

Medições em Imagens Médicas Tridimensionais utilizando VTK

Thiago Franco de Moraes¹, Tatiana Al-Chueyr Martins Rospendowski¹, Paulo Henrique Junqueira Amorim¹

¹Centro de Tecnologia da Informação Renato Archer(CTI)
Rod. Dom Pedro I, Km 146.6 – Campinas – SP – Brasil

{thiago.moraes,tatiana.martins,paulo.amorim}@cti.gov.br

Abstract. *This work proposes and compares different implementations of a tool for linear and angular measures in 3D medical models. Free-software tools, Python and VTK, were used in its development. The final result will be available in the new version of the free-software InVesalius.*

Resumo. *Neste trabalho são propostas e comparadas implementações distintas de uma ferramenta que permite a realização de medições lineares e angulares sobre modelos médicos 3D. Para seu desenvolvimento foram utilizadas ferramentas livres – Python e VTK, sendo que o resultado final será disponibilizado na nova versão do software livre InVesalius.*

1. Introdução

Cefalometria é a ciência responsável pela análise e medição de dimensões da cabeça humana. Amplamente empregada, principalmente na ortodontia, ela é utilizada tanto para diagnosticar quanto para avaliar resultados de um tratamento.

Tradicionalmente, a realização de uma análise cefalométrica ocorre por meio de uma radiografia bidimensional da face. Neste método, sobrepõe-se um papel de acetato sobre o filme radiográfico. Primeiramente, desenha-se sobre este papel o contorno da face do paciente, ressaltando algumas estruturas, tais como da mandíbula, maxila e dentes. Após esta etapa, determina-se um protocolo cefalométrico a ser seguido. Protocolos frequentemente escolhidos são: Ricketts, Steiner e Björk-Jarabak [ARAT *et al*, 2002]. Baseando-se nas diretrizes de tal protocolo, desenha-se sobre o papel os pontos e segmentos de retas, os quais serão a base para determinação de medidas lineares e angulares. No caso de diagnóstico, tais valores são comparados com os padrões de normalidade disponíveis na literatura. No caso de avaliação de resultado, tais valores são comparados com medidas prévias do próprio paciente. O grande limitante da cefalometria tradicional é ter como base dados exclusivamente bidimensionais.

Desde aproximadamente 1972, a Tomografia Computadorizada tem sido disseminada. Tal tecnologia possibilitou a obtenção de uma seqüência de imagens médicas bidimensionais que podem, posteriormente, ser reconstruídas em um único modelo tridimensional (3D).

Como conseqüência de tais avanços, hoje odontologistas e médicos podem utilizar softwares de reconstrução 3D para analisarem o caso clínico de seus pacientes. Dentre os softwares comerciais pode-se citar: Vitrea®, Analyze®, Mimics® e Dolphin Imaging®. Dentre os livres, há dois mais conhecidos: InVesalius® e Osirix®. Poucos oferecem recursos de medições que atendam as necessidades de ortodontistas.

Em 3D tem-se a vantagem de compreender com mais detalhes as estruturas anatômicas investigadas, sendo possível analisá-las sob diversos ângulos. O emprego de tais softwares é tido pela literatura como um recurso de grande acurácia e precisão [ADAMS *et al.*, 2004] [ACCORSI, 2007] .

Desde 2001 o CTI (Centro de Tecnologia de Informação Renato Archer), unidade do Ministério da Ciência e Tecnologia brasileiro, desenvolve o software InVesalius. A principal motivação da instituição é popularizar a tecnologia, melhorando com isso o atendimento em hospitais Brasileiros. Atualmente há duas versões disponíveis do programa, e todo conteúdo relacionado ao projeto é hospedado no Portal do Software Público [INVESALIUS, 2009].

Além de possibilitarem a reconstrução de imagens bidimensionais no formato DICOM, obtidas através de equipamentos de tomografia computadorizada ou ressonância magnética, ambas versões do InVesalius oferecem de medições 3D.

No InVesalius 1, lançado em 2003, há apenas o recurso de medida linear em volumes. O usuário não pode efetuar múltiplas medidas simultaneamente. Não é possível visualizar claramente onde está localizada a medida, dado que o segmento de reta que a representa se confunde com o modelo tridimensional estudado. O valor da medida não é exibido junto do volume.

A segunda versão do InVesalius, disponibilizada em novembro de 2007, não possui uma ferramenta para medidas 3D em sua estrutura, mas foi desenvolvido um módulo separado, denominado de InVesalius Craniometria 3D. Neste módulo, ocorreram progressos significativos em relação ao InVesalius 1. O usuário tem disponível além de medidas lineares, também medidas angulares. O módulo permite realizar múltiplas medidas simultaneamente. O problema de visualizar claramente onde estava localizada a medida (tanto o segmento de reta, quanto o valor propriamente dito) permaneceu, mas foram utilizados dois artifícios para contorná-lo: (1) o usuário pode aplicar transparência ao volume; (2) o valor da medida e seu segmento de reta são apresentados com cores, no próprio volume.

Dentre os maiores problemas desta ferramenta constam: (1) a dificuldade de ver onde está posicionada uma determinada medida, dado que o recurso de transparência dificultava sua localização; e (2) o fato de não estar acoplado ao InVesalius 2. Este módulo proporcionou aos odontologistas realizarem medidas com precisão semelhante ao do software comercial Vitrea® [Pinheiro, 2008].

O objetivo deste trabalho é comparar possíveis métodos de implementação da ferramenta de medidas 3D, sendo que o melhor resultado será apresentado no InVesalius 3, cujo lançamento está previsto para 2009. Ao longo do desenvolvimento do trabalho, a motivação foi implementar algo que mantivesse a precisão das medições das versões prévias do InVesalius, mas que viabilizasse melhor interatividade e usabilidade por parte do usuário. Para esta tarefa foi utilizada a linguagem Python e a biblioteca VTK (*Visualization Toolkit*) [VTK, 2009], ambos já empregados no desenvolvimento do InVesalius.

2. Fundamentação Teórica

2.1. Medições Lineares e Angulares em Espaço Tridimensional

Os fundamentos de geometria analítica [SANTOS, 2002] empregados neste trabalho são

apresentados a seguir.

Para calcular a distância entre dois pontos, foi usado o Teorema de Pitágoras. Onde a distância D entre os pontos A e B , localizados, respectivamente, nas coordenadas (X_a, Y_a, Z_a) e (X_b, Y_b, Z_b) , é dado pela equação i:

$$D = \sqrt{(x_b - x_a)^2 + (y_b - y_a)^2 + (z_b - z_a)^2} \quad (\text{i})$$

A fim de calcular o ângulo, dados três pontos marcados pelo usuário, foi utilizado o cálculo de produto interno vetorial. Dados dois vetores a e b , o produto interno deles é definido pela equação ii. Portanto, o ângulo Θ é expresso pela equação iii:

$$a \cdot b = |a| \cdot |b| \cdot \cos(\Theta) \quad (\text{ii})$$

$$\Theta = \arccos(a \cdot b \div |a| \cdot |b|) \quad (\text{iii})$$

2.2. Desenvolvendo Aplicativos Utilizando VTK

VTK é uma biblioteca *open-source* para computação gráfica 3D, processamento de imagens e visualização. É desenvolvida em C++, seguindo os princípios de orientação a objetos [VTK, 2009], e oferece suporte a várias linguagens, dentre elas Python - linguagem empregada no desenvolvimento do software InVesalius.

A biblioteca VTK segue a arquitetura de *pipelines*, onde a saída de um objeto é a entrada de um outro objeto. O *pipeline* pode conter *loops* e ramificações, sendo que ele pode ser composto pelos seguintes elementos:

Source: provê os dados iniciais, podendo ser lido de um arquivo ou ser criado a partir de classes do próprio VTK. Exemplos: *vtkSTLReader* e *vtkConeSource*.

Filter: modifica os dados trabalhados. É opcional. Exemplo: *vtkSmoothPolyDataFilter* e *vtkTriangleFilter*.

Mapper: Determina as geometrias que irão representar os dados e como será seu comportamento de interação com o usuário. Exemplo: *vtkPolyDataMapper*.

Actor: Controla a aparência (cor, visibilidade, tamanho, etc) do dado renderizado na tela. Exemplos: *vtkActor* e *vtkActor2D*.

Renderer: É o final do pipeline, sendo o momento em que o usuário visualiza o resultado do que processou previamente. Exemplo: *vtkRenderer*.

O *pipeline* pode representar tanto dados 3D quanto 2D, sendo que é possível limitar a interação do usuário com tal dado, através do *mapper* escolhido. É possível interagir com dados 3D utilizando recursos de rotação, translação e *zoom*. Dados 2D, a princípio, não são tratados com tanta liberdade, dependendo do modo que o *pipeline* é construído.

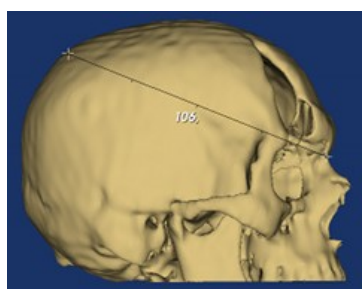
Por exemplo, é possível criar uma esfera utilizando a classe *vtkSphereSource*, que, por definição, é um dado tridimensional. Caso seja utilizado um *mapper* 3D, o usuário verá na tela uma esfera e poderá interagir em 3D com o volume. Entretanto, se o usuário utilizar um *mapper* 2D, o usuário verá na tela apenas um círculo.

3. Casos de Estudo

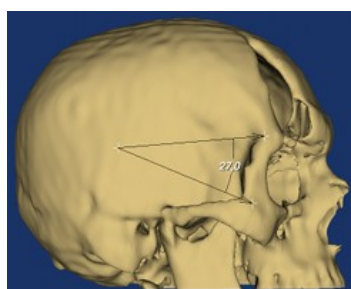
A seguir são apresentados os métodos implementados e avaliados.

3.1. Widgets para cálculo de distâncias

Este método utilizou duas classes do VTK “prontas” para a realização de medidas: *vtkDistanceWidget* e *vtkAngleWidget*. Para realizar medidas lineares a primeira delas foi utilizada, sendo que o usuário deveria definir dois pontos sobre o volume. Já para a realização de medidas angulares, a segunda classe foi empregada, e o usuário definir três pontos. Automaticamente eram apresentados os valores correspondentes a distância ou ao ângulo. A Figura 1 ilustra o resultado desta implementação.



(a) Medida linear



(b) Medida angular

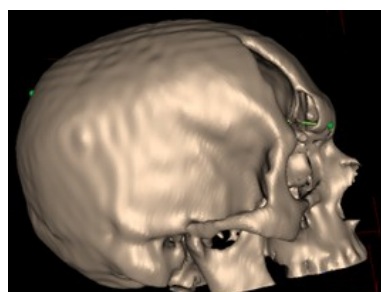
Figura 1 – Implementação de medidas 3D utilizando *widgets* do próprio VTK

3.2. Entes geométricos e texto utilizando *mapper 3D*

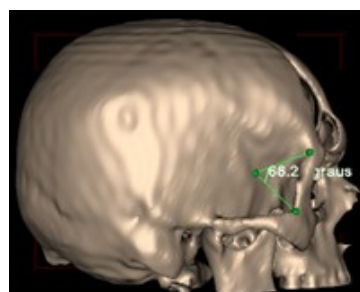
Considerando o exposto na sessão 2.2, quando planeja-se implementar o recurso de medidas 3D, sem a utilização de *widgets*, é intuitivo considerar o uso de *mappers 3D* não apenas no processamento do modelo médico, mas também nos entes geométricos utilizados para representação da medida.

Assim, como segunda implementação, foram empregados elementos geométricos primitivos oferecidos pelo VTK de acordo com seu propósito: (a) para demarcar pontos selecionados pelo usuário foram utilizadas esferas (*vtkSphereSource*); (b) para representar a distância linear, segmentos de reta (*vtkLineSource*); e (c) para representar o grau de abertura, arcos (*vtkArcSource*). Para interação com usuário, baseou-se no funcionamento dos *widgets* apresentados previamente. Para exibição do texto foi utilizada a classe *vtkVectorText*, por meio de um *mapper 3D*.

O resultado desta implementação pode ser analisado na Figura 2.



(a) Medida linear



(b) Medida angular

Figura 2 – Implementação de medições 3D com entes geométricos e *mapper 3D*

3.3. Entes geométricos e texto utilizando *mapper 2D*

A última implementação consistiu na reprodução da implementação anterior, 3.2, mas utilizando a classe *vtkTextSource* para representar o texto, um *mapper 2D* ao invés do 3D, tanto para estruturas geométricas, quanto para o texto. Apesar desta implementação aparentar ser uma contradição, por exibir em um mesmo *renderer* dados com interação 3D junto de dados mapeados em 2D, seus resultados foram bastante positivos, como apresentado na próxima sessão. O resultado é ilustrado na Figura 3.

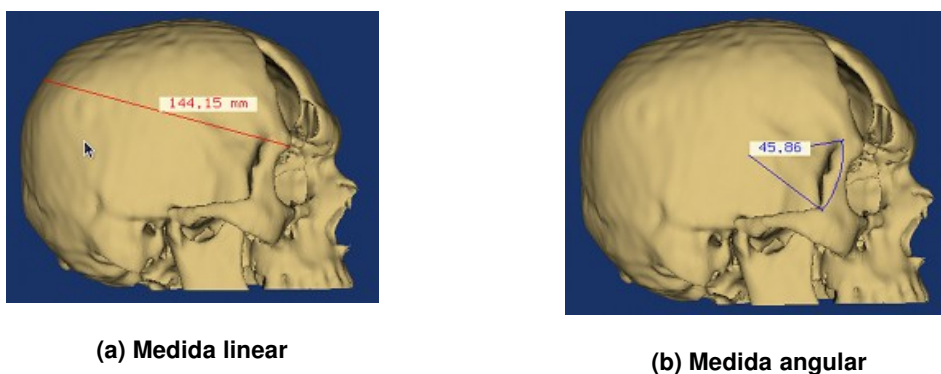


Figura 3 – Implementação de medições 3D com entes geométricos e *mapper 2D*

4. Resultados

Sob ótica do programador a primeira implementação, utilizando *widgets* do VTK é a mais simples, por não ser necessário realizar os cálculos apresentados na Fundamentação Teórica. Entretanto, tais *widgets* dão a liberdade de se selecionar qualquer ponto do espaço, independente de fazer parte da superfície do modelo analisado, o que dificulta a sua utilização por parte do usuário.

A segunda implementação proposta apresentou as mesmas falhas do módulo InVesalius Craniometria 3D: como os entes geométricos e os textos são mapeados tridimensionalmente, frequentemente eles são sobrepostos pelo volume de interesse, o que prejudica a visualização de onde estão posicionados e, muitas vezes, do valor de distância. Em relação a primeira implementação, houve grande avanço, pois as medidas foram tomadas apenas sobre a superfície existente.

A terceira implementação, por fim, uniu os aspectos positivos de ambas implementações anteriores. A utilização de *mappers 2D* possibilitou que nem os textos nem as entidades geométricas ficassem escondidas dentro do volume, assegurando legibilidade. Ao restringir a inserção de pontos apenas sobre a superfície, ela deu segurança ao usuário na marcação de pontos.

Ao compararmos o resultado da terceira implementação a ferramentas similares em softwares comerciais, tais como o Mimics®, constatou-se que seu funcionamento é equivalente, conforme ilustrado na Figura 4.

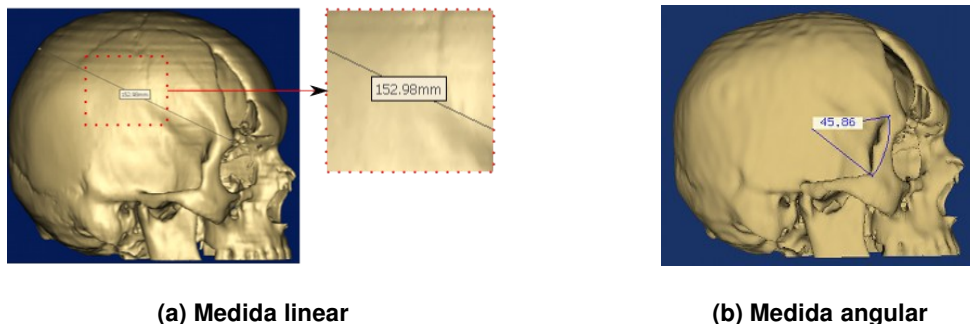


Figura 4 – Ferramenta de medições 3D disponível no software comercial Mimics®

5. Conclusão

Com este trabalho foram avaliados diversos modos de se implementar medições tridimensionais utilizando a biblioteca VTK. O método que se apresentou mais adequado para representar as medidas, considerando-se a legibilidade e interatividade por parte do usuário, foi realizar o cálculo da distância e ângulo manualmente e exibir o texto correspondente à medida utilizando um *mapper 2D*, em contraposição a *widgets* prontos e utilização de *mapper 3D* para representação dos dados. Os resultados desta implementação são comparáveis a ferramentas disponíveis em softwares comerciais da área, sendo que esta ferramenta estará disponível no software livre InVesalius 3.

Pretende-se, em trabalhos futuros, validar os resultados obtidos quantitativamente. Também há intenções de elaborar protocolos cefalométricos em três dimensões, em cooperação com cirurgiões, de modo que a ferramenta estimule o emprego de modelos 3D no cotidiano de ortodontistas, melhorando a precisão de suas análises.

Referências

- Accorsi, Mauricio Adriano de Olivério. (2007), “Comparação de grandezas cefalométricas obtidas por meio de telerradiografias e tomografias computadorizadas multislice em crânios secos humanos”, Universidade de São Paulo, 2007.
- Adams, Gregory L.; Gansky, Stuart A.; Miller, Arthur J.; Harrell, William E.; Hatcher, David C. (2004), “Comparison between traditional 2-dimensional cephalometry and a 3-dimensional approach on human dry skulls”. *American Journal of Orthodontics and Dentofacial Orthopedics*, Volume 126, Issue 4, Pages 397-409, 2004.
- Arat, Z. Mirzen, DDS, PhD; Rübendüz, Meliha, DDS, PhD; Akgül ,Ayça Arman, DDS, PhD. (2002) “The Displacement of Craniofacial Reference Landmarks During Puberty: A Comparison of Three Superimposition Methods”. *The Angle Orthodontist*: Vol. 73, No. 4, pp. 374–380, 2002.
- INVESALIUS (2009), <http://www.softwarpublico.gov.br>, acessado em 25/03/2009.
- Pinheiro, Debora (2008), *Revista FAPESP*, edição 148, junho 2008.
- Santos, Reginaldo J. (2002), “Geometria Analítica e Álgebra Linear” Parte 1.
- VTK (2009), <http://www.vtk.org>, acessado em 25/03/2009