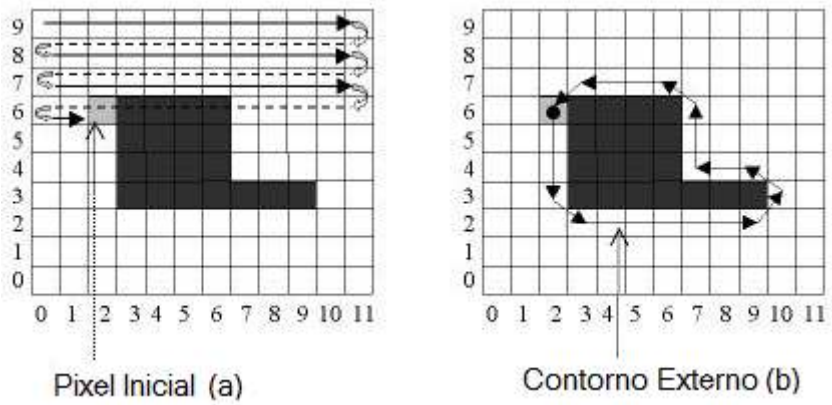


Figura 7. Esquemas ilustrando o algoritmo *Contour Following*.

Fonte: (COSTA; CESAR JUNIOR, 2001).

Uma vez que o pixel inicial tenha sido encontrado, o algoritmo deve decidir qual é o pixel de contorno seguinte (COSTA; CESAR JUNIOR, 2001). Para encontrar o próximo pixel da borda é feita a análise dos vizinhos do pixel inicial.

Um pixel inicial *p* na coordenada (*x*, *y*) tem quatro vizinhos horizontais e verticais cujas coordenadas são dadas por: (*x*+1,*y*), (*x*-1,*y*), (*x*,*y*+1), (*x*,*y*-1) e quatro vizinhos diagonais, com coordenadas: (*x*+1,*y*+1), (*x*+1,*y*-1), (*x*-1,*y*+1), (*x*-1,*y*-1) (GONZALEZ; WOODS, 2000).

Na Figura 8 são mostrados os vizinhos horizontais, verticais e diagonais de um pixel *p*.

(*x*-1,*y*-1)	(*x*,*y*-1)	(*x*+1,*y*-1)
(*x*-1,*y*)	***p*(*x*,*y*)**	(*x*+1,*y*)
(*x*-1,*y*+1)	(*x*,*y*+1)	(*x*+1,*y*+1)

Figura 8. Vizinhos de um pixel *p*.

Assume-se que os quatro vizinhos horizontais e verticais do pixel inicial *p* estão rotulados com direções E (Esquerda), D (Direita), C (Cima) e B (Baixo), conforme mostrado na Figura 9.

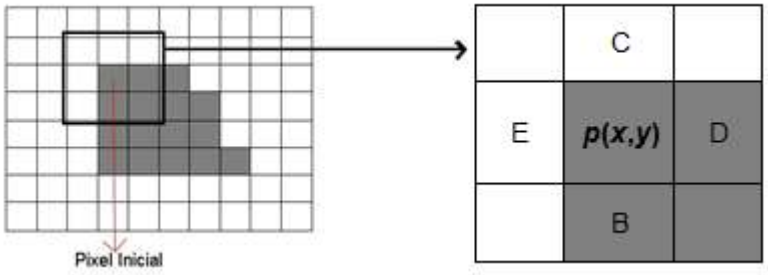


Figura 9. Vizinhos do pixel inicial *p*.

Define-se uma direção atual que é iniciada com D (Direita), pois a leitura para se encontrar o pixel inicial se fez da esquerda para direita. Definido o pixel inicial (posição central da matriz) e a direção atual (D) é possível iniciar a análise da matriz e encontrar o vizinho que será o próximo pixel da borda.

A análise para encontrar o próximo pixel é feita no sentido horário e sempre iniciada de algum vizinho diagonal.

Para encontrar a posição do vizinho diagonal e iniciar a busca, deve-se seguir o

sentido oposto da direção atual D (Direita) que é E (Esquerda) e a próxima diagonal lida no sentido horário será a posição inicial da análise, conforme mostrado na Figura 10.

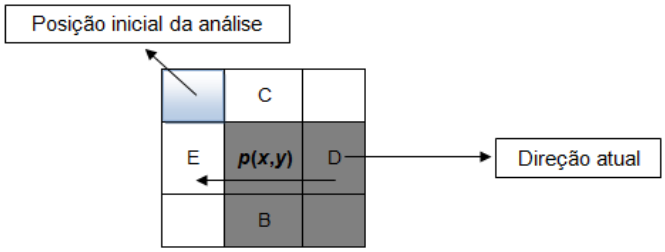


Figura 10. Posição inicial de análise da matriz.

A partir da posição inicial de análise, o método faz uma busca, no sentido horário, na matriz até encontrar o primeiro pixel da cor preta, assumindo esse, como próximo pixel da borda e novo pixel inicial conforme mostrado na Figura 11 e na Figura 12.

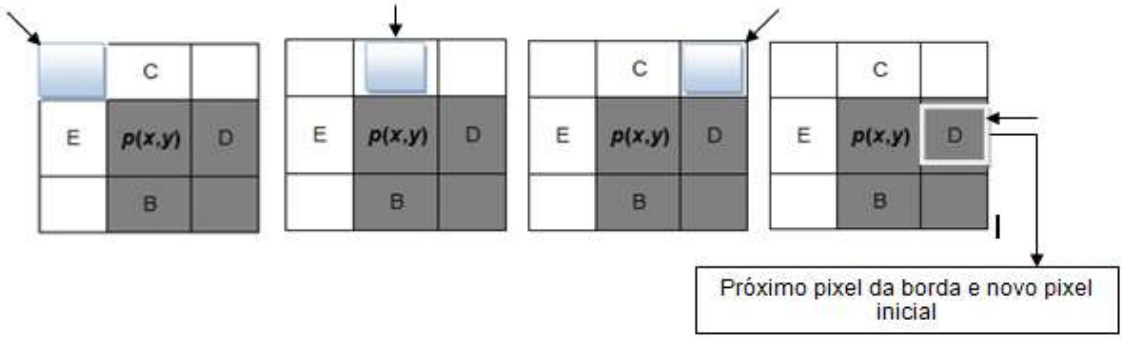


Figura 11. Busca próximo ponto da borda.

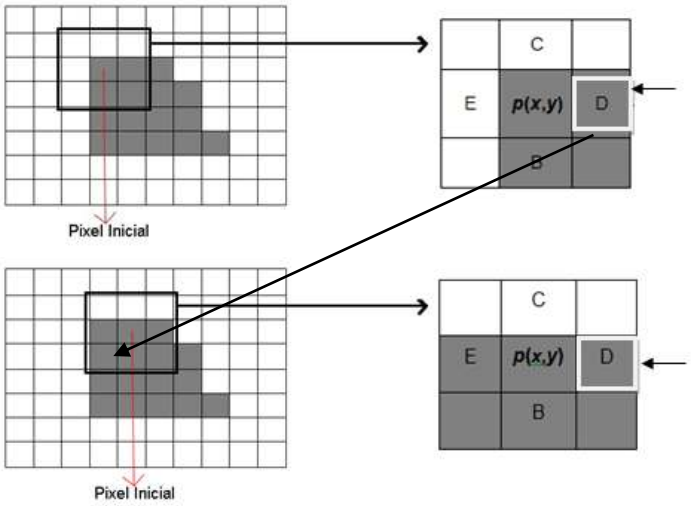


Figura 12. Comportamento do Algoritmo *Contour Following.*

O algoritmo segue todos os passos até o primeiro pixel armazenado da borda ser revisitado, conforme mostrado na Figura 13.

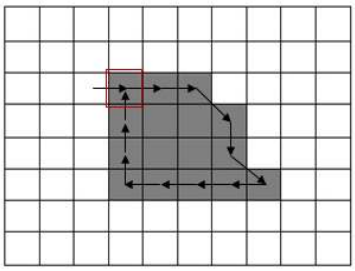


Figura 13. A borda do algoritmo *Contour Following* após percorrer todo objeto desejado.

Um dos problemas do algoritmo *Contour Following,* que pode ser notado, é que se a imagem contiver mais de um objeto, apenas em um deles será calculado o contorno. O outro problema é se o objeto contiver um buraco, o algoritmo não percorrerá toda a imagem, não sendo assim eficiente na determinação do contorno do objeto (NISHIYAMA JUNIOR; CESAR JUNIOR, 2003).

3.2 Modelo tridimensional do osso

Uma forma comum de representar objetos tridimensionais na Computação Gráfica é através de lista de vértices e lista de arestas e faces poligonais. Onde uma malha de polígonos representa uma superfície composta por faces planas, que podem ser triângulos (preferencialmente) ou quadrados, e a reunião desses elementos geométricos produz uma estrutura semelhante a uma "malha" que reproduz a superfície de um objeto, e assim permite o uso e manipulação do modelo 3D. A maioria dos objetos, desde um simples triângulo até o complexo modelo de um avião, é modelada desta maneira (MANSOUR; COHEN, 2006).

Para este trabalho foi modelado e criado um modelo tridimensional do osso da tíbia de um ser-humano, conforme mostrado na Figura 14.



Figura 14. Modelo tridimensional do osso da tíbia.

Os dados geométricos do modelo tridimensional são armazenados no formato .obj, que utiliza três tabelas: uma de vértices (*Vetex Table*), uma de arestas (*Edge Table*), e uma tabela de polígonos (*Polygon-Surface Table*), conforme apresentado na Figura 15. As coordenadas de cada vértice no objeto são armazenadas na tabela de vértices, a tabela de arestas contém ponteiros para a tabela de vértices de forma a identificar, para cada aresta, os vértices extremos. A tabela de polígonos contém ponteiros para a tabela de arestas para identificar as arestas que definem a fronteira de cada polígono (face poligonal). Na Figura 15 é ilustrada a representação para dois polígonos adjacentes na face de um objeto (TRAINA; OLIVEIRA, 2005).

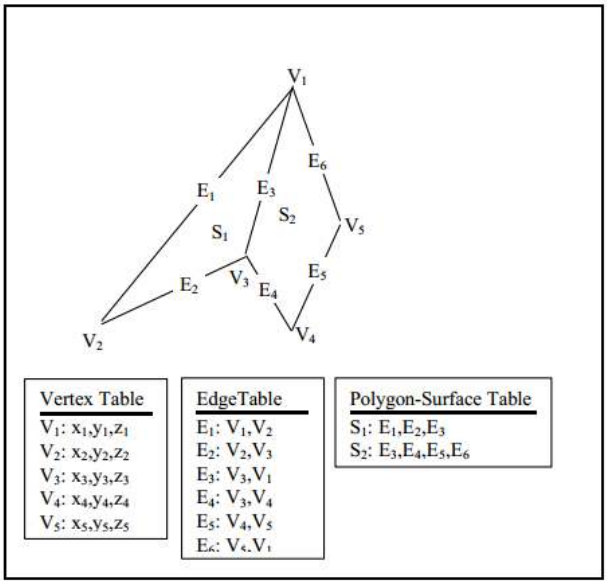


Figura 15. Esquema ilustrando um objeto poligonal.

Fonte: (TRAINA; OLIVEIRA, 2005).

As informações do modelo tridimensional (.obj), podem ser lidas e armazenadas em memória, logo é possível processar, visualizar e manipular dados (listas de vértices, arestas e faces) do objeto 3D.

3.3 Adequação dos Modelos

É necessário adequar o modelo tridimensional do osso, ou seja, deixá-lo do mesmo tamanho do Raio-X para posteriormente deformá-lo. Para adequar o modelo do osso, foi proposto neste trabalho, a utilização da transformação geométrica de escala, que consiste em multiplicar um fator de escala por cada vértice (*x*,*y*,*z*) do objeto tridimensional e assim o modelo terá seu tamanho aumentado ou diminuído.

Quando o objetivo da adequação é aumentar, deve-se aplicar um fator de escala maior que 1.0 e para diminuir deve-se aplicar um valor de escala entre 0.0 e 1.0 (MANSSOUR; COHEN, 2006).

O fator de escala é obtido através da divisão da altura do Raio-X (*ymax*-*ymin*) pela altura do modelo 3D (*ymax*-*ymin*), conforme ilustrado na Figura 16. Com a multiplicação do fator de escala por todos os vértices (*x*,*y*,*z*) do modelo, se obtém o modelo 3D do osso com a mesma altura da imagem de Raio-X conforme ilustrado na Figura 17.

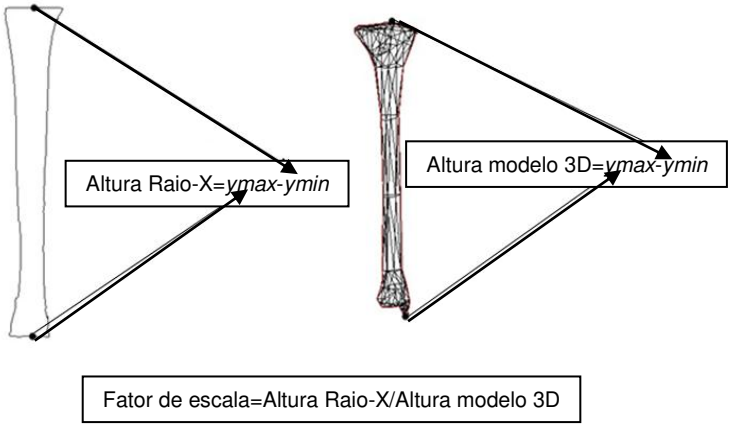


Figura 16. Obtenção de fator de escala.

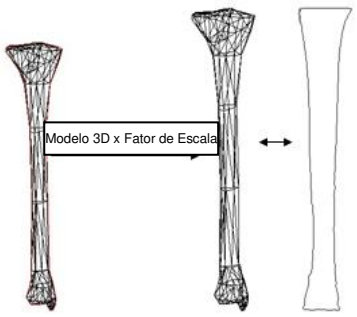


Figura 17. Resultado obtido ao aplicar fator de escala.

3.4 Deformação do Objeto 3D

O algoritmo de deformação proposto neste trabalho deforma inicialmente os pontos da borda e em seguida adequa o modelo tridimensional, deformando os vértices ligados aos pontos da borda. Na Figura 18 é mostrada a ideia básica do algoritmo.

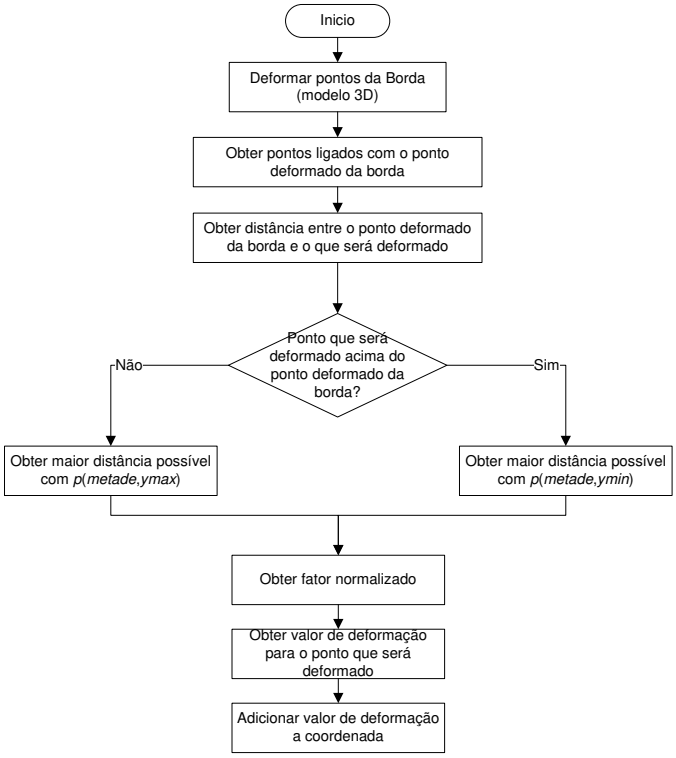


Figura 18. Fluxograma do algoritmo de deformação proposto.

Com o algoritmo *Contour Following (Seção 2.3)* é possível obter os valores de todos os pixels da borda da imagem do osso, conforme a Figura 19.

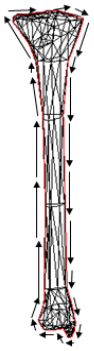


Figura 19. *Contour Following* aplicado a imagem do osso.

A deformação é iniciada manipulando os pontos do modelo 3D que estão na região de contorno. Para obter os vértices da região da borda do modelo 3D realiza-se uma busca na tabela de vértices no modelo (.obj), utilizando os pixels do contorno previamente armazenados e obtidos no passo anterior com o algoritmo *Contour Following.* Para isso verifica-se a existência do pixel na tabela de vértices do modelo 3D, e ao encontrar uma coordenada correspondente, esta é armazenada. Na Figura 20 estão representados os vértices da região da borda do modelo tridimensional.

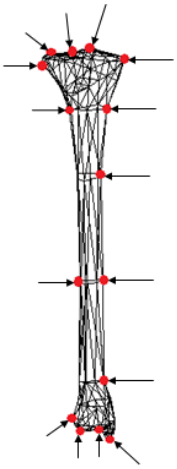


Figura 20. Vértices da região da borda do modelo tridimensional.

Para deformar todos os pontos da região da borda do modelo 3D, a coordenada *x* dos vértices (*x*,*y*,*z*) da borda é adequada ao valor de *x* do contorno do Raio-X, isso se os valores de *y* (vértice da borda e Raio-X) forem equivalentes, conforme ilustrado na Figura 21. Para todos os vértices da borda deformados deve-se realizar uma busca e encontrar o valor de *x* correspondente (mesmo *y*) na borda do Raio-X.

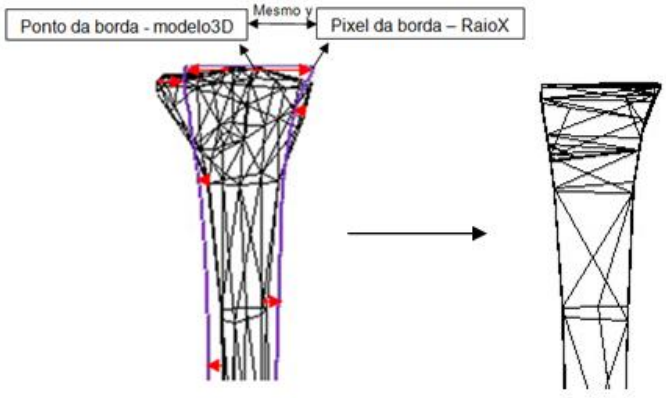


Figura 21. Resultado da deformação das bordas do modelo tridimensional.

Ao se deformar um ponto da borda (pd) do modelo 3D (composto por faces poligonais) os pontos ligados (pl1) e (pl2) também devem ser deformados proporcionalmente, conforme ilustrado na Figura 22.

É necessário encontrar os pontos conectados (pl1 e pl2) com o deformado (pd), para isso é feita uma análise nas informações da tabela de faces do modelo 3D. Pontos conectados e próximos ao deformado da borda, sofrem maior deformação do que os mais distantes.

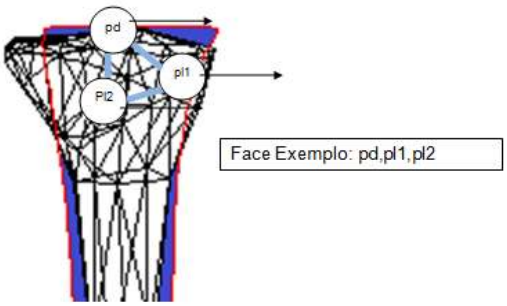


Figura 22. Pontos conectados com o deformado da borda.

Para se manipular os pontos ligados (pls) ao deformado da borda (pd) deve-se encontrar um valor de deformação proporcional e adicionar esse valor nos pontos (pls). Para isso são necessários alguns parâmetros como: regiões de interesse, distância entre o ponto deformado (pd) e o que será deformado (pl), maior distância possível entre pontos na região selecionada, fator de multiplicação normalizado e intensidade da deformação no ponto da borda.

Os pontos ligados (pls) ao deformado (pd) são manipulados de acordo com a região que se localizam. O modelo tridimensional é subdividido em cinco regiões para deformação, sendo R1, R2 e R3 definidas pela divisão da altura do modelo por três e M1, M2 definidas pela divisão da largura por dois, conforme mostrado na Figura 23.

São deformados somente os (pls) que se encontram na mesma região (R1, M2) do (pd) conforme mostrado na Figura 23.

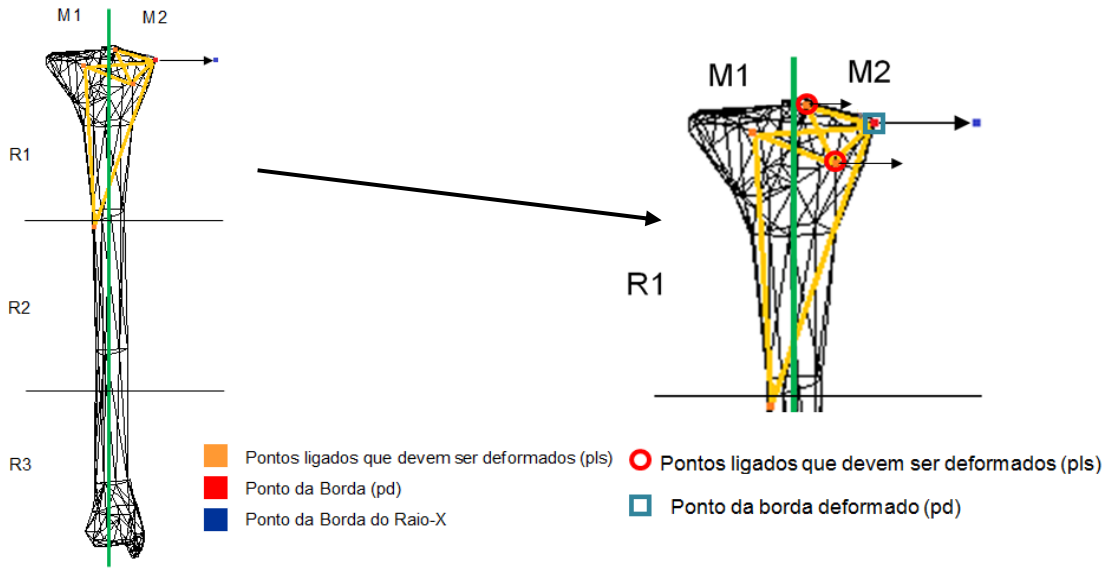


Figura 23. Esquema ilustrando pontos que devem ser deformados.

Outro parâmetro importante para encontrar a deformação proporcional é obter a distância entre o ponto da borda deformado (pd) e os pontos ligados que devem ser deformado (pl), como mostrado na Figura 24. Para isso é utilizada a distância euclidiana demonstrada na Equação (6).

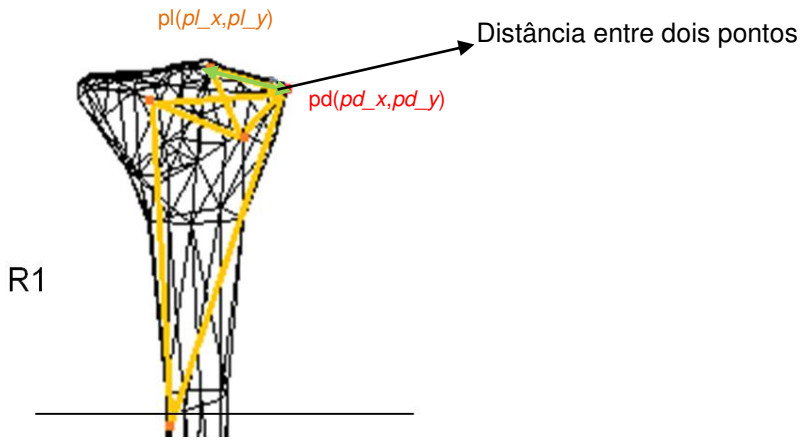


Figura 24. Distância entre dois pontos.

$$\text{Distância} = \sqrt{(x2 - x1)^{2} + (y2 - y1)^{2}} \qquad (6)$$

Para obter a maior distância possível do ponto da borda que foi deformado (pd) utiliza-se a Equação (6), deve-se criar um ponto (pmax) onde *x* assume o valor da metade (*xmetade*) do modelo tridimensional e *y* assume o menor valor de *y* do objeto tridimensional (*ymin*), e assim obter a distância entre (*pmax*) e (pd) conforme mostrado na Figura 25.

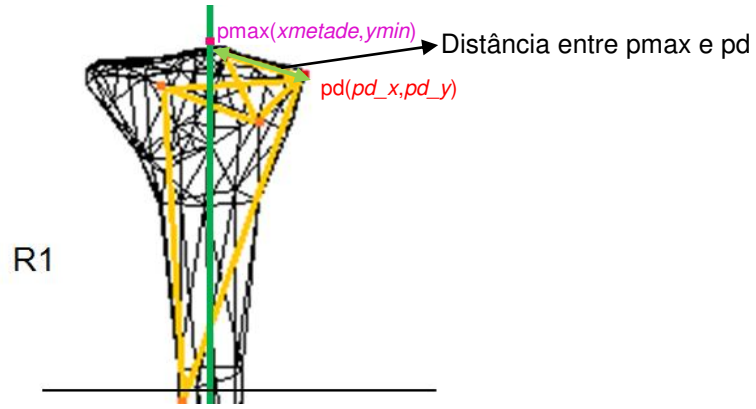


Figura 25. Distância máxima entre dois pontos.

O algoritmo, após encontrar a distância entre o ponto da borda (pd) deformado e os pontos ligado (pl), encontra a maior distância possível, aplicando a (7) e obtendo-se assim um fator normalizado.

$$FatorNormalizado = 1 - \left(\frac{\sqrt{(pd_x - pl_x)^2 + (pd_y - pl_y)^2}}{\sqrt{(pd_x - xmetade)^2 + (pd_y - y\min)^2}} \right) \quad (7)$$

Ao obter todos os parâmetros (regiões de interesse, distância entre o ponto deformado (pd) e o que será deformado (pl), maior distância possível entre pontos, fator normalizado) usa-se a Equação (8) para definir o valor da deformação proporcional no ponto.

$$\text{Deformação no Ponto} = (x_RaioX - pd_x) * FatorNormalizado \quad (8)$$

Assim, tem-se o valor da deformação proporcional, deve-se então adicionar o valor da deformação em todas as coordenadas (*x*,*y*,*z*) do vértice ligado ao deformado da borda.

Há algumas semelhanças entre o *Free-Form-Deformation* e o algoritmo de deformação proposto neste trabalho, em ambos são definidas regiões onde a deformação nos pontos é feita proporcionalmente.

4. EXPERIMENTOS E RESULTADOS

Foram digitalizadas imagens de Raio-X de um osso da tíbia de um ser humano tomadas de diferentes direções (frontal,

laterais esquerda e direita), para assim obter-se dados das imagens (bordas).

É feita a obtenção da malha tridimensional do osso da tíbia do ser humano e a deformação até corresponder ao Raio-X, e com isso é gerado um objeto do tipo .fbx. FBX é uma estrutura aberta para a transferência de dados 3D, que cria um alto grau de interoperabilidade em programas de manipulação tridimensional, para visualizar e editar os objetos, luzes, câmeras e materiais.

O visualizador para objetos tridimensionais do osso da tíbia foi implementado no Visual Studio 2010 usando a linguagem de programação C# e o framework de desenvolvimento XNA da Microsoft, para otimizar o processamento dos dados e adicionar textura, possibilitando uma melhor visualização do osso 3D. Na Figura 26 é ilustrado um osso em 3D sendo visualizado após aplicação da metodologia proposta.



Figura 26. Resultado obtido após aplicação da metodologia proposta.

5. CONCLUSÕES

Ao se manipular imagens na área médica, há uma série de fatores relevantes que devem ser levados em conta onde imagens são importantes para formular o diagnóstico correto. Imagens tornam-se assim decisivas no tratamento de algumas doenças e seus dados e informações precisam de uma margem segura de precisão.

Os resultados obtidos com a metodologia proposta neste trabalho demonstra que há a possibilidade de representar tridimensionalmente o osso humano, obtido a partir de imagens de Raio-X, e assim gerar um visão realística do osso do ser humano.

Ao se manipular imagens de Raio-X e obtenção de seus dados para geração de um modelo 3D alguns fatores são relevantes e as

imagens devem ser digitalizadas sem perda de informação.

Imagens radiológicas possuem alguns parâmetros importantes que devem ser levados em conta e tratados como: ruídos, diferentes contrastes, tons de cinza não uniformes, níveis de intensidade no filme variável, estruturas que não são de interesse.

Visualizar validade dos resultados em modelos tridimensionais com grande quantidade de dados é uma tarefa complexa e necessita da realização de testes.

REFERÊNCIAS

BERTOLO, A.L. **A física do diagnostico com raio-x**. Disponível em: http://www.bertolo.pro.br/Biofisica/Apostila_1/RAIO-X/RAIO-X.swf Acesso em: 20 mar. 2012.

COSTA, L.F.; CESAR JUNIOR, R.M. **Shape Analysis and classification**. Theory and practic. [s.l]; [s.n.], 2001.

COSTA, T.E. **Equipamentos médico-hospitalares e o gerenciamento da manutenção**. Capacitação a distância, Brasília – DF, 2002.

GONZALEZ, C.R.; WOODS, E.R. **Processamento de imagens digitais**. São Paulo: Blucher, 2000.

GUNAY, M.; SHIM, M; SHIMADA, K. **Cost- and time-effective three-dimensional bone-shape reconstruction from X-ray images**. Pittsburgh: Department of Mechanical Engineering, Carnegie Mellon University, 2007.

HIROTA, G.; MAHESHWARI, R.; LIN, M.C. Fast volume-preserving free form deformation using multi-level optimization. In: SIGGRAPH: ACM Special Interest Group on Computer Graphics and Interactive Techniques, 1999, Michigan. **Anais**... ACM Digital Library, Michigan, 1999.

NISHIYAMA JUNIOR, M.Y.; CESAR JUNIOR, R.M. **Implementar o algoritmo do ceguinho e calcular a curvatura multiescala.** [Apostila da disciplina Análise e Reconhecimento de Formas: teoria e prática]. 2003.

LORENSEN, W.E.; CLINE, E.H. Marching cubes: a high resolution 3D surface construction algorithm. **Comput Graphics**, 1987. http://dx.doi.org/10.1145/37402.37422

MANSSOUR, H.I,; COHEN, M. Introdução à computação gráfica. **RITA**, v.XIII, n.2, 2006.

MARENGONI, M.; STRINGHINI D. Introdução a visão computacional usando openCV. **RITA**, v.XIII, n.1, 2009.

OLIVETE, J.C.; RODRIGUES, L.L.E.; NASCIMENTO, Z.M. A correção do efeito heel aplicada em imagens radiográficas da mão. In: PROGRAMAÇÃO DO IX CBIS, 2004, 9. **Anais...** Escola de Engenharia de São Carlos (EESC).

PARENT, R. **Computer animation:** algorithms and techniques. Berkeley: Morgan Kaufmann, 2002.

RAJAMANI, T.K. et al. Statistical deformable bone models for robust 3D surface extrapolation from sparse data. **Medical Image Analysis**, 2006.

SEDERBERG, T.W.; PARRY S.R. Free-form deformation of solid geometric models. In: In: ANNUAL CONFERENCE ON COMPUTER GRAPHICS AND INTERACTIVE TECHNIQUES - SIGGRAPH '86, 13. **Proceedings...** Dallas, Texas, 1986. http://dx.doi.org/10.1145/15922.15903

SOUZA, S.G, **Qual a importância do diagnóstico por imagem para o tratamento das doenças**. 2012. Disponível em: http://www.santalucia.com.br/index.php/qual-a-importancia-do-diagnostico-por-imagem-para-o-tratamento-das-doencas/. Acesso em: 16 dez. 2012.

TRAINA, M.J.A.; OLIVEIRA, F.C.M. **Representação de objetos tridimensionais** – modelos poligonais. São Carlos: Departamento de ciências de computação e estatística, 2005.

VARGAS, S.K. **Implementação de deformação de malhas no humanoide v-art**. Blumenau: Universidade Regional de Blumenau, Centro de Ciências Exatas, 2008.