

# Java 2D API 技术及其实现方法\*

伍祥生\*

王克宏

(湖南师范大学物理系,长沙 410006)

(清华大学计算机科学与技术系,北京 100084)

**摘要** 介绍了 Java 2D API 的主要技术特点;并从几何图形、文本和图象三个方面对 Java2D API 的技术实现作了较详细的阐述。

**关键字** Java 2D API, 图形, 文本, 图象

Java 作为一种编程语言,与其它计算机高级语言一样,具有绘制图形和图象的功能。随着计算机设备都将联网、联网应用都将是“可视化”的大趋势,人们对图形、图象质量的要求不断提高,对独立于平台的图形、文本、图象的需求日愈增多,为此,由 SUN 和 ADOBE 两家公司联合开发了 Java 2D API,为用户提供了高级 2D 的编程接口。

## 1 Java 2D API 主要技术特点

### 1.1 绘制几何图形更有特色

Java 2D API 定义了一个图形子类 Graphics2D,它为特定的画图风格、定义复杂形状和控制绘画过程添加了特色。通过在 Graphics2D 对象中设置绘图属性,用户就可绘制所需要的形状、文本或图象。Java 2D API 定义了 2 个坐标空间,即用户坐标空间(the User Coordinate Space)和设备坐标空间(the Device Coordinate Space)。一般情况下,图形对象都是被描绘在与设备无关的用户坐标空间。一个与目标设备有关联的 Graphics2D 对象,其绘制状态中含有一个变换对象,可将该图形对象的用户空间坐标转换成设备空间坐标。Java 2D API 为绘制几何图形还提供了贝塞尔路径(Bezierpath)类,用户可利用直线和贝塞尔(Bezier)曲线的组合描绘一个形状。用 Bezierpath 定义对象的轮廓,可在用户坐标空间控制它的位置、大小和方向。

### 1.2 提供高级文本布局工具

在 Java 语言中,文本可作为一种特殊图形来处

理。Java 2D API 中许多类的绘制、变换和剪辑文本的功能,为处理文本提供了方便。与 java.awt 中的字体类相比较,Java 2D API 中的字体类控制了更多的字体,如对字体的规格、大小和轮廓等信息的控制、访问和处理。Java 2D API 提供的高级文本布局工具,用于处理包括具有混合字体的文本字符串、混合语言和双向文本等。

### 1.3 图象处理功能加强

Java 2D API 通过在 java.awt 和 java.awt.image 中增添了几个新的图象处理类,如 BufferedImage, Tile, Channel, ComponentColorModel 和 ColorSpace 等类,使得 Java 2D API 对图象处理的功能得到加强,如对图象进行大小的缩放、图象的旋转、倾斜或者像文本一样变换的操作。

### 1.4 向后兼容性好

Java 2D API 适用于 JDK 1.2 版本。但 Java 2D API 保持与 JDK 1.1 的兼容性,意味着现有的程序能在 Java 2D API 中运行。

## 2 Java 2D API 技术的实现

Java 2D API 中所有类都被编制在下面 7 个软件包里,如图 1(见下页)所示。每当我们解决某个具体问题时,就需要采取调包找类的编程方法。

### 2.1 几何图形的实现

为了使用 Java 2D API 绘制几何图形,可以调

\* 现为清华大学计算机科学与技术系访问学者

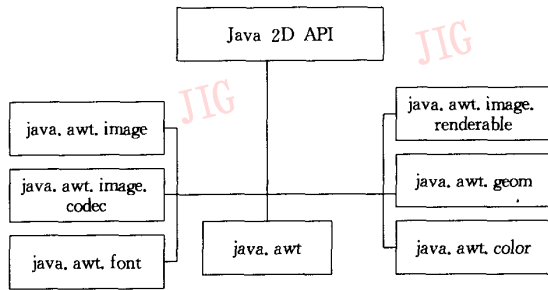


图1 7个软件包

用 `java.awt.geom` 包,在它中间寻找所需要的类。通常这些类定义了几何图形对象,包括点、直线、曲线和矩形等,人们可以利用这些类描绘任何二维的物体。在 `java.awt.geom` 包中有一个变换类: `AffineTransform`,它表示了一个二维的仿射变换,并支持像平移、定标、旋转和剪切等操作,还能完成从一个 2D 坐标系到另一个 2D 坐标系的线性映射,不发生变形;它为构造仿射变换对象提供完成标准转换的一组便利方法,如: `getTranslateInstance`、`getRotateInstance`、`getScaleInstance`、`getShearInstance` 等方法。用户能通过调用 `Graphics2D` 转换方法来修改图形范围内的变换:包括旋转、缩放、剪切和平移等。如要求实现绘制的矩形旋转 45 度,可用下述方法实现。

(1) 首先用构造方法创建一个 `Rectangle2D.Float` 对象;

(2) 其次调用仿射变换 `AffineTransform` 类的 `getRotateInstance` 方法,获得一个旋转变换;

(3) 再调用 `Graphics2D` 类的 `setTransform` 方法增加新变换到转换传输;

(4) 最后调用 `Graphics2D` 类的 `draw` 方法绘制矩形。

其程序段为:

```

Rectangle2D rect = new Rectangle2D.Float ( 1.0, 1.0, 2.0, 3.0);
AffineTransform rotate 45 = AffineTransform.getRotateInstance (Math.PI/4.0,0.0,0.0);
g2.setTransform (rotate 45);
g2.draw ( rect);

```

## 2.2 文本实现方法

Java 2D API 提供了相关的文本类,用于支持完善的文本布局和细密纹理字体控制。字体类加强了对详细字体信息规范的支持,并实现访问一个字体和它的电刻版 (glyphs) 的信息。字体类常用来表示一个字体的实例,通常字体的例子有 `Helveti-`

`caBold` 和 `CourierBold Italic` 等。有很多的信息能用来描述一个字体,包括:它的名字、使用的打字技术、版本和它的风格参数。每个字体对象内含以字体名、大小、变换和其它字体特征如权数和姿势的属性。用户可通过字体类提供的便利方法直接地访问这数据。字体类也提供了通过 `getGlyphMetrics` 和 `getGlyphOutline 1` 方法访问度规和轮廓信息。下面通过 Java 2D API 提供的 一个 `font.deriveFont` 方法,从一个存在的字体对象中创建一个新的字体对象为例,来说明其实现方法。(运用仿射变换,创造一个字体 `Helvetica` 的斜体版,并用新导出的字体绘出一个字符)。

(1) 首先创建一个字体对象;

(2) 其次创建应用到该字体的仿射变换;

(3) 再调用 `Font.deriveFont`,用于仿射变换。

其程序段为:

```

AffineTransform fontAT = new AffineTransform();
fontAT.setToShear (-1.2,0.0);
Font theFont = new Font ("Helvetica", Font.PLAIN,1);
theDerivedFont = theFont.deriveFont(fontAT);
ss = new StyledString("Java",theDerivedFont);
g2.drawString(ss,0.0f,0.0f);

```

## 2.3 图象实现方法

Java 2D API 提供了丰富的图象处理类,这些类可分为 6 个大组,即 `Image Data Classes` (包括 12 个类)、`Image Filter Classes` (包括 6 个类)、`Image Operation Classes` (包括 9 个类)、`Sample Model Classes` (包括 5 个类)、`JPEG Conversion Classes` (包括 5 个类) 和 `ColorModel Classes` (包括 5 个类)。

### 2.3.1 Java 2D API 图象处理模型

`java.awt.image` 包提供 2 个关键的图象类,即 `BufferedImage` 和 `Raster`。`BufferedImage` 类是一个提供访问它的图象数据的图象。它包括图象数据的一个 `ColorModel` 类 和一个 `Raster` 类。`Raster` 类在 `BufferedImage` 中定义了数据和数据布局,显示一个象素的正交数组。该图象包提供了一组类,用于定义对 `BufferedImage` 和 `Raster` 对象的操作。每个类能表示一个图象处理的操作,这些类的执行接口不是 `BufferedImageOp` 接口就是 `RasterOp` 接口。一幅真实的图象处理,需要经过仿射变换、幅度缩放、查找表变换、条纹的线性组合、颜色转变、旋转等操作,Java 2D API 提供了这些操作类,并定义了滤波方

法,能完成真实的图象处理操作,如图2所示。

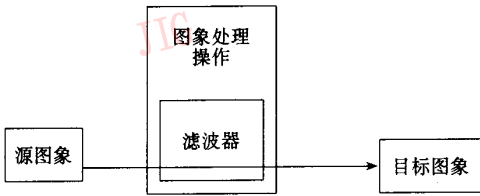


图2 图象处理示意图

执行这些操作的类包括:AffineTransformOp 和它的子类 BilinearAffineTransformOp、NearestNeighborAffineTransformOp、BandCombineOp、ColorConvertOp、ConvolveOp、LookupOp、RescaleOp 和 ThresholdOp 等。

### 2.3.2 处理一个图象

图象数据在计算机中是以位图的形式存在。位图是一个矩形点阵,它上面的每个点对应着一个象素。而描述每个象素需要8位数据,即一个字节。在图象数据中,每个字节是与画面上横着排列的8个象素相对应的。Java 2D API 提供了一个类 Kernel,该类是一个矩阵,用来描述一个特定的象素和它的周围的象素怎样影响在滤波操作中规定的象素的灰度值。另一个操作类 ConvolveOP,可使用 Kernel 类对源图象求卷积。卷积是构成大多数空间滤波算法基础的进程。在具有邻接象素灰度值的一个图象中,卷积是加权或平均每个象素的灰度值的过程。运用 Kernel 类能在数学上确定每个输出的象素受最邻近象素的影响。如设源图象中的每个象素被周围的8个象素等值平均,则有

$$\text{float weight} = 1.0f/9.0f; \text{(注:包括自身,即9个象素)}$$

```

float[]elements = new float[9];
for ( i=0;i<9;i++){
    elements[i] = weight;
}
private Kernel myKernel = new Kernel(3,3,1,1,
elements);
public ConvolveOp simpleBlur =
new ConvolveOp(myKernel);
simpleBlur.filter (sourceImage,desImage);
  
```

在程序段中,变量 simpleBlur 包含 ConvolveOp 的一个新的实例,用来实现对 BufferedImage 或 Raster 的一个模糊操作。假定 sourceImage 和 desImage 是 BufferedImage 的2个实例,当调用一个滤波器时,ConvolveOp 类的核心方法通过用9个象素的平均值设置目标图象中的每个象素的灰度值。

### 3 应用实例

在 Java 2D API 中,图象能为每个在图象中的象素传送透明信息,这个信息称为一个  $\alpha$  信道,它连同 Graphics2D 范围中的 Composite 对象一起把该图象与存在的图画混合。在图3中,包含3个具有不同透明信息的图象。在第1个图象中,小狗身体上的所有象素完全不透明,而图象的背景完全透明,这种效果经常应用在 WEB 页面上。第2个图象中,在小狗身体上的所有象素是用均匀、透明的  $\alpha$  值绘制,所以蓝色背景全部显示出来。在第3个图象中,围绕小狗脸部的象素是完全不透明的( $\alpha=1.0$ ),但随着离它脸部距离的增加,所对应的那些象素的  $\alpha$  值是减少的。



(1) (2) (3)

图3 3个不同透明信息的图象

### 4 结束语

本文所介绍的 Java 2D API 的技术特点、规范和实现方法均属于 Java 2D API β 版本的内容。Java 2D API 扩大了由 java.awt 所定义的图形和图象的类,并保持了对原有程序的兼容性;它使得用户把高质量的二维图形、文本和图象溶合到 JAVA 独立的

应用程序和 JAVA 小应用程序成为现实。

### 参考文献

- 1 Java 2D API Overview (<http://Java.sun.com/products/java-media/index.html>).
- 2 孙家广,杨长贵.计算机图形学.北京:清华大学出版社.

## Princess Cruise Lines 公司的皇家公主号游艇采用 VTEL 最先进的 SEAMED 远程医疗设备

应用于皇家公主号上的海上医疗视讯设备将游轮与洛杉矶的 CEDARS-SINAI 医疗中心急诊室的医生直接联系起来,创造出一个“模拟急诊室”。除了提供实时的双向视讯联系外,VTEL 数字视讯系统还把 X 光片,心电图及其它表示身体症状的信号通过卫星传输到地面医院。

这种“模拟急诊室”广泛应用于医疗领域,可治疗心脏、肺部、畸形及神经系统的急症。CEDARS-SINAI 的专家能够直接与皇家公主号上的医生一起工作,并可提供更多的医疗建议。

IMAGEVIEW 公司的海上远程医疗设备以 VTEL 的远程通讯技术为依托。带有 VTEL 公司生

产的高分辨率监视器的 TC2000 房间型多媒体会议系统 (LARGE GROUP CONFERENCING SYSTEM) 是此设备的核心。VTEL 的这套系统能提供速率 512Kbps,每秒 30 帧的高品质图象和一按即通的应用程序共享。该系统的另一特色即是通过装在机械臂上的小型摄像机,医生可以更真切地检查病人的伤情及病状。VTEL 的摄像机就是根据视讯系统的需要而设计的。

IMAGEVIEW 开发的海上远程医疗设备专门应用于远程医疗通讯,包括和私家游艇,商船和钻井平台等的联络。

(汪虹)

\*\*\*\*\*

### 怎样写参考文献

- 1 在文稿中引用过的参考文献,按在文中出现的先后用阿拉伯数字顺序统一编号。
- 2 参考文献中不应列入未公开发表的文章、著作等,也不应列入内部资料和密级资料。
- 3 在正文中,当需说明引用内容的出处时,应把引用的参考文献编号连同方括号写在所注部分末字的右上角,如:××××<sup>[1]</sup>。若所提及的参考文献作为正文中的直接说明语,则其编号连同方括号与正文并排,如:文献[2]所使用的方法是……。
- 4 参考文献的书写格式:
  - (1) 图书格式为:作者.书名.译者.版本(第 1 版可略).出版地:出版者,出版年.页数.  
例:1 钱伟长.穿甲力学.北京:国防工业出版社,1984.  
2 Bao Ting-Yu. The theory of potential equilibrium for interior ballistics and its application. Beijing: National Defence industry Press,1988.
  - (2) 期刊书写格式为:作者.文章题名.译者.刊名,年,卷(期):页数.  
例:1 朱自强.无激波机翼的设计方法.航空学报,1988,9(11):499~508.  
2 Gray R M. Vector quantization. IEEE ASSP Magazine,1984,1,4~29.  
3 Bezdek J C, et al. FCM: The fuzzy c-mean clustering algorithm. Comput. Geosciences,1984,10(2):191~203.