

# InVesalius: Software Livre de Imagens Médicas

Paulo H. J. Amorim<sup>1</sup>, Thiago F. de Moraes<sup>1</sup>, Fábio de S. Azevedo<sup>1</sup>, Jorge V. L. da Silva<sup>1</sup>

<sup>1</sup>Centro de Tecnologia da Informação Renato Archer - CTI  
Rodovia Dom Pedro I (SP-65) Km 143,6 – Bairro Amarais  
CEP:13069-901 – Campinas - SP

{paulo.amorim,thiago.moraes,fabio.azevedo,jorge.silva}@cti.gov.br

**Abstract.** *This work presents InVesalius, an open-source imaging software. Initially, the historical context and motivation for its development are discussed. Then, some software internals and development aspects are commented, as well as InVesalius main features and applications. At the end, the results of a survey realized within the user community are presented.*

**Resumo.** *Este trabalho apresenta o software livre para tratamento de imagens médicas InVesalius. Inicialmente, o contexto histórico e as motivações para o desenvolvimento do software são discutidos. Em seguida, comentam-se o desenvolvimento e aspectos internos do funcionamento do InVesalius, bem como seus principais recursos e aplicações. Por fim, são apresentados os resultados de uma pesquisa com seus usuários no Brasil e no exterior.*

## 1. Introdução

Imagens médicas têm como objetivo fornecer aos clínicos a possibilidade de visualizar internamente o corpo humano de forma não invasiva, além de tornar o diagnóstico mais preciso. Apesar de a técnica de raio-X existir desde o início do século XX, as imagens médicas em três dimensões surgiram somente em 1972 com o invento da tomografia computadorizada (TC). [Yoo 2004].

Além do raio-X, empregado na radiografia convencional e na tomografia computadorizada, outra forma de obter imagens do interior do corpo é a ressonância magnética (RM). A RM emprega um alto campo magnético em conjunto com ondas de rádio-frequência. Atualmente, as técnicas citadas geram imagens em formato digital. Em geral, essas imagens seguem o padrão DICOM (*Digital Imaging and Communications in Medicine*). Uma imagem DICOM, além dos voxels, contém metainformações, como nome do paciente, informações do equipamento, posição da imagem em relação ao espaço (no caso da TC e RM) [Pianykh 2007].

De acordo com o padrão DICOM, cada imagem da modalidade TC ou RM representa um "corte" do corpo. A reconstrução 3D é feita por meio do empilhamento desses "cortes", interpolando os espaços entre eles, formando um volume.

A maioria dos fabricantes de equipamentos de TC e RM vendem separadamente os softwares que realizam a reconstrução 3D a um preço inacessível para a atual realidade da saúde pública brasileira. Mesmo já no início do século XXI, não se encontrava software livre ou gratuito que realizasse a reconstrução 3D. Hoje, embora já existam softwares

para esse fim que são gratuitos ou livres, eles ainda possuem algumas limitações. Alguns apresentam uma interface gráfica pouco amigável. Outros, apesar de serem software livre, só podem ser executados em plataforma específica.

O Centro de Tecnologia da Informação Renato Archer - CTI, há mais de uma década, já possui equipamentos de prototipagem rápida (impressoras 3D) e tem como um de seus objetivos a produção de modelos físicos para auxiliar profissionais de saúde em planejamento cirúrgico. Diante desse cenário, em 2001, iniciou-se, no CTI, o desenvolvimento do software InVesalius. Atualmente na terceira versão, esse software de visualização de imagens médicas e reconstrução 3D é multiplataforma e multi-idioma. Com foco em usabilidade, pode ser executado em computadores pessoais.

## 2. Desenvolvimento do InVesalius

O InVesalius é um software livre, cuja licença é a GNU General Public License versão 2. Isso permite que pesquisadores e desenvolvedores de universidades e centros de pesquisas de todo o mundo ajudem no desenvolvimento, e ainda apliquem seus temas de pesquisa e interesse neste desenvolvimento.

O desenvolvimento é centralizado com o uso da ferramenta de controle de versão Subversion [Subversion 2011]. Essa ferramenta, basicamente, permite o controle de alterações e seus autores. Emprega-se a política de que novas ferramentas e erros sejam antes documentados, através da ferramenta *Trac* [Trac 2011] e depois seja feito o desenvolvimento e isto é feito não apenas pelos desenvolvedores, mas também pelos usuários do software.

## 3. Aspectos internos e Funcionamento

O software é desenvolvido na linguagem de programação *Python*. Python é uma linguagem multi-paradigma, pois permite o uso dos paradigmas orientado a objeto, procedural e funcional. No caso do InVesalius, o paradigma orientado a objetos foi utilizado quase que totalmente. Apenas em certos casos os paradigmas funcional e procedural foram utilizados. Também são utilizadas muitas bibliotecas que serão mencionadas no decorrer do texto.

O início da operação do software se dá pelos dados de entrada. O usuário do software seleciona os arquivos de imagens médicas, que podem ser no formato DICOM ou Analyze. Hoje, o InVesalius dá suporte a arquivos de tomografia computadorizada (CT) e ressonância magnética (RM). Para o suporte ao formato DICOM [Pianykh 2007], é utilizada a biblioteca *GDCM* [Malaterre 2011], e a biblioteca *Nibabel* [NiBabel 2011] para arquivos Analyze. É importante mencionar que a biblioteca *GDCM* permitiu dar suporte a arquivos DICOM com alterações do padrão e também a arquivos compactados que vêm se tornando comuns devido ao seu menor tamanho.

As imagens de entrada são convertidas em uma matriz com três dimensões. Essa matriz é mantida numa estrutura *memmap* em arquivo em disco e mapeada em memória RAM por meio da chamada de sistema *mmap*, comum a vários sistemas operacionais. Assim, apenas os dados da matriz necessários são mantidos em memória. Quando outros dados forem necessários, eles são trazidos do arquivo em disco para a memória RAM. Essa estratégia permite que arquivos de exames médicos com grande quantidade de fatias

ocupem menos memória do que se todas as fatias fossem carregadas em memória. A estrutura *memmap* é provida pela biblioteca *Numpy* [Kinser 2009]. *Numpy* é uma biblioteca numérica para Python que provê matrizes e vetores e operações matemáticas sobre estes. Poderosa e de fácil uso, ela permite o acesso aos dados das matrizes usando índices ao invés de chamadas explícitas a funções ou métodos.

Após a abertura das imagens, o usuário tem acesso à visualização do exame e a várias ferramentas. É possível visualizar o exame percorrendo suas fatias, nas orientações axial, sagital e coronal e também a sua reconstrução tridimensional. É importante salientar que o uso de uma matriz de três dimensões facilitou o acesso às orientações citadas, bastando para isso acessá-la de acordo com a orientação através de índices.

Para a construção da interface gráfica (figura 1) foi utilizada a biblioteca multiplataforma *wxPython* [Rappin and Dunn 2006]. Por utilizar a biblioteca de interface gráfica padrão do sistema operacional em que está sendo executado, o InVesalius mantém a aparência de uma aplicação nativa, sendo necessários apenas pequenos ajustes.

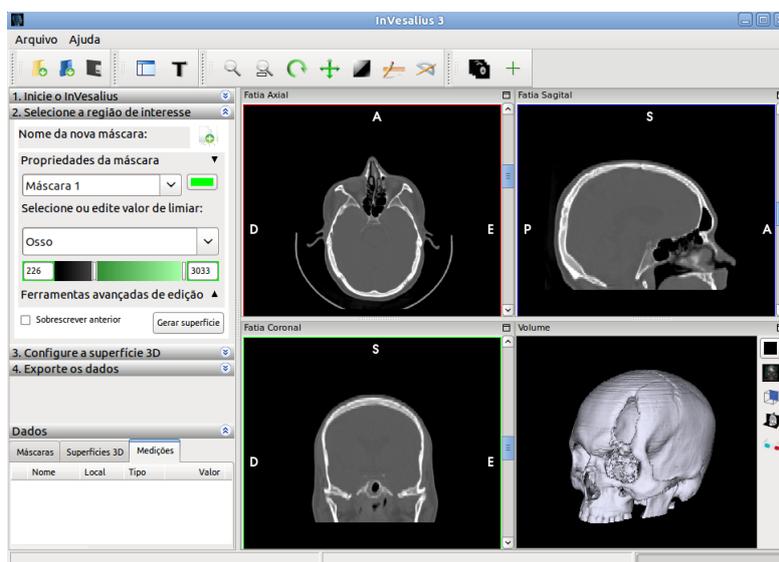


Figura 1. Software InVesalius

A biblioteca VTK (*Visualization ToolKit*) [Schroeder et al. 2004] foi utilizada para visualização, tanto 2D quanto 3D. Como os dados são armazenados em uma estrutura de matriz *memmap*, é necessário convertê-los para a estrutura VTK. Isso é feito apenas para as fatias do exame que estão sendo visualizadas no momento, ou seja, sob demanda. A biblioteca VTK também foi útil em outras funcionalidades que serão citadas no decorrer do texto.

## 4. Recursos

A seguir são apresentados os principais recursos do InVesalius na sua versão 3 e são citadas algumas de suas aplicações.

### 4.1. Segmentação de Imagens

Hoje, o InVesalius conta com duas técnicas de segmentação: *threshold* [González and Woods 2008] e manual. Na técnica de *threshold*, apenas os voxels

com valores entre um mínimo e um máximo são selecionados. A segmentação manual é uma adição à técnica por *threshold*. Com ela, é possível selecionar, interativamente, voxels de interesse do usuário não selecionados pela técnica de *threshold*, assim como eliminar da seleção voxels que não sejam de seu interesse. Adicionalmente, há ainda *presets* de *thresholds* para certas estruturas da anatomia humana cujos valores de voxels já são conhecidos, como por exemplo a estrutura óssea. O que é selecionado é demonstrado ao usuário por uma máscara colorida sobreposta às fatias do exame, como ilustra a figura 2. É possível criar várias máscaras, uma para cada segmentação. As máscaras também são armazenadas em uma matriz *memmap*.

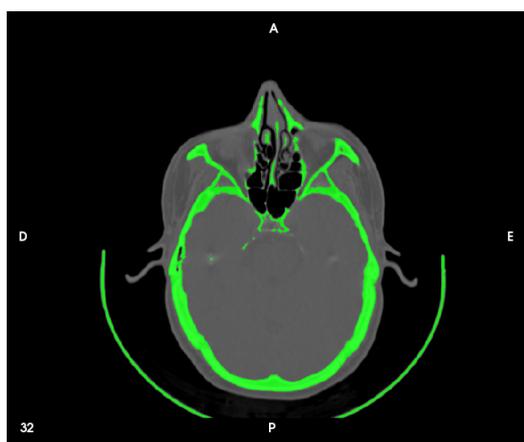


Figura 2. Imagem com *Threshold* para osso

#### 4.2. Superfície 3D

Após segmentar a região de interesse, o usuário tem a opção de reconstrução 3D. Isso é feito através do algoritmo de *Marching Cubes* [Lorenson and Cline 1987], utilizando a implementação disponível na biblioteca VTK. Baseado nos voxels do exame segmentado, o algoritmo gera uma superfície poligonal, constituída geralmente por triângulos. Essa superfície pode ser impressa fisicamente por meio de técnicas de prototipagem rápida [Volpato 2007], para uso posterior em planejamento cirúrgico por médicos e odontologistas, ou, ainda, para o ensino, principalmente de anatomia.

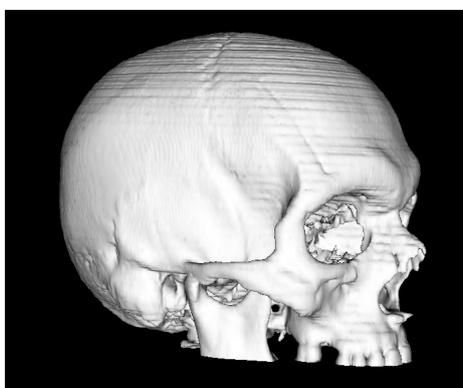


Figura 3. Superfície 3D do Osso

### 4.3. Visualização Volumétrica

Outro método de visualização 3D implementado no software é o *volume rendering*, cujas visualizações são mais realistas. No software foi utilizada a técnica conhecida como *Raycasting*. De modo simplificado, nesta técnica para cada pixel da tela é traçado um raio de luz (daí o nome *Raycasting*) em direção ao objeto. A cor do pixel será baseada na cor e transparência de cada *voxel* interceptado pelo raio de luz [Manssour and Freitas 2002]. Em TC os valores de cores e transparência são atribuídos aos níveis de cinza e estes valores dependem da necessidade do usuário que irá visualizar este volume e do estudo a ser realizado. Essa técnica foi implementada utilizando a biblioteca VTK. A figura 4 ilustra algumas projeções realizadas pelo software.

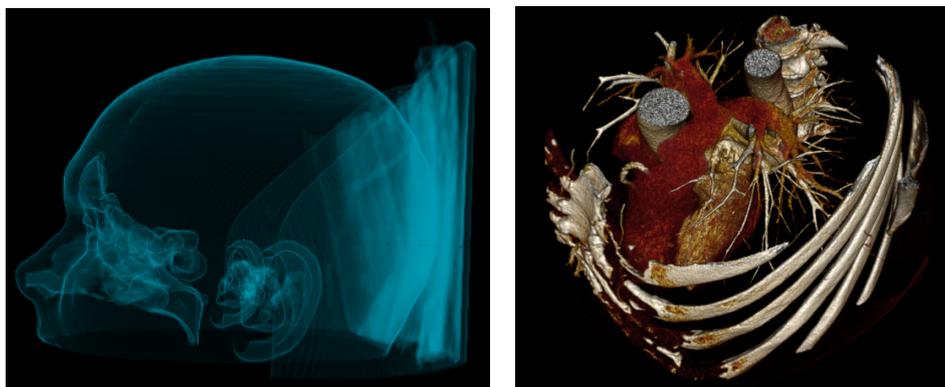


Figura 4. Visualização Volumétrica por *Raycasting*

### 4.4. Medidas

O InVesalius apresenta as ferramentas de medidas linear e angular, ambas disponíveis tanto nas fatias 2D quanto na reconstrução em 3D. A ferramenta de medida linear permite medir a distância entre dois pontos marcados pelo usuário. Já a ferramenta de medida angular mede o ângulo formado pelas duas semi-retas dos três pontos marcados pelo usuário. Essas ferramentas são importantes para, por exemplo, medir os avanços de um tratamento ou cirurgia, fazendo as medidas em exames antes e depois da cirurgia ou tratamento. São úteis também para medições de estruturas visíveis nos exames e para o planejamento cirúrgico. As representações como ilustra a figura 5, foram implementadas com a biblioteca VTK.

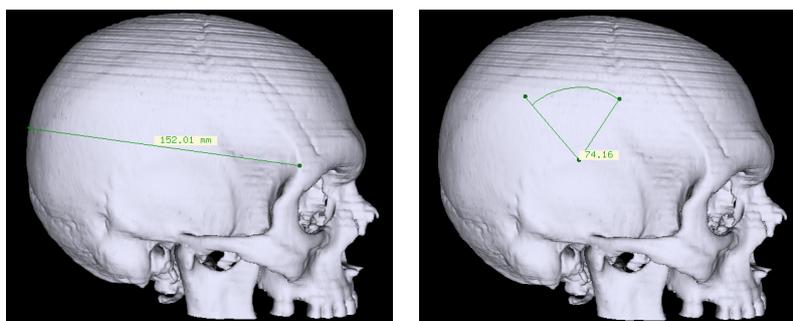


Figura 5. Medições linear e angular

## 5. Comunidade de usuários

O InVesalius, sua comunidade e todo o seu desenvolvimento estão centralizados no portal [www.softwarepublico.gov.br](http://www.softwarepublico.gov.br). Atualmente, a comunidade conta com 4833 membros, espalhados em 71 países, sendo Brasil, Portugal, Estados Unidos, Espanha e Itália os que concentram o maior número de usuários.

De acordo com pesquisa realizada em fevereiro de 2011, respondida por 120 usuários, o perfil da comunidade encontra-se assim distribuído: 54% dos usuários são da área de saúde; 25%, de computação; 9%, das engenharias, e 12%, de outras áreas (veterinária, gestão hospitalar, paleontologia entre outras). A pesquisa mostrou também que 63% dos usuários estão satisfeitos e 12% muito satisfeitos com o uso do software.

## 6. Conclusão

Neste trabalho foi apresentado o InVesalius, software multiplataforma, multi-idioma e executável em computadores pessoais, uma alternativa livre aos softwares proprietários de visualização médica e reconstrução tridimensional. Desde a sua concepção, o InVesalius tem sido bem aceito pela comunidade de usuários, tanto no Brasil quanto no exterior. Além disso, o fato de parte significativa da comunidade não ser da área de saúde demonstra a versatilidade do software e sua aplicação em outras áreas do conhecimento, contribuindo também para o fomento no desenvolvimento de novas aplicações no Brasil. O desenvolvimento do InVesalius é constante. Para o futuro tem-se como metas o suporte a outros formatos de exames médicos, suporte a outras técnicas de segmentação, novas maneiras de interação com o software, e transmissão de imagens por rede.

## Referências

- González, R. and Woods, R. (2008). *Digital Image Processing*. Pearson/Prentice Hall.
- Kinser, J. (2009). *Python for Bioinformatics*. Jones and Bartlett Publishers.
- Lorensen, W. E. and Cline, H. E. (1987). Marching cubes: A high resolution 3d surface construction algorithm. *SIGGRAPH Comput. Graph.*, 21:163–169.
- Malaterre, M. (2011). Grassroots dicom - <http://sourceforge.net/projects/gdcm/>.
- Manssour, I. H. and Freitas, C. M. D. S. (2002). Visualização volumétrica. *Revista de Informática Teórica e Aplicada IX*, pages 97–126.
- NiBabel (2011). Access a cacophony of neuro-imaging file formats - <http://nipy.sourceforge.net/nibabel/>.
- Pianykh, O. S. (2007). *Digital Imaging and Communications in Medicine (DICOM)*. Springer Publishing, 1th edition.
- Rappin, N. and Dunn, R. (2006). *wxPython in action*. Manning Pubs Co Series. Manning.
- Schroeder, W., Martin, K., Martin, K., and Lorensen, B. (2004). *The visualization toolkit*. Prentice Hall PTR.
- Subversion (2011). Subversion. <http://subversion.tigris.org/>. accessed 29-Mar-2011.
- Trac (2011). The trac project. <http://trac.edgewall.org/>. accessed 29-Mar-2011.
- Volpato, N. (2007). *Prototipagem Rápida: Tecnologias e Aplicações*. Edgard Blucher.
- Yoo, T. S. (2004). *Insight into Images*. A. K. Petters, 1th edition.