文章编号:1671-4598(2019)09-0151-05 DOI:10.16526/j.cnki.11—4762/tp.2019.09.033 中图分类号:TP391.9 文献标识码:A

基于抽象工厂模式的机载显示系统图形生成技术研究

王志乐[1]，董军宇[2]，胡文婷[1]
(1. 海军航空大学 青岛校区军用虚拟仿真研究与训练中心，山东 青岛 266041；
2. 中国海洋大学 信息科学与工程学院，山东 青岛 266100)

摘要：机载显示系统直接关系到先进航电系统的性能和成本，针对机载显示系统的特点、性能要求，分析了目前机载显示系统存在的复用率低、不易于移植、继承性差、无法重构、成本高等缺陷；提出了基于抽象工厂模式的图形生成设计新思想，对机载显示系统图形库进行分类、分级抽象，建立图形处理模型、图形绘制模型和显示模型三级架构，统一了字符、图形、窗口的模型处理方法；基于 OpenGL 实现了图形模型开发库和机载显示系统图形生成架构，以该方法实现了某型飞机显示系统的开发，实验证明，该方法实现了显示系统通用开发技术和软件重构技术。

关键词：机载显示系统；抽象工厂模式；图形模型；OpenGL；软件重构技术

Research on Airborne Cockpit Graphic Generation Technology Based on Abstract Factory Pattern

Wang Zhile[1], Dong Junyu[2], Hu Wenting[1]
(1. Naval Aeronautical University Qingdao Branch, Qingdao 266041, China;
2. Department of Computer Science and Technology, Ocean University of China, Qingdao 266100, China)

Abstract: The airborne display systems are directly related to the performance and cost of advanced avionics systems. According to the characteristics and performance requirements of the airborne display system, the shortcomings of current airborne display system are analyzed, such as low multiplexing rate, difficult to transplant, poor inheritance, unreconstructible, and high cost. A new idea of graph generation design based on abstract factory pattern is proposed. The graphics library of airborne display system is classified and described abstractly in different levels, and a three-level architecture of graphics: processing model, graphics drawing model and display model, is established. Unified model processing methods for characters, graphics, and windows. Based on OpenGL, the graphics model development library and the airborne display system graphics generation architecture are implemented, this method realizes the development of a certain aircraft flight display system. Experiments show that the method implements the general development technology and software reconstruction technology of the display system.

Keywords: airborne display system; abstract factory pattern; graphics model; OpenGL; software reconfigurable technology

0 引言

现代飞机航电系统已经由原先的分立式、联合式[1]变为现在的集成模块化[2]、分布式的航电系统，由于先进的电子信息技术在航电系统上的应用，使得航电系统功能越来越强大，交联越来越复杂，涵盖了飞控、显控、雷达、导航、通信、任务等子系统[3]，该系统也成为现代先进飞机的核心系统，成为衡量飞机先进性能的核心指标之一。所有子系统的数据可视化显示与处理、人机交互都是通过显示与控制系统完成，这也对座舱显控系统的开发和性能带来挑战。

座舱显控系统作为飞行员与机载传感器的人机交互系统，其中机载显示系统直接显示飞行数据、战术数据、叠加图像等，显示的视觉效果、图形生成的效率直接影响飞行员的视觉判断和作战时机[3]。然而飞行显示系统需要实时地获取飞行传感器指令，经过计算处理后动态地显示飞行仪表图形、交互菜单、任务图形、目标及外部环境等信息，并且对显示系统的实时性、稳定性、安全性有很高的要求。当前，主流的设计方式都是将显示图形预先制作成贴图，然后通过国外专业的仪表开发工具如 GL Studio、IData、VAPS 等[4-6]进行开发。由于显示的图形种类和数量庞大、字符类型多，很多是按画面整体显示的图形进行设计，这类方法入门比较简单，但是显示系统复用率低、集成性差、无法实现重构，类似机型或相同机型升级都需要重新设计研制机载显示系统。另外，采用贴图方式显示的图像符号视觉效果差，而且使用大量的贴图影响显示画面

收稿日期:2019-03-12； **修回日期**:2019-04-11。
基金项目:国家自然科学基金项目(41576011)。
作者简介:王志乐(1983-),男,江苏兴化人,硕士,副教授,主要从事军用仿真技术,虚拟现实技术方向的研究。
董军宇(1972-),男,教授,博士生导师,主要从事计算机视觉,虚拟现实技术方向的研究。

生成的效率。所以本文对机载显示系统各类图形进行分层级抽象建模，提出了基于抽象工厂模式[7]对机载显示图形的三级建模架构，最后采用 OpenGL[8]实现了图形模型开发库和机载显示系统图形生成架构，实验证明显示效果比贴图方式好、图形生成效率更高，并且可以实现机载显示系统的软件重构。

1 抽象工厂模式

当系统所需要的产品对象是多个位于不同产品等级结构中属于不同类型的具体产品时需要使用抽象工厂模式。抽象工厂模式是所有形式的工厂模式中最为抽象和最具一般性的一种形态，与普通工厂方法模式最大区别在于，工厂方法模式针对的是一个产品等级结构，而抽象工厂模式则需要面对多个产品等级结构。

当一个工厂等级结构可以创建出分属于不同产品等级结构的一个产品族[9]中的所有对象时，抽象工厂模式比工厂方法模式更为有效率。产品族和产品等级结构关系如图 1。

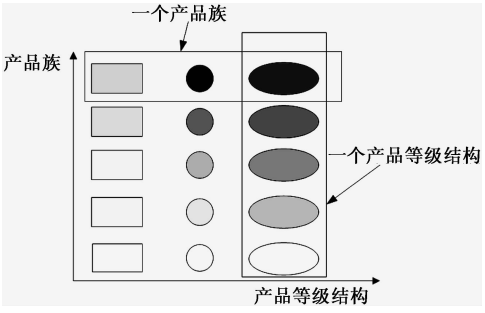


图 1 产品族与产品等级结构关系

因此可以采用抽象工厂模式将机载飞行显示系统所有图形库按产品族和产品等级进行抽象，将机载显示图形库进行分级建模形成标准的图形生成对象，对字符、图形、窗口进行抽象统一建模，再利用抽象工厂模式对图形处理模型、绘制模型、显示模型进行系统架构设计。首先对显示图形库进行分类抽象，形成机载显示图形类结构如图 2。

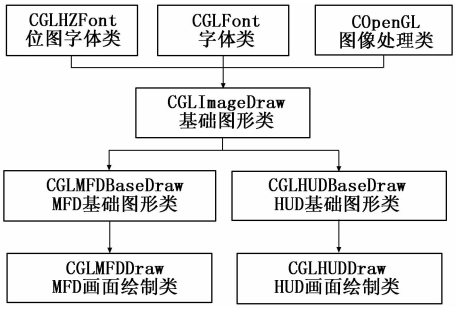


图 2 机载显示图形类结构

由于抽象工厂模式属于类创建型模式，它的目的是为一系列的、相互关联的具体类提供统一的创建接口[10]。抽象工厂从图形绘制模型代码中隔离出了创建具体对象的操作，把所有对相互关联的类的创建操作组织到一起，为其他类提供高层的、经过封装的对象创建操作[9]。为了方便阐述抽象工厂模式，引入标准化的产品族，将不同分级结构但功能相关联的图形对象组成家族，其抽象工厂接口类如图 3 所示。

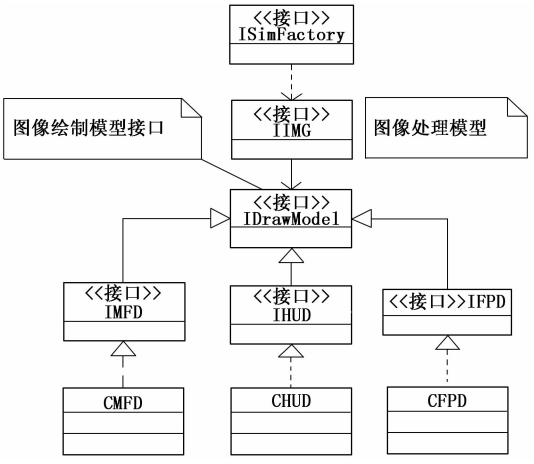


图 3 图形家族抽象工厂接口类

基于抽象工厂模式的思想创建统一的接口类，可以通过基类快速的创建图形产品族或者基于产品等级资源库创建新的图形样式，然后在图形内部完成新增功能和特性。

2 图形抽象建模

在显示图形抽象建模的设计过程中，采用产品族的概念进行分类分级描述，将图形的特征处理进行分类建模。将机载航空电子系统显示的图形图像分为字符模型、基本图形模型、复杂图形模型、窗口模型，其中字符模型描述汉字字符和英文字符，基本图形模型描述线、矩形、圆弧、三角形等，复杂图形模型在基本图形模型的基础上进行建模描述，如表示飞机姿态的天地圆、填充椭圆、填充带边框矩形等。

2.1 位图文字矢量显示法

字符模型作为一种基本的显示图形，其绘制如表 1 所表。字符结构模型＝｛字体｜坐标｜宽度｜高度｜旋转角｜下划线｜删除线｜斜体｝，字符模型绘制处理类分为英文字符处理、汉字字符处理，其中英文字符采用 ASCII 绘制，汉字字符采用位图显示方式，由于汉字字符的显示比较特殊，这里在 OpenGL 环境下以汉字绘制类进行抽象描述，如表 1 所示。

表 1 汉字绘制类

类名	抽象方法	类型	描述
CGLHZFont	CreateFont	HFONT	创建汉字句柄
	PrintText	bool	绘制输出汉字
	KillFont	bool	删除汉字句柄

其中汉字的绘制处理在 PrintText 接口中完成，在 Windows 环境下 OpenGL 常用的位图字符显示方法包括：(1)通过制作位图字符的显示列表来显示，但对于汉字字符将

大量消耗资源；（2）利用纹理贴图原理，将事先要显示的汉字制作成贴图，但运行时无法修改，只能适用于少量的汉子显示；（3）读取点阵字库信息，利用 glBitMap（）函数显示，读取显示效率高，但是放大或者缩小会存在锯齿现象。针对上述方法的特点，采用 GDI 提出的 TrueType 平面位图文字显示法。

利用 TrueType 矢量字体的与设备无关性、灵活性好等特点，结合 OpenGL 的位图显示技术，实现了矢量汉字灵活显示方法。其基本绘制模型如图 4 所示，在汉字矢量处理类中调用 OpenGL 位图数据结构及位图处理函数实现汉字显示处理。

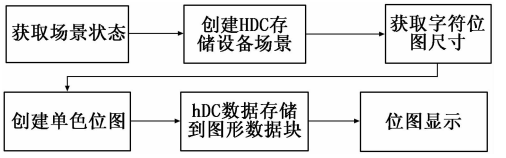


图 4　矢量汉字绘制模型

2.2　基本图形处理模型

在进行基本图形处理建模过程中，将汉字纳入到基本图形中，汉字绘制类实现之后，就可以和其他基本图形处理一起集成到基本图形处理类中，这里基本图形定义为：线、矩形、圆形、三角形、扇形或圆弧、多边形、窗口。基本图形定义的太详细或者太粗糙都不利于复杂图形的构建和图形的调用显示。因此在采用抽象工厂模式创建图形处理类时，首先创建基础的图形接口基类，如表 2 所示。

表 2　图形接口基类

图形模型接口类	抽象方法	类型	描述
IIMGModel	CreateIMG	virtual bool	创建图形对象
	InitIMG	virtual	初始化位置大小
	DrawIMG	virtual	绘制图形
	DeleteModel	virtual	删除对象模型

其中 CreateIMG 接口参数在完成创建图形对象过程中必然要使用 CDC 类，因此该接口的定义如下：

virtual bool CreateIMG(CDC * pDC, enIMGShowTypeDef enType,bool bIsColor)=0;

参数 enIMGShowTypeDef 表示载机显示系统的某种类型的物理显示设备，如 MFD1 \ MFD2 \ HUD \ HMD 等。

在进行基础图形类库的开发过程中，需要使用 Opengl32. dll 和 Glu32. dll，因此这里基于 OpenGL 图形接口，结合基础图形绘制模型的特点，以及开放式航电显示系统设计的技术要求，设计专用的图形绘制类接口 CBaseDraw，主要包括图形绘制类函数和图形控制类函数，该接口类的模型如表 3 所示。

2.3　复杂图形绘制模型

复杂图形处理类模型 CGLImageDraw 的定义是基于 IIMGModel 进行派生的，由于复杂图形也是基于基本图形库产生的，因此需要引用 CBaseDraw，其模型定义如表 4 所示。

表 3　基础图形绘制模型

模型绘制类	抽象方法	主要参数	描述
CBaseDraw	DrawString	X,Y：基准点坐标 Num：字符数 Str:字符数组	顺序绘制若干字符
	SetColor	Color：颜色字	设置颜色
	SetLineStyle	LineWidth:线宽 LineStyle:线形特征	控制绘制图形的线宽和线形
	SetBlink	Blink:闪烁特征字	控制图形闪烁
	DrawSymbol	X,Y：基准点坐标 Symbol:符号编码 Character:特征字	绘制单个符号
	……	……	……
	DrawCircle	Fill_In:是否填充 X,Y：基准点坐标 Radius:半径	绘制圆形
	DrawLine	X1,Y1：起点坐标 X2,Y2：终点坐标	绘制直线
	……		

表 4　复杂组合图形模型

模型处理类	方法及对象	类型	描述
CGLImageDraw	CreateIMG	virtual bool	创建图形对象
	InitIMG	virtual	初始化位置大小
	DrawIMG	virtual	绘制图形
	DeleteModel	virtual	删除对象模型
	SetViewPara	void	设置视窗参数
	m_pBaseDraw	CBaseDraw	基本图形类对象
	m_HzFont	CGLHZFont	汉字处理类对象
	DrawCompass	bool	绘制罗盘符号
	DrawWpt	bool	绘制航路点
	DrawPlane	bool	绘制飞机符号
	DrawHorizIndicator	bool	绘制地平仪
	……	……	……

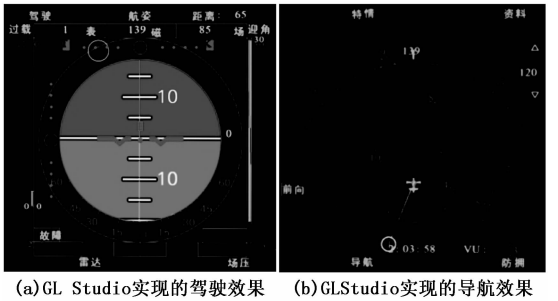
CGLImageDraw 类的具体定义如表 5 所示。

在复杂图形模型的绘制过程中，可以采用贴图实现，这种方法对开发人员的技术水平要求不高，并且开发过程简单，但是灵活性和重用性不强，显示效果也不好。一般战斗机的图形显示符号达到 100 余种，都采用贴图之后在显示过程中系统占用的资源比较高。因此基于设计的基本图形处理类通过图形绘制算法创建复杂图形，既可以实现图形符号的通用性，也实现了图形绘制算法的重构，通过参数可以构建不同飞机同一类别的图形符号，而且显示效果和软件效率都得到大大提高。如图 5 是基于 GL Studio 开发工具利用预先处理好的贴图实现的 MFD 显示效果，其特点是显示的每一个字符及图形符号都是基于贴图图像

实现的，因此显示效果受图像分辨率影响较大，当画面进行缩放时图像容易发虚；对于可变图形（如航线）利用 GL Studio 进行动态绘制后由于没有进行反走样处理，锯齿明显。

表 5 CGLImageDraw 类模型定义

```
class CBaseDraw;
    class CGLImageDraw: public IIMGModel
    {
    public:
      CGLImageDraw ();
      virtual bool CreateIMG(CDC * pDC, enIMGShowTypeDef enType,bool bIsColor); //创建图形对象
       virtual void InitIMG(SHORT nInitX, SHORT nInitY, USHORT nWidth, USHORT nHigh); //初始化位置及宽高
       virtual void DrawIMG(const IMGINFO pData);
       virtual void DeleteModel();
       void SetViewPara(float xOffsetRate = 1.0f,float yOffsetRate = 1.0f,float xScaleRate = 1.0f,float yScaleRate = 1.0f);
    //设置视图窗口参数
       void ShowChar(unsigned short ChineseChar);
       void ShowChar(char AsciiChar);
       void ShowSymbol(unsigned SymbolCode);
    bool DrawCompass(int deviceID,unsigned SymbolCode,PosX,
    PosY,float angle); //绘制罗盘
    ......
    bool DrawHorizIndicator (int deviceID,unsigned SymbolCode,
PosX,
    PosY,float angle);
      GLHZFont m_HzFont; //位图汉字处理类
      GLFont  m_Font; //英文字符处理类
    CBaseDraw * m_pBaseDraw; //基本图形处理类
```



(a)GL Studio实现的驾驶效果 (b)GLStudio实现的导航效果

图 5 GL Studio 实现的驾驶效果和导航效果

利用 CGLImageDraw 和 CBaseDraw 类进行画面绘制时，通用基本图形符号可以通过 CBaseDraw 类实现，对于复杂组合图形符号在 CGLImageDraw 类设计时实现，通过调用 OpenGL 函数库、CBaseDraw 类和相关图形处理算法实现。以图 1 画面中地平仪（天地圆）的实现为例，其组成及变化比较复杂，其特征如表 6。

在圆形窗口区域绘制可变天地圆是比较困难的，通过

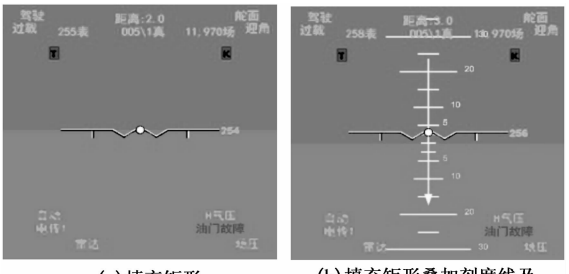
表 6 天地圆特征表

符号名称	组成	运动规律	特征
地平仪（天地圆）	蓝色半圆	上下平移、旋转	绕圆心左右旋转
	土黄半圆	上下平移、旋转	绕圆心左右旋转
	俯仰刻度线	与半圆同步运动	−90 至+90；

分析知道天地圆实际是表示飞机俯仰和横滚姿态的刻度带，因此采用分层设计的思路，首先以 MFD 屏幕分辨率宽度为基本参数分别绘制蓝色和土黄色的填充矩形作为天地圆的姿态刻度带，如图 6（a），这里坐标圆点在屏幕中心，选定 MFD 的分辨率为 600×600 像素，中心坐标为（300，300），屏幕坐标圆点到 MFD 坐标圆点的转换为：$x=\frac{X}{300.\ 0F}$，$y=\frac{Y}{300.\ 0F}$。然后通过调用 CBaseDraw 的填充矩形绘制函数完成姿态刻度带的绘制，其绘制函数实现过程如表 7。

表 7 姿态刻度带绘制

```
#define MFD_GL_XY (x) ((x)/300.0) //坐标转换宏定义
x1= X; y1 = Y; //X = − 300, Y = 0; Width = 600, Height =1000;
x2=X+Width;y2=Y+Height; //通过刻度带宽度和高度计算屏幕坐标
glBegin(GL_QUADS); //绘制矩形
glVertex2f(MFD_GL_XY(x1),MFD_GL_XY(y1));
glVertex2f(MFD_GL_XY(x2),MFD_GL_XY(y1));
glVertex2f(MFD_GL_XY(x2),MFD_GL_XY(y2));
glVertex2f(MFD_GL_XY(x1),MFD_GL_XY(y2));
glEnd();
```



(a)填充矩形 (b)填充矩形叠加刻度线及刻度值绘制效果

图 6 填充矩形与填充矩形叠加刻度线及刻度值绘制效果

在填充矩形显示层上面调用 CBaseDraw 类的绘线接口完成刻度线的绘制，以及刻度值的绘制，其显示效果如图 6（b）显示。最后为了构成天地圆的显示效果，需要对填充矩形进行遮挡层绘制，均分成四等分绘制遮挡图形，遮挡图形由多边形和圆弧组成，主要计算出圆弧半径和四个点的坐标，如表 8 关键点坐标。圆弧的绘制算法如下，遮挡圆弧实现后的显示效果如图 7（a）所示。

定义：$circle_r=173$。

$circle_x$ 表示圆弧上某点 X 坐标，$circle_y$ 表示圆弧上某点

Y 坐标

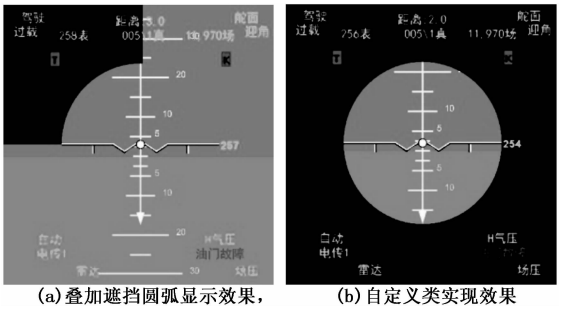
$$circle_x = \sum_{-\frac{\pi}{2}}^{\leqslant 0.05} circle_r * \sin(circle_q);\ (circle_q = circle_q + 0.1)$$

$$circle_y = \sum_{-\frac{\pi}{2}}^{\leqslant 0.05} circle_r * \cos(circle_q);\ (circle_q = circle_q + 0.1)$$

表 8　关键点坐标

坐标点	X	Y	变换
$P1$	-300	1200	$(X)/300.0,(Y)/300.0$
$P2$	-300	0	$(X)/300.0,(Y)/300.0$
$P3$	$-R$	0	$-R,0$
$P4$	0	1200	$0,(Y)/300.0$

采用遮挡圆弧绘制算法绘制完其他三份，至此就完成了天地圆的绘制，当飞机姿态变化时，天地圆及刻度带随动，而遮挡部分始终保持不动，这样的机理也完全符合地平仪的结构。通过自定义类绘制效果如图 7（b）所示。基于 CBaseDraw 基本图形类和复杂图形的绘制函数、管理函数构建 CGLImageDraw 类，这样就实现了机载显控系统的可视化图形抽象建模。



(a)叠加遮挡圆弧显示效果，　(b)自定义类实现效果

图 7　叠加遮挡圆弧显示效果与自定义类实现效果

2.4　显示画面实现

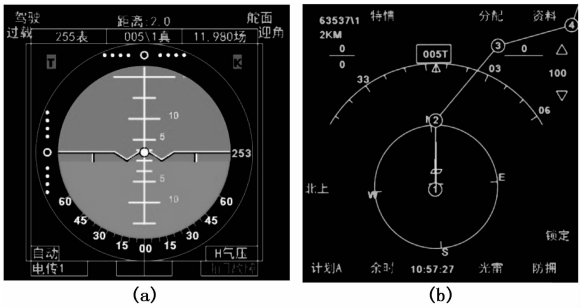
以某型固定翼飞机飞行驾驶画面为例，该画面包括地平仪、飞行参数指示等[11]。利用自主开发的图形模型库和显示系统图形生成架构，快速的实现画面绘制。该架构首先是画面绘制准备工作，其次是利用图形模型库提供的 SDK 进行字符、图形和窗口的显示。

准备工作过程中，主要实现着色模式选择、目标像素深度设置、指定颜色和纹理坐标的差值质量、启用点线反走样和抗锯齿[12-13]，定义像素运算算法等，其关键代码如表 9。

字符和图形的显示根据机载航电子系统输出的状态数据、传感器测量数据、任务解算数据[14]以及显控系统当前所显示的画面状态判断下一刻应该显示的画面。根据画面图形元素类型及图形位置布局等特点，调用 CBaseDraw 类和 CGLImageDraw 类提供的图形生成算法实现字符、通用图形、复杂图形的实时绘制和显示。图 8 是采用该架构提供的抽象图形模型库实现的某型飞机驾驶画面和导航画面的效果。

表 9　显示初始化功能

```
glShadeModel(GL_SMOOTH);  //着色模式
glEnable(GL_DEPTH_TEST); //
glDepthFunc(GL_LEQUAL);
glHint(GL_PERSPECTIVE_CORRECTION_HINT,GL_NICEST);
glEnable(GL_POINT_SMOOTH); //启用点线反走样
glHint(GL_POINT_SMOOTH,GL_NICEST);
glEnable(GL_LINE_SMOOTH); //反走样
glEnable(GL_BLEND); //抗锯齿
glBlendFunc(GL_SRC_ALPHA,GL_ONE_MINUS_SRC_ALPHA);
..... //定义像素运算算法
    glViewport(0,0,m_width,m_height); //设置视口尺寸
```



(a)　(b)

图 8　自定义架构和图形库实现的驾驶和导航显示效果

3　仿真应用分析

国外专用开发工具（如 GL Studio、IData 等），往往不支持汉字显示，汉字和图形都是通过贴图实现，因此当显示比例变化时显示效果会受到影响，对于一些动态变化的图形，需要依靠工具提供的控件和 API 来实现，但工具未提供图形反走样处理，如图 1 显示效果。基于国外专用工具开发后的软件重用性、重构性都大大降低，即使是贴图的重复使用率也不高，优点是对开发人员的编程技术要求不高。

基于抽象工厂模式实现的座舱图形生成库，解决了国外专用工具面临的问题，所有的显示符号都是图形，并进行了反走样和锯齿的处理，图 6 对比图 1 的显示效果明显提高。利用上述平台架构和图形模型库开发的任意机型的显示系统，通过调整视窗、修改绘制参数、增加或删除某些图形等可以快速的实现代码的重构，而开发人员无需掌握图形的绘制机理，只需要调用对应图形的绘制函数，输入位置和特征参数。表 10 给出了图形生成技术与国外专用工具的对比。

4　结论

通过对比传统显示系统的软件开发以及利用国外软件

（下转第 162 页）