

# 湖南科技大学课程教案

## (章节、专题首页)

授课教师: 王志喜

职称: 副教授

单位: 计算机科学与工程学院

课程名称	计算机图形图像技术
章节、专题	计算机图形学概述
教学目标及基本要求	了解计算机图形学的研究内容和应用领域, 熟悉用Dev-C++开发OpenGL应用的方法。
教学重点	用Dev-C++开发OpenGL应用
教学难点	用Dev-C++开发OpenGL应用
教学内容与时间分配	(1) 计算机图形学的研究对象和内容(0.4课时) (2) 计算机图形学的部分应用领域(0.3课时) (3) OpenGL简介(0.3课时) (4) 用Dev-C++开发OpenGL应用(1课时) 共计2课时。
习题	第1.5节(练习题)。

# 第1篇 计算机图形学

## 第1章 计算机图形学概述

### 1.1 计算机图形学研究的内容

#### 1.1.1 计算机图形学的研究对象

## 1. 图形的含义

计算机图形学的研究对象是图形。广义的图形是指能够在人的视觉系统中形成视觉印象的客观对象。所以，以下所列都可以称为图形。

- 人眼看到的自然景物。
- 用摄像机、照相机等获得的照片和图片。
- 用绘图机或绘图工具绘制的工程图、设计图和方框图。
- 各种人工美术绘画、雕塑品。
- 用数学方法描述的图形（包括几何图形、代数方程、分析表达式或列表所确定的图形）。

## 2. 图形的构成要素

- 几何要素：刻画形状的点、线、面、体等。
- 非几何要素：反映物体表面属性和材质的灰度、颜色等。

例如，方程  $x + y = 1$  确定的图形由满足该方程且具有颜色信息的点构成。

### 3. 图形的表示方法

一般来说，在计算机中有下列两种表示图形的方法。

(1) 参数法。用图形的形状参数和属性参数来表示图形，参数法描述的图形叫做参数图或简称为图形。

- 形状参数。指描述图形的方程或分析表达式的系数、线段的端点坐标或多边形的顶点坐标等。
- 属性参数。包括颜色和线型等。

(2) 点阵法。通过列出图形中所有的点来表示图形，点阵法描述的图形叫做像素图或图像。点阵法强调的是图形由哪些点组成，每个点具有怎样的颜色。

## 1.1.2 计算机图形学的研究内容

### 1. 图形的输入

如何开发和利用图形输入设备及相关软件把图形输入到计算机中，以便进行各种处理。

### 2. 图形的处理

包括对图形进行变换（如几何变换、投影变换）和运算（如图形的并、交、差运算）等处理。

### 3. 图形的生成和输出

如何将图形的特定表示形式转换成图形输出系统便于接受的表示形式，并将图形在显示器或打印机等输出设备上输出。

### 1.1.3 计算机图形学与相关学科的关系

与计算机图形学密切相关的学科主要有图像处理、计算几何、计算机视觉和模式识别等。它们之间的关系如图1-1所示。

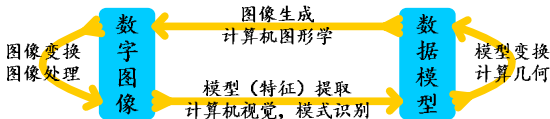


图1-1 计算机图形学与相关学科的关系

- 计算机图形学着重讨论怎样将数据模型变成数字图像。
- 图像处理着重研究图像的压缩存储和去除噪音等问题。
- 模式识别重点讨论如何从图像中提取数据和模型。
- 计算几何着重研究数据模型的建立、存储和管理。

随着技术的发展和应用的深入，这些学科的界限越来越模糊，各学科相互渗透、融合。一个较完善的应用系统通常综合利用了各个学科的相关技术。

## 1.2 计算机图形学的部分应用领域<sup>14</sup>

### 1.2.1 计算机辅助设计与制造

计算机辅助设计与制造是计算机图形学在工业界应用最重要、最成功的领域，广泛应用于飞机、汽车、船舶的外形设计、超大规模集成电路（VLSI）设计以及建筑、服装、印染和玩具设计等领域。

应用CAD系统进行设计，不仅可以获得产品的精确表示和显示结果，还可以在计算机中建立对象的数据模型，对它进行各种性能分析计算，设计人员可以根据计算结果对产品设计进行修改。

应用CAD系统进行设计，可以将制造过程与设计结果联系起来，设计结果直接传送至后续工艺进行加工处理，大大缩短了设计周期，降低了设计成本。

## 1.2.2 科学计算可视化

科学计算可视化就是利用计算机图形生成技术，将科学及工程计算中的计算数据和测量数据等以图形的形式显示出来，使人们能观察到用常规手段难以观察到的自然规律和自然现象。

目前，可视化技术已广泛应用于流体力学、有限元分析、医学、天气预报、海洋和空间探测等领域，用于生成宇宙飞船表面的气流、雷暴雨的数值模型、金属内部断裂的传递研究、空气薄层的流体密度的图形、数据集的交叉切片、蛋白质建模、分子结构的立体视图、海平面模型、空气污染情况等。



## 1.2.3 虚拟现实

### 1. 含义

虚拟现实技术是一种利用计算机生成逼真的三维虚拟环境，通过自然技能使用传感设备与虚拟环境相互作用的新技术。

虚拟现实技术将模拟环境、视景系统和仿真系统合三为一，利用头盔显示器、图形眼镜、数据衣服、立体声耳机、数据手套及脚踏板等传感装置，把操作者与计算机生成的三维虚拟环境连结在一起。

### 2. 应用范围

虚拟现实技术的应用范围很广，包括航空航天、建筑、医疗、娱乐、教育等领域。例如，建筑设计师可以运用虚拟现实技术向客户提供三维虚拟模型，外科医生可以在虚拟的病人身上施行一种新的外科手术。

## 1.2.4 计算机艺术

### 1. 用计算机软件从事艺术创作

可用于艺术创作的软件很多，如二维平面的画笔程序（如CorelDraw, Photoshop, PaintShop）、专门的图表绘制软件（如Visio）、三维建模和渲染软件包（如3DMAX, Maya）以及一些专门生成动画的软件（如Alias, Softimage）等。

### 2. 优点

上述软件不仅提供多种风格的画笔画刷，而且提供多种多样的纹理贴图，甚至能对图像进行雾化，变形等操作，很多功能是一个传统艺术家无法实现的。

### 3. 缺点

传统艺术的一些效果是上述软件不能达到的，比如钢笔素描、中国毛笔书法等的效果，而且在传统绘画中许多个人风格的效果也是上述软件无法企及的。

## 1.2.5 计算机动画

### 1. 计算机动画的含义

计算机动画是指用绘制程序生成一系列景物画面，其中当前帧画面是对前一帧画面的部分修改。

计算机动画是计算机图形学和艺术相结合的产物，它综合利用计算机科学、艺术、数学、物理学和其它相关学科的知识在计算机上生成绚丽多彩的连续画面，给人们提供了一个充分展示个人想象力和艺术才能的新天地。

### 2. 计算机动画的应用范围

计算机动画不仅可以应用于电影特技、商业广告、电视片头、动画片、游艺场所，还可以应用于计算机辅助教育、军事、飞行模拟等。

## 1.2.6 图形用户接口

用户接口是人们使用计算机的第一观感。一个友好的图形化的用户界面能够大大提高软件的易用性。

1980年代，X-Window标准的提出，苹果公司图形化操作系统的推出，以及微软公司Windows操作系统的普及，标志着计算机图形学已经全面融入到了计算机的方方面面。操作系统和应用程序中的图形和动画比比皆是，程序直观易用。很多软件几乎可以不看说明书，而根据图形或动画的提示就可以使用这些软件。

## 1.3 OpenGL简介

### 1.3.1 OpenGL概述<sup>1</sup>

计算机图形学经过几十年的发展，已经达到了比较高的水平，目前已经有很多种图形软件开发技术（本书只介绍OpenGL）。利用这些开发技术，编写图形程序就容易多了，不必从底层开始，而只需将精力集中到图形程序本身。

OpenGL（Open Graphics Lib）是一套三维图形处理库，也是该领域事实上的工业标准。

OpenGL独立于硬件、操作系统和窗口系统，能运行于不同操作系统的各种计算机，并能在网络环境下以客户/服务器模式工作，是专业图形处理、科学计算等高端应用领域的标准图形库。

以OpenGL为基础开发的应用程序可以十分方便地在各种平台间移植；OpenGL与C/C++接合紧密，便于实现图形的相关算法，并可保证算法的正确性和可靠性；OpenGL使用简便，效率高。

### 1.3.2 OpenGL的常用组成部分

目前，OpenGL常用的组成部分主要有以下三部分。

(1) OpenGL核心库 (GL)。包含OpenGL最基本的命令函数，提供建立几何模型、进行坐标变换、产生光照效果、进行纹理映射和雾化等操作。函数名的前缀为gl。

(2) OpenGL实用库 (GLU)。利用低层OpenGL命令编写的一些执行特殊任务的函数，如纹理映射、坐标变换、NURBS曲线曲面等。函数名的前缀是glu。

(3) OpenGL实用函数工具包 (GLUT)。包括窗口操作函数、回调函数、创建复杂三维物体函数、菜单函数、程序运行函数等。函数名的前缀为glut。

在Windows中，相关的库以动态链接库的形式存在。在Dev-C++中，libopengl32.a、libglu32.a、libglut32.a分别表示opengl库、实用库和实用函数工具包，相应的头文件分别是gl.h、glu.h和glut.h

本书使用FreeGLUT替代GLUT，libfreeglut.dll.a和freeglut.h分别是实用函数工具包和相应的头文件。

### 1.3.3 一个简单的OpenGL程序

下列程序绘制了一个填充的白色矩形。运行结果如图1-2所示。相关函数在后续章节说明。

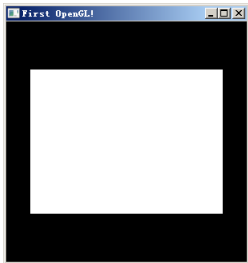


图1-2 一个简单例子

```
// FirstGL.c
#include<GL/freeglut.h>

void Paint() // 对象的描述
{
    glClear(GL_COLOR_BUFFER_BIT); // 清除颜色缓冲区
    glRectf(-0.8, -0.6, 0.8, 0.6); // 由顶点(-0.8, -0.6)和(0.8, 0.6)定义矩形
    glFlush(); // 强制OpenGL命令序列在有限的时间内完成执行
}

int main(int argc, char *argv[])
{
    glutInit(&argc, argv); // 初始化GLUT, 记录main()的参数
    glutCreateWindow("First OpenGL!"); // 创建程序窗口, 指定窗口标题
    glutDisplayFunc(Paint); // 指定场景绘制循环函数, 必需
    glutMainLoop(); // 开始循环执行OpenGL命令
}
```



## 1.4 用Dev-C++开发OpenGL应用<sup>♫</sup>

### 1.4.1 必备工具的下载和安装

#### 1. Dev-C++的下载和安装

Dev-C++原始版本为Bloodshed Dev-C++，最高版本号为4.9.9.2，4.9.9.2以后的版本为Orwell Dev-C++。Orwell Dev-C++的下载地址为

<http://orwelldevcpp.blogspot.com/>

从该网站下载最新的不带编译器的Orwell Dev-C++。下载完成后直接运行安装程序，按照提示完成安装。注意，安装完成以后不要立即运行Orwell Dev-C++。

## 2. 编译器的下载和安装

使用Orwell Dev-C++编写程序通常使用GCC编译器。这里选用MinGW-W64 GCC-8.1.0的i686-posix-dwarf（编译OpenCV必须使用POSIX线程支持的版本），下载地址是

[https://sourceforge.net/projects/mingw-w64/files/Toolchains%20targetting%20Win32/Personal%20Builds/mingw-builds/8.1.0/thread-s-posix/dwarf/i686-8.1.0-release-posix-dwarf-rt\\_v6-rev0.7z](https://sourceforge.net/projects/mingw-w64/files/Toolchains%20targetting%20Win32/Personal%20Builds/mingw-builds/8.1.0/thread-s-posix/dwarf/i686-8.1.0-release-posix-dwarf-rt_v6-rev0.7z)

下载完成后解压该文件，将解压得到的mingw32文件夹复制到Dev-C++的安装目录（例如C:\Dev-Cpp）。

然后运行Orwell Dev-C++，按照提示完成Orwell Dev-C++的配置和编译器的默认配置。

**【注】**为了方便编译FreeGLUT和OpenCV，这里需要将MinGW的可执行文件目录（例如C:\Dev-Cpp\mingw32\bin）加入Path环境变量，并且，还需要保证Path环境变量中只有这一份GCC的可执行文件目录。

### 3. cmake-3.13.2-win32-x86

用于编译FreeGLUT和OpenCV，下载地址为

<https://github.com/Kitware/CMake/releases/download/v3.13.2/cmake-3.13.2-win32-x86.msi>

下载完成后直接运行安装程序，按照提示完成安装。

## 1.4.2 FreeGLUT的编译和安装

### 1. 下载源程序

OpenGL和GLU是Windows的标准组成部分，而且libopengl32.a、libglu32.a已经集成到了GCC中，只有GLUT需要另外下载安装。由于GLUT最后一次更新是2001年，这里使用2019年发布的FreeGLUT 3.2.1替代GLUT。FreeGLUT 3.2.1源程序的下载地址为

<https://sourceforge.net/projects/freeglut/files/freeglut/3.2.1/freeglut-3.2.1.tar.gz/download>

下载完成后解压缩该文件，记住解压缩位置，例如D:\freeglut-3.2.1。

## 2. 配置项目

使用CMake完成。

(1) 启动CMake (cmake-gui)。可以从开始菜单启动。

(2) 分别指定FreeGLUT源文件位置(例如D:\freeglut-3.2.1)和目标文件位置(例如D:\glut321)，如图1-3所示。

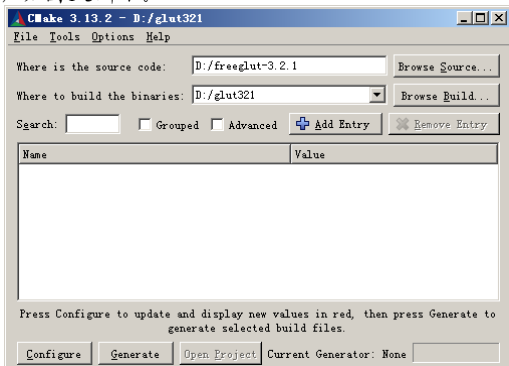


图1-3 源文件和目标文件位置

(3) 点击Configure，弹出选择环境和编译器的对话框。在列表中选择MinGW Makefiles，编译器选用本机默认的编译器。如图1-4所示。

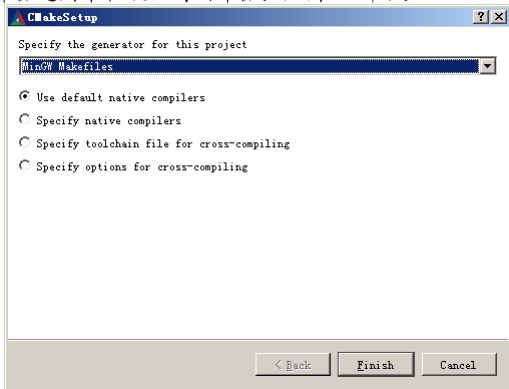


图1-4 选择环境和编译器

(4) 点击Finish，等待选项窗格中出现一片红色的选项。如图1-5所示。

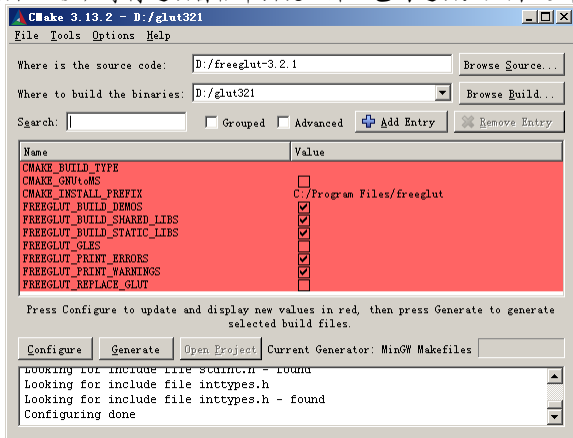


图1-5 初始选项

(5) 找到CMAKE\_INSTALL\_PREFIX选项，将该选项的值改成install。如图1-6所示。

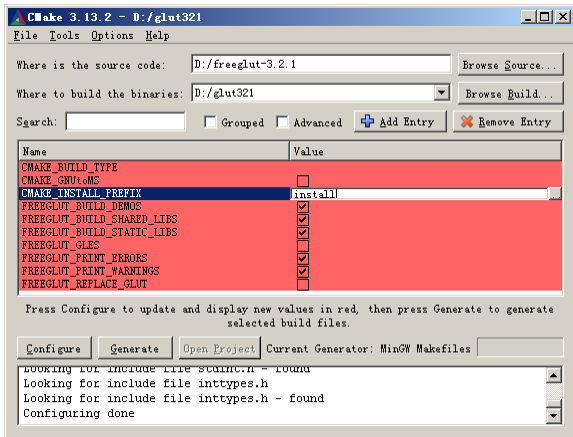


图1-6 修改CMAKE\_INSTALL\_PREFIX选项



(6) 再次点击Configure，等待红色选项消失，显示Configuring done。如图1-7所示。

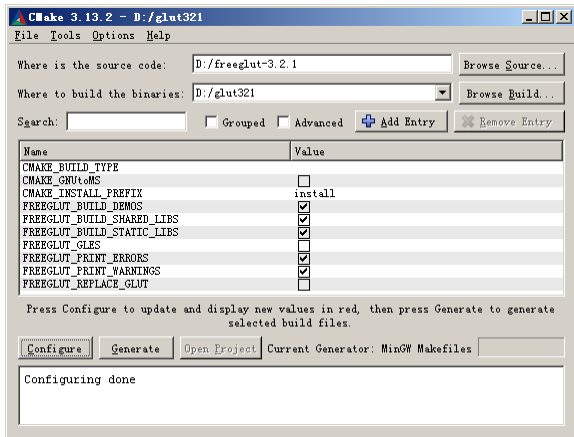


图1-7 完成配置

(7) 点击Generate，生成项目相关文件。完成后会显示Generating done。如图1-8所示。

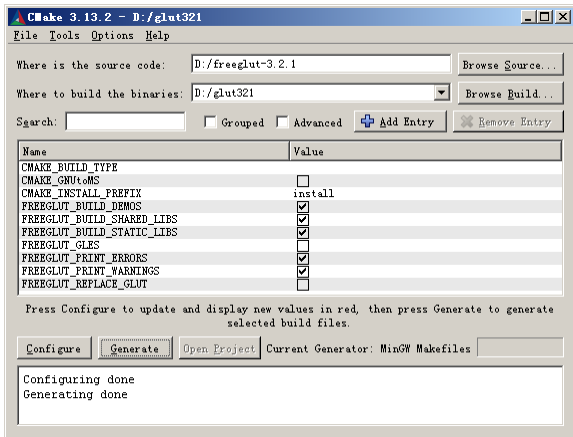


图1-8 生成相关文件

### 3. 编译项目

打开命令行窗口，将工作目录切换到目标文件夹（例如D:\glut321）。输入命令mingw32-make install，开始编译。如图1-9所示。

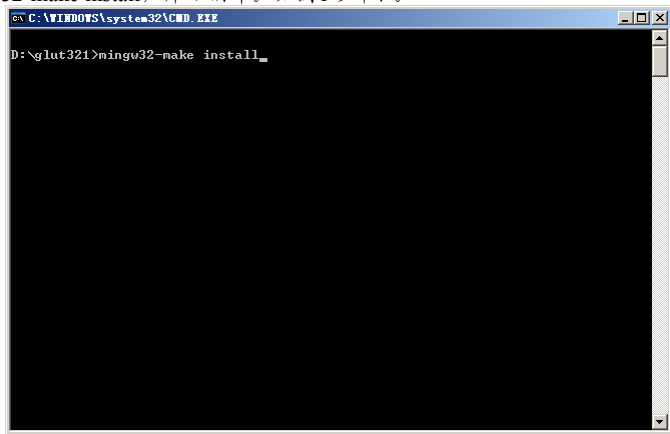


图1-9 开始编译

编译完成以后，mingw32-make install命令将自动完成相关配置。如图1-10所示。

```
C:\WINDOWS\system32\CMD.EXE
[ 98%] Linking C executable bin\timer.exe
[ 98%] Built target timer
Scanning dependencies of target timer_static
[ 99%] Building C object CMakeFiles/timer_static.dir/progs/demos/timer/timer.c.o
bj
[100%] Linking C executable bin\timer_static.exe
[100%] Built target timer_static
Install the project...
-- Install configuration: ""
-- Installing: D:/glut321/install/lib/libfreeglut.dll.a
-- Installing: D:/glut321/install/bin/libfreeglut.dll
-- Installing: D:/glut321/install/lib/libfreeglut_static.a
-- Installing: D:/glut321/install/include/GL/freeglut.h
-- Installing: D:/glut321/install/include/GL/freeglut_ucall.h
-- Installing: D:/glut321/install/include/GL/freeglut_ext.h
-- Installing: D:/glut321/install/include/GL/freeglut_std.h
-- Installing: D:/glut321/install/lib/pkgconfig/freeglut.pc
-- Installing: D:/glut321/install/lib/cmake/FreeGLUT/FreeGLUTTargets.cmake
-- Installing: D:/glut321/install/lib/cmake/FreeGLUT/FreeGLUTTargets-noconfig.cmake
-- Installing: D:/glut321/install/lib/cmake/FreeGLUT/FreeGLUTConfig.cmake
-- Installing: D:/glut321/install/lib/cmake/FreeGLUT/FreeGLUTConfigVersion.cmake
D:\glut321>
```

图1-10 完成配置

## 4. 安装

完成以后，所有目标文件位于目标文件夹的install子目录中（例如，如图1-11所示的D:\glut321\install）。可以从该目录或其子目录中找到bin、include和lib这三个文件夹，将这三个文件夹复制到待安装目录即可（例如C:\Dev-Cpp）。

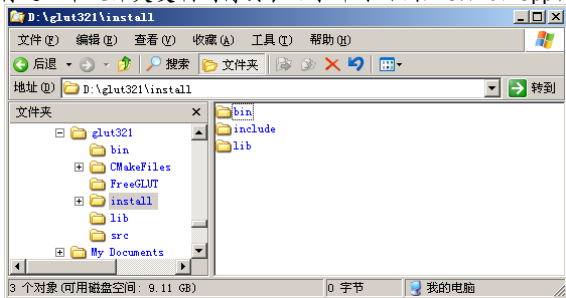


图1-11 目标文件位置

## 1.4.3 编译器设置与IDE的使用

### 1. 编译器设置

(1) 设置方法。从Tools菜单中选择Compiler Options打开Compiler Options，在Compiler Options中完成编译器设置。如图1-12所示。

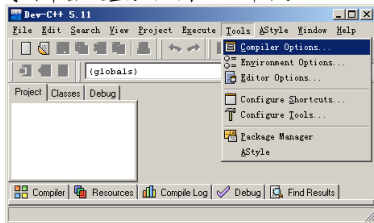


图1-12 编译器设置

(2) 语言支持。后续章节中有很多程序使用到了C99和C11新增的特性，可以使用选项-std=c11使得编译器支持C99和C11（如图1-13所示）。

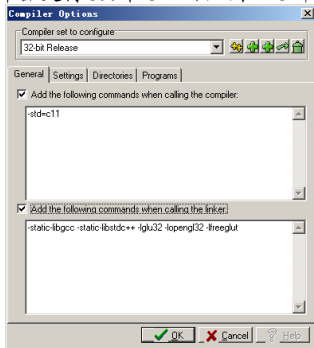
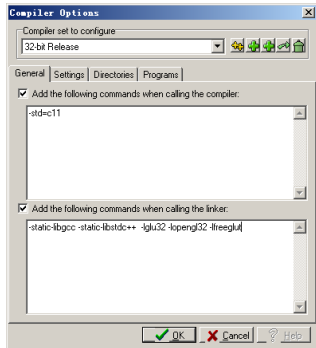


图1-13 语言支持和链接库设置

(3) 链接库。在Dev-C++中编译OpenGL应用程序需要的静态链接库包括libopengl32.a、libglu32.a和libfreeglut.dll.a，可以使用选项-lopengl32、-lglu32和-lfreeglut设置这些链接库文件（如图1-13所示）。





(4) 可执行文件路径。首先从Directories页中选择Binaries，然后将编译和运行应用程序所需要的可执行文件的路径添加到列表中，最后调整这些路径的顺序（如图1-14所示）。

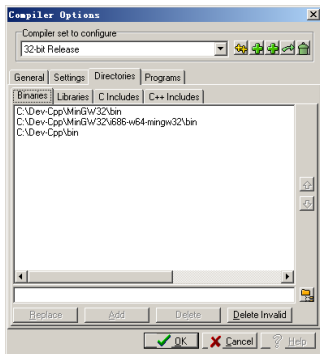


图1-14 可执行文件路径设置

(5) 链接库文件路径。首先从Directories页中选择Libraries，然后将编译应用程序所需要的链接库文件的路径添加到列表中，最后调整这些路径的顺序(如图1-15所示)。

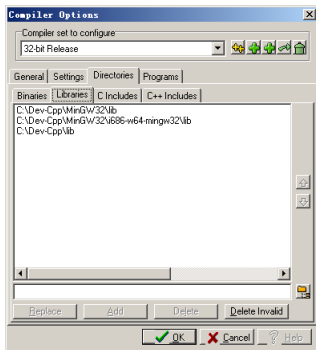


图1-15 链接库文件路径设置

(6) C Include文件路径。首先从Directories页中选择C Includes，然后将编译应用程序所需要的C Include文件的路径添加到列表中，最后调整这些路径的顺序（如图1-16所示）。

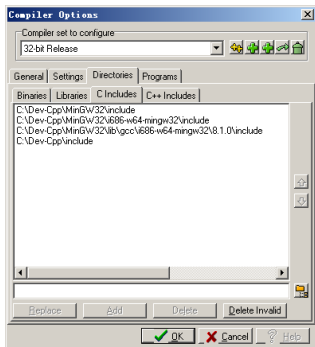


图1-16 C Include文件路径设置

(7) C++ Include文件路径。首先从Directories中选择C++ Includes，然后将编译应用程序所需要的C++ Include文件的路径添加到列表中，最后调整这些路径的顺序（如图1-17所示）。

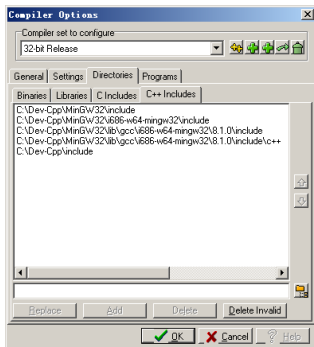


图1-17 C++ Include文件路径设置

(8) 编译器文件。在Programs页中指定编译器各主要组成文件的文件名(如图1-18所示)。

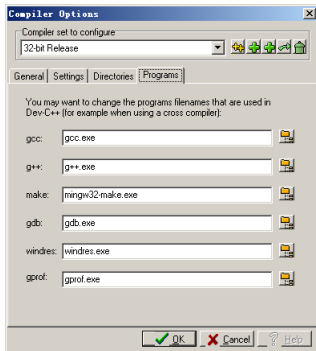


图1-18 编译器文件路径设置

## 2. IDE的使用方法

在完成开发环境的安装和编译器的设置以后，使用Dev-C++开发OpenGL应用的方法非常简单，打开Dev-C++集成开发环境，创建一个空白源程序文件，编写源程序、编译、调试、运行。

## 1.5 练习题

请查阅教材和互联网完成。

1. 第一届ACM SIGGRAPH会议是哪一年在哪里召开的？
2. 计算机图形学之父是谁？
3. 列举一些计算机图形学的应用领域（至少5个）。
4. 简要介绍计算机图形学的研究内容。
5. 简要说明计算机图形学与相关学科的关系。
6. 简要介绍几种计算机图形学的相关开发技术。
7. 图形的构成要素有哪些？
8. 计算机图形学的最高奖以哪位科学家的名字命名，获得第一届和第二届该奖的分别是哪些科学家？