

# 基于 COM 的 Matlab 参数处理与图像嵌入在 VC 中的实现\*

王 维, 刘 斌, 柳炳利, 魏友华

(成都理工大学 数学地质四川省重点实验室, 四川 成都 610059)

**摘 要:**通过运用 Matlab 与 VC 的语言和平台特性,解决了二维数组参数的传递与输出;介绍两种 Matlab 图像嵌入 VC 界面的方法,并以主成分分析为实例,讲解了相关细节与难点。同时,介绍了一种更加灵活、简便的读取 Excel 文件的方法。

**关键词:** COM 组件; Matlab; VC; 二维数组; 图像嵌入

中图分类号: TP311

文献标识码: A

文章编号: 1674-7720(2011)24-0005-04

## Realization of parameters and figures processing between Matlab and the VC based on COM

Wang Wei, Liu Bin, Liu Bingli, Wei Youhua

(Sichuan Province Key Laboratory for Mathematical Geology, Chengdu University of Technology, Chengdu 610059, China)

**Abstract:** We solved the two-dimension array parameters passing and output by means of the characteristic of development language and platform. And, we will take PCA as an example so as to interpret how to embed Matlab figure in VC interface in detail. On the other hand, a flexible and simple way to read data from Excel file into VC program will be presented.

**Key words:** COM component; Matlab; VC; two-dimensions array; embedded images

基于 COM 组件的 Matlab 与 C++ 混合编程方式因拥有独立的运行环境和两种语言的互补优势而被众多科研人员和编程人员所接受,同时也是 MathWorks 公司推荐使用的混编方式。但在程序设计过程中,通常会遇到两个难点问题:(1)二维数组参数的传递与输出;(2)Matlab 生成图形嵌入 VC 工程界面中。对于第一个问题, Bruce McKinney<sup>[1]</sup>在 MSDN 上指出:“如果对一维数组进行操作,则 SAFEARRAY 函数变的简单且易操作。但是对于多维数组,同样的操作要复杂得多”,造成这一问题的根源在于 Matlab 与 C++ 对多维数组元素的存储方式不同。而在本项目开发过程中回避了这个复杂的过程,转而利用 Matlab 强大的数组处理功能将其解决。Matlab 完善的图形处理功能是其被科研人员所推崇的原因之一,但因其图形拥有独立的窗口,严重影响了整体界面美观和用户交互体验,所以 Matlab 图形嵌入是混合编程中不可回避的问题。对此将通过实例介绍两种图形嵌入方法并分析说明其优缺点和相关细节。

本文示例均在 Matlab R2008a 和 VC 6.0 平台下完成,且通过调试能够正常运行。

### 1 Excel 文件读取方法

VC 平台下的 Excel 文件读取方式复杂难懂,但如果使用 Matlab 与 VC 的混合编程方式编写将更加简便,而且拥有较强的可扩展性。下面代码详细讲解了该过程。

(1)在 Matlab 平台下建立 CExcelRead.m 文件,代码如下:

```
function [output, row, col] = CExcelRead(filePath)
//filePath 包含了指定文件的路径和文件名,由 VC 平台函数 GetPathName()获取
[typ, desc] = xlsinfo(filePath);
//xlsinfo()可获取指定 Excel 文件中各个 sheet 工作区的名称,返回值存入名为 desc 的 cell 数组中
sheet1 = cell2mat(desc(1));
//默认打开"sheet1"(名称任意),并将其转换为字符串
output = xlsread(filePath, sheet1);
//返回该 Excel 文件中的数据,返回值存入 output 数组
```

\* 基金项目:中国地质调查局工作项目(1212011120212)

中

```
[row,col] = size(output);
```

//获取录入数据的行列值,返回到 VC 平台以供其拷贝指定内存大小的数据

```
output = reshape(output,row*col,1);
```

//将该二维数组转换为一维。这是第二部分二维数组处理的第一步

保存文件后,使用 deploytool 工具将其制作成名为 FileOpen 的 COM(封装 MCR),注册该控件并将相应的文件拷贝到 MFC 的工程中,对于基础操作此处不再赘述。

(2)在 VC 平台下创建名为 PCA 的基于对话框的 MFC 工程,添加一打开文件的按钮控件, ID 为 ID\_FILE\_OPEN。篇幅所限只给出部分核心代码,首先在 Dlg 类中添加 3 个私有成员变量用以保存 Matlab 读取的数据及行列值,即:

```
double *m_originData; long m_row, m_col;
```

在响应函数中添加如下代码

```
CFileDialog fileDlg(TRUE);
```

```
fileDlg.m_ofn.lpstrTitle="请选择你的 excel 数据";
```

```
fileDlg.m_ofn.lpstrFilter = "Text Files (*.xls)\0*.xls\0All Files(*.*)\0*.*/\0";
```

```
if(IDOK==fileDlg.DoModal() {
```

```
CString fileName; IFileOpenclass *pfile;
```

```
VARIANT filePath,output,rowOutput,colOutput;
```

//m 文件的输入参数

```
fileName = fileDlg.GetPathName();
```

//获取文件路径并存于 fileName 中

```
CoInitialize(NULL); //COM 初始化
```

```
HRESULT hr = CoCreateInstance (CLSID_FileOpenclass, NULL,CLSCTX_ALL,IID_IFileOpenclass,(void*)&pfile);
```

//新建 COM 实例

```
VariantInit(&filePath); //VARIANT 数据初始化
```

```
:
```

```
filePath.vt = VT_BSTR; //指定 filePath 变量类型
```

//将 fileName 中保存的指定文件路径名保存至 filePath 参数中

```
filePath.bstrVal = fileName.AllocSysString();
```

```
pfile ->CExcelRead (3,&output,&rowOutput, &colOutput, filePath);
```

```
m_row=(long) rowOutput.dblVal;
```

//取出行列值及录入数据

```
m_col = (long) colOutput.dblVal;
```

```
m_originData = (double *)malloc(sizeof(double)*m_row*m_col); memcpy(m_originData,output.parray->pvData,m_row*m_col*sizeof(double));
```

//将 matlab 读入数据保存到 m\_originData 供后面程序使用

```
} 上述 C++代码中省略了异常处理和相关的内存、COM 释放代码,由于代码比较简单所以不做进一步解
```

析。请注意,下面将省略 COM 初始化及实例化等相同代码。

(3)需要重点说明的是该方法的扩展性。通常 Excel 文件中保存多个工作区,有时用户可能需要打开同一 Excel 文件中不同工作区中的数据,常规方法实现过于复杂,但对于本文介绍的方法可以通过修改添加几条语句即可实现。首先,新建一个 m 文件用来处理工作区的选择,代码如下:

```
function [sheet,col] = CSheetSelected(filePath)
```

```
[typ, sheet] = xlsinfo(filePath);
```

```
[row,col] = size(sheet);
```

由于返回值是一个 cell 数组,所以 VC 平台要使用 CStringArray 数据结构保存返回值,并显示各工作区名称供用户选择。然后,通过人机交互将用户选择的工作区参数保存并传递至 CExcelRead.m,通过在 CExcelRead.m 增加一个工作区选择参数,并对代码稍作修改即可。

## 2 二维数组参数的传递与输出<sup>[2]</sup>

下面以主成分分析为例介绍基于 COM 的 Matlab 与 VC 混合编程中二维数组参数处理。

(1)主成分分析 pcamat.m 代码如下:

```
function [eigenvector,eigenvalue] = pcamat(oriData,row, column)
```

//在 Excel 读入时已经完成了二维数组输出的关键步骤,即输出时将二维数组转换为一维数组。但在 VC 平台接收还原为二维时要注意,Matlab 数组存储方式是按列存储,而 VC 平台下数组是按行存储,所以转换时 0~row-1 为第一列,row~2\*row-1 为第二列,以此类推。本文输入参数 oriData 是一维数组,所以要将其还原为二维数组使 Matlab 程序能够正常运行,即下一行代码所示。

```
oriData = reshape(oriData,row,column);
```

```
dataSTD=std(oriData,0,1); dataMean = mean(oriData);
```

```
dataSR = (oriData - dataMean (ones (row,1),:))./dataSTD (ones(row,1),:);
```

```
[eigenvector,newdata,eigenvalue,Exa]=princomp(dataSR);
```

//第三行至此处均为主成分分析内容

```
eigenvector = reshape(eigenvector,column*column,1);
```

//与 Excel 文件读取时类似,将二维输出转换为一维数组进行输出

(2)保存后,封装打包为 COM 组件,并完成注册等相关操作。在 PCA 工程对话框上添加一个名为 PCATest 的按钮控件,核心代码如下:

```
VARIANT oriData,row,column,eigVector,eigValue;
```

```
VariantInit(&oriData); //参数初始化
```

```
:
```

```
oriData.vt = VT_R8|VT_ARRAY;
```

//定义SAFEARRAY 类型的一维数组

```
SAFEARRAYBOUND rgsadound[1];
```

```
rgsadound[0].lLbound = 0;
```

《微型机与应用》2011 年 第 30 卷 第 24 期

```
rgsadound[0].cElements = m_row*m_col;
oriData.parray = SafeArrayCreate(VT_R8,1,rgsadound);
oriData.parray->pvData = m_oriData;
//完成相关设置后,将第一步读入的数据录入到 oriData
中,即赋给 pcamat 的 oriData。到此完成了二维数组的传递
row.vt = VT_I4; col.vt = VT_I4;
row.lVal = m_row; col.lVal = m_col;
pca->pcamat(2,&eigVector,&eigValue,oriData,row,col);
memcpy(result, eigVector.parray->pvData,m_col*m_col*
sizeof(double));
```

综上,二维数组参数处理就是使用 reshape() 函数对输入输出数据维数进行变换来完成操作。

### 3 Matlab 图像嵌入 VC 界面<sup>[3]</sup>

#### 3.1 基于 CWnd 类的图像嵌入

在 Windows 操作系统下,所有应用程序的窗口都是基于 MFC 中的 CWnd 类。所以可以通过调用该类或其派生类中的方法实现图像嵌入。基本思想:在 Matlab 平台下用 COM 封装产生图形窗口的程序;在 VC 平台获取 Figure 窗口的句柄,将 Figure 窗口设为 VC 程序的子窗口;(3)将 Figure 窗口移动到指定显示位置。

程序实现如下:

(1)将原 pcamat.m 进行修改,添加生成图像的相关代码

```
function [eigenvector,eigenvalue] = pcamat(oriData,row,col-
umn,picName)
//增加 picName 参数,VC 平台下 hFig 将通过该名称获
取生成图像的句柄
figure ('NumberTitle','Off','MenuBar','None','ToolBar','Fig-
ure','Name',picName,'Units','Points');
//图像参数预设,保留工具栏。使用 Matlab 提供的工具
栏的所有功能是使用该方法的最大优点
percent = 100*eigenvalue /sum(eigenvalue);
```

```
pareto(percent); //计算贡献率 //画图
xlabel('主成分'); //画图
ylabel('方差占的比重(%)');
```

(2)封装打包成名为 figure 的 COM 组件,并完成注册等相关操作。因为修改后的 m 文件运行结果包含输出结果和图像两部分,所以下面有关图像处理的代码依然在 PCATest 控件的响应函数中。

```
CString WNDName = "Demo"; //自定义窗口名称
Ifigureclass *pic;
VARIANT oriData, row, col, picName,eigVector, eigValue;
VariantInit(&picName);
.....
```

```
picName.vt = VT_BSTR; //将自定义窗口名称赋予
Matlab 生成图像
```

```
picName.bstrVal = WNDName.AllocSysString();
```

```
HWND hFig; int timer = 50;
```

//用死循环确保可以获取到图像句柄,注意此处必须使

用 sleep(),给予系统足够的响应时间

```
while(1){
pic->pcamat(2,&eigVector,&eigValue,oriData,row,col,
picName);
Sleep(timer);
hFig = ::FindWindow(NULL,FIGName);
if(hFig != NULL){
break;
}
timer += 10;
pic->Release();
}
```

long lStyle = ::GetWindowLong (hFig,GWL\_STYLE); //设置 Figure 窗口样式。

//注意 SetWindowLong() 和 SetWindowPos() 先后顺序,详见 MSDN

```
::SetWindowLong (hFig,GWL_STYLE,lStyle& (~WS_CAP-
TION)&(~WS_THICKFRAME))
```

```
::SetWindowPos(hFig,NULL,0,0,0,SWP_NOMOVE|SWP_
NOSIZE|SWP_NOZORDER|SWP_NOACTIVATE|SWP_
FRAMECHANGED);CRect PlotRec;
```

```
CWnd *PlotArea = GetDlgItem(IDC_STATIC_FIGURE); //设置图像显示区域
```

```
PlotArea->GetWindowRect(&PlotRec);
```

```
long Width = PlotRec.right - PlotRec.left;
```

```
long Height = PlotRec.bottom - PlotRec.top;
```

```
::SetParent(hFig,PlotArea->GetSafeHwnd());
```

//设置图像的父窗口

```
::SetWindowPos (hFig,NULL,0,0,Width,Height,SWP_NO-
ZORDER|SWP_NOACTIVATE);
```

运行结果如图 1 所示。该方法的缺点是,在图像生成时会有闪烁现象。而优点是前面提到的可以继续使用 Matlab 提供的工具栏。鉴于该缺点影响整体美观,所以引入下面第二种方法。

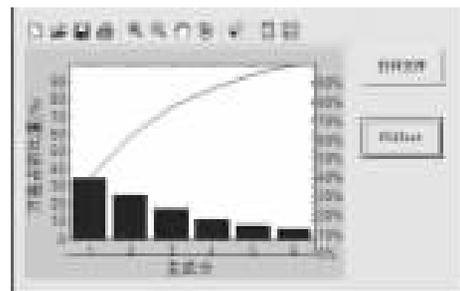


图 1 方法 1 的运行结果

#### 3.2 基于 Bitmap 类的图像嵌入

通过 Bitmap 类将 Matlab 生成的 .bmp 文件加载到 VC 工程中,使用 Bitmap 类中的成员函数对图像进行处理。由于相关函数可以通过帮助手册获取,所以此处不再给出相应代码,运行结果如图 2 所示。该方法避免了方法

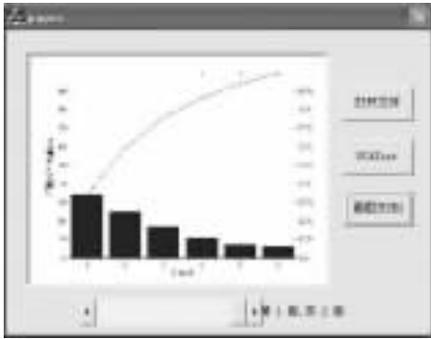


图2 方法2的运行结果

一生成图像时的闪烁现象,但是 BMP 图像显示效果较差而且无法使用 Matlab 提供的工具菜单,这是其不足之处。

使用 COM 组件进行混合编程时,往往习惯于在 VC 平台下思考所遇到的问题,但是这样不仅使得问题可能变得复杂化,或得不到妥善解决,而且也违背了“混合”的初衷。二维数组参数处理就是一个很好的例证。其次,充分利用 Matlab 特性可以使得程序具备良好的扩展性和稳定性,对 Excel 文件读取方式进行的扩展,明显使软

件更加人性化。对于图像嵌入问题,虽然文中提出的两种嵌入方式可以满足基本需求,但是仍然存在一些瑕疵,还需要进一步研究。

#### 参考文献

- [1] MCKINNEY B. Article 5. The Safe OLE Way of Handling Arrays. MSDN, 2001.
- [2] 刘维.精通 Matlab 与 C++ 混合程序设计(第2版)[M].北京:北京航空航天大学出版社,2008.
- [3] 董维国.深入浅出 Matlab 7.X 混合编程[M].北京:机械工业出版社,2006.

(收稿日期:2011-09-19)

#### 作者简介:

王维,男,1985年生,在读硕士研究生,主要研究方向:科学工程计算。

刘斌,男,1987年生,在读硕士研究生,主要研究方向:科学工程计算。

柳炳利,男,1981年生,在读博士生,主要研究方向:数学地质。