

# 基于ITK、VTK和MFC的DICOM图像读写及显示\*☆

吕晓琪, 任晓颖, 贾东征

## Image reading, writing and display for DICOM files based on ITK, VTK and MFC

Lü Xiao-qi, Ren Xiao-ying, Jia Dong-zheng

School of Information Engineering, Inner Mongolia University of Science and Technology, Baotou 014010, Inner Mongolia Autonomous Region, China

Lü Xiao-qi<sup>\*</sup>, Doctor, Professor, School of Information Engineering, Inner Mongolia University of Science and Technology, Baotou 014010, Inner Mongolia Autonomous Region, China  
lxiaoqi@imust.cn

Correspondence to: Ren Xiao-ying, Lecturer, Master, School of Information Engineering, Inner Mongolia University of Science and Technology, Baotou 014010, Inner Mongolia Autonomous Region, China  
huhurenxy@sohu.com

Supported by: the National Natural Science Foundation of China, No. 60761002\*

Received:2010-11-15  
Accepted:2010-01-28

内蒙古科技大学信息工程学院, 内蒙古自治区包头市 014010

吕晓琪<sup>\*</sup>, 男, 1963年生, 内蒙古自治区包头市人, 汉族, 2003年北京科技大学毕业, 博士, 教授, 主要从事医学图像处理的研究。  
lxiaoqi@imust.cn

通讯作者: 任晓颖, 讲师, 硕士, 内蒙古科技大学信息工程学院, 内蒙古自治区包头市 014010  
huhurenxy@sohu.com

中图分类号:R318  
文献标识码:B  
文章编号:1673-8225(2011)13-02416-05

收稿日期: 2010-11-15  
修回日期: 2011-01-28  
(20101115025/WL-Z)

### Abstract

**BACKGROUND:** The DICOM files include not only the image data but also a plenty of the medical information in the file header, which makes the work of the DICOM image correct reading, writing and display very important.

**OBJECTIVE:** To make use of the advantage of the toolkit and development platform to reach a basic purpose that achieves the correct reading, writing and display of the DICOM standard image.

**METHODS:** This article firstly integrated the ITK, VTK with MFC and then read and displayed the DICOM files by the classes and functions offered by ITK and VTK in the integration environment.

**RESULTS AND CONCLUSION:** This article achieved the correct reading, writing and display of the DICOM files by using the method that was mentioned in the article and a flexible, friendly, practical interface offered to users by the integration environment of ITK, VTK and MFC, and made a preliminary attempt on the software development in the integration environment and lay a good foundation for realizing more functional on this basis in medical image processing.

Lü XQ, Ren XY, Jia DZ. Image reading, writing and display for DICOM files based on ITK, VTK and MFC. Zhongguo Zuzhi Gongcheng Yanjiu yu Linchuang Kangfu. 2011;15(13): 2416-2420. [http://www.crter.cn http://en.zglckf.com]

### 摘要

**背景:** DICOM 文件不仅包含了图像本身的信息, 同时在文件头中还携带了大量的医疗相关信息, 这使得 DICOM 标准图像的正确读写和显示工作变得尤为重要。

**目的:** 有效地综合利用工具包和开发平台的优点, 借以达到一个实现 DICOM 标准图像的正确读写和显示。

**方法:** 首先将 ITK 工具包、VTK 工具包与 MFC 深度集成, 然后在该集成环境下利用这两个工具包所提供的类和函数读写和显示 DICOM 文件。

**结果与结论:** 在 ITK、VTK 与 MFC 深度集成环境下, 通过其所提供给用户的一个灵活、友好、实用的交互界面, 采用文中所述具体的方法实现了 DICOM 图像文件的正确读写及显示, 为三者集成环境下的软件开发做了一个初步的尝试, 为在此基础上三者医学图像处理方面更多功能的实现打下良好的基础。

**关键词:** 医学图像; DICOM 文件; VTK; ITK; MFC

doi:10.3969/j.issn.1673-8225.2011.13.032

吕晓琪, 任晓颖, 贾东征. 基于 ITK、VTK 和 MFC 的 DICOM 图像读写及显示[J]. 中国组织工程研究与临床康复, 2011, 15(13):2416-2420. [http://www.crter.org http://cn.zglckf.com]

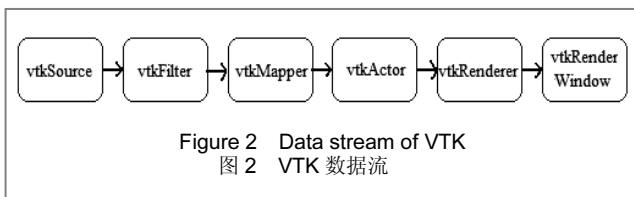
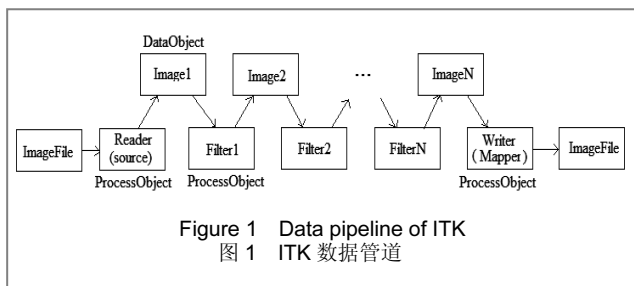
## 0 引言

ITK是一个基于C++语言设计和开发的跨平台、开放源码、面向对象的软件系统<sup>[1]</sup>, 提供医学图像处理、图像分割与配准的算法平台, 它起源于美国可视化人体项目(visible human project, VHP)<sup>[2]</sup>。因为ITK完全是面向医学图像的, 因此, 算法针对的数据对象是有限的数据来源。主要包括常见的二维图像格式: DICOM、BMP、JPEG、PNG等; 也包括三维数据格式: Raw、Meta。ITK的主要设计思想就是采用管道结构来管理数据流程, 即: 以数据处理为中心, 将数据对象与处理对象连结起来就构成了管道模型。①ITK中的数据对象(itk::DataObject类)有两种类型: 图像Image(itk::Image类)和网格Mesh(itk::Mesh类)。图像表示的是n维、规则采样的数据。采样的方向平行于x-y-z坐标轴,

采样起点、各方向的采样间距和采样大小都可以指定。像素类型和图像的空间维数可确定一个图像对象。网格则是由点和单元组成。②ITK中的处理对象分为3种类型, 包括Source、Filter和Mapper。Source类型对象的输出为数据对象, 包括各种图像的读取类(itk::ImageFileReader类, 图像类型由用户自己指定); Filter类型对象以一个或多个数据对象作为输入、然后输出一个或多个数据对象, 包括实现各种数字图像处理算法的类; Mapper类型对象是管道处理的最后一步, 它将最终的处理结果存到硬盘上。数据处理流程见图1。

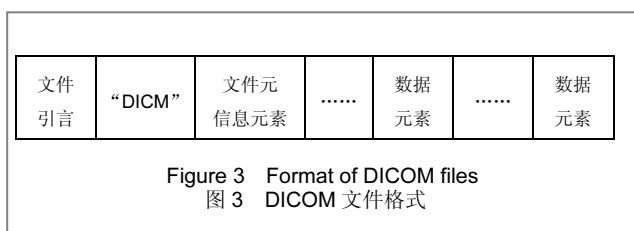
VTK是一个开源、自由获取的、支持多平台的软件系统, 由美国的Kitware公司负责维护。目前最新的版本是VTK 5.6.0<sup>[3]</sup>。它利用C++语言构建, 支持跨平台开发, 提供了超过1 000个C++类, 包括三维计算机图形学、图像处理和可视化三大部分, 提供了丰富的医学影像处

理与分析工具。VTK支持包括数量、向量、张量、结构和测定体积等方法一系列可视化算法，VTK还支持包括建模。VTK的数据流同样采用流水线作业方式，见图2。



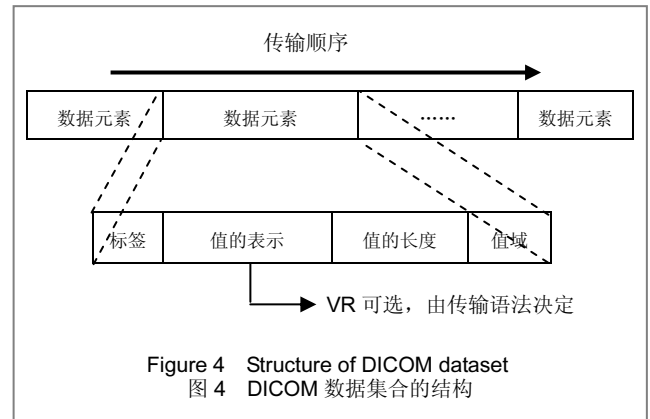
ITK的特点是不提供用户接口(UI)。VTK软件开发包虽具有强大的图形处理和可视化功能，但缺乏实用、灵活的交互界面。所以两者结合应用时，必须借助其他图形用户接口(GUI)软件包，比如Console、Qt、FLTK、MFC，才能完成实际意义上的数据三维显示或图像处理系统。在ITK、VTK所提供的范例中，大多数的程序都是基于Console为主的。在Windows操作系统下，Microsoft Foundation Class(MFC)具有较好的实用性、灵活性，是Windows平台上使用最为流行的辅助软件开发包，具有强大的UI开发能力，可利用它开发出专业级的软件。

DICOM文件是目前国际通用的医学图像存储和传输格式，读取DICOM文件是处理医学图像的必备步骤之一<sup>[4]</sup>。DICOM文件的格式复杂，所以DICOM图像的读取和显示要比其他格式困难得多。下面先简单介绍一下DICOM文件的结构。该协议包括了数字成像和通信传输两个方面的内容，涵盖了医学数字图像的采集、归档、通信、显示以及查询等几乎所有信息交换的协议，目前的版本为3.0。DICOM文件是DICOM数据集的封装，由DICOM文件元信息(MetaInformation)和数据集组成<sup>[5]</sup>，其结构见图3。



DICOM文件元信息包含设备信息、患者信息、检查信息、扫描参数、图像信息等数据，这些医疗相关数据

为医生的诊断提供了重要的依据，并且根据这些信息，医学图像就可以在不同的系统之间实现传输与共享。DICOM数据集(DICOM Dataset)是由一系列的数据元素(Data Element)按照标签(Tag)从小到大的顺序排列而成，其结构见图4。



由于DICOM文件是医学图像的专用存储格式，作为医学图像处理的专业开发工具包，ITK也提供了完整的读取DICOM图像的方法，包括读写单张、序列二维DICOM图片，读取二维序列图片并存储为体数据，读取文件头信息的方法等等。对于需要进行图像后处理的情况，使用ITK来读取DICOM图像更为方便，便于读取图像后直接对图像使用ITK的其他类进行分割、配准等处理。但因为ITK并不具备可视化方面的功能，所以在应用ITK进行图像读取和处理后，可以利用VTK进行可视化，以便观察结果及进行交互显示。

文章主要介绍了如何在VTK、ITK和MFC集成环境下正确读写和显示DICOM文件。

## 1 集成环境的实现

首先，通过CMake工具把ITK和VTK软件开发包构建在Windows操作系统下运行<sup>[6-7]</sup>。

其次，在VS2005环境下建立MFC工程(文章采用单文档工程)，并完成IDE下的相应配置。在MFC中，为了能够使用ITK、VTK所提供的类库，需要设置头文件及库文件的路径。便于在编译和链接时能搜索要使用的ITK、VTK库的程序。在VS2005环境下找到项目→属性→配置属性→C/C++→附加包含目录，将两者的头文件路径输入其中。在链接器目录下将两者库文件路径输入，并添加依赖项。至此完成了ITK、VTK与IDE的基本集成。

最后，实现VTK与MFC的深度集成。

在完成了以上ITK、VTK与IDE的集成之后，来运行一个显示图像的程序见图5(为踝关节DICOM图像)，不难发现VTK窗口是独立的，浮于MFC单文档窗口之上，不利于界面的统一布局与管理。



Figure 5 Unintegrated display effect  
图 5 未集成显示效果

对于VTK与MFC的集成已有相关文献介绍,分为快速集成和深度集成两种集成方式。图6是作者将图5对应程序做了快速集成后DICOM图像的显示。集成过程只需设置vtkRenderWindow类中的SetParentId函数为当前视图或对话框句柄即可,关键代码如下所示:

```
vtkRenderWindow
*renWin=vtkRenderWindow::New();
vtkRenderWindowInteractor
*iren=vtkRenderWindowInteractor::New();
this->renWin->SetParentId(this->m_hWnd);
this->iren->SetRenderWindow(this->renWin);
```

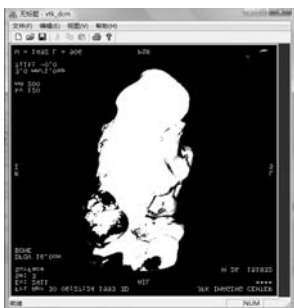


Figure 6 Display effect of fast integration  
图 6 快速集成显示效果

快速集成屏蔽了MFC的鼠标响应和消息传递,不利于后续编程实现更复杂的功能。VTK与MFC集成的根本目的是实现更实用、更灵活的交互界面,以弥补VTK自身在该方面的缺憾,那么就要求不仅要要将VTK的渲染窗口嵌入于MFC应用程序视图区域,而且还要使其融为一体,即拥有一个统一的具有友好交互能力的界面。

在此,重点介绍一下作者尝试实现VTK与MFC深度集成的另一种方法,具体步骤如下:首先,在视类中增加可视化管道中会用到的类,用这些类实例化对象,文章用到的对象有:渲染对象(ren)、渲染窗口(renWin)、交互对象(iren)、演员(BrainActor)、映射(BrainMap)等,

文章是在视类的构造函数及其析构函数中实现它们的初始化及善后工作的。其次,改写视图类的虚函数OnCreat(),以下为关键代码:

```
renWin->AddRenderer(ren);
renWin->SetWindowId(m_hWnd);
iren->SetRenderWindow(renWin);
```

最后,在视图类的OnDraw()函数中添加绘制代码:  
renWin->Render();

当然窗口的尺寸也可以在这里进行调整。

至此,可以在MFC应用程序视图区域中实现与VTK渲染对象的交互和鼠标响应操作,并可以利用MFC界面控件来进行参数控制,本程序中鼠标按下会弹出一个显示图像上像素坐标及像素值的消息框(见图7, 8)借以说明鼠标响应交互操作的实现,至此完成了VTK与MFC的深度集成。读者也可以参考VTK工具包Examples\GUIWin32\vtkMFC下的例程,实现更丰富的集成。该种集成环境下的界面显示将结合后续DICOM文件显示部分给出。



Figure 7 Display effect of VTK depth integration  
图 7 深度集成环境下 VTK 显示效果



Figure 8 Display effect after flipping  
图 8 翻转后 VTK 显示效果

## 2 DICOM图像的读写与显示

2.1 DICOM图像的读写 文章中的数据对象是二维DICOM图像。读写DICOM文件的功能使用ITK提供的范

函数来实现, 在ITK中, DICOM的范函性由GDCM库提供。这个开放的源码库由INSA\_Lyon的creatis团队发展<sup>[8]</sup>。实验使用itk::GDCMImageIO类来建立与GDCM库的连接, 实现读取DICOM图像的功能。ITK数据处理管道中的三种处理对象具体为: Source 采用itk::itkImageSeriesReader类; Filter 采用了itk::CurvatureFlowImageFilter; Mapper 采用了itk::itkImageFileWriter类。

**单张二维DICOM图像的读写:** 本部分实现单张二维DICOM图像以DICOM格式读入与以DICOM格式写入的工作。首先声明像素类型和图像的维数, 以它们作为模板参数来实例化读入图像的类型, 用该图像类型作为模板参数再实例化读取器Reader的类型并创建一个Reader, 将读入图像的文件路径和文件名作为参数传给Reader, 并为Reader关联GDCMImageIO对象, 调用Update()函数更新读取过程, DICOM图像就已读入内存中。用Reader的GetOutput()方法可以获得图像数据的指针。目前的焦点是如何再一次将图像在新文件中保存成DICOM格式。具体的, 先实例化一个ImageFileWriter类型, 并创建指针Writer, 将写入的图像的文件路径和文件名作为参数传给Writer, 并为其关联GDCMImageIO对象, 这个对象是知道DICOM格式的内部结构的, 最后用Update()来触发写入程序。文章在读写过程中对图像进行了保留边缘的平滑去噪处理, 以下为主要代码:

```
ImageIOType::Pointer
gdcmlImageIO=ImageIOType::New();
reader->SetImageIO(gdcmlImageIO);
reader->SetFileName(argv[1]); //参数为文件路径
及文件名
writer->SetImageIO(gdcmlImageIO);
rwriter->SetFileName(argv[2]); //将平滑后图像以
DICOM格式写入到参数所指定的文件中。
...
CurvatureFlowImageFilterType::Pointer
smoother=rvatureFlowImageFilterType::New(); //
实例化一个平滑滤波器的对象。
smoother->SetNumberOfIterations(5); //平滑滤波
器的迭代次数设为5
smoother->SetTimeStep(0.125); //平滑滤波器的
时间步长设为0.125
smoother->SetInput(reader->GetOutput());
writer->SetInput(smoother->GetOutput());
至此完成DICOM单张图像的读写。
```

**序列二维DICOM图像的读写:** 这部分主要实现以文件集的方式读取一个DICOM序列图像, 并在保持头文件信息不变的情况下保存为另一个DICOM序列图像的工作。

写DICOM文件是一件非常精密的操作, 因为所要处理的是大量重要的有关患者详细而精确的数据, 并且涉及到患者的隐私问题。所以在写的过程中, 一定要核实由你的代码所生成的头文件信息不会给患者的治疗和诊断带来风险, 要确保沿着数据处理管道正确地传递DICOM文件的细节信息, 从而才能保证正确地将原DICOM文件的这些信息写入新的DICOM序列图像文件。以下便是实现该部分内容的主要代码:

首先定义图像类型, 并明确所选像素类型和维度, 从而定义序列图像读取类型。

```
typedef signed short PixelType;
const unsigned int Dimension=3;
typedef itk::Image< PixelType, Dimension >
ImageType;
typedef itk::ImageSeriesReader< ImageType >
ReaderType;
```

接下来声明并实例化实际读写DICOM图像的itk::GDCMImageIO对象和为所有切片生成文件名并排序的itk::GDCMSeriesFileNames对象, 并创建相应智能指针gdcmlIO和namesGenerator。

```
ImageIOType::Pointer
gdcmlIO=ImageIOType::New( );
NamesGeneratorType::Pointer
namesGenerator=NamesGeneratorType::New( );
namesGenerator->SetInputDirectory(argv[1]); //
生成读入文件的文件名。
```

```
const ReaderType::FileNamesContainer &
filenames=namesGenerator->GetInputFileNames( );
实例化ReaderType对象并创建智能指针reader, 读
取DICOM格式序列图像。
```

```
reader->SetImageIO(gdcmlIO);
reader->SetFileNames(filenames);
最后用reader的Update()触发读取程序, 最好放在
try/catch询问块中, 在此不再赘述。
```

至此, 可以开始写数据了。首先把确定存在的输出目录传给outputDirectory指针。

```
const char * outputDirectory=argv[2];
然后, 实例化SeriesWriterType对象并创建智能指
针seriesWriter, 将reader的输出写为DICOM格式序列
图像。
```

```
seriesWriter->SetInput(reader->GetOutput());
seriesWriter->SetImageIO(gdcmlIO);
现在可以设置GDCMSeriesFileNames在
outputDirectory下产生新的文件名, 然后将最新生成
的文件传递给seriesWriter。
```

```
namesGenerator->SetOutputDirectory(outputDire
ctory);
```

```
seriesWriter->SetFileNames(namesGenerator->
GetOutputFileNames());
```

最后一步也是最关键的一步, 用 reader 读取 MetaDataDictionary 并传递给 seriesWriter。之所以关键是因为 MetaDataDictionary 包含了所有输入的 DICOM 文件的头文件。

```
seriesWriter->SetMetaDataDictionaryArray(reade
r->GetMetaDataDictionaryArray());
```

未了调用 seriesWriter 的 Update() 触发写程序, 放在 try/catch 问询块中, 在此不再赘述。

**2.2 DICOM 图像的显示** 本部分主要介绍在前述方法实现的 VTK、ITK 和 MFC 集成环境下 DICOM 图像的显示方法。在此, 将利用 ITK 读取的 DICOM 文件用 VTK 显示在 MFC 单文档视图区, 因为旨在说明方法所以未给出复杂的交互界面。

VTK 数据处理管道中采用了 vtkDICOMImageReader 类、vtkRenderer 类、vtkRenderWindow 类、vtkRenderWindowInteractor 类、vtkActor2D 类、vtkImageMapper 类, 并实例化各个类的对象并创建相应智能指针。图7为显示效果, 相应关键代码如下:

```
BrainMap->SetInput(vtkreader->GetOutput());
BrainActor->SetMapper(BrainMap);
ren->AddActor(BrainActor);
```

ITK 所读取的图像一经 VTK 显示后图像发生了翻转, 并且是上下翻转而左右并不发生翻转, 原因是 ITK 的坐标系统和 VTK 的坐标系统 Y 轴是相反的, 所以为了能正确显示图像还需要将 Y 轴进行翻转, 可见这种翻转可以发生在两种情形, 一种是在 ITK 读入图像的时候翻转即改变文件的读取方式, 该种方式可参考 ITK 的 FlipImageFilter 类, 该类可以实现翻转任何坐标轴中的图像, 对于本文中情形可以使用一个 [0, 1] 翻转排列, 表示 X 轴保持不变时对 Y 轴进行翻转; 另一种是在 ITK 图像数据已经转换为 VTK 图像数据后将新的 VTK 图像数据 Y 轴翻转, 这也正是本文实现的图像翻转方法。相应关键代码:

```
flipY->SetFilteredAxis(1);
flipY->SetInputConnection(vtkreader->GetOutput
Port());
BrainMap->SetInput(flipY->GetOutput());
```

以上便是集成环境下的 DICOM 图像显示。

### 3 结论

ITK 是算法设计者的一个有力助手, 是医学图像处理工作者的案例研究算法仓库, 遗憾的是不提供 UI, VTK 软件开发包强大的图形处理和可视化功能, 却缺乏实用、灵活的交互界面, 而 MFC 具有强大的 UI 开发能力。结合三者的特点, 作者通过对 DICOM 医学图像的读取与显示, 将 ITK、VTK 和 MFC 有机的结合起来, 扬长避短, 为三者集成环境下的软件开发做了一个初步的尝试, 而对于在此基础上三者更强大功能的实现(诸如医学图像的分割与配准、三维可视化、虚拟手术导航等)还有待深入探讨与研究。

### 4 参考文献

- [1] Kitware Inc. NLM Insight Segmentation and Registration Toolkit [EB/OL]. 2005. <http://www.itk.Org>.
- [2] Ackerman MJ. The visible human project. Proceedings of the IEEE, 1998;86(3):501-504.
- [3] <http://www.vtk.org/VTK/resources/software.html>.
- [4] 黄嘉丽, 王博亮, 黄晓阳. 基于 DICOM 格式的肝脏肿瘤 CT 图像分割[J]. 计算机技术发展, 2008, 18(1):48-51.
- [5] NEMA. The DICOM standard. Technical report, NEMA, <http://medical.nema.org,2004>.
- [6] <http://www.itk.org/CourseWare/Training/GettingStartedI-WebPage/index.html>.
- [7] <http://www.kitware.com/products/books/vtkguide.html>.
- [8] Benoit Regrain Eric Boix, Mathieu and Jean Pierre Roux. The GDCM Library. CNRS, INSERM, INSALyon, UCBLyon, <http://www-creatis.insa-lyon.fr/Public/Gdcm>.

#### 来自本文课题的更多信息——

**基金资助:** 国家自然科学基金资助项目(60761002)。

**作者贡献:** 第一作者进行实验设计及评估, 实验实施为第二作者, 资料收集为第三作者, 第二作者成文, 第一作者审核并对文章负责。

**利益冲突:** 课题未涉及任何厂家及相关雇主或其他经济组织直接或间接的经济或利益的赞助。

**本文创新性:** 于 2011-01 以文章关键词为检索关键词, 在 CNKI 数据库、维普中文科技期刊数据库、万方学位论文全文数据库、ELSEVIER 数据库等多个数据库中检索未见具有相同创新点文章。

**方法创新:** 为医学图像读写及显示后一系列的医学图像处理工作搭建了灵活实用的实验平台, 进而得到相关基金资助。