

可视化编程数字图像处理平台的界面设计与实现^①

吕 敏 沙 莎 (中南大学 信息科学与工程学院 湖南 长沙 410083)

摘 要: 可视化编程语言的出现给人机界面的设计带来了巨大的变化,通过对当前可视化编程现状的研究,利用 VB 的 OOP 和 TreeView 控件,提出并实现了将图形控件应用到了图像处理平台的界面设计当中。使用封装技术把算法信息封装在这些图形控件中,用 VC 中的 DLL 实现算法,避免了代码的编写,用户通过使用这些图形控件组成流程图开发出自己需要的软件,提高了图像处理的速度。该图形算法控件开发方法能增强组态的灵活性并提高开发效率,实现了软件重用。

关键词: 可视化编程; 图像处理平台; 图形算法控件; 封装技术; 流程图

Interface Design and Implementation of Digital Image Processing Platform

LV Min, SHA Sha (School of Information Science and Engineering, Central South University, Changsha 410083, China)

Abstract: The appearance of visual programming language has brought great changes to the design of man-machine interface. In the paper, through the research on current visual programming, a graph control is proposed and used as an interface design of an image processing platform. Algorithm information, which is implemented by DLL of VC, is encapsulated in these graphical controls through the use of packaging technology, and users can develop their own software by using the graphical controls, in order to avoid the compilation of code and improve the speed of image processing. The development methods of graphical controls can enhance the flexibility of configuration and improve development efficiency, and to achieve software reuse.

Keywords: visual programming; image processing platform; graphic algorithm control; packaging technology; flow diagram

1 引言

随着图像处理领域的应用越来越广泛,各种图像处理方法的算法也越来越多,不同环境下的图像和不同的应用场合需要一个或多个算法模块的协同工作,每次实验都重新编写这些模块,是一件费时又费力的工作。提供一个便于实验和开发软件的用户界面可以避免代码的编写,加快软件开发过程,而图形技术可以快速的生成可视化的语素,供信息调用者使用。所以利用封装技术将复杂的数据处理信息封装于这些可视化的语素中,不用编写代码就完成图像处理的算法实验工作,进而省去大量的调试工作。

本平台利用现有的可视化开发语言,提出并实现

了把各种图像处理算法以图形控件^[1]形式集成到平台中,双击图形算法控件界面栏中的图形算法控件,用户便可以在流程编辑器界面中以“画图”的方式灵活开发应用程序,一个图形化算法控件流程就构成了一个小型的软件,不用编写相应的代码,方便又简单。

2 图形控件编程概述

可视化编程用流程图^[2]表示数据的处理过程。图形控件可以生成明了的可视化语素,供用户使用。较之于传统的文本语言,可视化编程具有直观形象和缩短开发周期的优势。流程是一组相互依赖的节点的集合,集合的元素可以为空也可以是多个,相互依赖是

① 收稿时间:2010-01-04;收到修改稿时间:2010-02-01

指数据的传送和交换关系，打个比方，一个节点的输出是另一个节点的输入，则它们之间就存在一种相互依赖的关系，并且后者必须晚于前者执行。节点在本文当中指的是具有一定算法功能的图形控件。现如今比较有代表性的可视化流程开发有 HP 开发的 HP VEE 和测控领域的 LabView 等软件。

图形控件编程语言^[3]的基本元素是控件和连接控件走向的边，控件代表各个功能模块，边用来表示控件走向。用户可以在使用的时候拖拉这些图形控件和连接控件走向的边，设置一些图形控件的属性参数，从而实现应用程序的开发。图形控件^[4]和连接控件走向的边可以动态的加入和删除，这样就极大的减少了代码的输入，对于开发人员可以省去编写代码这样繁重的工作，对于非计算机人员也可以进行掌握和使用，降低了平台的使用门槛。

3 平台界面设计与实现

流程^[5]是一组相关的图形算法控件的集合，本平台界面设计的主要任务就是让用户在流程编辑器界面中通过拖拉工具栏中的图形控件并设置图形控件的属性参数来绘制流程，从而实现应用程序的开发。界面主要包括三大部分：流程编辑器、图形控件栏、属性和帮助窗体。其中流程编辑器中的流程运行和图形控件栏的界面设计和实现是本文研究的重点和难点。

3.1 流程编辑器

流程编辑器界面是一个子窗体，并使用图形编辑组件 General 提供的接口实现图形的编辑功能。对于图形化实验流程开发环境，它的主要功能是绘制实验流程图，一个实验流程图的基本元素可以分为以下三种，图形控件、节点和连线。流程编辑器就是提供绘制基本元素的容器。

流程编辑分为图形控件的拖曳、图形控件的连线和其他图形编辑功能。图形控件的拖曳指的是在图形算法控件界面中选择所需的图形控件，用鼠标将其拖曳到流程编辑器界面中进行绘制。它的实现包括以下三部分：首先，在图形控件界面库中单击鼠标左键选择图形控件；其次，拖动控件到流程编辑界面的某一位置；最后，抬起鼠标左键，完成对图形控件的绘制。

用户通过对图形控件进行边的绘制定义其先后顺序。边绘制的实现可以分为以下三个步骤：首先，按下 shift 键；其次，在一个图形控件 M 的节点上单击

鼠标左键开始绘边；最后，在图形控件 N 节点上单击鼠标左键，完成图形控件 M 到 N 的边的绘制。

当所绘图形控件并不是所需的功能控件时，可用编辑器界面提供的流程删除操作进行相应的删除。用带箭头的线连好绘制的控件，一个按照箭头走向运行的应用程序也即开发完成。用户还可以保存当下所绘的流程和流程中控件和边所设置的属性参数，平台中用写文件的方式实现。打开流程功能可以打开保存的流程，平台中用读文件的方式实现。

运行流程是平台的重要工作，本文采用数据驱动机制^[6]并用栈的结构实现流程中节点的运行，流程运行过程图如图 1 所示，其中 TopNode 代表栈顶节点，top 代表栈顶指针，语言描述如下所述：

- (1) 初始化保存图形控件和边信息的节点队列。
- (2) 存储流程编辑器中存在的图形控件和边，计算各图形控件的入度和出度，更新节点队列信息，入度为零的节点进栈。
- (3) 判断栈是否为空。如果空，运行结束。否则，栈顶节点 TopNode 出栈，根据该节点所存储的图形控件的类型进行相应的操作。若是简单算法控件或者是复合算法控件则执行该控件所封装的算法，如若是控制流程图图形控件，则根据解析控制图形控件封装的解析函数来决定走流程图中的哪个分支。如若是控制循环图形控件，则根据解析控制图形控件封装的解析函数来决定执行循环体或者是跳出循环体。

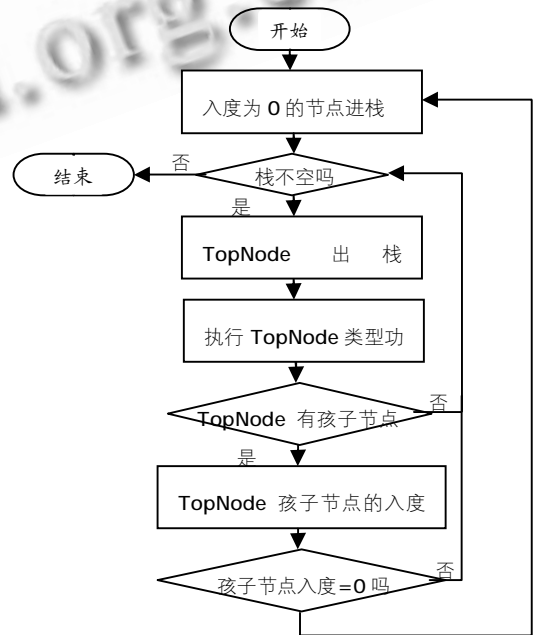


图 1 运行流程图

(4) 判断 TopNode 的孩子节点的入度是否为零, 如果为零让其孩子节点入栈, 转到 3 执行, 否则直接转到 3 执行。

3.2 图形控件栏

图形算法界面提供图形控件的显示功能, 并使用图形编辑界面中的 TreeView 控件为控件的显示提供相关功能。本平台把具有图像处理功能的算法等做成图形控件。

实验需求不同, 实验流程也就各不相同, 但所有实验流程大体上可以分为顺序结构、分支结构和循环结构。根据上述基本结构, 将图形控件分为三类: 简单算法控件、逻辑算法控件和复合算法控件。简单算法控件封装了某个图像处理算法; 逻辑算法控件封装了解析表达式的功能函数, 主要用来控制流程的走向; 复合算法控件封装了某些算法集合。封装算法的图形控件节点可以用一个四元组来表示: $ImageNode = (NodeIndex, InDegree, OutDegree, NodeType, F)$, 其中 $NodeIndex$ 是图形控件节点的索引值, $InDegree$ 表示该控件的入度, 也即该图形控件的父节点个数, $OutDegree$ 表示控件的出度, 也即是该节点的孩子节点的个数。 $NodeType$ 表示该图形节点的类型, F 表示该节点封装的功能算法或者算法集合。连接控件走向的边的接口用一个四元组表示: $Edge = (Index, Start_Node, End_Node)$, $Index$ 是该边在流程图线队列中的索引值, $Start_Node$ 表示数据流输出节点接口, End_Node 表示数据流输入节点接口。边的结构如图 2 所示。边控件的接口既可以传输函数信息, 也可以控制流程走向的选择。算法以控件的形式出现, 控件之间通过接口联系, 算法控件内部的实现独立, 所以平台中的算法控件具有高内聚低耦合的优点。

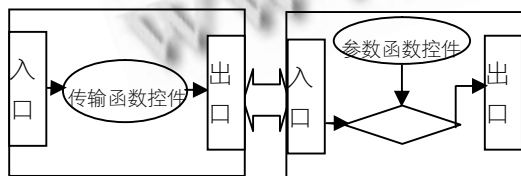


图 2 控件结构

图形算法控件是流程图中特定功能的实现, 封装了图像处理的算法, 图形控件栏加载时的工作包括以下几部分:

(1) 工具栏控件初始化: 在图形控件运行之前, 初

始化原始数据, 包括图形控件在流程中的索引值, 图形控件的类型, 图形控件的入度和出度, 图形控件上显示的功能函数名, 图形控件关联的图像的索引值等。

(2) 控件属性初始化: 初始化各图形控件关联的属性面板。

(3) 控件封装信息的实现: 每个控件封装了不同的功能, 是本机上的一个 DLL, 在节点之间通信时, 传递的就是该 DLL 的实现。

本平台图形控件节点的数据类型是图像的数据信息, 节点数据的传输实现是通过把每个节点的处理结果放在一个文件中, 当子控件节点需要时根据边控件信息再取出所需文件数据。

3.3 属性窗体和帮助窗体

属性窗体有两种, 一种是属性控制窗体, 一种是属性参数设置窗体。属性控制窗体包括 if、while、case 等结构窗体。属性设置窗体主要是用来设置图形控件相关联的参数。

帮助窗体包括算法的适用范围和平台的使用规则文档, 是开发人员重要的参考资料。

4 系统界面仿真实例

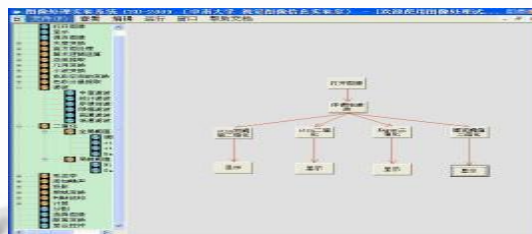


图 3 平台绘制流程界面



图 4 运行结果图

系统提供了包括灰度变换、直方图处理、算术逻辑运算等七十多个基本的图像处理算法。用户在平台中能完成基本的图像处理仿真实验。以图像分割为例, 用户首先在左边的图形控件栏里选取“打开图像”控件、“滤波”里

(下转第 218 页)

面的“序惯性滤波”控件、各种“二值化”控件和“显示”控件拖曳到流程编辑器中,根据处理过程把各个控件用带箭头的线连接起来并设置好各个控件的属性,构成了图 3 所示的实验流程,运行结果如图 4 所示。

5 结束语

本文论述了可视化编程图像处理平台的界面设计及实现。封装算法的图形控件方便了用户,可视化的平台为从事图像处理软件开发的人员提供一个检验,对比算法效果的环境,加快了开发速度,提高了科研效率。

参考文献

1 董新法,沈夏炯.基于图形控件的可视化编程研究.研究与开发,2008,11:26-28.

2 Rasure J, Young M. Data flow visual languages. Potentials, IEEE,1992,2 (11):30-33.

3 张智,鲁强.一种图形化流程语言.计算机与数字工程,2006,34(1):80-83.

4 王建新,陆炜妮,王伟平.基于组件的数字图像处理仿真系统的设计与实现.系统仿真学报,2004,16(6):1213-1216.

5 Wauters P, Engels M, Lauwereings R. Cyclo-dynamic dataflow.IEEE Proceedings of the Fourth Euromicro Workshop on Parallel and Distributed Processing. Braga,Portugal,1996,319-326.

6 王瑞荣,汪乐宇.面向图形化编程的事件触发并发数据流模型.浙江大学学报,2002,36(5):535-539.