

# 基于 JGraph 的图像处理编程平台的实现

杨林蛟<sup>1</sup>, 张 鑫<sup>2</sup>

(1. 沈阳师范大学 计算机与数学基础教学部, 辽宁 沈阳 110034; 2. 沈阳师范大学 网络管理中心, 辽宁 沈阳 110034)

**摘要:** 为了实现一个图像处理可视化编程平台, 首先在数据流模型的基础上, 构建了一个树形数据流模型和调度算法, 并使用该模型作为图像处理可视化编程平台的基础运行模型; 然后在开源软件 JGraph 的基础上, 给出了平台的总体设计、基本图元设计和具体实现方法。实验结果表明, 与传统的图像处理编程软件相比较, 该平台易于掌握, 创建图像处理程序更简单、更快速, 结果一目了然。

**关键词:** 组件; 可视化编程; 数据流模型; 图像处理

中图分类号: TP317.4

文献标志码: A

文章编号: 1001-4551(2011)08-0978-05

## Implement of a image processing platform based on Jgraph

YANG Lin-jiao<sup>1</sup>, ZHANG Xin<sup>2</sup>

(1. Department of Computer and Mathematics Foundation Teaching, Shenyang Normal University, Shenyang 110034, China;

2. Network Management Center, Shenyang Normal University, Shenyang 110034, China)

**Abstract:** In order to solve the problems of creating a visual programming platform of image processing, firstly, a tree type data flow and scheduling algorithm on the basis of data flow model was created. Then the general design of the platform, basic pixel design and the specific implement method were put forward on the basis of an open source software JGraph. By experiments and the contrast with traditional image processing softwares, this platform can be easy grasped. The process of creating graph is made simpler and quicker, and the results can be fully understood at first glance.

**Key words:** component; visual programming; tree-dataflow model; image processing

## 0 引 言

对于图像处理和图像分析技术在新世纪的广泛应用, 除了研究人员在图像处理算法方面的不断创新以外, 高质量的图像处理平台的研发也是必不可少的。传统的图像处理过程都是基于文本编程的, 随着 LabVIEW<sup>[1]</sup>等可视化编程软件的出现, 许多公司和研究机构开始研发图像处理可视化编程平台<sup>[2]</sup>。可视化编程的图像处理平台是指该平台以拖拽图标的方式提供一个交互界面, 每一个图像处理的算法都是由带有输入/输出端口的图形组件来表示, 图形组件根据需要进行相互连接、编辑和删除。图像的输入/输出和处理分别由输入组件、输出组件和算法处理组件来实现。每

一个组件代表一个操作, 每一个组件的输入端口代表一个函数的输入参数, 输出端口代表返回值。一个可视化程序是由一组组件和它们之间的连线组成的有向图, 数据从一个组件的输出端口沿着连线的方向传递到另一个组件的输入端口<sup>[3-5]</sup>。

为了加快开发进度, 本研究以树形数据流模型为基础, 基于 JGraph 框架设计并实现一个能够可视化编程的图像处理平台。

## 1 树形数据流模型

组件和组件间的连线是可视化编程平台的主要对象, 完成各种数据处理的功能。目前大部分可视化编程平台的组件和连线都是采用数据流模型来实现

收稿日期: 2010-12-24

作者简介: 杨林蛟(1976-), 男, 山东平度人, 主要从事教育技术学方面的研究. E-mail: ylj@synu.edu.cn

的<sup>[6]</sup>。

### 1.1 数据流模型的概念

数据流模型以有向图来表示计算过程,在图中节点代表数据操作,边代表数据的流动。当每个操作的输入数据都到达时,该操作被激活。一个简单的数据流图如图1所示,一个无环的数据流程序代表数学表达式 $1 * 2 + 3 * 4$ ,在该图中运算 $1 * 2$ 和 $3 * 4$ 很明显能并行执行,同时执行的顺序、数据的运算和流动过程都一目了然。

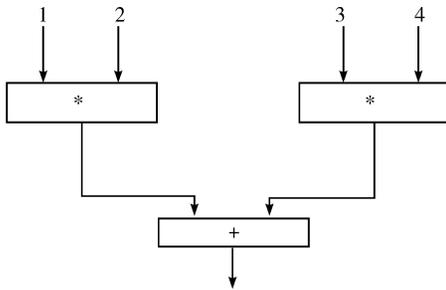


图1 一个简单的数据流图

### 1.2 树形数据流模型的形式化描述

由于数据流模型结构简单,许多可视化编程台都是基于动态纯数据流模型的<sup>[7-8]</sup>。但由于其内建的数据驱动机制需要一个动态调度器,必须连续对节点进行扫描以判断其是否能够激活,这种连续扫描的调度机制几乎费尽了宝贵的CPU资源。事件并发数据流模型<sup>[9]</sup>虽然提高了系统的相应速度和运行效率,但模型比较复杂,不容易实现。通过对图像领域和现有的面向对象编程语言的分析,本研究提出了一种树形数据流模型。实验表明该模型具有以下特点:①明显降低了CPU的占用率,提高了程序运行的效率;②采用拓扑排序来一次性遍历节点,不需要反复扫描节点集和边集;③引入树模型来进一步明确节点端口和节点的关系,便于使用现有的面向对象语言来实现。

定义1 一个数据流图可以表示为一个2元组 $P = (N, E)$ ,其中:

(1) $N$ —节点集,每一个节点包括一个输入端口集合 $T_i$ 和一个输出端口集合 $T_o$ 。

(2) $E$ —边集,是一个有序积 $(n, T_o) \times (n, T_i)$ 的一个子集, $n$ 为节点。

每个节点代表一个操作,每条边代表一条数据流通路。

定义2 一个边可以表示为 $e = (u, oi, n, it)$ ,其中:

(1) $u, oi$ —表示该边的源节点 $u$ 的输出端口 $oi$ 。

(2) $n, it$ —表示该边的目的节点 $n$ 的输入端口 $it$ 。

定义3 一个节点组件是一个2层结构的多叉树,可表示为一个五元组 $c = (v, P, f, A, s)$ ,其中:

(1) $v$ —组件的唯一根节点,因此也可以用 $v$ 来代表该组件。

(2) $P$ —端口集,分为输入端口集 $P_i$ 和输出端口集 $P_o$ 。

(3) $f$ —该组件的功能函数,当组件被激活时执行。

(4) $A$ —组件的属性集。

(5) $s$ —组件的状态,表示该组件是否被激活。

在逻辑上一组件 $c$ 包含一系列输入和输出端口,这些端口都是 $v$ 的孩子,可以使用 $c, P_i$ 或者 $v, P_i$ 来表示一个组件的输入端口部分,用 $c, P_o$ 或 $v, P_o$ 表示组件的输出端口部分。

定义4 一个端口可以表示为 $p = (i, c, f, s, A, e)$ ,其中:

(1) $i$ —该端口在该组件中的序号,即该组件的第几个孩子。

(2) $c$ —该端口的类型,表示该端口是输入端口还是输出端口。

(3) $f$ —该端口的祖先,即组件。

(4) $s$ —端口的状态,表示该端口是否被激活,初始值为false。

(5) $A$ —端口的属性集。

(6) $e$ —端口所连接的边。

一个组件的激活条件是当该组件的所有输入端口的状态 $p, s$ 都为true时该组件被激活。

### 1.3 树形数据流模型的调度算法

遍历算法的作用是在树型数据流模型的基础上,采用拓扑排序的方法依次访问组件,算法如下:

(1)扫描所有组件,把所有没有输入端口的组件的根节点入激活组件队列;

(2)判断激活组件队列是否为空,空则退出;

(3)从队列取一个组件,执行组件的执行函数,并将结果传递到所有输出端口;

(4)取组件一个输出端口;

(5)端口是否有输出边,没有则跳到步骤(8)执行;

(6)将该端口的值传递给目的端口,并设目的端口的状态为true;

(7)由目的端口的得到目的组件,如果目的组件为可激活状态,将目的组件加入激活队列;

(8)取组件另一个未访问的端口,如果都访问了则跳到步骤(10)执行;

(9)跳到步骤(5)执行;

- (10) 将组件从队列中删除;
- (11) 跳到步骤(2)执行。

## 2 系统的设计与实现

### 2.1 系统结构

整个平台的模块结构和模块之间的关系如图 2 所示,系统模块由组件库、可视化程序编辑器、可视化语法分析器、可视化程序执行器 4 个部分组成,最下面的框图分别对应各个模块的实现技术<sup>[10-11]</sup>。下面将分别介绍各个模块:

(1) 组件库。在平台中,有一个组件库,其中包括了一系列用于图像处理的算法和流程控制的类。它们与实际图像处理中的滤波器相对应,完成一些图像处理的功能。在可视化程序编辑器中,这些模块能够更加自由和灵活地组合成可视化程序。

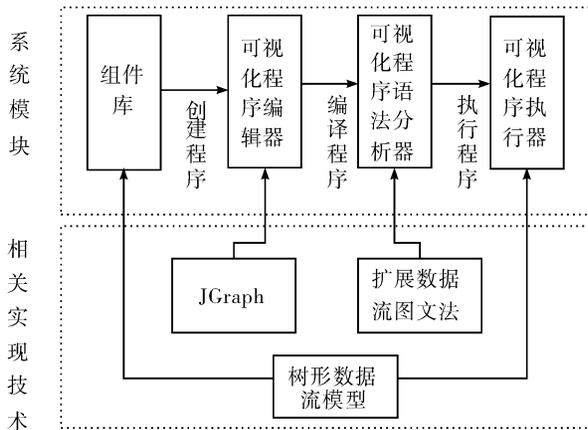


图 2 平台总体结构

(2) 可视化程序编辑器。在可视化语言中,可视化程序编辑器处在一个相当重要的地位,其直接反映可视化编程环境是否被实际用户所接受。用户通过鼠标可以选中组件浏览器上的组件,通过拖拽的方式在可视化编辑器中创建相应的组件,并用这些组件图像处理程序。

(3) 可视化程序语法分析器。在可视化编程平台中,语法分析器的作用相当于文本编程平台的编译器的作用,当用户使用组件表达出自己的程序时,必须用图语法来判断程序中是否有错,这与在文本语言编程中编译程序用字符串文法来检查程序中的语法错误相似。

(4) 可视化程序执行器。可视化程序执行器负责运行经过语法分析的可视化程序,并最终显示程序执行的结果。可视化程序执行器根据数据的有效和数据的需要来决定哪些组件需要被激活。

### 2.2 基本图元设计

本研究用 UML 建立的组件、端口和边的对象模型如图 3 所示。根据树形数据流模型的定义,组件和端口之间的关系是父节点与孩子节点之间的关系,采用 Java 类库中的 DefaultMutableTreeNode 类作为基类可以很容易地实现这种关系,DefaultMutableTreeNode 是树数据结构中的通用节点,组件类和端口类作为其子类可以直接使用其 getParent() 和 getChildAt() 函数来相互访问。当一个组件要添加其孩子端口时可以直接使用 DefaultMutableTreeNode 类中的 add() 函数。

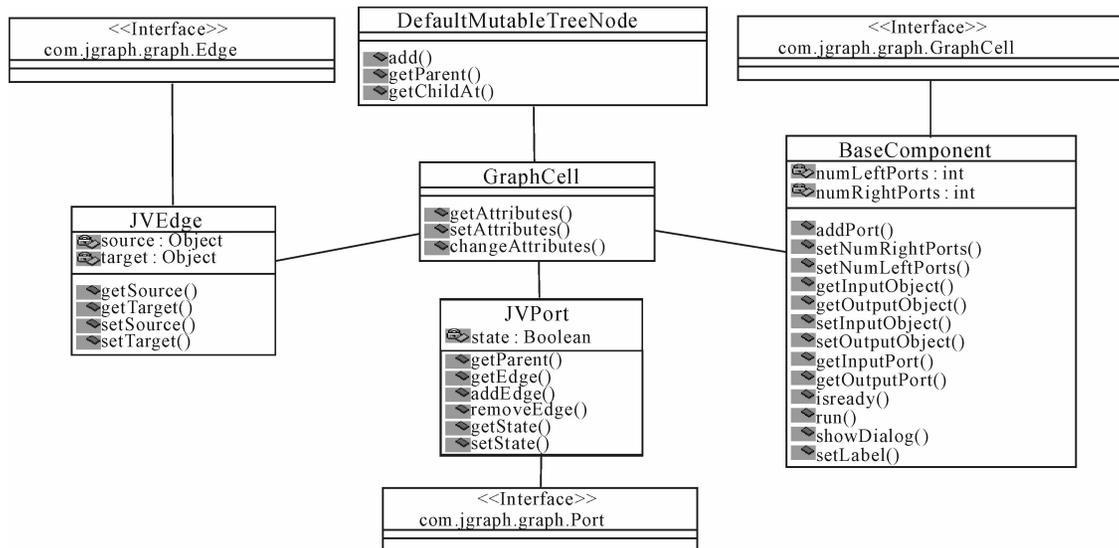


图 3 组件、端口和边的对象模型

可以把组件、端口和边分别看作具有不同形状的

图元,GraphCell 类定义了组件、端口和边的共有的属

性,如大小、颜色、形状等,而具体实现需要根据具体实现类的功能而定。BaseComponent 类是组件的基类,具体元件类继承该基类。该类是一个抽象类,提供两个抽象函数 run() 和 showDialog(), 具体子类通过实现这两个函数来实现不同的图像处理功能。JPort 类是端口类,根据树形数据流模型定义了 6 个函数,同时作为 DefaultMutableTreeNode 的子类,其实例可以被 BaseComponent 类 add() 函数添加为孩子节点。JVEdge 类是组件间的边类,如图 4 所示,在树形数据流模型中边不是直接连到组件上,而是从一个组件的输出端口连接到另一个组件的输入端口。

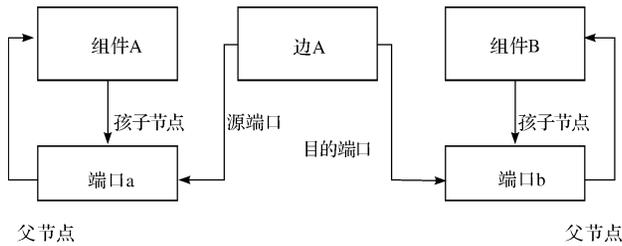


图 4 组件、端口和边的对象模型之间的关系

### 2.3 创建具体组件

在创建具体组件类时,只需继承 BaseComponent 类和实现 run() 函数和 showDialog() 函数,同时还需要在具体类的构造函数中给出具体类的端口数目、类型(输入端口还是输出端口)和组件的各种属性。

现以一个图像相加组件实现过程为例,图像相加组件有两个输入端口和一个输出端口,输入端口用来从外界获得图像,输出端口输出图像相加后的结果图,run() 函数执行图像相加处理,代码如下所示:

```
public class ImageAdd extends BaseComponent {
    public ImageAdd() {
        setNumRightPorts(1); //设置输出端口个数
        this.setNumLeftPorts(2); //设置输入端口个数
        SetLabel("imageAdd") //设置组件的标签
    }
    public void run() {
        //从输入端口 0 和输入端口 1 获取图像
        RenderedOp im0 = (RenderedOp) this.getInputObject(0);
        RenderedOp im1 = (RenderedOp) this.getInputObject(1);
        //进行图像相加处理
        RenderedOp im21 = JAI.create("add", im0, im1)
        //把处理结果送到输出端口
        this.setOutputObject(im2);
    }
    public void showDialog(Component parent) {
```

//鼠标左键双击弹出的对话框

### 2.4 平台运行效果

首先本研究使用平台构建对一幅图像用各种算子进行边缘检测,这些算子包括 frei-chen 算子、prewitt 算子、robert 算子和 sobel 算子,如图 5 所示,然后构建一个对乳房 CT 进行反转运算的编程,如图 6 所示。在示例中,本研究从组件浏览器中把需要的组件托拽到可视化程序编辑器中,通过鼠标动态地在各个组件之间建立连线。然后,用鼠标左双击 imageInput 组件弹出图像选择对话框,选择一幅图像。最后,编译和运行程序并显示结果。可以看到,与传统的文本语言函数只能有一个输出不同,该程序的组件可以有多个输出,这样可以便于图像的比较和分析。

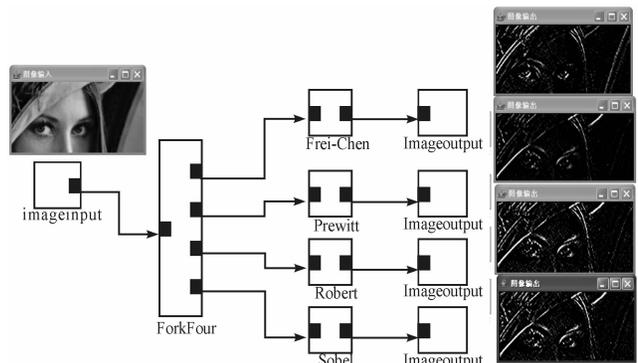


图 5 对一幅图像用各种算子进行边缘检测

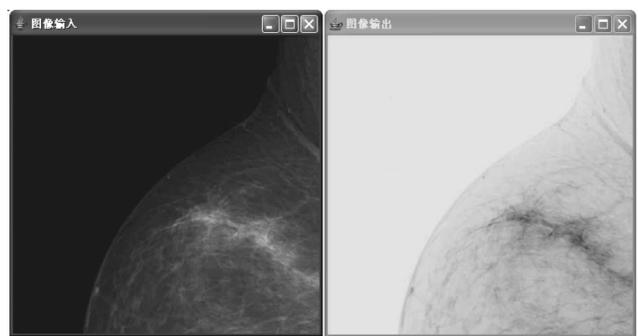


图 6 对乳房 CT 影像反转编程

## 3 结束语

本研究基于开源软件 Jgraph 设计并实现了一个图像处理平台,该平台能够可视化编程,能处理包括 DICOM 等各种格式的医学影像,通过继承该平台组件的基类能够自由扩展平台的组件库,并提供了一个友

好的编程界面。经实验表明,与传统的基于文本编程相比,该平台易于使用,几乎所有的编程工作都只用鼠标完成;构建程序更快速、更容易,所见即所得;易于维护,组件及它们间的连线可以任意添加、删除和修改,而不用修改源程序代码。

**参考文献 (References) :**

[1] National Instruments. LabVIEW[EB/OL]. [2008-12-01]. <http://en.wikipedia.org/wiki/labVIEW>.

[2] WHELAN P F, SADLEIR R J T. A visual programming environment for machine vision engineers [J]. **Sensor Review**, 2004, 24(3):265-270.

[3] SMITH B J. Conceptual graphs as a visual programming language for teaching programming[C]// Visual Languages and Human-Centric Computing (VL/HCC), 2009. Corvallis: [s. n. ], 2009:258-259.

[4] TU Pin-ying, CHEN W K, CHENG Y C. A Visual Programming Language for Data Transformation [C]. International

Symposium on Computer Science and its Applications, 2008; Australia: [s. n. ], 2008.

[5] 韩秀清,曾晓勤,邹 阳,等. 图文法综述[J]. 计算机科学, 2008, 35(8):10-16.

[6] 徐小良,汪乐宇,周 泓. VPP 虚拟仪器元件库的实现框架[J]. 工程设计学报, 2003, 10(2): 75-79.

[7] 周 泓,汪乐宇,陈祥献. 虚拟仪器系统软件结构的设计[J]. 计算机自动测量与控制, 2000, 8(1): 21-24.

[8] 方 林,谢 立. 关系图文法及其应用[J]. 软件学报, 1997, 8(2): 87-92.

[9] 王瑞荣,汪乐宇. 事件并发数据流模型[J]. 软件学报, 2002, 14(3): 409-414.

[10] 耿晨歌. 面向虚拟仪器系统的可视化编程研究[D]. 杭州:浙江大学仪器系, 1999.

[11] 王瑞荣. 基于事件触发并发数据流模型的可视化编程语言研究[D]. 杭州:浙江大学仪器系, 2003.

[编辑:李 辉]

(上接第 964 页)

经相关计算可得到  $z$  值的测量均值、均值误差、绝对误差均值、误差标准方差,如表 2 所示<sup>[11]</sup>。

**表 2 误差分析比较表(单位:mm)**

$z$ 值	本研究数据
标准平面值	1 630
测量值均值	1 618. 83
绝对误差均值	13. 51
误差标准方差	7. 15

通过表 2 中的对比数据得出,本研究中以 1 630 mm 作为检测对象,可以看出本研究中的  $z$  测量值均值为 1 618. 83 mm,绝对误差为 13. 51 mm,相对误差为 0. 83%。在保证标定速度的同时达到了较高的标定精度,达到了预期的目标。

**4 结束语**

感知二步插值标定方法让计算机具备感知 CCD 光轴与标定平面法线间的偏差能力,本研究通过实验证明了该方法的有效性。在标定过程中该方法不必对摄像机的内外参数和系统结构参数进行标定,而是通过摄取标定板上的已知相对关系点的图像,建立成像面上的角点坐标与世界坐标之间的对应关系,然后根据已知角点插值得到平面上任一点世界坐标来完成系统标定,从而提高了标定速度与标定精度。

**参考文献 (References) :**

[1] 邱茂林,马颂德,李 毅. 计算机视觉中摄像机标定综述[J]. 自动化学报, 2000, 26(1):43-55.

[2] 韦争亮,钟约先,袁朝龙,等. 单摄像机单投影仪三维测量系统标定技术[J]. 清华大学学报:自然科学版, 2009, 49(2):202-205.

[3] 李宏伟. 结构光三维视觉检测关键技术研究[D]. 黑龙江:哈尔滨工程大学自动化学院, 2006.

[4] CAI Ji-hua, LIANG Cheng-song. Mathematical model for the system calibration of digital camera [J]. **Mathematics In Praticce And Theory**, 2009:40 (14):52-59. .

[5] 任延俊,郭 霏,姜淑华,等. 基于径向约束的 CCD 相机标定参数的整体优化[J]. 长春理工大学学报:自然科学版, 2009, 32(2):13-17.

[6] 陈 工. 基于单目视觉的自主移动机器人检测系统[J]. 机电产品开发与创新, 2007, 20(1):4-6.

[7] 郝丽俊,程 胜. 基于结构光亚像素提取的三维重建[J]. 中国医疗器械杂志, 2009, 33(5):336-338.

[8] 杜艳可,郝飞龙,段龙江. 系统标定的一种新方法[J]. 机械工程学院学报, 2009, 21(6):75-78.

[9] 刘 超,高井祥,杨化超. 基于平面控制格网的相机标定算法研究[J]. 测量科学, 2009, 134(14):50-52.

[10] 徐士良. 数值分析与算法[M]. 北京:机械工业出版社, 2007:188-200.

[11] 张振祥,陈永清. 基于线阵 CCD 的轴承外圈表面缺陷检测[J]. 轻工机械, 2010, 28(4):70-72.

[编辑:张 翔]