文章编号:1674-7070(2014)06-0530-09

于连庆[1]

基于触控操作方式的大气科学数据可视化系统技术研究与实现

摘要

设计并实现了一种基于触控操作方式,运行于移动计算设备上的大气科学数据可视化软件系统 MeteoExplorer Touch,它是世界上首款提供触控操作方式体验,运行于 Windows8 和 Windows RT 操作系统上的大气科学数据可视化软件.该软件系统的技术特色有:1)针对 Windows 8 和 Windows RT 操作系统所带来的崭新特性进行设计,实现了诸如使用手指滑动、捏掐伸展完成底图的缩放和漫游,在操作系统从低功耗运行状态下再次激活后恢复用户会话状态等功能,从而为广大气象科研和业务人员在平板电脑上分析、处理气象数据提供了全新的体验;2)MeteoExplorer Touch 不仅仅是一个图片查看器,它还实现了主流桌面版专业气象软件所提供的大部分功能,包括数据资料解析、观测场客观分析、物理量计算、图形产品生成、地理信息和图层管理等;3)MeteoExplorer Touch 注重程序性能和用户体验.为实现这一目标,软件采用 C++语言开发功能,使用 XAML 标记语言设计用户界面,使用 DirectX 11 硬件加速渲染技术进行图形显示,因此,它具有原生程序所特有的性能高、速度快、系统资源需求低的优点.

关键词

触控操作;移动计算;气象数据;可视化系统;软件系统

中图分类号 TP31

文献标志码 A

收稿日期 2013-09-12

作者简介

于连庆,男,博士,高级工程师,主要从事气象预报业务系统的开发.yulq@ cma.gov.cn

1 国家气象中心,北京,100081

0 引言

随着信息科学技术的飞速发展,新兴的具备屏幕触控功能的计算设备,如智能手机、平板电脑和触屏式一体机等已经与传统的个人计算机一样,成为一个重要的计算平台.这些移动计算设备凭借自身携带方便、待机时间长、触控操作方式方便灵活等优点,在个人计算和企业计算领域发挥着越来越大的作用.

在国内外大气科学研究领域和气象预报与服务业务中,移动计算设备与触屏式计算设备的使用率逐年增加,并且发挥着越来越大的作用.例如德国的 MeteoGroup 公司开发的三维气象信息软件 MeteoEarth 和 WeatherPresenter 的运行环境,均是以超大屏幕作为显示设备,预报员以手指触控的操作方式完成天气分析、产品制作等常见业务[1].美国 CNN 电视台天气预报节目中,主持人通过手指操纵屏幕上出现的各种气象图形向观众传达气象预报信息.在 2010 年的上海世博会上,由国家气象中心开发的基于触控操作的三维气象信息显示系统充分展示了气象科技对社会、经济发展的推动力量,引起了参会人员的极大兴趣.然而,上述介绍的系统都是面向特定业务和应用的专业平台,而不是面向广大气象科研和业务人员的通用系统,因此研究针对普通用户、低端计算设备的系统技术是十分必要的.

要发挥移动计算设备的作用,软件系统必须跟进支持,这体现在将专门为桌面计算机开发的大气科学软件移植到移动计算设备平台上.移植的必要性如下:

1)由于移动计算设备操作系统的限制,导致传统的桌面气象软件在这些设备上根本无法运行,因此需要软件开发人员针对特定设备及其操作系统进行开发.目前,国内外气象工作者经常使用的桌面气象软件如 MICAPS[2]、AWIPS[3]、GrADS[4]、NCL[5]、IDV[6]、MetView[7]和 Google 地球[8]等都不能在移动设备上运行.

2)传统的桌面气象软件针对鼠标、键盘这些输入设备而设计,而移动计算设备主要使用手指触控和手写笔作为主要输入设备,因此,即使传统的桌面气象软件能够运行在移动设备上,输入方式的差别导致在移动设备上运行桌面软件时,用户的工作效率必然受到极大的影响.

3) 与桌面计算机、工作站和服务器相比,移动计算设备硬件资源十分有限,具体体现在处理器的计算速度较低,内、外存储器容量较小,图形适配器的渲染效率不高等.如果不对程序进行优化,那么在移动计算设备上运行传统的桌面气象软件将无法保证较高的运行效率和流畅的用户体验.

本文的主要工作是针对这一移植过程中存在的主要问题,从系统框架设计、用户交互界面和技术实现难点等几个方面入手,进行了深入的研究和探讨.为了体现系统设计和技术路线的可行性,本文实现了一种运行于 Windows 8 和 Windows RT 操作系统上,用于科研交流用途的软件系统 MeteoExplorer Touch,该软件实际上是将桌面版软件 MeteoExplorer 移植到移动计算平台上的一项尝试.通过这种努力,希望能够在开拓气象信息移动计算技术的同时,让气象科研和业务人员享受到新型移动计算设备所带来的全新体验,提高自己的工作效率.

1 可视化系统的技术特色和主要功能

1.1 技术特色

本文所实现的可视化系统 MeteoExplorer Touch 软件在技术实现上具有以下特色:

1) 针对具有触控功能屏幕的移动计算设备设计,为用户提供了使用触控方式处理气象数据的全新体验.

MeteoExplorer Touch 支持大部分常用的触控手势,包括轻敲(tap)、按下并保持(hold)、捏掐与伸展(pinch and stretch)、滑动(slide)以及刷卡式滑动(swipe),并使用这些手势作为用户与系统交互的主要方式,来实现诸如视图的漫游与缩放、功能控件的操纵等功能.

MeteoExplorer Touch 还具备移动设备所特有的从待机状态自我激活的功能.众所周知,为了保持移动设备的续航时间,如果一段时间内用户没有对设备进行操作,那么操作系统一般会关闭当前所有正在运行的程序,然后进入低功耗待机状态.当用户再次操纵设备时,操作系统会退出待机状态,重新启动待机前关闭的所有程序.这样给用户一种系统始终在无间断运行的感觉.在 Windows 8 和 Windows RT 下,操作系统是通过给应用程序发送消息的方式通知应用程序在启动时恢复设备待机前的会话状态.至于如何完成恢复待机前会话状态的任务需要应用程序自己完成.MeteoExplorer Touch 具备这一功能,即它能够自动打开所有的图层,同时保持所有图形对象的显示属性.

此外,MeteoExplorer Touch 的用户界面设计遵循 Windows 应用商店程序的设计风格,这样可以给用户带来一致性的感觉.图 1 给出了在 MeteoExplorer Touch 中打开多个图层的显示结果.由图 1 可见,

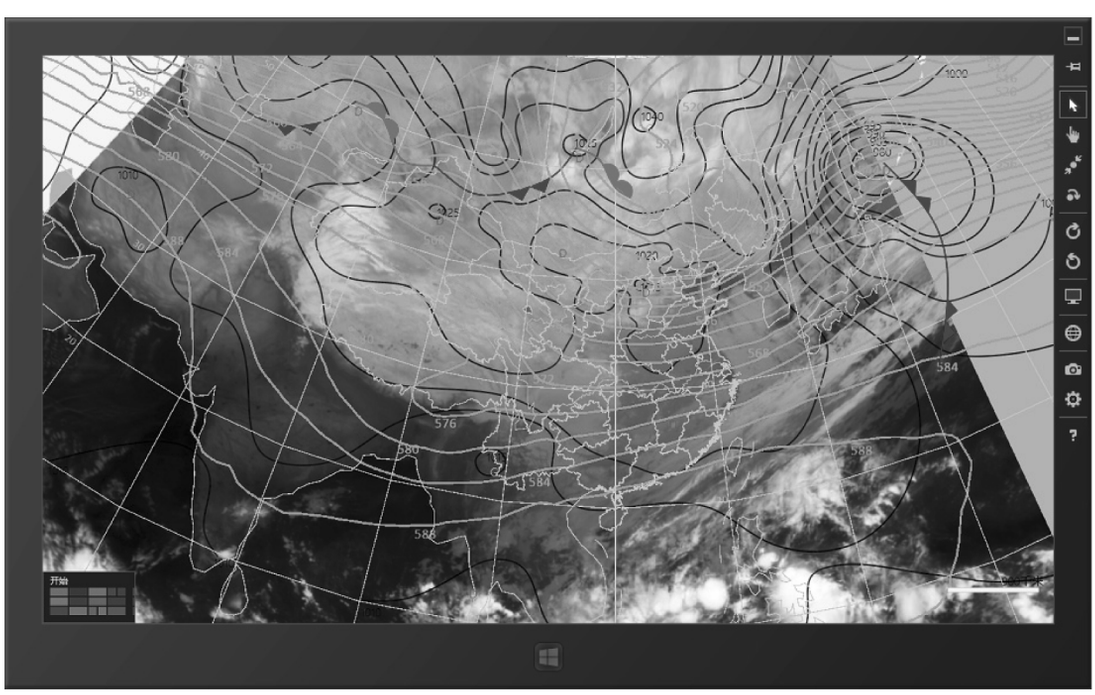


图 1 MeteoExplorer Touch 的用户界面

Fig. 1 User interface of MeteoExplorer Touch

Windows 应用商店程序设计风格的主要特点是程序主窗口占据整个屏幕,传统 Windows 程序所特有的标题栏、菜单栏、工具栏和状态栏等控件默认情况下处于隐藏状态,程序支持键盘、鼠标、手指触控和手写笔输入.

2) MeteoExplorer Touch 提供了桌面版气象软件中的常见功能,是一款能够为用户带来工作效率的工具软件,而不仅是一个文字或图像浏览器.

为了让软件系统能够跨越多个平台,本文采用了层次化、结构化、模块化的设计思路,以提高代码的可复用比例.实际上,除了使用 XAML 实现的用户界面交互功能之外,MeteoExplorer Touch 与桌面版 MeteoExplorer 软件共用了其他所有功能,包括地理信息、数据资料解释、图形图像渲染、气象算法等,此设计可以更方便地将系统功能移植到新的平台上.

3) 使用 C++原生语言和 DirectX 硬件加速渲染技术,保证程序的运行效率和图形显示性能.

随着气象科学技术的迅速发展,科研业务工作人员对气象软件的图形显示速度和效率提出了更高的要求,要求能够在尽可能短的时间内完成大量数据的分析和可视化功能.MeteoExplorer Touch 从设计之初就强调程序运行性能,目的是给用户带来流畅的体验,从而提高工作效率.因此,本文未选择 C#或者 Visual Basic 这样的解释类型的程序设计语言,而是使用 C++这样的原生语言进行开发,以此希望降低程序对系统资源的要求.此外,为了能够快速地处理、显示气象数据,MeteoExplorer Touch 使用了 Direct3D 11 硬件加速渲染技术,以提供较快的图形图像显示速度,从而保证流畅的用户体验.

1.2 主要功能

MeteoExplorer Touch 的主要功能有:

1) 图层管理功能,包括图层的显隐、删除、刷新、属性设置、翻页与动画;

2) 地面、探空观测场的客观分析;

3) 格点场的等值线分析和填色;

4) 风场的流线分析;

5) 常用的物理量计算;

6) 常见数据资料的解释,支持的数据格式有 MICAPS、NetCDF、GRIB1/GRIB2、GrADS 等;

7) 卫星云图资料显示和动画功能;

8) 地图缩放、漫游和投影切换;

9) 页面布局及设置;

10) 缩放到指定区域功能;

11) 闭合区域(国界、省界)内显示功能;

12) 屏幕截图保存功能,格式为 BMP、JPG、PNG;

13) 支持简体中文和英文 2 种用户界面语言.

以上功能的具体介绍可参考 MeteoExplorer Touch 软件中附带的用户手册文档.

2 系统框架设计

在软件系统开发中,当需要把已有应用程序移植到新的计算平台上时,开发人员总是希望能够尽可能多的利用已有代码,因此,代码的可复用性决定了程序移植的难度和工作量.

解决应用程序的跨平台问题一般有以下 2 种方案:

1) 第一种方案是使用第三方提供的跨平台开发框架,如 J2EE、Qt 和 ArcMap 等.这些开发框架提供了从底层数据结构、图形显示到高层用户界面交互、数据库存取等在内的众多功能.这种方案的优点是一次性程序开发完毕后,可以部署到开发框架所支持的所有计算平台上,缺点是应用程序过度依赖开发框架,继承了开发框架中的问题和错误,程序代码由于包含了开发框架中未用到的功能而导致臃肿,并且无法把应用程序移植到开发框架所不支持的计算平台上.

2) 第二种方案是分析应用程序功能,按照层次化、模块化的原则设计系统框架.根据功能之间的逻辑关系将系统分解为多个层次,每个层次又包括多个相互独立的模块.除了用户界面交互功能使用第三方开发框架实现之外,其他的功能模块使用像 C/C++这样的跨平台编程语言实现.这种方案的优点是程序内核短小精悍,基础功能不依赖于第三方开发框架,能够灵活地把应用程序移植到任意计算平台,并且选择目标平台上最好的开发框架实现用户界面交互功能,缺点是当应用程序所支持的计算平台数目增加时,需要重复实现用户界面交互功能,从而增加了开发的工作量.

当前,一些技术实力雄厚,运营资金充足的企业往往选择第二种方案,因为这些企业更为看重自身的技术积累,减少对第三方开发框架的依赖性,从而增强市场竞争实力.

如图 2 所示,MeteoExplorer Touch 软件系统的框架结构采用了第二种方案,主要原因有:

1) 当前还没有任何一个第三方开发框架能够

支持多个移动计算平台,如 IOS、Android 和 Windows 8(包括 Windows Phone 8 和 Windows RT);

2) 开发者希望系统核心功能能够应用在尽可能多的计算平台上,或者以多种服务方式提供,例如对于那些没有任何开发框架支持,甚至没有图形用户界面的系统,程序功能可以终端命令行的形式提供,或者以面向服务的后台服务方式提供.

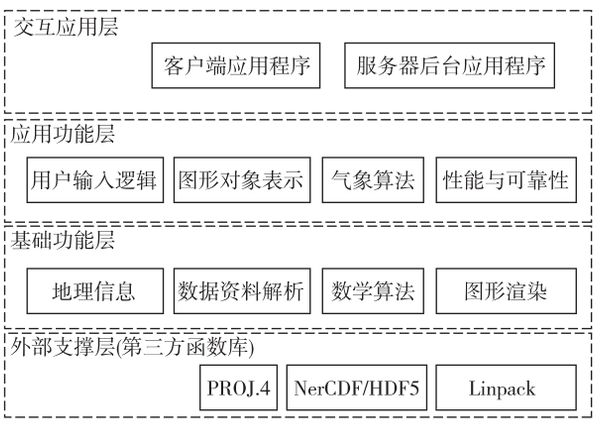


图 2 MeteoExplorer Touch 软件系统的框架结构

Fig. 2 System architecture of MeteoExplorer Touch

软件系统的框架包括 4 个层次,按照从底到上的顺序是外部支撑层、基础功能层、应用功能层和交互应用层.外部支撑层包括软件系统使用的所有第三方函数库,如提供地图投影功能的 PROJ.4,提供常用科学数据解析功能的 NetCDF、HDF5 和数学算法库 Linpack 等.基础功能层按照功能可以分为地图投影、数据资料解析和图形图像渲染等多个模块.应用功能层也按照功能划分为用户输入逻辑分析、图形对象表示、大气科学算法实现、性能与可靠性等多个模块.交互应用层是实现应用程序的地方,它既可以是提供图形用户界面的桌面端应用程序或 Web 应用程序,也可以是以后台进程形式运行的服务程序.

采用层次化、模块化原则设计的系统框架具有以下优点:

1) 位于下一层次的模块提供服务接口供上层模块调用,反之则不然.这种设计方法约束了模块之间的依赖关系.

2) 位于同一层次内的模块之间是相互独立的关系,不存在调用关系,这样就减少了模块之间的耦合性.

3) 系统中任何一个模块提供的接口规范是固定的,但其实现功能的方法是可替换的.这便于开发者采用最新的技术替换系统中过时的技术,保持系统技术的先进性.

4) 系统中任何一个模块都可以根据应用程序是否需要该模块提供的功能来加载或卸载.例如,对于一个后台数据服务器,由于它不需要地图投影功能和图形渲染功能,因此在创建该程序时就可以不用加载这 2 个模块.带来的好处是程序代码精炼,运行效率高.

3 跨平台图形渲染引擎的设计与实现

随着气象科学技术的迅速发展,科研和业务工作人员对相关软件的图形显示速度和效率提出了更高的要求,要求能够在尽可能短的时间内完成大量数据的分析和可视化功能,同时要求能够将所显示的图形保存成矢量格式的图形文件,便于将业务产品或者研究成果进行演示和发表.这种客观需求要求一个软件系统的图形渲染引擎既要提供效率和多种功能,也要有跨平台的灵活性.

从开发人员的角度上看,单独使用一种图形渲染技术不存在很大的难度.例如可以选择 DirectX 中的 Direct3D 技术[9]、OpenGL 技术[10]和 GDI/GDI+技术[11],甚至可以直接使用集成开发框架(如 J2EE、.NET、Qt、MFC)里面封装的图形渲染技术,还可以使用更为高级的图形图像函数库,如 Magics.这些选择各有优缺点,如 Direct3D 技术在 Windows 操作系统下具有最好的兼容性,市面上销售的显示卡都支持 Direct3D,缺点是只支持 Windows 平台.OpenGL 技术则在跨平台上具有优势,即几乎所有平台都或多或少地支持 OpenGL 技术,缺点是各个平台之间实现的功能差异较大.例如在 Windows 操作系统下 Visual C++只支持到 OpenGL v1.2 版本,开发人员需要借助第三方函数库来实现对最新版本的支持.在手机等移动设备上只实现了 OpenGL 的一个子集 OpenGL ES.与之相反,在图形工作站上,厂商往往提供了各种 OpenGL 的扩展,以增强 OpenGL 的功能.GDI/GDI+技术则是 Windows 下矢量图形的基础.集成开发框架中的图形渲染技术抽象层次更高,使用起来更为方便,且与整体框架结合得更为紧密.

单一使用某一图形渲染工具库带来的问题包括:

1) 图形图像显示功能的实现受限于所使用的图形渲染工具库.如使用 GDI/GDI+时无法提供硬件加速功能.

2）当需要把应用程序移植到图形渲染工具库不支持的操作系统或硬件设备时,需要重新实现图形图像显示功能模块.如将使用 DirectX 的应用移植到 Unix 操作系统下时,需要使用其他图形渲染工具库(如 OpenGL)来实现图形图像显示功能.

使用集成开发框架中的图形渲染技术的问题是程序依赖性强,可移植性较弱.

3.1 图形渲染抽象层的概念设计与功能实现

针对单一使用某一图形渲染工具库带来的问题,同时考虑到目前多种操作系统、计算设备和图形渲染工具库并存的事实,本文在设计图形渲染引擎时提出了以下目标:

1）相对于计算平台、操作系统和图形渲染工具库的独立性,即图形渲染引擎的实现不依赖于某一计算平台、操作系统和图形渲染工具库.

2）可扩展性,即在需要时能够使用新的图形渲染工具库来实现图形渲染引擎,或者能够较容易地将图形渲染引擎移植到新的计算平台和操作系统上.

3）多态性,即根据用户选项或者程序需要,实时地在不同图形渲染工具库之间切换,以实现图形渲染引擎所提供的功能,同时保证图形图像的显示属性在切换后保持不变.

根据上述的设计目标,本文提出了图形渲染抽象层(Graphics Rendering Abstract Layer,GRAL)这一模型.模型的逻辑结构如图 3 所示.在图形渲染抽象层的设计中,本文使用了抽象工厂(Abstract Factory)设计模式[12].抽象工厂模式为创建一组在逻辑上相互关联或者相互依赖的实体对象提供了接口,同时隐藏了这些待创建实体对象的细节.抽象工厂设计模式为创建跨越不同平台的函数库提供了可能.

在实现图形渲染抽象层模型中,首先将所有实体对象的共同功能提取出来,对每一个功能加以抽象描述,作为抽象基类的一个接口,然后对每一个实体对象定义一个类作为抽象基类的子类,在每一个子类中需要实现接口所指定的功能,另外,还需要定义一个面向客户的接口类来调用抽象基类的接口函数,从而完成用户要求的功能.这样,客户只要调用接口类所提供的接口函数来实现自己需要的功能,而无需知道具体每一个实体对象内部是如何定义和实现的.

下面结合图 3 具体说明基于抽象工厂模式进行设计的图形渲染抽象层模型.图 3 中虚线表示引用

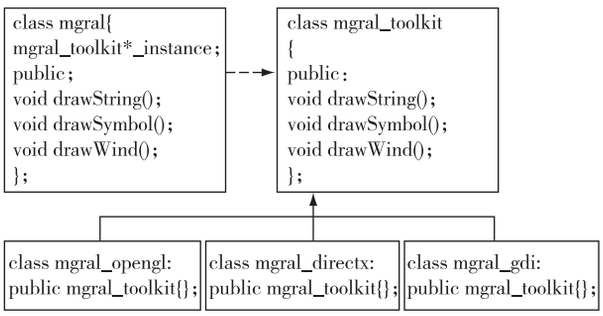


图 3 图形渲染抽象层模型的结构关系

Fig. 3 Data structure of the graphics rendering abstract layer

关系,实线代表 C++类继承关系.首先定义一个面向客户的功能接口类 mgral,它是图形渲染抽象层的抽象描述.该类中声明了所有实现客户需求功能的函数声明(在 COM 技术中称为接口).如显示字符串的函数 drawString、显示符号的函数 drawSymbol 等.在类 mgral 中,需要定义一个封装了图形函数工具库的抽象基类 mgral_toolkit 的指针型成员变量 instance.抽象基类 mgral_toolkit 同样声明了所有实现客户需求功能的函数声明,这些函数将在其子类中给出具体定义.下面的程序代码示例了类 mgral 与类 mgral_toolkit 之间的关系:

```
mgral_toolkit * instance;
void mgral::drawString(const char * str,float x,float y,float dx,
float dy,float theta,float rscale,float gscale)
{
  if (instance)
     instance -> drawString (str, x, y, dx, dy, theta, rscale,
gscale);
  else
    fprintf(stderr,"no toolkit selected\n");
}
```

在此例中,当客户需要在程序窗口绘制字符串时,它调用接口类 mgral 中提供的接口函数 drawString,在此函数的定义中,根据成员变量 instance 是否已被初始化,或调用抽象基类 mgral_toolkit 中的接口函数,或打印输出错误提示信息.

当变量 instance 被实例化后,它总是指向抽象基类 mgral_toolkit 某一子类的实例对象.由于一个时刻只能使用一种图形函数工具库进行绘图,因此在定义变量 instance 时使用了设计模式中的单态(singleton)模式.采用单态模式的优点是保证任意时刻只有 mgral_toolkit 的一个子类实例存在,并且只存在唯一的一个存取点对这个子类实例进行读写操作.这样,通过用户指定的参数,或者程序预先的配置,类

mgral 生成 mgral_toolkit 子类的一个实例,并将变量 instance 指向这个实例.例如,当客户指定当前使用 OpenGL 图形函数工具库时,类 mgral 生成 mgral_toolkit 子类 mgral_opengl 的一个实例,并将变量 instance 指向这个实例.

抽象基类 mgral_toolkit 的子类必须实现使用相应图形函数工具库时接口的定义.例如,实体类 mgral_opengl 中实现了使用 OpenGL 图形函数工具库时接口的定义.这样,当需要在不同的图形渲染函数工具库之间切换时,只需要首先删除变量 instance,然后生成对应于新的图形渲染函数工具库的实例并将变量 instance 指向这个实例即可.例如,当从 OpenGL 图形函数工具库转换到 DirectX 图形函数工具库时,先删除 mgral_opengl 的实例指针 instance,再创建 mgral_directx 的实例并将变量 instance 指向这个实例.

图形渲染抽象层模型的优点:

1) 使客户程序与实体对象的创建和功能实现相分离,降低了客户程序与实体对象之间的耦合性和依赖性.

2) 便于灵活地切换实体对象.某一类实体对象的创建由面向客户的接口类 mgral 根据用户选项或者程序参数决定,当从一类实体对象变换到另一类实体对象时,只要删除前一类实体对象的实例,然后创建后一类实体对象的实例即可.所有功能函数的实现也随之一起切换.例如,在本文所提模型中,当从 OpenGL 转换到 GDI 时,只需要删除 mgral_opengl 类的实例,然后创建 mgral_gdi 的实例即可.

3) 便于加入新的实体对象.当需要加入新的实体对象时,只需定义抽象接口类 mgral_toolkit 的实体子类即可.原有实体对象的实现和客户程序均不受影响.

3.2 混合图形渲染引擎:图形渲染抽象层的实现

本文在实现图形渲染抽象层模型中选择了直接使用相对底层的图形渲染工具库来实现混合图形渲染引擎,而不是使用集成开发框架中封装的图形渲染技术.这样做的好处有:

1) 不依赖于某一集成开发框架,集成开发框架的开发进度和其中的程序错误不会影响图形渲染引擎的发展和功能的正确性;

2) 便于将系统移植到所选集成开发框架不支持的操作系统或计算平台上.

在桌面版 MeteoExplorer 软件的多功能图形渲染引擎实现中,完成了对 OpenGL 和 GDI 图形函数工具库的封装,它们分别用抽象基类 mgral_toolkit 的实体子类 mgral_opengl 和 mgral_gdi 表示.由于 Windows 应用商店程序只支持 DirectX,本文还实现了 mgral_toolkit 的实体子类 mgral_directx.抽象基类 mgral_toolkit 中声明了如下功能的函数声明:

1) 窗口视图设置;

2) 图形渲染工具库坐标系与设备坐标系之间的转换;

3) 背景颜色、前景颜色设置;

4) 几何图形属性设置;

5) 字体属性设置;

6) 灯光和材质功能;

7) 纹理生成与显示;

8) 文字、几何图形的渲染;

9) 显示列表;

10) 多边形三角化(tesselation).

总之,通过设计图形渲染抽象层,可以方便地将图形渲染引擎移植到 Windows 8 系统下.所做的工作只是定义抽象接口类 mgral_toolkit 的实体子类 mgral_directx,而无需更改图形渲染模块内的其他代码和外部模块的代码.

4 图层管理方案

在图形图像编辑软件中,图层是指逻辑上构成一个整体的图形对象的集合.一个图层既可以是由几个天气符号组成的集合,也可以是包括地面站点观测数据、等值线、天气系统(如槽线、高低压中心等)的天气图.在实际操作中,常见的气象软件一般把一个数据文件中所创建的所有图形对象看作一个图层.

在传统的图层管理方式中,通常使用一个数据文件保存某一层次和时次下的某一个要素场的信息.在气象数据服务器中的文件组织方式上,通常按照数值模式、产品类型、要素、层次来创建多级文件目录,以起报时间和预报时效来命名文件.在这种组织方式下,一个文件对应某一要素在给定层次和时次下的物理场.对每个图层,其常用的操作包括翻页、动画、显隐、修改(仅对可交互图层有效)、查看文件内容、删除、刷新和显示属性设置,然而,当引入数值模式输出的数据文件时,上述传统的图层管理方式就显得不太适合.因为一个模式数据文件往往包括多个要素或集合成员、层次和时次的数据.以前针对单个要素场的操作,如翻页、动画、修改、查看,对于一个庞大的模式数据

文件来说,已经失去了原有的意义.

为此,本文系统对图层管理功能进行了增强,使之适合多种类型的气象数据文件.在如图 4 所示的图层管理窗口中,以树状结构组织图层,每个图层对应树中一个节点,树中的节点按照深度分为两类,深度为一的节点对应由数据文件创建的图层,例如由 MICAPS 数据文件或者模式数据文件创建的图层,深度为二的节点对应由一个模式数据文件中某一要素在给定层次和时次下的数据创建的图层.以图 4 为例,用户先后打开了 3 个文件:第一个是 T639 模式 850 hPa 温度场;第二个是 GRIB 格式的 NCEP 数据文件,其中又从该数据中分析得到的 500 hPa 位势高度场和温度场图层;第三个图层是一个交互制作得到的天气图.系统将此 3 个文件生成的图形对象看作 3 个图层进行管理,而又把由 NCEP 数据文件计算得到的 2 个图形看作子图层进行管理.

由于 Windows 应用商店程序设计风格的要求,MeteoExplorer Touch 不再使用辅助窗口作为图层管理的界面,取而代之的是利用顶部的应用程序工具栏作为图层管理的界面.如图 5 所示,每个图层以缩略图加上标题的形式显示在顶部应用程序工具栏中,显示的顺序与图层生成的顺序一致.实现图层操作功能的按钮放在顶部应用程序工具栏的右侧.以图 5 为例,用户先后打开了 3 个数据文件:第一个是

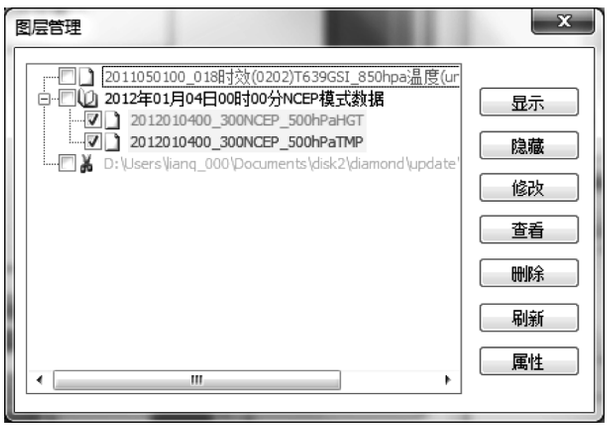


图 4　以树状结构组织图层的图层管理方案

Fig. 4　The graphics layer management in tree structure

卫星云图数据;第二个是 GRIB 格式的 NCEP 数据文件,其中又从该数据中分析得到的 850 hPa 位势高度场和温度场;第三个文件存放了 T639 模式 850 hPa 温度场.

相比于图 4 中的图层管理界面,MeteoExplorer Touch 的图层管理界面具有以下优点:

1) 以缩略图的方式显示图层内容,便于用户一目了然地区别各个图层;

2) 每一个图层以图像加文字的方式组织,由于增大了可触面积,因此便于手指触控操作;

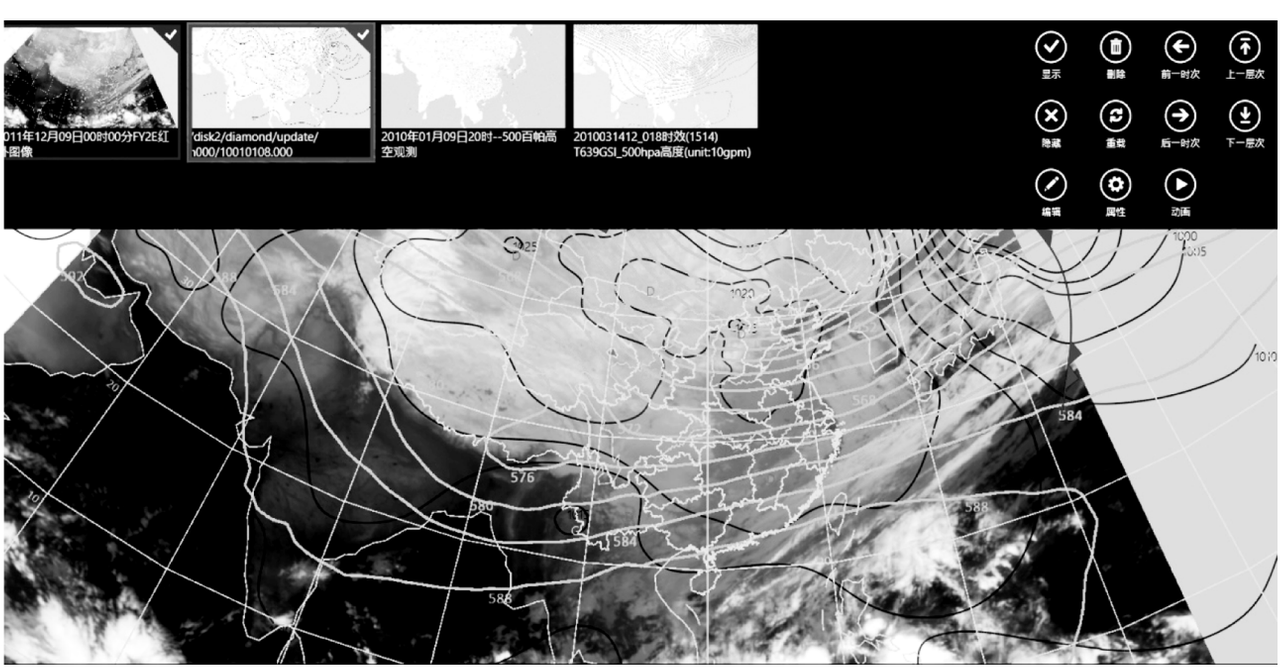


图 5　MeteoExplorer Touch 的图层管理界面

Fig. 5　MeteoExplorer Touch's user interface for graphics layermanagement

3) 以丰富的显示效果装饰每个图层项,便于用户快速查看每个图层的状态;

4) 遵循 Windows 应用程序商店设计风格,带给用户一致的操作体验.

为了支持如上所述的第三个优点,MeteoExplorer Touch 提供了以下非常实用的功能:

1) 用彩色和灰度显示表示一个图层的显隐状态,当一个图层处于显示状态时,其缩略图为正常颜色,当一个图层被隐藏时,其缩略图以灰度图像显示.如在图 5 中,第三个图层处于隐藏状态,其余均为显示状态.

2) 所有被选中的图层外围均显示一个选择框控件,以表示其处于选中状态.如在图 5 中,第一、二个图层处于选中状态.

3) 用户单击某一缩略图,则该缩略图所对应的图层在选中/非选中状态之间切换.

5 总结

气象信息处理技术在移动计算设备上的应用具有广阔的前景,然而这种向异构平台的技术移植并非一蹴而就的事情,需要解决诸多系统设计和技术实现上的难题.为此,本文详细讨论了在 Windows 8/RT 设备上开发基于触控操作的大气科学数据可视化系统的框架设计和关键技术,并结合原型系统 MeteoExplorer Touch 示例了将桌面软件系统移植到移动计算设备开发过程中一些可行的设计策略和实现方法.

1) 在系统框架设计中,本文所提的原型系统采用了层次化、模块化的设计原则,从而保证了在充分利用已有技术积累的基础上,通过新增用户界面交互功能来快速实现程序功能.

2) 通过设计图形渲染抽象层及其程序实现——多功能图形渲染引擎,示例了如何解决在 Windows 8/RT 操作系统下进行图形图像显示的技术难题,即将原有的基于 OpenGL 技术的图形显示模块,替换为具备同等功能的用 DirectX 技术实现的模块.通过这一设计,使软件系统具备了在无需修改已有模块代码的情况下,通过新增功能模块添加新功能的能力.

3) 提出了具有创新性的图层管理方案,即使用图层显示内容缩略图加上图层标题文字的形式,作为用户操作图层的控件.示例了如何在移动计算设备的图形界面交互操作上,为用户提供既有直观易懂,又方便快捷的交互方式.

由于移动计算平台上提供的应用程序开发技术与传统的桌面应用开发技术有较大的差异,再加上作者水平有限,许多常用功能如天气图图表的交互制作、三维图形显示等还未能在原型系统上实现.这也是今后技术研发的主要内容.

总之,本文讨论了将桌面气象数据可视化技术向移动设备移植过程中的框架设计和技术实现,旨在抛砖引玉,促进大气科学信息处理系统技术在新型计算设备上的研究与发展,从而更好地为地理信息、气象科学及相关领域的科学研究和实际业务工作服务,提高科研业务人员的工作效率.

参考文献

References

[1] MeteoGroup.MeteoGraphics:Weather solutions for modern broadcasting[EB/OL].[2013-09-05].http://www.meteographics.de/home.html

[2] 李月安,曹莉,高嵩,等.MICAPS 预报业务平台现状与发展[J].气象,2010,36(7):50-55
LI Yue'an, CAO Li, GAO Song, et al. The current state and development of MICAPS [J]. Meteorological Monthly,2010,36(7):50-55

[3] Unidata Program Center. AWIPS II overview[EB/OL].[2013-09-05].http://www.unidata.ucar.edu/software/awips2/

[4] Doty B E,Kinter I E.Geophysical data analysis and visualization using GrADS [M] // Szuszczewicz E P, Bredekamp J H.Visualization techniques in space and atmospheric sciences. Washington D C: NASA, 1995: 209-219

[5] CISL's NCAR Command Language (NCL).Boulder,Colorado: UCAR/NCAR/CISL/VETS [EB/OL].[2013-02-07].http://dx.doi.org/10.5065/D6WD3XH5

[6] Nogueira R,Cutrim E M.Applications of "Integrated Data Viewer" (IDV) in the classroom[J].Advanced in Geoscience,2006,8:63-67

[7] Janssen P. METVIEW-meteorological data visualization and processing software at ECMWF[J].ECMWF Newsletter,1999(86):6-18

[8] Google Inc.Google earth[EB/OL].[2013-10-10].http://www.google.com/earth/index.html.

[9] Luma F D.Introduction to 3D game programming with DirectX 10 [M]. Sudbury, MA: Wordware Publishing Inc,2008

[10] Woo M, Neider J, DavisT, et al. OpenGL programming guide:The office guide to learning OpenGL[M].Version 1,2,3rd Ed.Addison-Wesley Professional,1999

[11] Petzold C. Programming windows[M].5th Ed. Microsoft Press,1998

[12] Gamma E,Helm R,Johnson R,et al.Design patterns:Elements of reusable object-oriented software[M].Addison Wesley Professional,1994

Research and implementation of an atmospheric science data visualization system manipulated with touch gestures

YU Lianqing[1]
1 National Meteorological Center, Beijing 100081

Abstract In this paper, the research and implementation of MeteoExplorer Touch software, a scientific data visualization system designed especially for mobile computing devices that use touch gestures to manipulate, is introduced. This software system is also the world's first Windows store application that offers touch manipulation experience. The system enjoys the following technical advantages. First of all, by making use of the innovative technologies introduced in Windows 8 and Windows RT operating system, MeteoExplorer Touch manages to provide a number of exciting features. To name a few, it allows users to zoom and pan the base map using slide, pinch, and stretch gestures. It is able to restore the last session after the operating system recovers from the standby state. Next, MeteoExplorer Touch is not merely a picture viewer. Instead, it provides a large proportion of the functionalities offered by the professional meteorological applications designed for desktop and workstations. These functionalities include data analysis, objective analysis, physics computation, graphics product generation, geographic information, and layer management. Third, MeteoExplorer Touch concentrates on performance and user experience. To accomplish the goal, C++programming language is used for implementing functionalities, XAML markup language is used to design user interface, and DirectX 11 hardware accelerating rendering technology is used for graphics display. All the proposed software system shares the same strengths as those provided by a typical native application in that it offers high performance, while at the same time does not require much system resources.

Key words touch gesture manipulation; mobile computing; meteorological data; visualization system; software system