

Socket 及 Matlab 引擎技术在工程软件中的应用

孙冬, 明军, 吴先良, 王伟

(安徽大学 计算机智能与信号处理教育部重点实验室, 安徽 合肥 230039;

安徽大学 电子科学与技术学院, 安徽 合肥 230039)

摘要:为减少工程软件开发的复杂度, 充分利用 Matlab 工具箱提供的各类功能强大的函数, 文中在客户端与服务器端远程通信系统模型的基础上, 通过对服务器端程序的进一步配置, 使其具有自动调用本地 Matlab 工具箱函数进行数据分析的功能, 充分发挥 C++ 和 Matlab 各自的优势。并将此技术应用到烟叶大片率实时检测系统中, 获得了很好的效果, 证明了有效性和可行性。

关键词:分布式技术; Socket; Matlab 引擎

中图分类号: TP311.5

文献标识码: A

文章编号: 1005-3751(2006)02-0029-03

Application of Socket and Matlab Engine Technology in Engineering Software

SUN Dong, MING Jun, WU Xian-liang, WANG Wei

(Ministry of Edu. Key Lab. of Intelligence Computing and Signal Processing, Anhui Univ., Hefei 230039, China;

School of Electronic Science and Technology, Anhui Univ., Hefei 230039, China)

Abstract: In order to reduce the complexity of engineering software development, and make full use of many powerful Matlab toolbox functions, put further configuration in this paper to the server which is based on the client-server telecommunication system model. So the server has the ability to call local Matlab toolbox functions automatically to do data analysis, and takes advantage of C++ and Matlab respectively. In addition, apply this technology to a real time processing of detecting the quality of the tobacco leaves, which proves the validity and feasibility.

Key words: distributional technology; Socket; Matlab engine

0 引言

随着计算机技术的飞速发展及广泛应用, 分布式软件变得越来越重要, 在很多情况下, 软件设计者需要把软件分为客户端和服务端两部分, 其中客户端程序工作在数据采集现场, 而服务器端程序负责对远程客户端采集的数据进行处理。

C++ 是常用的软件开发语言, 在实现交互界面、数据采集和端口操作等方面具有其它语言无法比拟的优势, 但是 C++ 在数据分析处理和算法工具等方面却有着明显的弱点。Matlab 作为当今世界上应用最广泛的数学软件之一, 拥有着功能强大的各类工具箱, 可以运用简单的几行代码完成复杂的数据分析处理操作。因此, 文中将 C++ 同 Matlab 结合运用, 取长补短, 实现优势互补, 在实际工程软件中取得了很好的效果。

收稿日期: 2005-05-09

基金项目: 安徽省自然科学基金资助项目(01042105)

作者简介: 孙冬(1982—), 男, 安徽亳州人, 硕士研究生, 研究方向为视频图像处理及模式识别; 明军, 硕士生导师, 研究方向为视频图像处理及模式识别。

1 运用 Socket 进行远程通信

Windows Socket 是 Windows 下的一套网络编程接口^[1,2], 它定义了 API 与 Internet 协议族(通常是指 TCP/IP)连接的方法。在进行点对点通信时, 将会用到一对 Socket, 它们分别应用于客户端和服务端, 只要这对 Socket 接通, 那么客户端和服务端就可以相互进行通信。

直接使用 Socket API 进行开发比较麻烦, 而事实上也不必这样做, 因为当今主流的 C++ 开发工具 Visual C++ (VC) 和 Borland C++ Builder (BCB) 等都提供了对这些 API 进行了很好封装的类。下面以 BCB6.0 为例, 对服务器端和客户端程序的建立及其通信细节进行研究。

1.1 服务器端与客户端配置

首先把 Internet 组件页上的 TServerSocket 组件添加到工程中, 以对应用程序进行配置, 使其成为 TCP/IP 服务器, 然后对 Socket 的 Port 的属性值进行设置, 为其指定通信时所使用的端口号, 最后运用 Open() 方法打开服务器, 就可以等待客户端的连接请求了^[1]。用同样的方法, 把 Internet 组件页上的 TClientSocket 组件添加到工程, 以

完成对工程的 TCP/IP 客户端的配置。为使其与服务端能够进行通信,还需指定通信端口号和待通信服务器的 IP 地址,这可以通过对 Port 属性和 Socket 的 Address 进行设置来完成,注意 Port 的值必须与服务端端的 Port 值相同。设置完成后,就可以使用 Open() 方法向服务器提出连接请求了,当然,此时必须保证服务器端程序已经运行。

1.2 数据传送

默认情况下,远程数据的传送是以“非阻塞”方式进行的:当客户端的 Socket 准备对服务器端(或者是逆过程)进行读或写操作时,服务器端的 Socket 便会触发 OnClientRead 或 OnClientWrite 事件,可以在事件响应函数中加入数据发送和接收的代码。

Socket 提供了 SendBuf(void * buff, int n) 和 SendText(String str) 方法来发送数据,前者发送由 buff 指定的长为 n 字节的数据块,而后者仅发送一个字符串。

同样地,对于数据的接收,可用 ReadBuf(void * buff, int n) 和 ReadText(String str) 方法,参数的含义同上。

2 C++ 本地调用 Matlab

ActiveX 是 Windows 对象集成的一个标准协议,它是组件对象模型(COM)的一个子集,其目的是通过一个通用机构,实现软件之间的相互提供服务的目的。

2.1 Matlab 引擎

Matlab 提供了 ActiveX 自动化技术,它允许其它的 ActiveX 组件能对 Matlab 进行控制。5.0 版本以后的 Matlab 都提供了引擎组件,可以由它来完成控制 Matlab 的任务,因此可以通过服务器端程序对该引擎的操纵来实现对 Matlab 的控制,从而完成对客户端数据的处理。

有 6 个常用的操纵 Matlab 引擎的函数^[3]:

* Engine * engOpen(const char * startcmd)

功能:打开一个 Matlab 引擎。

* void engClose(Engine * ep)

功能:关闭打开的 Matlab 引擎。

* engEvalString(Engine * ep, const char * cmd)

功能:把 cmd 所代表的 Matlab 命令送入 Matlab 进程中,并立即执行。

* engOutputBuffer(Engine * ep, char * p, int n)

功能:把 Matlab 的文本输出送回由 p 指定的长为 n 的缓冲区。

* mxArray * engGetVariable(Engine * ep, const char * var)

功能:用于从 Matlab 工作空间(WorkSpace)中获得变量 var。

* int engPutVariable(Engine * ep, const char * name, mxArray * var)

功能:把变量 var 送至 Matlab 工作空间中,并重新命名为 name。

在实际开发过程中,实现应用程序与 Matlab 之间的

数据交换是尤为重要的一个环节,下面通过一个实例来说明具体的实现步骤。

假设需要把应用程序中的一个数组传给 Matlab,该数组事先定义为 double data[10]。

第一步:定义一个“矩阵变量”T,并对其进行初始化(尺寸为 1 * 10,双精度,仅存放实数):

```
mxArray * T = mxCreateDoubleMatrix(1, 10, mxREAL);
```

第二步:把 data 的值赋给第一步中定义的矩阵变量 T:

```
memcpy((char *) mxGetPr(T), (char *) time, 10 * sizeof(double));
```

第三步:通过 engPutVariable(ep, "T", T) 指令把矩阵变量 T 送入 Matlab 的 WorkSpace 中。

完成以上三步之后,应用程序中的数组“data”就以“矩阵变量”的形式送入到 Matlab 的工作空间中了。同样地,数据的读回操作与此类似,在此不做赘述。

2.2 应用程序的引擎配置

为了让服务器端程序能够使用这些引擎库函数和诸如 mxArray 这样的数据类型,需要 libeng.dll, libmat.dll 及 libmx.dll 的支持。加载 dll 的方法有两种:分别为动态和静态方法,下面以常用的静态加载方法为例来说明。

静态加载需要与 dll 相对应的 lib 文件,此文件称为导入库,它包含了指向函数在 dll 中具体位置的索引指针,这可以通过 BCB 提供的工具 implib.exe 来制作。以生成 libeng.dll 的导入库 libeng.lib 为例:

```
<implib.exe 所在路径>implib libeng.lib <libeng.dll  
所在路径>libeng.dll
```

导入库制作完毕后,接下来的工作与调用普通的.dll 库函数的步骤相同。首先把定义其函数原型的头文件 Engine.h 用“#include”指令加到代码中,然后使用“Add to Project...”选项把上面生成的 3 个导入库添加到工程。

为使编译器能找到 Engine.h 及其所包含的相关头文件,需要对“#include”指令的搜索路径进行配置。假定“<Matlab>”为 Matlab 的安装路径(如 D:\Matlab6p5),则可使用菜单中的“Project→Option→Directories→Conditionals”选项,在打开对话框后,把“<matlab>\extern\Include”路径加到“Include Path”文本框中。

同样地,为了使程序在运行时能找到所需的.dll 动态库,还需要把“<matlab>\extern\Include”和“<matlab>\bin”这两个路径加到“Library Path”文本框中。

经过这样的配置后,服务器端程序就可以调用 Matlab 来进行辅助的数据分析了,需要指出的是,运行此服务器端程序时,并不需要事先手动运行 Matlab,因为 engOpen() 方法会自动建立一个 Matlab 进程。

3 应用实例——烟叶的大片率检测

在烟叶大片率实时检测系统中,工作在现场的客户端

程序负责对样本烟叶进行图像采集,并通过网络把图像发至控制室中的服务器端。当服务器端程序收到图像后,便会自动调用 Matlab 程序对其进行大片率分析。

文中使用的叶片图像分析算法^[4]如下:

1)图像除噪。此环节使用函数 wiener2()进行自适应的维纳滤波。

2)叶片轮廓提取。此环节分3步进行:

A、使用函数 edge()进行边缘检测。

B、考察检测到的边缘的邻域,以此邻域是否包含背景为依据,使用自编函数去除由粗大脉络生成的虚假边缘。

C、使用函数 imdilate()实行膨胀操作,连接叶片轮廓的间断点。

3)填充轮廓。此环节分两步进行:

A、使用函数 imfill()进行轮廓填充,获得背景和烟叶完全分离的二值图像。

B、使用函数 imopen()进行开运算,扫除细小碎片的干扰。

4)叶片面积统计。此环节分两步进行:

A、使用函数 bwlabel(),确定每个叶片所占据的图像区域,并分别对其进行标识。

B、使用函数 imhist()统计每个叶片所占据像素的数目,获得面积。

需要指出,以上所列函数均来自 Matlab 的图像处理工具箱,这使得此算法的实现异常简洁。以第2步的A操作“边缘检测”为例,可以在C++中直接做如下的调用:

```
engEvalString(ep, "image = edge(image, 'canny');");
其中“ep”为此前建立的 Matlab 引擎指针,而引号内的则为将要执行的 Matlab 指令,其中字符串‘canny’表明使用 canny 算子进行边缘检测[5]。
```

表1 叶片分布统计表

叶片面积(像素)	2000以下	2000~3000	3000~4000	4000~5000	5000~6000
数目:	0	1	3	1	3
叶片面积(像素)	6000~7000	7000~8000	8000~9000	9000以上	/
数目:	1	5	1	2	/

(上接第28页)

作集中在:“中游”公共编辑平台的进一步建立,现在的窗口有5个:建立数据库;生成界面;生成代码;文字性文档管理;构件库重用。需要增加向图形转换的窗口。另外类之间复杂调用关系的代码生成能力不够,必须增加功能更加强大的编辑窗口。下一步准备用C#来实现该平台,增强图形绘制功能。相信在未来的几年内,该平台将被完善成一个强大的软件自动生成工具,成为ERP软件开发的重要利器。

参考文献:

[1] 王萍,蔡雨阳,黄丽华.面向对象建模方法——上海卷烟

如图1所示的烟叶图像,使用上述算法,可求出叶片面积的分布,如表1所示。

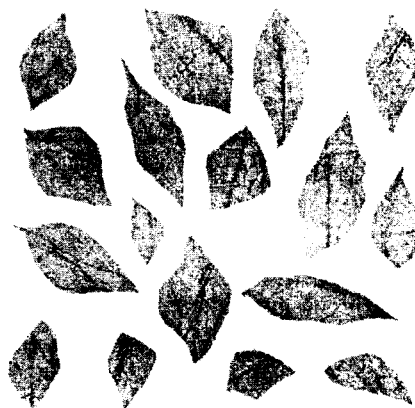


图1 叶片图像(512×512,缩小2倍显示)

4 结束语

文中所讨论的混合语言开发是一种各取所长的技术: Matlab 中许多耗时或者无法完成的工作对于C++来说是轻而易举的,而某些C++非常难以实现的算法对于 Matlab 又是易于实现的。分布式技术又使得一台服务器上的 Matlab 可以被多个客户端共享,既有效地利用了资源,又降低了解决方案的复杂性。

参考文献:

- [1] 徐科,杨朝霖,李滨涛.C++ Builder 实用技术与经典案例[M].北京:清华大学出版社,2002.283-303.
- [2] 李幼仪,甘志.C++ Builder 高级应用开发指南[M].北京:清华大学出版社,2002.395-401.
- [3] MATLAB, External Interfaces, User's Guide, Version 6[Z].2002.
- [4] MATLAB, Image Processing Toolbox, User's Guide, Version 3.2[Z].2002.
- [5] Castleman K R. 数字图像处理[M].朱志刚等译.北京:电子工业出版社,2002.

厂计调系统分析与设计[J].中国管理科学,2000,8(S):459-466.

- [2] 邵维忠,杨美清.面向对象的系统设计[M].北京:清华大学出版社,2003.
- [3] 邵维忠,蒋严冰,麻志毅.UML 现存的问题和发展道路[J].计算机研究与发展,2003,40(4):509-516.
- [4] 田丽从,张莉,周伯生.基于UML的集成化软件开发环境的研究与实现[J].北京航空航天大学学报,2003,29(10):935-938.
- [5] 陈英,施兴华,赵小林.基于UML软件开发过程的CASE平台研制[J].北京理工大学学报,2001,21(1):49-51.
- [6] 李留英,王戟,齐志昌.UML statecharts 的测试用例生成方法[J].计算机研究与发展,2001,38(6):691-697.