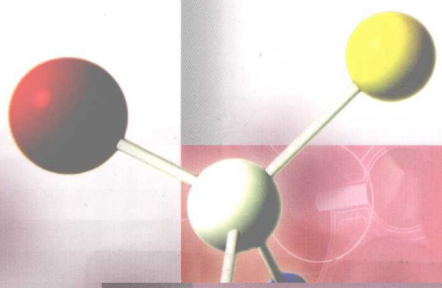
医学图像编程技术

周振环　伍云智　赵　明　编著



電子工業出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
http://www.phei.com.cn

医学图像编程技术

周振环　伍云智　赵　明　编著

電子工業出版社
Publishing House of Electronics Industry
北京·BEIJING

内 容 简 介

本书是医学图像编程的入门级教材和参考书。本书通过一个个由浅入深的编程范例，介绍了如何使用三维可视化工具箱 VTK 和医学图像分割与配准工具箱 ITK 进行三维医学图像编程。本书的主要内容包括 VTK 与 ITK 的联合安装和使用、VTK 编程入门范例、VTK 的数据结构、VTK 的可视化算法（包括颜色映射、抽取轮廓、剪切、纹理等）、VTK 的医学图像处理功能（包括图像分割、图像平滑、重切分、体绘制等）、VTK 的综合应用等。本书最后介绍了常用医学图像处理软件 MIPAV、3D Slicer 在结构像、功能像、脑图谱、弥散张量成像和纤维束跟踪等方面的应用。

本书可作为医学影像学专业高年级本科生和研究生教材，也可作为大学教师、公司研发人员、硕博研究生进行医学图像研究时的技术参考书。

本书中的医学图像数据和编程范例在随书携带的光盘中，可供读者运行和上机实验。

图书在版编目（CIP）数据

医学图像编程技术 / 周振环，伍云智，赵明编著. —北京：电子工业出版社，2010.6
ISBN 978-7-121-10881-5

Ⅰ. ①医… Ⅱ. ①周… ②伍… ③赵… Ⅲ. ①医学图像－图像处理－程序设计 Ⅳ. ①R445

中国版本图书馆 CIP 数据核字（2010）第 087918 号

责任编辑：侯丽平　　文字编辑：谭丽莎
印　　刷：北京东光印刷厂
装　　订：三河市皇庄路通装订厂
出版发行：电子工业出版社
　　北京市海淀区万寿路 173 信箱　邮编：100036
开　　本：787×1 092　1/16　　印张：19.25　　字数：493 千字　　彩插：2
印　　次：2010 年 6 月第 1 次印刷
定价：49.80 元（含 CD 光盘 1 张）

凡所购买电子工业出版社图书有缺损问题，请向购买书店调换。若书店售缺，请与本社发行部联系，联系及邮购电话：（010）88254888。

质量投诉请发邮件至 zlts@phei.com.cn，盗版侵权举报请发邮件至 dbqq@phei.com.cn。

服务热线：（010）88258888。

前　言

医学图像编程门槛较高，入门较难。一般是采用 VC++6.0 来开发的，其开发周期很长，代码难以维护，特别是三维显示速度较慢，质量较差。使用 OpenGL 以后，其三维绘制速度有较大程度提高，显示质量也得到了不断改善，但医学图像编程困难的问题并没有得到根本解决。要想解决这一问题，首先程序员应对医学图像文件格式有一定的认识。医学图像是二维序列组，原来各厂商的图像格式均不一样，但现在已制订了统一的标准 DICOM，因此要想进行图像编程先要懂得如何读/写 DICOM 图像。其次，用 VC 来显示一幅图像并不容易，好在现在已有许多数字图像编程的书籍介绍如何来显示一幅二维图像。当然，能显示一幅就能显示多幅，这里所介绍的医学图像便是序列二维图像组。再次，医学图像需要显示三维，这就需要在二维序列图像间插值，以进行三维重建和绘制。因此，程序员还要懂计算机图形学。现在已经有一些书中用 OpenGL 来讲解和演示计算机图形学了。

美国 Kitware 公司于 1998 年推出了“三维可视化工具箱”Visualization Toolkit（VTK），希望用简单的代码来实现三维可视化编程工作。2002 年，它又推出了“医学图像分割与配准工具箱” Insight Segmentation and Registration Toolkit（ITK），封装了最常用的医学图像处理算法。不过它没有可视化算法，因此 ITK 必须与 VTK 联合使用，只有这样才能实现医学图像的三维可视化编程。Kitware 公司于 1998 年使用 VTK 和 ITK 成功开发出了 3D Slicer，以应用于图像引导下的治疗。2003 年，该公司成功开发出“图像引导下的手术工具箱”Image-Guided Surgery Toolkit（IGSTK），将医学图像应用于手术计划和手术导航场合。

本书的详细内容如下所示。

第 1 章详细介绍了 VTK 和 ITK 的混合安装使用，并给出了一个简单编程例子，演示了如何使用 VTK、ITK、CMake 来编程，从而最终显示出一幅医学图像。

第 2 章通过编程范例来说明了三维可视化编程的一些概念，如相机、观察员、渲染、光照、交互器、读取器、过滤器等编程对象，并对每个对象给出了其 1～2 个范例。另外，本章还分别在 API 和 MFC 下给出了医学图像编程的例子。

第 3 章描述了 VTK 和 ITK 使用的数据集结构和数据属性，并对由各种数据集构成的对象进行了显示，还演示了读/写和导入/导出数据集。

第 4 章讲解了 VTK 可视化算法，演示了颜色映射、标量映射、抽取轮廓、等值面着色、图形符号、流线、流面、剪切、剪裁、探测、纹理等绘制方法，并给出了运行结果。

第 5 章演示了图像数据及其处理。本章主要介绍了医学图像经常用到的一些技术，如数据创建、图像显示、图像直方图、图像分割、图像平滑、重采样、轴排列、轴翻转、重切片等。本章最后演示了体绘制，并给出了运行结果。

第 6 章使用 VTK+MFC 开发了一个医学图像软件界面，其主要功能包括读写 DICOM 图像、显示三个正交面（横断面、矢状面、冠状面）、三维体绘制。

第 7 章介绍了医学图像软件，其中包括 VTK Designer、MIPAV、3D Slicer。首先，本章演示了如何用 VTK Designer 创建 VTK 流水线应用程序。其次，演示了如何使用 MIPAV 软件将脑图谱与结构像进行融合。最后，应用 3D Slicer 软件演示了两个例子：一是三维可视化（脑组织和血管），包括图像装载、分割、三维显示、保存场景；二是图像引导下的治疗，包括肿瘤模型、结构像、功能像、脑图谱融合、弥散张量成像和纤维束跟踪等。

本书中的医学图像数据和编程范例在随书携带的光盘中，可供读者运行和上机实验。

本书既可作为医学影像学专业高年级本科生和研究生的教材，也可作为大学教师、公司研发人员、硕博研究生进行医学图像研究时的参考书。限于作者水平，书中难免存在错误，敬请读者批评指正。

周振环　伍云智　赵明

2010 年 2 月于深圳

目　录

第 1 章　安装 VTK 和 ITK

关于 VTK 的安装，笔者看过一些书上的介绍，也查阅过网上的资料。按照这些说明，笔者确实成功安装了 VTK，但在安装过程中也产生了一些疑问。带着这些疑问，通过尝试不同的设置，进行多次的安装，并对安装结果进行测试，笔者得到了一些心得和体会。希望笔者的经验能让大家少走一些弯路。

1.1　VTK 的安装

1.1.1　获取安装资源

安装 VTK 的相关文件可从 http://vtk.org/VTK/resources/software.html 上进行下载，其当前最新版本是 5.4.2。从该网页上可以看到，供下载的文件分成 4 类：Windows（Installer），Source，Data 和 Documentation。

① Windows（Installer）指的是一个二进制安装程序。它只支持 Tcl 语言，可以不用下载。

② Source 指的是 VTK 的源代码文件。它有两种压缩格式，根据 PC 系统的支持情况，可选择一种格式进行下载。

③ Data 指的是一些图像数据文件，在 VTK 程序中需要用到。它和 Source 一样，也有两种压缩格式，根据 PC 系统的支持情况，可选择一种格式进行下载。

④ Documentation 指的是一些帮助文件，包含了所有类及其方法的说明，以及类继承关系、相关类的关系说明等。这些文件按网页的形式组织，利用网页的超链接功能，可从一个类的网页方便地跳转到相关类的网页中，对学习 VTK 很有帮助，推荐下载。不过对于安装 VTK 而言，这些文件不是必需的。

为生成一个安装 VTK 的工程，还必须先安装 CMake 程序，可以从 http://www.cmake.org/cmake/resources/software.html 进行下载。该网页提供了对应不同 PC 系统的多个文件。对于 Windows 用户而言，可选择 Windows（Win32 Installer）（安装文件）或 Windows ZIP（压缩文件）进行下载。

最后，还必须要安装一个 C++编译器。编译器的版本可以是 Visual Studio 6.0（VS 6.0）或者是.NET。本书采用的是 VS 6.0 编译器。

1.1.2　安装步骤

首先解压下载的 VTK 文件并放置于适当目录下。假设下载的源代码文件名为 vtk-5.4.2.zip，数据文件名为 vtkdata-5.4.2.zip。将 vtk-5.4.2.zip 解压在 E:\vtksrc\下，将 vtkdata-5.4.2.zip 解压在 E:\vtkdata\下，则将分别产生 E:\vtksrc\Common 和 E:\vtksrc\Common 等路径。

然后安装并运行 CMake。如图 1-1 所示是一个 cmake-gui.exe 程序的简单界面，用户可以根据自己的 PC 系统和需要来配制 VTK 的组建选项。此时首先必须告诉 cmake-gui VTK 源代码树的位置（即 vtk-5.4.2.zip 的解压路径），以及存放编译后的 VTK 二进制文件的路径。如果是用“Browse Source…”或“Browse Build…”按钮来指定这些目录，则“Browse Source…”的按钮必须选择 E:\vtksrc\路径，而“Browse Build…”按钮必须选择用户指定的路径。这两个路径也可以在编辑框手动输入，本书中将它们分别设为了 E:\vtksrc\和 E:\vtkbin\。其中 E:\vtksrc\就是 vtk-5.4.2.zip 的解压路径。之后，单击“Configure”按钮（首次单击会提示选择编译器，接受默认选项即可），这时将会用 CMake 缓存中的一系列变量和值来填充 cmake-gui 界面。首次运行时，所有的变量显示为红色，如图 1-2 所示。红色表示在先前定制步骤中，缓存项有新增或被改变（此处显示为图中的深色部分）。如果没有看到期望修改的缓存项，可将下拉列表框中的值 Simple View 改为 Advanced View，这时将出现更多缓存项。

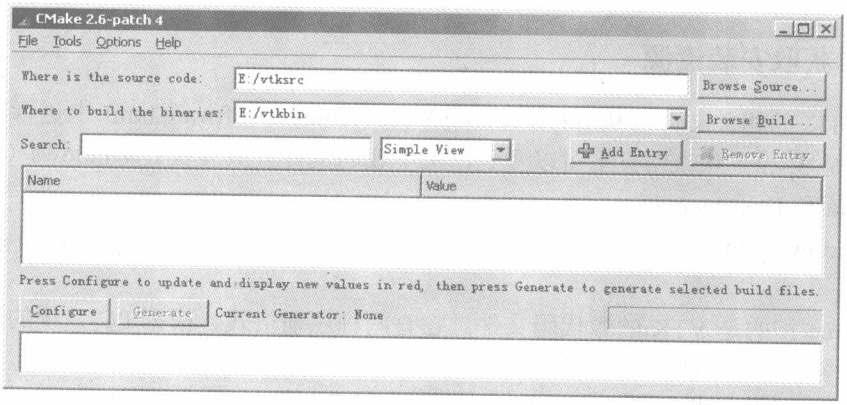


图 1-1　cmake_gui.exe 界面

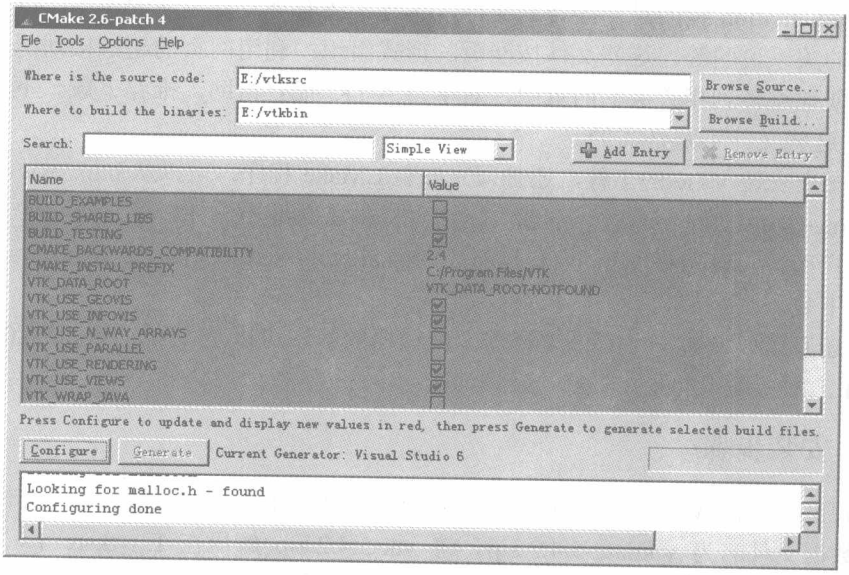


图 1-2　第一次 Configure 后的结果

这时，用户可以自定义 VTK 组建方式了。要设置 CMake 界面中的值，需单击变量的右边项。依据变量的类型，将会出现文件选择器、编程框或下拉框，用户可以修改其值。

VTK 有一些重要的缓存值，如下所示。

① BUILD_EXAMPLES——指明是否编译 VTK 中的例子。当它处于未激活状态时可以减少编译时间，之后也可以再进行单独编译。

② BUILD_TESTING——指明是否编译 VTK 中的测试代码。当它处于未激活状态时可以减少编译时间，之后也可以再进行单独编译。

③ BUILD_SHARED_LIBS——指明是否创建共享库。如果激活，那么 DLL 或共享库将会被创建，这样之后生成的 VTK 应用程序将会比较小，从而可以节省不少空间；如果未激活，那么静态库将被创建。默认设置是静态库。静态库易于执行。当可执行程序运行时，静态库不必包含在路径中。可执行程序在生成时会将静态库包含进去，这对基于应用的 VTK 分布比较好。

④ CMAKE_INSTALL_PREFIX——VTK 的安装路径。安装后，源代码中的头文件及编译生成的库文件和可执行程序将被分别复制到安装路径的不同子目录下，这将使 VTK 应用程序的工程设置方便一些。即使不安装到这个路径，源代码中的头文件及编译生成的库文件和可执行程序仍然会在 e:/vtkbin（本书的设置）的目录下，用户仍然可以编译 VTK 程序。

⑤ VTK_DATA_ROOT——vtkdata-5.4.2.zip 的解压路径。CMake 可能不会自动找到这个解压路径，因此可能需要手动设置，本书将其设为 E:\vtkdata\。

⑥ VTK_USE_GUISUPPORT——指明是否支持 GUI。激活后，Configure 会产生两个新缓存项，激活其中一个缓存项 VTK_USE_MFC，就可以进行 VTK 与 MFC 的混合编程，可以编写类似于 VTK 自带的例子 Examples\GUI\Win32\vtkMFC 中的程序了。

为了得到 CMake 变量的在线帮助，可将光标停靠在变量名上，此时提示信息就会自动出现。

继续单击“Configure”按钮直到不再出现红色的值，并且都是用户期望的值为止，如图 1-3 所示。此时，再单击“Generate”按钮，CMake 会按所选的组建类型写出组建文件。对于 Microsoft 而言，在所选的二进制目录中将创建一个工程文件，用 Visual Studio 装载这个工程文件 VTK.dsw，在左侧的工作空间窗口的 ClassView 视图中，便可以看到该工作空间里的所有工程了，包括生成库文件的工程（如 vtkCommon、vtkMFC 等），以及编译测试代码的工程（如 CommonCxxTests、FilteringCxxTests 等）。其中最重要的两个工程是 ALL_BUILD 和 INSTALL——唯一两个名字字母全是大写的工程。先对 ALL_BUILD 进行组建：单击 BUILD 菜单下的 Set Active Configuration 菜单项，在弹出的对话框中可以看到对 ALL_BUILD 工程有四种组建方式，即 Debug、Release、MinsizeRel 和 RelWithDebInfo。如果用户使用 VC 编过程序，则一定对前两种方式不陌生，其中 Debug 生成的程序因带有调试信息而显得比较大，Release 则正好相反。对于后两种方式，也可以从名字中看出它们的含义来，一个是将程序最小化的 Release，一个是带调试信息的 Release。这里需要用户注意的是，如果用带调试信息的方式组建，之后在调试自己的程序时，可以使用“Step into”进入 VTK 库函数中，这适合于类开发者；而对于 VTK 用户来说，就不需要用带调试信息的方式编译了。一般可使用前两者，如果磁盘空间比较紧张，可以考虑后两者。选好后，按快捷键 F7 进行组建即可。

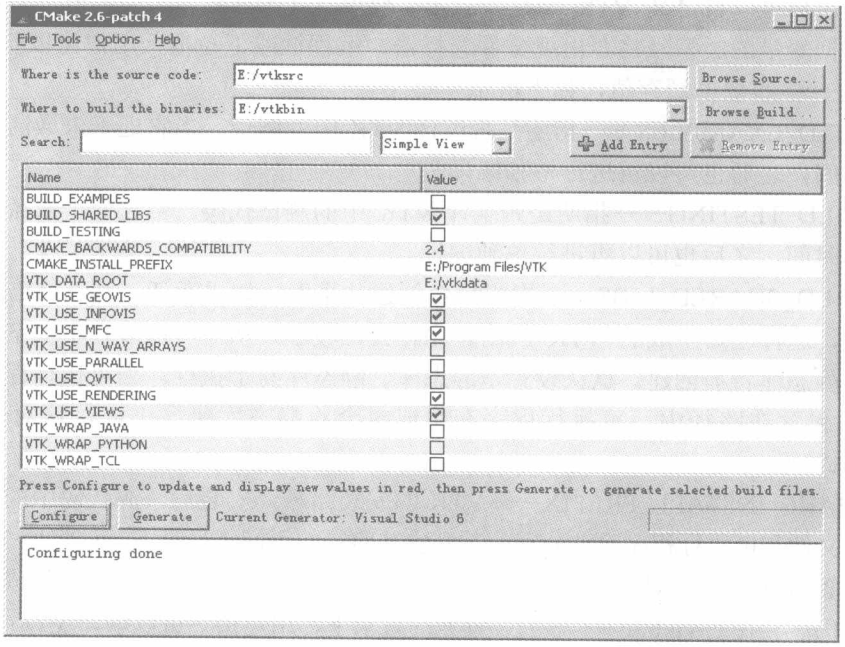


图 1-3　配置 VTK 安装

经过较长时间的等待后，组建好的所有库和可执行文件都会存放在一个 bin 子目录下，该 bin 子目录是在 CMake 中设置的二进制文件目录下的一个子目录，如本书的 E:\vtkbin\bin。之后可以再选择 INSTALL 工程，并对其组建一次以进行安装（该步可不执行）。

注意： 不要使用 MSVC++的“Rebuild All”菜单项来重建源代码，这会删除 CMakeLists.txt 文件，而这些文件是先前自动产生的并成为组建过程的一部分。MSVC 会重新装载它们，这将导致错误提示的产生。如果想重新创建，必须先删除 VTK 二进制目录，重新运行 CMake，然后再重建。

下面给出在几种不同设置下进行安装后，所产生的文件大小（基于版本 5.4.2）。

① 设置静态库，用 Release 方式组建——123MB。

② 设置静态库，用 Debug 方式组建——206MB。

③ 设置动态库，用 Release 方式组建——62.3MB。

④ 设置动态库，用 Debug 方式组建——206MB。

如果激活了 BUILD_SHAREDLIBS，创建 VTK 后，必须让 Windows 知道 DLL 文件的路径，可以采用两种方法：一种方法是将产生的 DLL 复制到 windows 会自动查找的路径，如 Windows/system32 或 Winnt/System（不推荐这种方法，因为复制的路径可能会被忘记，之后复制新版本的 DLL 文件时也不一定就会将原来的覆盖掉）；另一种方法是修改 path 环境变量，将库文件存放路径包含进去。如果要复制 DLL 和可执行程序，需要复制 bin/selected configuration/路径中的所有文件（selected configuration 是用户之前选择的组建方式，如 Debug、Release 等）。如果修改环境变量，则在 Windows95/98 中可以使用 sysedit 在 autoexec.bat 文件中加一行。下面给出与此前列出的四个不同组建方式相应的四个例子，假

设 4 个例子都组建在 VTK 的 E:\vtkbin 目录下。

① PATH = E:\vtkbin\bin\Debug。

② PATH = E:\vtkbin\bin\Release。

③ PATH = E:\vtkbin\bin\MisizeRel。

④ PATH = E:\vtkbin\bin\RelWithDebInfo。

如果组建了 INSTALL 工程，并且假设安装路径 CMAKE_INSTALL_PREFIX 为 E:\Program Files\VTK，那么有

```
PATH = E:\Program Files\VTK \bin
```

在 WindowsNT/ME/2000/XP 中，右击“我的电脑”，选择“属性”选项，会弹出一个对话框，选择环境变量标签，像上面一样添加或修改环境变量。如果 PATH 环境变量已经存在，就将 VTK 路径加在它的前面，以分号分隔。例如，假设原始路径为

```
C:\winnt\system;C:\some\other=dir;%Path%
```

则类似下面修改

```
E:\vtkbin\bin\Debug;C:\winnt\system;C:\some\other=dir;%PATH%
```

或

```
E:\Program Files\VTK \bin;C:\winnt\system;C:\some\other=dir;%PATH%
```

这样就可在 PC 上成功安装 VTK 了。

1.1.3　测试安装结果

VTK 自带的每个 C++例子都有一个相应的 CMakeLists.txt 文件，被 CMake 用来创建该例子的 VC++工作空间，也正好可用来测试 VTK 的安装。这里以 Examples\Tutorial\Step1\Cxx 为例子进行测试。

运行 CMake，首先设置其源代码位置为 CMakeLists.txt 所在路径，如 E:\vtksrc\Examples\Tutorial\Step1\Cxx。然后设置产生的工作空间的存放路径，并单击“Configure”按钮，操作类似 VTK 的安装，之后将会出现一些缓存变量。其中需要注意的变量有 VTK_DIR，该变量会指示 VTKConfig.cmake 文件所在路径，其默认值可能是编译后的 VTK 二进制文件路径，如 E:/vtkbin。如果对编译后的 VTK 文件进行了安装，则路径为 PREFIX/lib/vtk，如 E:/Program Files/VTK/lib/vtk-5.4。建议使用后者，如图 1-4 所示。

设置完变量后继续单击“Configure”按钮直到不再出现红色的值为止，再单击“Generate”按钮，CMake 就会在指定路径下产生一个工作区。之后用 VC++打开该路径下的 Cone.dsw 文件，就可以编译运行，检测是否安装成功了。编译后会生成 Cone.exe，其运行结果是一个旋转的圆锥，如图 1-5 所示。

以后如果要创建自己的应用程序，而又不想使用 CMake，则可以复制该工程的一些设置，如单击 Project 菜单下的 Settings，在弹出的窗口中选择“C/C++”选项卡，将“Category”选择为“Preprocessor”，记录下“Additional include directories”里的内容。

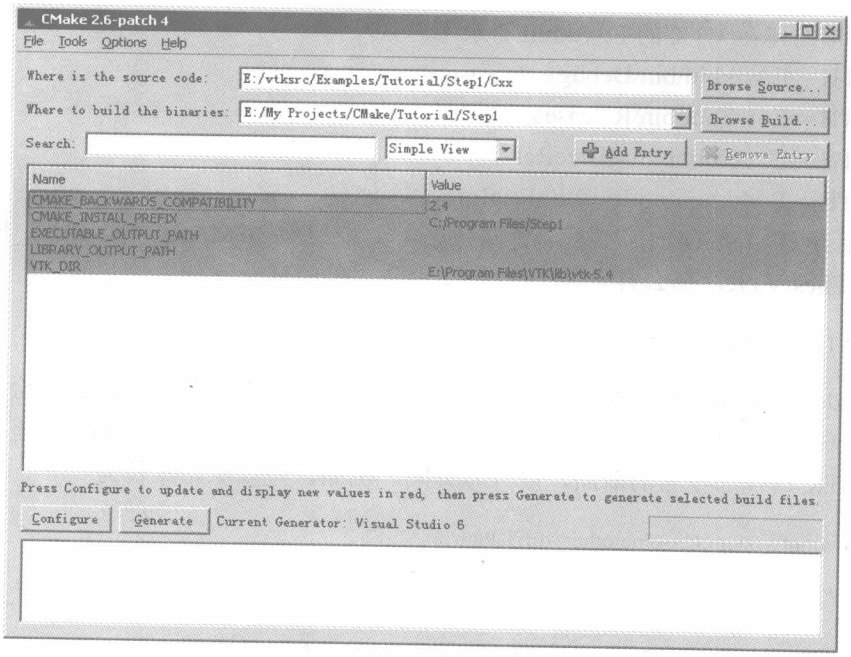


图 1-4　配置 VTK 自带范例 Examples/Tutorial/Step1

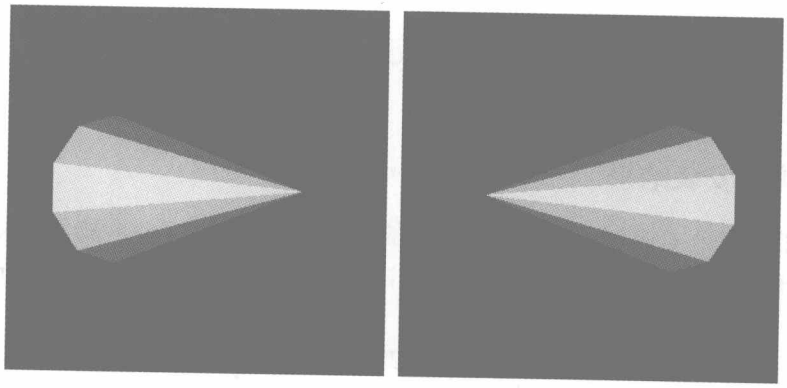


图 1-5　Cone.exe 运行结果的两帧画面

如果 VTK_DIR 设置的是“E:\vtkbin”，则其内容为 E:\vtkbin,E:\vtkbin\Common,E:\vtkbin\Utilities,E:\vtkbin\VolumeRendering,E:\vtkbin\Rendering,E:\vtkbin\GUISupport\MFC,E:\vtkbin\Utilities\vtkalglib,E:\vtksrc\Infovis,E:\vtksrc\Geovis,E:\vtksrc\Views,E:\vtksrc\VolumeRendering,E:\vtksrc\Hybrid,E:\vtksrc\Widgets,E:\vtksrc\Rendering,E:\vtksrc\Rendering\Testing\Cxx,E:\vtksrc\IO,E:\vtksrc\Imaging,E:\vtksrc\Graphics,E:\vtksrc\GenericFiltering,E:\vtksrc\Filtering,E:\vtksrc\Common,E:\vtksrc\Utilities,E:\vtksrc\Common\Testing\Cxx,E:\vtkbin\Utilities\vtklibproj4,E:\vtksrc\Utilities\vtklibproj4,E:\vtkbin\Utilities\DICOMParser,E:\vtksrc\Utilities\DICOMParser,E:\vtkbin\Utilities\vtkfreetype\include,E:\vtksrc\Utilities\vtkfreetype\include,E:\vtkbin\Utilities\vtknetcdf,E:\vtksrc\Utilities\vtknetcdf,E:\vtkbin\Utilities\vtkexodus2\include,E:\vtksrc\Utilities\vtkexodus2\include,E:\vtkbin\Utilities\MaterialLibrary,E:\vtksrc\Utilities\MaterialLibrary,E:\vtkbin\Utilities\verdict,E:\vtksrc\Utilities\verdict,E:\vtksrc\GUISupport\MFC,E:\vtksrc\Utilities\vtkalglib。

如果 VTK_DIR 设置的是“E:\Program Files\VTK\lib\vtk-5.4”，则其内容为 E:\Program Files\VTK\include\vtk-5.4。

单击“Link”选项卡，将“Category”选择为“Input”，记录下“Object/library module”里的内容，为 vtkRendering.lib，vtkGraphics.lib，vtkverdict.lib，vtkImaging.lib，vtkIO.lib，vtkFiltering.lib，vtkCommon.lib，vtkDICOMParser.lib，vtkNetCDF.lib，vtkmetaio.lib，comctl32.lib，wsock32.lib，vtksqlite.lib，vtkpng.lib，vtktiff.lib，vtkzlib.lib，vtkjpeg.lib，vtkexpat.lib，vtksys.lib，ws2_32.lib，vfw32.lib，vtkftgl.lib，vtkfreetype.lib，opengl32.lib，kernel32.lib，user32.lib，gdi32.lib，winspool.lib，comdlg32.lib，advapi32.lib，shell32.lib，ole32.lib，oleaut32.lib，uuid.lib，odbc32.lib，odbccp32.lib。

将上述内容与用户自己新建的空 win32 Console Application 工程相比，可看出它仅增加了 kernel32.lib 之前的库。在对用户自己的应用程序进行设置时，只需要加入前面的库，再记录下“Additional library path”里的内容即可。

如果 VTK_DIR 设置的是“E:\vtkbin”，则其内容为 E:\vtkbin\bin\$(IntDir),E:\vtkbin\bin\。

如果 VTK_DIR 设置的是“E:\Program Files\VTK\lib\vtk-5.4”，则其内容为 E:\Program Files\VTK\lib\vtk-5.4\$(IntDir),E:\Program Files\VTK\lib\vtk-5.4。

总体来说，还是安装后的设置简单些。对于 VTK 自带的大部分例子，这些设置还是一样的。编写用户自己的应用程序时，主要应对这三个地方进行设置。另外要注意，如果用户自己的程序编译不成功，可先对 VTK 中类似的例子运行 CMake，然后再参考其设置进行编写。

1.2　ITK 的安装

ITK 的安装方式大体上和安装 VTK 的方式类似，下面简要介绍其安装过程。

1.2.1　获取安装资源

可以免费从网站 http://www.itk.org/ITK/resources/software.html 上下载 ITK 源代码文件（当前最新版本为 3.14.0），如 InsightToolkit-3.14.0.zip。

1.2.2　安装步骤

首先解压下载的 ITK 文件 InsightToolkit-3.14.0.zip 并将其放置于适当目录下。如解压在 E:\下，则将产生 E:\InsightToolkit-3.14.0\Code 等路径。

然后安装并运行 cmake-gui.exe 程序。此时首先必须告诉 cmake-gui ITK 源代码树的位置，以及存放编译后的 ITK 二进制文件的路径。这里分别设为 E:\InsightToolkit-3.14.0 和 E:\itkbin\。之后再单击“Configure”按钮并根据提示选择编译器。

与 VTK 一样，ITK 也有一些重要的缓存值，如下所示。

① BUILD_EXAMPLES——指明是否编译 ITK 中的例子。当它处于未激活状态时可以减少编译时间。

② BUILD_TESTING——指明是否编译 ITK 中的测试代码。当它处于未激活状态时可以减少编译时间。

③ BUILD_SHARED_LIBS——指明是否创建共享库。这里可以不激活，因为即使激活也只生成一个 DLL 文件 ITKCommon.dll。

④ CMAKE_INSTALL_PREFIX——ITK 的安装路径。安装后，源代码中的头文件及编译生成的库文件将被分别复制到安装路径下的不同子目录下，这将使 ITK 应用程序的工程设置方便一些。当然即使不安装，仍然可以编译 ITK 程序（情况和安装 VTK 时一样）。

继续单击“Configure”按钮直到不再出现红色的值，并且都是用户期望的值为止，如图 1-6 所示。此时再单击“Generate”按钮，CMake 会按所选的组建类型写出组建文件。对于 Microsoft 而言，在所选的二进制目录中将创建一个工程文件，用 Visual Studio 装载这个工程文件 ITK.dsw，单击 BUILD 菜单下的“Set Active Configuration”菜单项，在弹出的对话框中对 ALL_BUILD 工程选择一种合适的编译方式，然后组建它（快捷键 F7）即可。

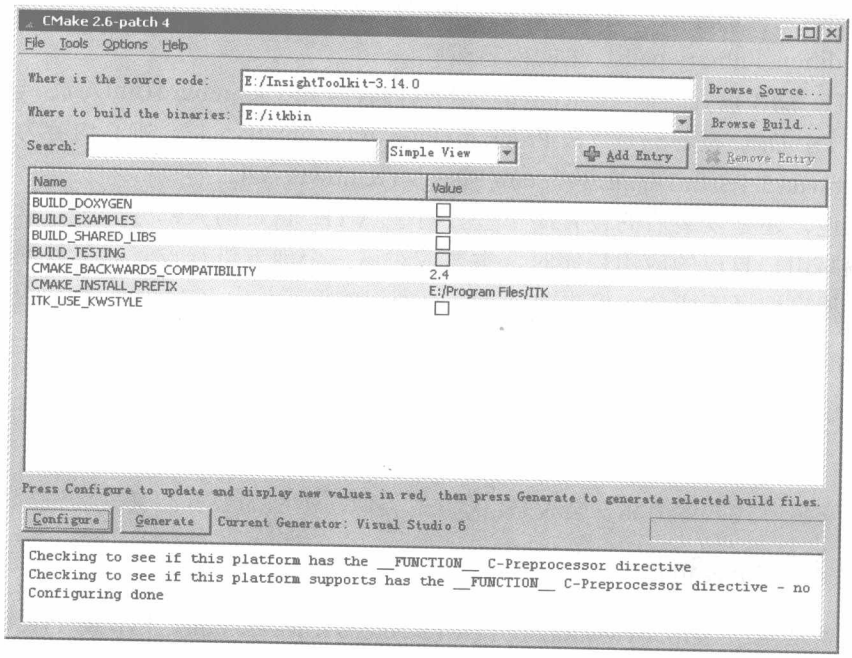


图 1-6　配置 ITK 安装

编译建立的所有库和可执行文件都会存放在一个 bin 子目录下，该 bin 子目录是在 CMake 中设置的二进制文件目录下的一个子目录，如 E: \InsightToolkit-3.14.0\bin。之后可以再选择 INSTALL 工程，并对其组建一次以进行安装。

至此，ITK 的安装过程就结束了。

1.2.3　测试安装结果

和 VTK 的测试类似，可以对 ITK 源代码路径下的 Examples/Installation/HelloWorld.cxx 运行 CMake 来进行测试，如图 1-7 所示。

测试操作过程与 VTK 完全一样，这里就不详述了。最后运行的结果是在命令提示符窗口中输出字符串“ITK Hello World !”，如图 1-8 所示。