

基于 WPF 的交互式绘图系统的开发

李成刚, 冯静, 凌玲

(华中科技大学 机械科学与工程学院, 湖北 武汉 430074)

摘要: 开发了一个 VS2008 环境下, 基于 WPF 平台的交互式绘图系统。该系统利用计算机图形学的基本原理、图论知识和 Windows 表现层基础类库, 实现了基本图形元素的绘制、编辑与修改, 以及图形文件的打开、保存、打印等操作, 并提供了自定义的图形数据文件格式与 DXF 格式相互转换的功能。

关键词: 图形; WPF; 系统

中图分类号: TP317.4

文献标识码: A

文章编号: 1674-7720(2011)04-0050-03

Development of interactive graphics system based on WPF

Li Chenggang, Feng Jing, Ling Ling

(School of Mechanical Science & Engineering, Huazhong University of Science & Technology, Wuhan 430074, China)

Abstract: An interactive graphics system based on WPF platform and VS2008 environment was developed. The system uses the basic principles of computer graphics, graph theory and Windows Presentation Foundation, it achieves some basic function such as to draw, to edit and to modify basic graphic elements, as well as to open, to save and to print graphic files and so on, aslo it provides conversion functions between custom graphic data file format and DXF format.

Key words: graphics; WPF; system

交互式绘图系统是指能为用户提供交互式操作支持, 使用户通过鼠标、键盘和手写笔等在图形终端上输入数据和图形信息, 进行绘图, 并对生成的图形进行修改、编辑等操作, 设计完成后, 保存设计结果, 最终在显示屏或其他外部设备上图形输出的一整套设计及其应用软件, 在 CAD、办公自动化和其他领域得到了非常广泛的应用。现有绘图软件绘制复杂图形时会占用大量 CPU, 消耗大量内存, 从而导致界面失去响应。WPF 是一个新的编程模型, 与传统 GDI/GDI+ 绘图模型相比, WPF 将 CPU 和 GPU 分开来利用, 由于 GPU 的浮点计算能力远高于 CPU, 从而保证了大量计算的优先性, 同时也能最小化 CPU 使用率。WPF 以 DirectX 作为核心绘图函数, DirectX 绘图速度高于 GDI/GDI+, 能产生更好的视觉效果。对于需要大量浮点运算的图形图像来说, WPF 是一个很好的选择。本文介绍了在 VS2008 环境下, 以 WPF 为平台, 用 C# 编写开发的一个交互式图形系统。

1 交互式绘图系统的设计

1.1 系统的功能设计

基于 WPF 的交互式绘图系统主要由输入单元、基本绘图单元、图元编辑单元、图元修改单元、属性设置单元、输出单元组成。系统模块结构图如图 1 所示。

本系统实现了以下功能:

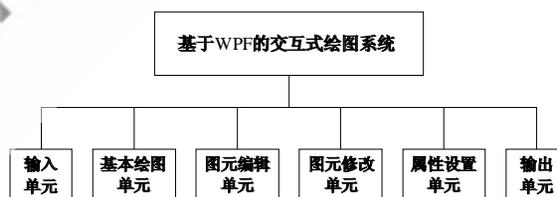


图 1 系统模块结构图

(1) 基本图形元素(点、直线、折线、圆形、圆弧、椭圆、矩形、多边形、B 样条、闭合 B 样条)以及文本的动态绘制与生成功能;

(2) 编辑修改功能: 对象的拾取、删除、复制、镜像、阵列、移动、旋转、填充、裁剪, 平移缩放视图, 测量封闭区域的面积;

(3) 属性的设置: 包括对图元的图层、线型、线宽、颜色; 系统的背景色, 文字的高度等属性进行设置;

(4) 图形数据文件的读取、保存、打印等操作。

1.2 开发平台的选择

目前 Windows 环境下的图形程序接口主要有 GDI/GDI+、Direct3D、OpenGL、WPF 四种。GDI/GDI+ 通常是一个静态的显示系统, 只有有限的动画支持, 此外 GDI/

《微型机与应用》2011 年 第 30 卷 第 6 期

图形、图像与多媒体

Image Processing and Multimedia Technology

GDI+只支持二维图形开发。OpenGL是一个开放的三维图形软件包,由于它是图形的底层图形库,没有提供几何实体图元,不能直接用以描述场景。Direct3D以COM接口形式提供,所以较为复杂,稳定性差。WPF是微软新一代图形系统,运行在.NET Framework 3.0架构下。WPF的核心是一个与分辨率无关并且基于向量的呈现引擎,旨在利用现代图形硬件的优势来最小化CPU使用率。在WPF中,底层的图形技术不再是GDI/GDI+,而是DirectX。

WPF在图形与界面开发方面,主要有以下几个优势^[1]:

(1)丰富的绘图模型与文本模型。不同于以往的逐个像素的绘制,在WPF中可以直接处理基本形状、文本框以及其他UI元素。

(2)独立的图形解析。WPF图形系统的基本度量单位是与设备无关的像素,它等于1英寸的1/96。不管实际的屏幕分辨率是多少,每个与设备无关的像素都会自动缩放,以符合呈现该像素的系统上的每英寸点数(dpi)设置。

2 交互式绘图系统的实现

2.1 系统的功能界面

交互式绘图系统的界面窗口主要由标题栏、菜单栏、工具栏、绘图区、工具箱、消息框和状态栏组成,如图2所示。

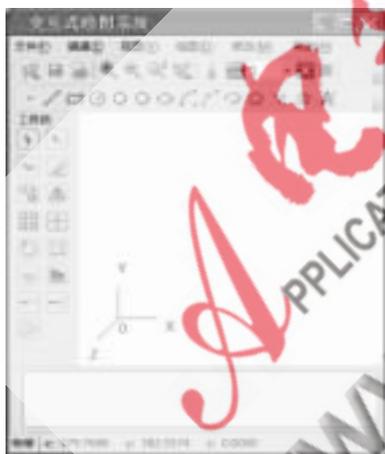


图2 交互式绘图系统界面

2.2 系统的功能实现

2.2.1 系统的窗口框架

WPF包含两个相互关联的编程接口,通常使用可扩展应用程序标记语言(XAML)标记实现应用程序的外观,而使用托管编程语言(代码隐藏)实现其行为。WPF中的Menu控件、ToolBar控件、StatusBar控件、GroupBox控件、TextBox控件以及Canvas面板^[2]分别构成了应用程序主窗口顶端的菜单栏、工具栏、底端的状态栏与客户区左边的工具箱、下方的消息框以及绘图区。系统使用鼠标和键盘输入来绘制图形,Canvas面板是用于创建复杂绘图的特别理想的选择,因为它支持对其子对象的绝对定位。框架窗口的逻辑树简图如图3所示。

《微型机与应用》2011年第30卷第6期

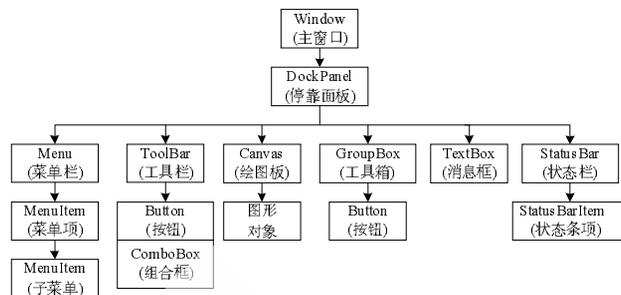


图3 框架窗口的逻辑树

2.2.2 图形的绘制与编辑功能

WPF提供了Shapes库来显示矢量图,WPF中的基本图形类主要位于System.Windows.Shapes命名空间。Shape类是所有图形类的基类,它定义了Stroke、StrokeThickness、Fill等通用属性,用来绘制形状轮廓的颜色、粗细以及形状的内部填充。具体的图形元素类(Line、Path、Polygon、Polyline、Ellipse、Rectangle)都由其派生而来。2D相关类型的层次结构图如图4所示。

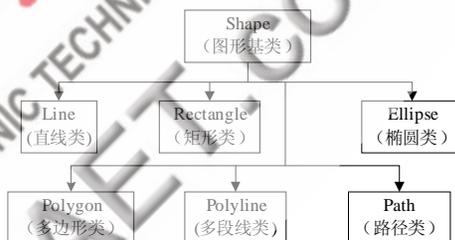


图4 2D图形类的层次结构

Canvas是顶级布局控件,Canvas面板类似传统的图形环境,利用坐标位置指定对象的摆放地点。Canvas的左上角为原点,X坐标向右,Y坐标向下。而一般的图形系统的坐标原点设定在绘图区的左下角,X坐标水平向右,Y坐标竖直向上,因此可利用WPF的Transform类,通过设定Canvas的RenderTransform来实现^[3]。为了使图元大小根据窗口的大小进行缩放,可以将Canvas放在Viewbox内。以下是实现坐标变换的XAML片段:

```
<Viewbox>
<Canvas RenderTransformOrigin="0.5,0.5">
<Canvas.RenderTransform>
<TransformGroup >
<ScaleTransform ScaleY="-1" />
</TransformGroup>
</Canvas.RenderTransform>
</Canvas>
</Viewbox>
```

图形元素的编辑与修改功能主要依靠计算机图形学的基本原理以及Windows表现层基础类库(WPF)来实现。WPF的System.Printing命名空间下的PrintDialog类主要用来实现打印单页或多页文件的功能,PrintDialog类显示一个打印对话框,与用户交互操作来完成打印功能。

欢迎网上投稿 www.pcachina.com 55

2.2.3 曲线曲面造型功能

曲线曲面造型^[4]是计算机辅助几何设计(CAGD)的核心,也是计算机图形学的一项重要内容。三次 B 样条曲线是曲线学中一个重要部分,它不但保留了 Bezier 曲线的优点,而且克服了其不具备局部性质的缺点,能更好地表示与设计自由型曲线曲面,通过几个控制点的移动能形成不同的曲线。系统提供了绘制 B 样条曲线与闭合 B 样条曲线的功能。绘制图形如图 5 所示。

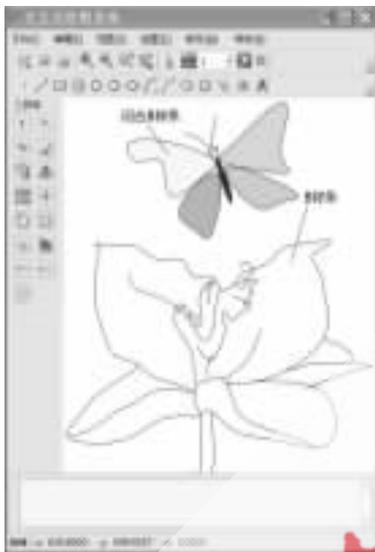


图 5 B 样条与闭合 B 样条

动态生成 B 样条曲线的实现步骤如下:

- (1) 设定一个点集合 $List<Point>$ mousepts 保存鼠标左键所按下的序列点;
- (2) 在鼠标移动过程中把鼠标的当前位置作为 B 样条曲线的临时终点,加入点集合 mousepts;
- (3) 由点集合 mousepts 中的这些序列点反求 B 样条曲线的控制顶点^[5];
- (4) 利用求出的 B 样条曲线的控制顶点得到 B 样条曲线上的序列离散点;
- (5) 将序列离散点依次加入多段线 Polyline 的 Points 集合,从而生成 B 样条曲线。

3 图形数据文件的存储与读取

系统图形数据文件的输出包含 JPG 和 DXF 两种格式。由于 AutoCAD 在全世界应用非常广泛,它的数据文件格式 DXF 已经成为事实上的一种标准,因此系统提供了自定义的图形数据文件格式与 DXF 格式相互转换的功能。DXF 文件本质上由代码及关联值对组成。代码(通常称为组码)表明其后的值的类型。使用这些组码和值对,可以将 DXF 文件组织到由记录组成的区域中,这些记录由组码和数据项目组成。在 DXF 文件中,每个组码和值都各占一行。DXF 文件完整的结构^[6]如图 6 所示。

程序中设计了两个基类 ReadDXF 和 WriteDXF 来分别读取和写入 DXF 文件。ReadDXF 读取 DXF 文件,并

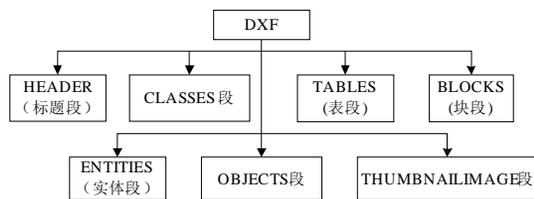


图 6 DXF 文件的完整层次结构

且从给定段的给定对象中提取特定代码和值。与读取 DXF 文件相比,写入 DXF 文件更困难一些,因为必须保持图形的一致性,这样 AutoCAD 才能找到可接受的文件。AutoCAD 允许省略 DXF 文件中的很多项目,仍然能够获得可用的图形,但文件结尾处必须存在 EOF 项目。为了简便,WriteDXF 省略了整个 HEADER 段、TABLES 段和 BLOCKS 段,只包含 ENTITIES 段。

随着计算机系统、图形输入、图形输出设备的发展,计算机图形软件系统也有了很大的发展^[7]。本文详细介绍了基于 WPF 的交互式绘图系统的功能与设计,并探讨了系统实现过程中的若干关键技术。该系统实现了其他图形软件具有的基本操作,并提供了自定义的图形数据文件格式与 DXF 格式相互转换的功能,操作方便,灵活,具有可扩展性,经过改进还可以成为适用于其它行业的通用的图形工具软件。

参考文献

- [1] MACDONALD M. Pro WPF in C# 2008: Windows Presentation Foundation with .NET 3.5[M]. New York: Apress, 2008.
- [2] NATHAN A. WPF 揭秘[M]. 瞿杰,译.北京:人民邮电出版社,2008:109-111.
- [3] PETZOLD C. Windows Presentation Foundation 程序设计指南[M].蔡学镛,译.北京:电子工业出版社,2008:844-845.
- [4] 朱永强,鲁聪达.自由曲线曲面造型技术的综述[J].中国制造业信息化,2003,32(5):110-113.
- [5] 孙家广,杨长贵.计算机图形学[M].北京:清华大学出版社,1995.
- [6] 谭超,张群会.基于 AutoCAD 的 DXF 格式文件读写方法研究[J].计算机与网络,2009,35(21):46-49.
- [7] 曾庆云.计算机绘图技术的发展趋势[J].机械研究与应用,1996(4):37-38.

(收稿日期:2010-08-26)

作者简介:

李成刚,男,1956年生,博导,博士,主要研究方向:计算机图形学、计算机辅助设计、智能设计与优化设计。

冯静,女,1985年生,硕士研究生,主要研究方向:智能设计。

凌玲,女,1975年生,副教授,主要研究方向:智能设计、知识工程。