

MATLAB工程应用丛书

4

基于 MATLAB 6.X

MATLAB 6.X

图形编程与图像处理



陈杨 陈荣娟 郭颖辉 等编著



西安电子科技大学出版社

<http://www.xduph.com>

MATLAB 工程应用丛书

MATLAB 6.X

图形编程与图像处理

陈杨 陈荣娟 郭颖辉 等编著

西安电子科技大学出版社

2002

内 容 简 介

随着计算机技术的迅猛发展,图像和图形技术不断融合,产生了各种图像处理、CAD 软件。这些软件被广泛应用于计算机科学、工程学、统计学、物理学、信息科学、化学、生物学、医学乃至社会科学等领域,取得了令人瞩目的成就。本书从解决一般工程和科研问题的基础知识出发,主要介绍使用 MATLAB 进行图形编程和图像处理技术。

本书立足于基础的 MATLAB 图形编程和图像处理技术讲解,通过穿插简单的理论介绍以及大量丰富的演示实例,帮助读者快速、清晰地掌握相关的内容,为读者更深入了解和研究相关的技术提供了很好的前提。

本书理论讲解简单清晰,实例讲解重点突出,可以作为高校数字图像处理、计算机图形学等教学参考用书,也可以作为使用 MATLAB 进行图形编程及图像处理的开发人员的指导书籍。

图书在版编目(CIP)数据

MATLAB 6.X 图形编程与图像处理/陈杨等编著. —西安:西安电子科技大学出版社,2002.10

(MATLAB 工程应用丛书)

ISBN 7-5606-1179-6

I. M... II. 陈... III. 图像处理-计算机辅助计算-软件包, MATLAB 6.X IV. TP391.41

中国版本图书馆 CIP 数据核字(2002)第 072886 号

策 划 毛红兵

责任编辑 夏太平 王素娟

出版发行 西安电子科技大学出版社(西安市太白南路 2 号)

电 话 (029)8227828 邮 编 710071

http://www.xduph.com E-mail: xdupfxb@pub.xaonline.com

经 销 新华书店

印 刷 西安文化彩印厂

版 次 2002 年 10 月第 1 版 2002 年 10 月第 1 次印刷

开 本 787 毫米×1092 毫米 1/16 印张 14.25

字 数 333 千字

印 数 1~4 000 册

定 价 19.00 元

ISBN 7-5606-1179-6/TP·0611

XDUP 1450001-1

如有印装问题可调换

本书封面贴有西安电子科技大学出版社的激光防伪标志,无标志者不得销售。

前 言

视觉是人类感知外部世界的最主要和最直接的途径。随着科学技术的发展,视觉信息的表达和处理越来越被人们所重视。图像和图形技术就是在研究人类视觉信息的基础上发展起来的。本质上,图像和图形是用以表示视觉信息的两种不同方式,图像采用最直接朴素的方式记录信息,而图形则是人们在现有知识上对视觉信息的一种抽象的表示方法。

一般来说,图像是最直接的视觉信息。它包含着最原始的巨大信息,人类对于这类信息的感受是本能的。对这些原始信息的记录、再现、修改、传播以及理解就是图像处理过程,而图形技术则是在抽象后的视觉信息的基础上进行的记录、再现、修改、传播以及理解过程。

随着计算机技术的迅猛发展,图像和图形技术不断融合,产生了各种图像处理、CAD 软件。这些软件被广泛应用于计算机科学、工程学、统计学、物理学、信息科学、化学、生物学、医学乃至社会科学等领域,取得了令人瞩目的成就。从解决一般工程和科研问题的基础知识出发,本书着重于介绍最基础的图像处理技术,以及简单的图形显示技术,而对于图形技术中的最前沿 CAD 技术并不涉及。

MATLAB 是 MathWork 公司于 1984 年推出的用于数值计算的软件。自推出之后,MathWork 公司不断接收和汲取各学科领域权威人士为之编写的函数和程序,并将它们转换为 MATLAB 的工具箱。这样,MATLAB 得以不断发展和扩充。2000 年 10 月,MathWork 公司推出 MATLAB 6.0,2001 年 6 月推出 MATLAB 6.1。MATLAB 6.X 的推出是一次飞跃,它的可视化界面焕然一新,风格比以前更平易近人,而且还添加了对 JAVA 的支持,函数库也得到了进一步扩充。每一次新版本的推出都使 MATLAB 有了长足的进步,界面越来越友好,内容越来越丰富,功能越来越强大。现在世界上越来越多的人在使用 MATLAB,工程和研究人員已经离不开 MATLAB,国外许多著名大学的理工科教材也基于 MATLAB 软件编写。使用 MATLAB 语言编写的程序结构简单,容易理解,能够清晰地表达程序的思路和意图,因此越来越多的专业技术书籍都采用 MATLAB 语言进行讲解。另外,MATLAB 软件具有强大的图形功能以及丰富的图像处理工具函数,MATLAB 语言使得工程人员从复杂繁琐的算法以及细节编程中解放出来,从而达到快速实现设计思路的目的。

本书主要介绍 MATLAB 图形编程和图像处理技术。全书由图形编程基础篇、图像处理技术篇、综合应用篇三部分组成。图形编程基础篇侧重于介绍一些 MATLAB 图形编程中的基本概念,比如图形句柄、图像的表达等等。图像处理技术篇重点介绍 MATLAB 中的基本图像处理技术和技巧,比如图像变换、设计 FIR 滤波器、图像增强、二叉树分解、边缘检测、二值图像的操作等,涵盖了在工程实践中经常遇到的一些相关的图像处理问题。本篇适当穿插了相关理论的简单介绍,以使读者对这些技术的使用有较为全面的认识。综合应用篇介绍了综合图形图像技术应用中有关用户界面设计的相关内容,并给出了一个较为完整的 MATLAB 图形图像技术应用的实例。本书内容涉及到了 MATLAB 在图形编程和图像

处理中的大部分基础技术，对需要了解相关技术的读者是很好的指导教材。

本书由陈杨、陈荣娟、郭颖辉、刘凯、李军等编著；文字由李明、张东、刘涛、李凯、王华等录入；图像由李燕、胡利明、曾飞、刘丽、谢涛等编辑处理；书稿由陈杨、林哲辉审校。毛红兵女士为本书的策划与编辑工作付出了大量的心血与汗水。另外还有很多同志在本书的排版、校对过程中付出了大量的劳动，在此一并表示衷心的感谢！

由于时间仓促以及水平有限，书中错误之处在所难免，敬请各位专家和广大读者批评指正，本人不胜感谢！欢迎通过 E-mail:zhuri2032@sina.com 同作者联系和交流。

编 者
2002 年 7 月

目 录

图形编程基础篇

第一章 MATLAB的句柄图形对象 1	2.2 条形图的绘制 26
1.1 MATLAB的图形对象..... 1	2.2.1 二维垂直的条形图..... 26
1.1.1 Root对象..... 2	2.2.2 三维垂直的条形图..... 28
1.1.2 Figure对象..... 2	2.2.3 二维水平的条形图..... 29
1.1.3 Uicontrol对象..... 4	2.2.4 三维水平的条形图..... 30
1.1.4 Uimenu对象..... 6	2.3 柱状图的绘制 30
1.1.5 Axes对象..... 8	2.3.1 迪卡尔坐标系中的柱状图..... 30
1.1.6 Image对象..... 9	2.3.2 极坐标系中的柱状图..... 31
1.1.7 Light对象..... 9	2.4 区域图的绘制 32
1.1.8 Line对象..... 11	2.5 饼图的绘制 33
1.1.9 Patch对象..... 11	2.5.1 二维饼图的绘制..... 33
1.1.10 Rectangle对象..... 12	2.5.2 三维饼图的绘制..... 35
1.1.11 Surface对象..... 13	2.6 离散数据的图形绘制 35
1.2 图形对象的属性..... 14	2.6.1 二维枝干图..... 35
1.3 图形对象属性值的设置和查询..... 15	2.6.2 三维枝干图..... 36
1.3.1 属性值的设置..... 15	2.6.3 阶梯图..... 38
1.3.2 对象的默认属性值..... 17	2.7 方向和速度矢量图的绘制 38
1.3.3 属性值的查询..... 18	2.7.1 罗盘图..... 38
1.4 图形对象句柄的访问..... 20	2.7.2 羽状图..... 39
1.4.1 图形对象句柄的取值..... 20	2.7.3 二维箭头图..... 40
1.4.2 句柄图形的当前性..... 21	2.7.4 三维箭头图..... 41
1.4.3 通过属性值查找对象..... 21	2.8 轮廓图的绘制 42
1.4.4 图形对象的拷贝..... 23	2.8.1 二维轮廓图..... 42
1.4.5 图形对象的删除..... 24	2.8.2 三维轮廓图..... 43
第二章 MATLAB的图形绘制 24	2.9 动画的绘制 44
2.1 基本曲线的绘制..... 24	2.9.1 电影动画..... 44
2.1.1 plot函数..... 25	2.9.2 程序动画..... 45
2.1.2 plot3函数..... 25	第三章 MATLAB中的图像 48
2.1.3 loglog函数、semilogx函数 和semilogy函数..... 25	3.1 MATLAB中的图像文件格式..... 48
2.1.4 plotyy函数..... 25	3.2 图像类型..... 48
	3.2.1 索引图像..... 49

3.2.2	灰度图像	50	4.1.2	显示灰度图像.....	66
3.2.3	RGB图像	50	4.1.3	显示二值图像.....	66
3.2.4	二值图像	51	4.1.4	显示索引图像.....	67
3.2.5	图像序列	51	4.1.5	显示真彩图像.....	67
3.3	图像类型转换	52	4.1.6	显示图形文件中的图像	67
3.3.1	dither函数	52	4.2	特殊图像显示技术.....	68
3.3.2	gray2ind函数	53	4.2.1	添加颜色条.....	68
3.3.3	grayslice函数	54	4.2.2	显示多帧图像阵列.....	69
3.3.4	im2bw函数	54	4.2.3	图像上的区域缩放.....	71
3.3.5	ind2gray函数	55	4.2.4	纹理映射.....	72
3.3.6	ind2rgb函数	55	4.2.5	在一个图形窗口中显示多幅图像	73
3.3.7	mat2gray函数	56	4.3	MATLAB中的颜色模型	74
3.3.8	rgb2gray函数	56	4.3.1	颜色模型的分类.....	74
3.3.9	rgb2ind函数	57	4.3.2	颜色模型的转换.....	76
3.4	MATLAB中的8位和16位图像	57	第五章 图像的几何操作及基于		
3.4.1	8位和16位索引图像	57	区域的处理	80	
3.4.2	8位和16位灰度图像	58	5.1	图像插值的基本原理.....	80
3.4.3	8位和16位RGB图像	58	5.1.1	最近邻插值.....	81
3.5	图像文件的读写和查询	59	5.1.2	双线性插值.....	81
3.5.1	图像文件信息的查询	59	5.1.3	双三次插值.....	82
3.5.2	图像文件的读取	60	5.2	图像的插值缩放.....	82
3.5.3	图像文件的保存	61	5.3	图像的插值旋转.....	84
3.6	图像对象及其属性	62	5.4	图像的剪切.....	85
3.6.1	图像对象的CData属性	62	5.5	基于区域的图像处理.....	86
3.6.2	图像对象的CDataMapping属性.....	62	5.5.1	多边形选择法.....	86
3.6.3	图像对象的XData和YData属性.....	63	5.5.2	灰度选择法.....	87
第四章 MATLAB中的图像显示技术	65		5.5.3	其它选择方法.....	88
4.1	标准的图像显示技术	65	5.5.4	对指定区域的滤波.....	88
4.1.1	imshow函数	65	5.5.5	对指定区域的填充.....	89

图像处理技术篇

第六章 图像变换	91	6.3 Radon变换.....	100
6.1 傅立叶变换.....	91	第七章 FIR滤波器设计	103
6.1.1 二维连续傅立叶变换.....	91	7.1 FIR滤波器设计基础.....	103
6.1.2 二维离散傅立叶变换(DFT).....	95	7.1.1 freqz2函数.....	103
6.1.3 快速傅立叶变换(FFT).....	96	7.1.2 freqspace函数.....	104
6.1.4 傅立叶变换的应用.....	96	7.2 窗口方法.....	106
6.2 离散余弦变换.....	98	7.2.1 fwind1函数.....	107

7.2.2 fwind2函数	111	9.2 边缘检测	157
7.3 频率采样法	115	9.2.1 边缘检测的基本原理及处理函数 ...	157
7.4 二维FIR滤波器设计的频率变换法	117	9.2.2 各种边缘检测算子的效果比较	161
第八章 图像增强	121	第十章 二值图像操作	162
8.1 空域变换增强	121	10.1 二值形态学基本运算	163
8.1.1 直接灰度调整	122	10.1.1 膨胀	163
8.1.2 直方图处理	129	10.1.2 腐蚀	167
8.1.3 图像间的代数运算	134	10.1.3 膨胀与腐蚀的对偶性	169
8.2 空域滤波增强	137	10.1.4 开启和闭合	170
8.2.1 基本原理	137	10.2 二值形态学进行图像处理的	
8.2.2 平滑滤波器	138	综合应用	172
8.2.3 锐化滤波器	143	10.2.1 噪声滤除	172
8.3 频域增强	146	10.2.2 边界提取	173
8.3.1 低通滤波	146	10.2.3 对象标注	176
8.3.2 高通滤波	148	10.2.4 图像的特性度量	178
第九章 四叉树分解与边缘检测	150	10.2.5 细化与骨架提取	181
9.1 四叉树分解	150	10.2.6 查找表操作	182
9.1.1 四叉树分解的基本原理以及		10.2.7 区域填充	184
MATLAB工具箱函数	150	10.2.8 对象提取	186
9.1.2 四叉树分解应用	156		

综合应用篇

第十一章 综合应用实例	188	12.1.3 激活图形	198
11.1 对不均匀亮度的校正	188	12.2 GUI设计	198
11.2 基于特征的逻辑	190	12.2.1 指导原则	198
11.3 对钢纹(steel grain)的区域标识	193	12.2.2 动态界面的设计	203
第十二章 图形用户界面设计	195	12.2.3 开发流程	203
12.1 GUIDE开发环境介绍	196	12.3 GUI实现	204
12.1.1 控制面板添加按钮	197	12.3.1 GUIDE开发实例	205
12.1.2 使用回调函数编辑器		12.3.2 边缘检测实例	208
编写回调函数	198		

参考文献	219
-------------------	-----

图形编程基础篇

第一章 MATLAB 的句柄图形对象

句柄图形(Handle Graphics)是 MATLAB 中用于创建图形的面向对象的图形系统。图形句柄提供了多种用于创建线条、文本、网格和多边形等的绘图命令，以及交互式图形界面，如菜单、按钮和对话框等。

利用图形句柄，用户可以直接操纵线条、表面和其它图形对象。MATLAB 中的绝大多数图形函数和高级绘图命令正是利用这些图形元素来产生各种类型的图形的。利用图形句柄，我们可以在 MATLAB 的命令行中修改图形的显示效果，也可以在 M.文件中创建用户自定义的绘图函数。

本章主要内容：

- ★ MATLAB 的图形对象
- ★ 图形对象属性值的设置和查询
- ★ 图形对象的属性
- ★ 图形对象句柄的访问

1.1 MATLAB 的图形对象

图形句柄对象是 MATLAB 用来显示数据和创建图形用户界面(GUI)的基本绘图元素。对象的每个实例(instance)均对应于一个惟一的标识符，即句柄(handle)。利用该句柄，用户就可以非常容易地操纵现有图形对象的各种特征(即对象属性)。

MATLAB 中，对象的从属关系如图 1.1 所示。

从图 1.1 可以看出，所有 MATLAB 的图形对象都是彼此联系的。因此，典型的 MATLAB 图形显示均包含多种对象，而不是单一的。

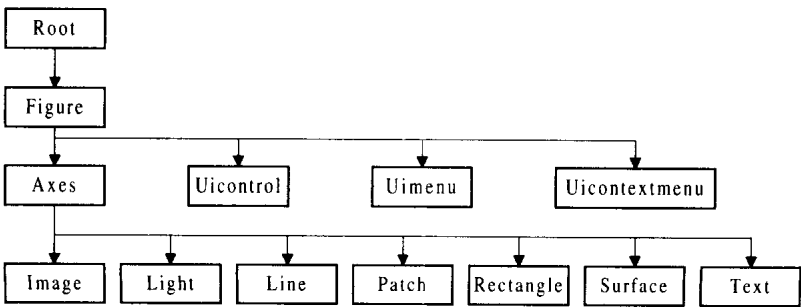


图 1.1 MATLAB 的对象从属关系图

MATLAB 遵循面向对象的原则, 每个图形对象都有一个相应的构造函数, 用于创建该对象类的实例。对象构造函数的名称与其构造的对象名相同。例如, `text` 函数创建 `Text` 对象(即文本对象)的实例; `figure` 函数创建 `Figure` 对象的实例等等。

下面的各小节将对图 1.1 中的各个对象进行详细的说明。

1.1.1 Root 对象

Root 对象即根对象。它位于 MATLAB 的对象继承图的最上层, 因此在 MATLAB 中创建图形对象时, 只能创建惟一的一个 Root 对象, 而其它的所有的对象都从属于该对象。

需要注意的是, 根对象是由系统在启动 MATLAB 时自动创建的, 而用户可以对根对象的属性进行设置, 从而影响图形的显示效果。

1.1.2 Figure 对象

Figure 对象即图形窗口对象, 是在根屏幕(即 Root 屏幕)上显示 MATLAB 图形的独立窗口。用户可以在系统允许的前提下, 在 MATLAB 中创建任意多个图形窗口。所有的图形窗口都是根对象的子对象, 而所有其它的图形对象又是图形窗口对象(即 Figure 对象)的子对象。

如果当前还没有创建图形对象(即 Figure 窗口), 则调用任意一个绘图函数或图像显示函数(如 `plot` 函数、`imshow` 函数等)均可以自动创建一个图形窗口。如果当前的根对象已经包含了多个图形窗口, 则总有一个窗口被指定为“当前”窗口, 且该窗口为所有绘图函数的输出窗口。

在 MATLAB 中, 用 `figure` 构造函数创建图形窗口的格式如下:

窗口句柄=`figure`('属性 1', 属性值 1, '属性 2', 属性值 2, ...)

`figure`(窗口句柄)

其中, 第一种格式既产生了一个窗口句柄, 同时又对窗口对象的各个属性进行了具体的设置, 且按照该设置生成了一个图形窗口。

例 1.1.1 创建图形窗口。

```
h=figure('color',[1 1 1],'position',[0 0 200 100],...
         'name','image processing','NumberTitle','on',...
         'menu','figure','WindowButtonDown',...
         'disp("welcome to this program)");
figure(h);
```

结果产生如图 1.2 所示的窗口。

另外, 对于第二种格式, 即 `figure`(窗口句柄)格式, 如果窗口句柄是存在的, 则将调用该窗口并使其处于可视状态且设定为“当前”窗口; 如果窗口句柄不存在, 则该函数将针对这一句柄生成一个新的窗口, 并设定为“当前”窗口。

例如, 执行下面的语句:

```
figure(h);
```

将以系统默认的设置生成一个图形窗口, 如图 1.3 所示。

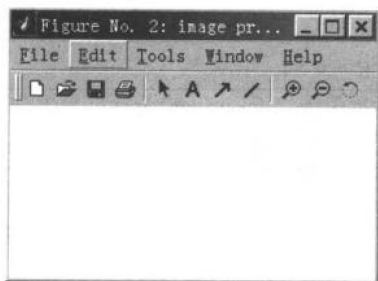


图 1.2 用户设置的图形窗口

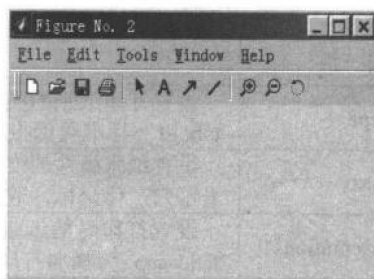


图 1.3 系统默认的图形窗口

另外，对图形窗口属性的设置还可调用 `set` 函数，其格式如下：

`set(窗口句柄, '属性 1', 属性值 1, '属性 2', 属性值 2, ...)`

要注意的是，在调用 `set` 函数之前，该窗口句柄应该是已经存在的。

图形窗口的各种常用属性的设置及其可设置的属性值如表 1.1 所示。

表 1.1 图形窗口的常用属性与属性值

属 性	说 明
Color	图形的背景颜色，可设置为三元素的 RGB 向量或者是 MATLAB 自定义的颜色名，MATLAB 的缺省值为黑色。RGB 向量分量的取值范围为[0 1]，取极端值可得到以下八种颜色：[0 0 0] 黑色，[1 1 1] 白色，[0 0 1] 蓝色，[0 1 0] 绿色，[0 1 1] 天蓝色，[1 0 0] 红色，[1 0 1] 粉红色，[1 1 0] 黄色
CurrentAxes	当前图形坐标轴的句柄
CurrentMenu	最近被选择的菜单项的句柄
CurrentObject	图形内最近被选择的对象的句柄。可由 <code>gco()</code> 函数获得
MenuBar	设置图形窗口菜单条形式，'figure'显示缺省的 MATLAB 菜单，'none'为不加菜单。在选择'none'后，可以用 <code>uimenu()</code> 函数设置自定义菜单；在选择'figure'后，还可以用 <code>uimenu()</code> 函数添加新菜单
Name	设置图形窗口的标题栏内容，它的属性为一个字符串，在创建窗口时，字符串将添入标题栏
NameTitle	设置图形窗口标题栏图形标号的有无，取'on'将自动地给图形窗口标题前加上 Figure No 字样；取'off'则将不再给标题前加标号
Position	位置向量[left, bottom, width, height]，(left, bottom)代表窗口左下角的坐标，width 和 height 分别表示窗口的宽度和高度
Units	设置尺寸的单位。'inches'，即 in；'centimeters'，即 cm；'normalized'为归一化坐标，取值范围为[0 1]；'point'为排字点，取值为 0.353 mm；pixel 为屏幕像素，这是 MATLAB 默认的单位
Resize	设置是否可以用鼠标调整窗口的大小，'on'为可调；'off'为不可调
Pointer	设置窗口下指示鼠标光标的显示形式，'crosshair'表示十字型；'arrow'表示箭头；'watch'表示沙漏；'topl'指向左上方的箭头；'topr'指向右上方的箭头；'bopl'指向左下方的箭头；'bopr'指向右下方的箭头；'circle'表示圆形光标；'cross'表示双线十字；'fleur'表示带有箭头的十字花形
Visible	设置窗口初始时刻是否可见，选项有'on'和'off'，其中 MATLAB 的缺省值为'on'，在编程的过程当中，不必看见中间的过程，可以首先使用'off'，在完成设置后，再选取'on'来显示窗口
PaperSize	设置打印纸张的大小。向量[width, height]代表了打印纸张的宽度和高度

续表

属 性	说 明
PaperType	设置打印纸张的类型。'usletter'为标准美国信纸；'uslegal'为标准美国法定纸张；'taboid'为标准美国报纸；'a3'为 A3 纸；'a5'为 A5 纸；'b4'为 B4 纸；'b5'为 B5 纸
PaperUnits	设置纸张属性的度量单位。'inches'，即 in；'centimeters'，即 cm；'normalized'表示归一化坐标，取值范围为[0 1]；'points'即像素点，取值为 0.353 mm
PaperOrientations	设置打印时的纸张方向。'portrait'为纵向打印，这是 MATLAB 的缺省值；'landscape'为横向打印
PaperPosition	设置打印页面上的图形位置，位置向量[left, bottom, width, height]，(left, bottom)代表打印页面图形左下角的坐标，width 和 height 分别表示打印页面图形的宽度和高度

1.1.3 Uicontrol 对象

Uicontrol 对象即控制框对象，是图形用户界面(GUI)控件，当用户激活该对象时，系统将执行相应的回调函数。Uicontrol 对象的控件有多种类型，包括按钮、无线电按钮、检查框、编辑框、静态文本框、列表框、滑标等。一个良好的用户界面是与这些控件分不开的。

表 1.2 给出了控制框的控件类型及相应的属性值。

表 1.2 控制框的控件类型

控 件	说 明
按钮 (button)	按钮是对话框中最常用的控制图标，按钮上通常有提示的文本，其通常表示执行一个动作，而不是用来改变对象的状态和属性。其'style'属性值为'pushbutton'
无线电按钮 (radio)	无线电按钮是一组带有文字提示的选择项，通常用于一组互斥的选项当中，每次只能选一项。当一项被选中后，被选中按钮的圆中心显示为黑色。其'style'属性值为'radiobutton'
检查框 (checkbox)	检查框通常是由具有标志的一个小方框组成的，其作用与无线电按钮极相似，主要区别是检查框可同时选择多个选项。其'style'属性值为'checkbox'
编辑框 (editbox)	编辑框是一个含有初始值的方框或无初值的空框。用户可以在里面输入自己的数据，相关的程序将会获取框中的数据信息。其'style'属性值为'checkbox'
静态文本框 (statictext)	静态文本框的主要作用是显示标志、用户提示信息和一些特定数据的当前值。其'style'属性值为'edit'
滑标 (slider)	滑标又称滚动条，它是由滚动槽、长方条区域、指示器三部分组成的。其中，长方条区域代表有效值范围，指示器代表滑标的当前值。其'style'属性值为'slider'
列表框 (listbox)	列表框用来列出可以选择的所有选项，用户可以方便地从中选择任意选项，如果选项太多，可采用垂直滚动条加以控制。其'style'属性值为'listbox'
弹出式菜单 (popupmenu)	弹出式菜单通常用以向用户提出一系列互斥选项的清单，以使用户选择，弹出式菜单不受菜单条的限制，可以出现在窗口的任何位置。其'style'属性为'popupmenu'

用户可以将 Uicontrol 对象控件组合使用，以构建控制面板和对话框。

控制框的创建用 uicontrol()函数。该函数用法如下：

控制框句柄=uicontrol(父对象的句柄，'属性 1'，属性值 1，'属性 2'，属性值 2，...)

表 1.3 给出了 Uicontrol 对象的一般属性。

表 1.3 Uicontrol 对象的一般属性

属 性	说 明
ForegroundColor	设置对象的前景颜色
BackgroundColor	设置对象的背景颜色
Callback	设置回调字符串, 当控制框被选中时, 执行响应的函数
HorizontalAlignment	设置当前控件在水平方向的对齐方式, 可选的值有'left'、'center'、'right', 分别代表左、中、右的对齐方式, 'right'为默认值。该属性对于单个控件不起作用, 只有在多个控件共存时才起作用
Max	设置属性'value'的最大有效值, 且该最大值取决于 Uicontrol 对象的'type'属性。当控制框对象无线电按钮和检查框处于'on'状态时, 将 value 设定为 max; 它设定了弹出式菜单最大下标值和滑标的最大下标值。当 $\max \cdot \min > 1$ 时, 编辑文本框为多行文本。其默认值为 1
Min	设置属性'value'的最小有效值, 且该最小值取决于 Uicontrol 对象的'type'属性。当控制框对象无线电按钮和检查框处于'off'状态时, 将 value 设定为 min; 它设定了弹出式菜单最小下标值和滑标的最小下标值。当 $\max \cdot \min > 1$ 时, 编辑文本框为多行文本。其默认值为 0
Position	设置控件的位置属性, 用位置向量[left bottom width height]来表示
String	设置控件上的字符串, 在按钮、无线电按钮、检查框和弹出式菜单上的显示标志
Style	设置控件类型, 细节请参照表 1.2
Type	设置可读对象的辨识串, 通常是用户指定的数据
Value	设置 uicontrol 的当前值。无线电按钮和检查框处于'on'状态时, 将 value 设定为 max; 无线电按钮和检查框处于'off'状态时, 将 value 设定为 min。滑标把 value 设置为 $\min \leq \text{value} \leq \max$, 弹出式菜单则把 value 设置为 $1 \leq \text{value} \leq \max$
Visible	设置对象的可视状态, 'on'为可视; 'off'为不可视

下面举一个创建控制框对象的简单例子, 更为详细的过程请参照本书的最后部分——用户界面(GUI)的设计。

例 1.1.2 创建控制框对象。

```
H=figure('Color',[0,1,1],'Position',[0,0,400,400],'Name',...
    '关于控制框设计的程序','NumberTitle','off',...
    'MenuBar','none');
uicontrol(H,'Style','text','Position',[0.05,0.74,0.9,0.1],...
    'Uints','normalized','String','WELCOME TO THIS...
PROGRAM!','Back',[0,1,1],'Fore',[1,0,0]);
uicontrol(H,'Style','frame','Position', ...
[0.15,0.35,0.5,0.40],'Uints','normalized','Back',[1,1,0]);
uicontrol(H,'Style','edit','Position', ...
```

```

[0.42,0.65,0.2,0.06],...
    'Uints','normalized','String','256','Back',[0,1,0]);
uicontrol(H,'Style','slide','Position', ...
[0.05,0.05,0.9,0.08],...
    'Uints','normalized','Back',[1,1,0],'max',1,'min',0);
radio=uicontrol(H,'Style','radio','Position', ...
[0.2,0.65,0.2,0.06],...
    'Uints','normalized','String','输入 N','Back',[1,1,0]);
checkbox(1)=uicontrol(H,'Style','checkbox','Position', ...
[0.2,0.54,0.2,0.06],...
    'Uints','normalized','String','N=100','Back',[1,1,0]);
checkbox(2)=uicontrol(H,'Style','checkbox','Position', ...
[0.2,0.43,0.2,0.06],...
    'Uints','normalized','String','N=200','Back',[1,1,0]);
change=[disp("改变窗口的颜色");
uicontrol(H,'Style','push','Position',[0.7,0.35,0.2,0.06],...
    'Uints','normalized','String','cancel','Call','close(H));
uicontrol(H,'Style','push','Position', ...
[0.05,0.17,0.85,0.06],...
    'Uints','normalized','String','CHANGE WINDOWS ...
COLOR','Call',change);

```

执行的结果如图 1.4 所示。

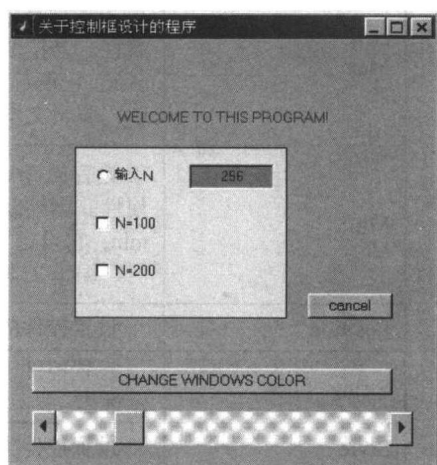


图 1.4 控制框的设计

1.1.4 Uimenu 对象

在一个方便简洁的用户图形界面设计中，菜单的设计是至关重要的。有了一个完备的菜单，用户就可以方便地从菜单中选择各条命令和各个选项。MATLAB 为图形用户界面提供了一个标准的菜单，其中包括四个下拉式菜单：File 菜单、Edit 菜单、Windows 菜单和 Help 菜单。

除了上面已经提到的标准菜单外，MATLAB 还提供了 `uimenu()` 构造函数来创建 Uimenu 对象(即下拉式菜单对象)，从而设计用户感兴趣的个性化菜单。

`uimenu()`函数的用法如下：

菜单项句柄=`uimenu`(窗口句柄, '属性 1', 属性值 1, '属性 2', 属性值 2, ...)

子菜单句柄=`uimenu`(菜单项句柄, '属性 1', 属性值 1, '属性 2', 属性值 2, ...)

菜单对象可以设置的属性如表 1.4 所示。

Uimenu 对象属性中，最重要的是 'Label' 和 'Callback' 属性。'Label' 属性决定了菜单的名称，'Callback' 属性决定了菜单项的响应，如果不设置 'Callback' 属性，菜单项将失去意义。

表 1.4 菜单对象的常用属性

属 性	说 明
ForegroundColor	设置菜单的前景色, 可以设置为 RGB 向量或 MATLAB 自定义的颜色名称
BackgroundColor	设置菜单的背景色, 可设置为 RGB 向量或 MATLAB 自定义的颜色名称, 默认值为黑色
Callback	设置 MATLAB 的回调字符串, 当该菜单被选中时, 回调字符串传给函数 eval(), 其初值为空矩阵
Checked	设置被选项的校验标记, 如果为'on', 则校验标记出现在所选菜单的旁边; 若为'off', 则不出现校验标记
Enable	设置菜单使能状态。若为'on', 则可用, 当选中该菜单时执行回调函数; 若为'off', 则菜单项不可用, 菜单标志变灰
Label	设置菜单条名称, 通常是一个字符串, 在菜单项字符中可使用'&'标志以表示该符号后面的字符在显示时有一个下划线修饰, 以便用户可以使用键盘 Alt+ 该字符来操作菜单
Position	设置菜单项的相对排列位置, 可以用来测试菜单项处于整个菜单系统的第几级
Separator	设置菜单项的分界符, 'on'为在该菜单项的上面加上一条横线作为分界符, 'off'为不加分界符, MATLAB 的默认值为'off'
Parent	父对象的句柄, 如果 uimenu 对象是顶层菜单, 则其为图形对象; 如果 uimenu 对象是子菜单, 则其为父菜单的句柄值
Visible	设置 uimenu 对象在屏幕上的可见形, 'on'为可见, 'off'为不可见

例 1.1.3 创建菜单对象。

```

H= figure('Color',[1 1 1],'position',[0,0,300,400],...
    'Name','signal processing','NumberTitle','off',...
    'MenuBar','none');
mfile=uimenu(H,'label','&File');
mtran=uimenu(H,'label','&Transforms');
mhelp=uimenu(H,'label','&Help');
uimenu(mfile,'label','&New','call',...
    'disp("new file selected)");
uimenu(mfile,'label','&Open','call',...
    'disp("open file selected)");
msave=uimenu(mfile,'label','Save','Enable','on');
uimenu(msave,'label','Mfile','call','savem.m');
uimenu(msave,'label','Matfile','call','savemat.m');
uimenu(mfile,'label','Save & As','Enable','off');
uimenu(mfile,'label','&print','separator','on', ...
    'call','print.dwin.loose.f');
uimenu(mfile,'label','&Exit','separator','on', ...
    'call','close(H)');
uimenu(mtran,'label','&Czt','call','help czt');

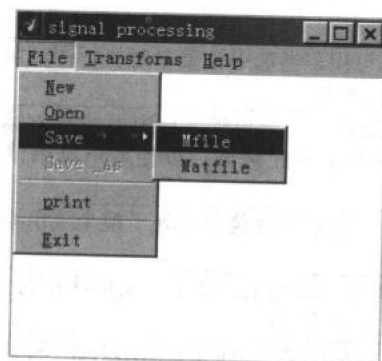
```

```

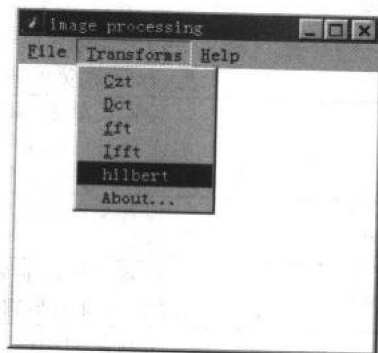
uimenu(mtran,'label','&Dct','call','help dct');
uimenu(mtran,'label','&fft','call','help fft');
uimenu(mtran,'label','&Ifft','call','help ifft');
uimenu(mtran,'label','hilbert','call','help hilbert');
uimenu(mtran,'label','About...','call',...
      'disp("This is a menu.making example")');

```

代码执行的结果如图 1.5 所示。



(1)



(2)

图 1.5 用 uimenu 函数创建的菜单

这只是一个简单的创建菜单的例子，关于如何创建一个功能更加完善的菜单，请参照本书的最后一部分——用户界面(即 GUI)的设计。

1.1.5 Axes 对象

Axes 对象即坐标轴对象。它在图形窗口中定义一个区域，并将其所有子对象都限制在该区域内。坐标轴对象是图形窗口对象(Figure 对象)的子对象，同时还是 Image 对象(图像对象)、Light 对象(灯光对象)、Line 对象(线条对象)、Patch 对象(片块对象)、Surface 对象(表面对象)和 Text 对象(文本对象)的父对象。

如果当前运行的 MATLAB 输出窗口中还没有坐标轴对象，则任意一个绘图函数(如 plot 函数、surf 函数等)都可以创建坐标轴对象。如果在当前图形窗口中包含多个坐标轴对象，则总有一个坐标轴对象被指定为“当前”坐标轴，并且，该“当前”坐标轴即为所有绘图函数输出的坐标基准。

Axes 对象是由 axes()构造函数创建的。

axes 函数的调用格式如下：

坐标轴句柄=axes('position', rect)

其中，rect=[left, bottom, width, height]定义了坐标系的位置和尺寸。单位归一化后，(0, 0)即为当前图形窗口中左下角点的坐标，而(1.0, 1.0)则为当前图形窗口中右上角点的坐标。

例如，执行下面的代码：

```
axes('position',[0.1,0.2,0.5,0.5]);
```

将在(0.1, 0.2)处，创建一个宽度和高度均为 0.5 个单位的坐标系，结果如图 1.6 所示。

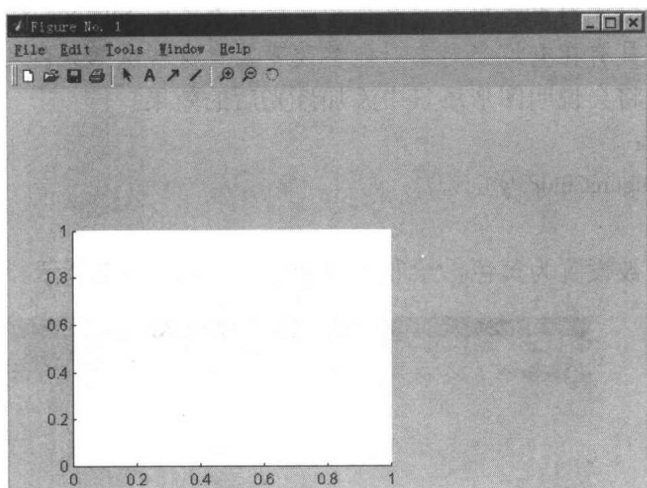


图 1.6 用 `axes` 构造函数创建的坐标系

另外，在 MATLAB 中，与坐标轴对象相关的函数还有如下几个：

(1) `axis`——该命令用于设置影响当前坐标轴对象的相关值。如执行下面的命令：

`axis([xmin, xmax, ymin, ymax])`

可以设置当前坐标轴的坐标值范围。

(2) `get`——该命令用于获取现有的坐标轴的各项属性值。

(3) `set`——该命令用于设置现有的坐标轴的各项属性值。

(4) `gca`——该命令用于返回当前坐标轴的句柄(即标识号 ID)。

1.1.6 Image 对象

Image 对象即图像对象。它包含一个数据矩阵，有时还包含一个颜色映像表。根据数据矩阵元素代表意义的不同，MATLAB 中的 **Image** 对象有三种基本的类型，即索引图像、灰度图像和真彩图像。

Image 对象是由构造函数 `image()` 来创建的，其具体用法可参见第三章。

1.1.7 Light 对象

灯光设置是 MATLAB 中为增加图形场景的视觉效果而提供的一种处理技术。该项技术是通过模拟对象实体在自然光下的明暗特征来实现的。在 MATLAB 中，如果希望产生灯光效果，必须创建一个“灯光”对象。

MATLAB 中利用 `light()` 构造函数来创建灯光对象。每个灯光对象都包含三个非常重要的属性：

(1) **Color**——该属性用于指定来自于光源方向的灯光颜色。其可以影响到图形场景中对象的显示颜色。

(2) **Style**——该属性用于指定光源类型是点光源还是无限远光源。当将 **Style** 设置为 `local` 时，表示是点光源；设置为 `infinite` 时，表示是无限远光源。默认值为后者。

(3) **Position**——该属性用于指定光线方向(对于无限远光源)或光源位置(对于点光源)。