

文章编号: 1008-1542(2005)04-0309-03

利用 MATLAB 图像处理工具箱进行 图形数字化的研究

周广芬¹, 李 鹏², 杨久义¹

(1. 河北科技大学理学院, 河北石家庄 050018; 2. 军械工程学院火炮工程系, 河北石家庄 050003)

摘 要: 针对研究的利用 MATLAB 图像处理工具箱处理实验输出图形、获取图线坐标值、使图形数字化的过程, 通过对一构件振动位移测试结果图形的处理, 证明此方法取点密集、返回坐标值精确度高, 且整个过程可以通过 MATLAB 语言自动完成, 比人工取点的处理方法省时、省力, 数据误差小。

关键词: 图像处理; 数字化; MATLAB

中图分类号: TP391.41 文献标识码: A

Study of using the image processing toolbox of MATLAB to digitize graph

ZHOU Guang-fen¹, LI Peng², YANG Jiur-yi¹

(1. College of Sciences, Hebei University of Science and Technology, Shijiazhuang Hebei 050018, China; 2. Department of Guns Engineering, Ordnance Engineering College, Shijiazhuang Hebei 050003, China)

Abstract: The paper studies the process using the image processing toolbox of MATLAB to digitize graph and get coordinates of the curve. Through applying it to an output graph of component vibration displacement experiment, it is proved that this method could get serried and precise coordinates of points on curve. The whole process can be finished automatically by using the MATLAB language. Compared with the manual method, this method is facile and makes little error of experiment data.

Key words: image processing; digitizing; MATLAB

在实验室进行的各种测试中, 许多仪器输出的结果仅仅是一条曲线, 而没有这条曲线上点的坐标数据。例如: 使用测振仪测量构件振动位移的测试结果就是一条绘制在坐标纸上的曲线。在随后的数据处理过程中, 通常的做法是手工标出曲线上的若干点, 然后根据坐标纸读出这些点的坐标值, 作为实验得到的初始数据, 以此为基础进行计算和分析。这种处理方法不仅繁琐、工作量大, 而且在曲线上取点的多少和坐标值的判读都因人而异, 给测试结果带来了较大的误差。

作为一种应用广泛的编程工具, MATLAB 在图形处理方面有着明显的优势: 具有强大的矩阵运算功能, 在进行图像处理时可以避免许多繁琐的运算; 图形显示方便, 可以在调试过程中随时观察图形的变化; 有丰富的图像处理函数库, 其图像处理工具箱(image processing toolbox)几乎涵盖了所有常用的图像处理函数^[1]。笔者利用 MATLAB 的图像处理工具箱, 对实验输出的图形进行处理, 获得实验图线的二进制图像矩阵, 进而通过对二进制矩阵的分析获得实验图线的坐标值。实践证明, 该过程简单实用, 获取坐标

收稿日期: 2004-10-10; 修回日期: 2004-12-16; 责任编辑: 王士敏

作者简介: 周广芬(1976), 女, 山东陵县人, 讲师, 硕士, 主要从事物理化学和计算机应用方面的研究。

值精度高,是一种解决上述问题的有效方法。

1 原始输出图形的处理

1.1 图形的输入

实验中输出的图形,一般可通过扫描仪扫入计算机,使其成为可进一步编辑的计算机图像,见图1。

在 MATLAB 中,通过 `imread` 函数可以将图形读入内存,返回一个大小与原图形大小(图像的大小以像素计)相同的矩阵,输入语句:

```
ori= imread(' ori_pic.tif')
```

通常情况下,获得的原始图像是一个 RGB 图像,由于涉及的图像处理函数多针对灰度图像,因此需要将 RGB 图像 `ori` 转换为灰度图像 `bw`,输入语句:

```
bw= rgb2gray( ori)
```

灰度图像矩阵中每个元素代表一个像素点,如果矩阵为 `uint8` 类型,其数据范围为 $[0, 255]$,不同的元素数值代表不同的亮度或灰度级,数值 0 代表黑色,数值 255 代表白色,数值越大,表明该位置像素点亮度越高^[2]。

通过 `imshow` 函数,可以将矩阵代表的图形显示在图形窗口上,可以看到,显示出来的图形与原图形是完全一样的。

1.2 图形的处理

为了消除图形中无关像素的干扰,需要除去图形的背景,仅仅保留输出结果图线。将图形转换为二进制图形,再施以相应的变换,就可以达到这个目标,满足数字化的需要。

笔者首先利用 MATLAB 中双阈值选择的方法,有效地清除图形中的次要区域。用小波分析的方法处理扫描图形,根据图形扫描强度和明暗的变化,可以确定 2 个阈值^[2],用这 2 个阈值对原图形矩阵进行筛选,如阈值 `threshold_1= 100`, `threshold_2= 210`,输入语句:

```
bw1= bw > threshold_1
```

```
bw2= bw > threshold_2
```

筛选后返回的 `bw1` 和 `bw2` 矩阵是 2 个二进制矩阵,与原图形矩阵大小相同,其中的元素仅有 0 和 1 组成。数字 0 代表原图形矩阵该元素所在位置像素小于等于设定的阈值(即“否”),而数字 1 代表大于设定的阈值(即“是”)。与灰度图像相同,在显示为图形时,数值 0 代表黑色,数值 1 代表白色^[3],因此二进制图像可以看作是一种特殊的灰度图像,见图 2。

由于阈值设定的不同,`bw1` 中黑色像素点的数量远远少于 `bw2` 中的数量,人们可以选择合适的阈值,使 `bw1` 中包含的全是结果曲线上的像素点,再以 `bw1` 为基础对 `bw2` 进行选择,可以很好地清除图形中实验曲线之外的背景像素,如图形记录纸上的网格线等均可被清除。

首先确定 `bw1` 中像素点的位置,输入语句:

```
[ r, c ] = find( bw1 == 0)
```

`r` 和 `c` 是 2 个数组,分别代表 `bw1` 矩阵中 0 元素所在的行值和列值,也就是黑像素点所在位置。

对 `bw2` 进行选择,输入语句:

```
bw_clean= bwselect( bw2, c, r, 8)
```

函数 `bwselect` 的作用是寻找 `bw2` 中与指定位置(`c, r`)相重叠的物体。需要指出的是, MATLAB 二进制图像中的“物体”指的是白色像素点组成的区域,因此函数 `bwselect` 中的选择源应该是 `bw2` 图形的“负片”。

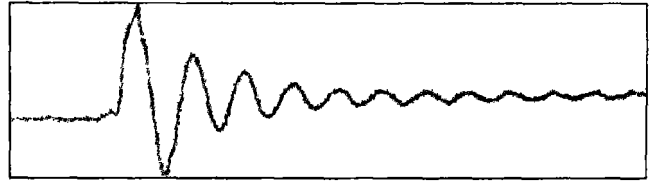


图 1 原始实验图形

Fig. 1 Original image

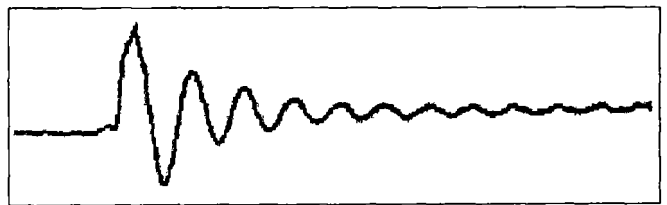


图 2 利用较大阈值 `threshold_2` 筛选后的图形

Fig. 2 Return image of using bigger `threshold_2` to choose

即矩阵中的元素 0 和 1 互换。如前所述, (c, r) 指示的都是曲线上点的位置, 因此返回的 bw_clean 就是原图形中的结果曲线, 见图 3。

通过 imshow (bw_clean) 可以看到, 获得的 bw_clean 曲线还并不完善, 曲线较粗且不光滑, 需要对它作进一步的处理, 才能满足数字化的需要。

MATLAB 提供了一个强大的二进制图像处理函数 bwmorph, 其用法为 bwmorph(pic, operation, n), 通过指定“operation”为不同的参数, 可以完成对二进制图像不同的处理。

这里首先使用“skel”参数, 将曲线进一步细化, 使像素点的位置更加精确, 见图 4。

从图 4 可以看出返回的 bw_skel 曲线已经很细, 只是仍不够光滑, 一些微小的“毛刺”需要进一步去除。

再次使用 bwmorph 函数, 选用“spur”参数, 将 bw_skel 曲线上的“毛刺”去除。观察返回的图形, 曲线已经非常精确, 图形中不包含无关的像素点, 见图 5。

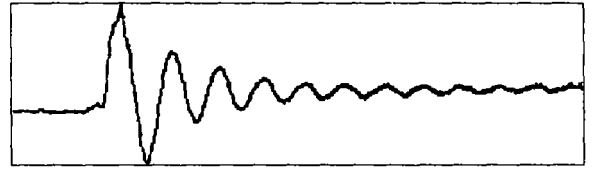


图 3 矩阵 bw_clean 所代表的图形

Fig.3 Image of matrix bw_clean

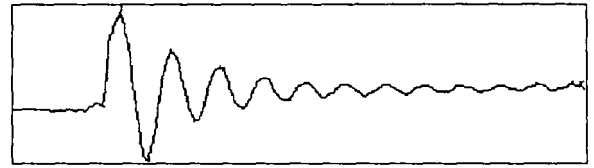


图 4 矩阵 bw_skel 所代表的图形

Fig.4 Image of matrix bw_skel

2 图形的数字化

2.1 利用二进制矩阵将图形数字化的原理

经过图像处理, 最终得到的 bw_pruned 矩阵为二进制格式, 矩阵中元素为 0 和 1, 原图片中每一个像素点对应矩阵中的一个元素。由于处理后的图形已经没有任何无关像素点, 因此矩阵中 0 (显示为黑色像素) 所在位置就是原图片中曲线所在位置, 而图片的空白处在矩阵中则全部显示为 1。bw_pruned 矩阵的大小与原图片的大小总是相同的, 所以矩阵中某一个 0 所在相对位置和其对应的曲线上像素点在原图片中的相对位置也是相同的^[4]。确定了矩阵中元素 0 的位置, 就相当于确定了曲线在原图片中的位置。将确定的相对位置在图片的真实坐标中定位, 即可知道曲线上每个点的坐标值。例如: 原曲线上一点 a, 其对应的数值 0 出现在二进制矩阵的 i 行 j 列, 原图形的大小 (即矩阵大小) 为 m × n, 而原图形对应的真实坐标尺度为 x × y, 则可求出 a 点在真实坐标中的位置为 $x \cdot (j - 1) / (n - 1)$, $y \cdot (i - 1) / (m - 1)$, 再根据真实坐标的起点进行坐标平移, 即可得到 a 点的坐标值。

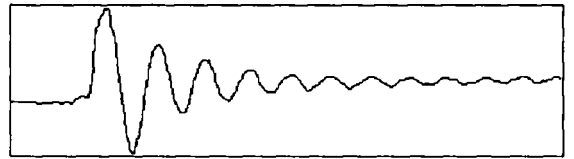


图 5 最终得到的实验结果图形

Fig.5 Final image

2.2 在 MATLAB 中计算坐标值的过程

在 MATLAB 中, 判断一特定元素在矩阵中的位置是非常简单的, 完全不必像其他语言一样使用循环嵌套判断, 只需使用一个 find 命令即可, 输入语句:

```
[ tempy, tempx ] = find ( bw_pruned == 0 )
```

返回的 tempy 和 tempx 是 2 个数组, 分别记录了矩阵中每个 0 元素所在的行与列, 由于矩阵的行数计算是自上而下的, 而实际的纵坐标通常是自下而上的, 输入下列语句将 tempy 转换为相对于矩阵最后一行的相对值, 可表示曲线的纵向相对位置:

```
tempy = matix_height - tempy
```

其中 matix_height 为矩阵的总行数。

将获得的横向位置 tempx 和纵向位置 tempy 与图片的真实坐标比较, 进行相应的缩放和平移, 即可获得曲线上每点的坐标值, 见表 1。

(下转第 321 页)

毒和脆弱性评估等技术的优点与自动阻止攻击的功能融为了一体。尽管在技术上仍有很多未克服的问题,但正如攻击技术不断发展一样,IPS 也会不断更新、成熟。

参考文献:

[1] DESAI, NEIL. Intrusion Prevention Systems: the Next Step in the Evolution of IDS[EB/OL]. <http://securityfocus.com/printable/infocus/1670>, 2003-02-27.

[2] TED H. Understanding IPS and IDS: Using IPS and IDS Together for Defense in Depth[EB/OL]. <http://www.sans.org>, 2004-02-23.

[3] 宋献涛, 纪 勇. 网络防护: 从 IDS 到 IPS[J]. 计算机安全, 2003, (11): 26-28.

[4] 崔晓垣. 网络入侵防御系统研究[J]. 信息安全与通信保密, 2005, (3): 109-111.

[5] CASWELL B, BEALE J. Snort 2.0 Intrusion Detection[M]. 宋劲松译. 北京: 国防工业出版社, 2004. 367-386.

[6] 卿 昊, 原宏春. 入侵防御系统(IPS)的技术研究及其实现[J]. 通信技术, 2003, (6): 101-103.

(上接第 311 页)

表 1 获取的曲线上部分坐标值

Tab.1 Part of coordinates of the curve

横坐标(时间/s)	1.094	1.797	2.734	4.492	5.898	6.367	...
纵坐标(位移/mm)	0.168	0.170	0.173	0.175	0.178	0.180	...

比较图 6 与原始曲线图 1 可以看出,通过获得的坐标值重新绘制的图线与原扫描图线符合得非常好,应用坐标值计算的结果也准确地反映了实验结果。

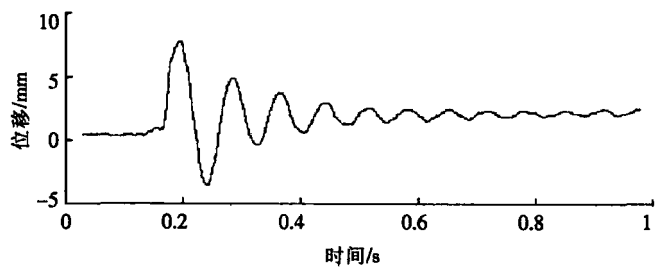


图 6 通过获取的坐标值重绘的图线

Fig. 6 Redrawing image of coordinates

3 结束语

通过上述图形数字化的过程可以看出,这种方法与计算机图像处理紧密结合,取点密集,返回坐标值精确度高,且整个过程可以通过 MATLAB 语言自动完成,比人工取点的方法省时,省力,实验数据处理误差小,是一种实用、快捷的实验图形数字化方法。

参考文献:

[1] 张宜华. 精通 MATLAB5[M]. 北京: 清华大学出版社, 1999.

[2] 陈桂明, 张明照, 戚红雨. 应用 MATLAB 语言处理数字信号与数字图像[M]. 北京: 科学出版社, 2000.

[3] HARALICK R M, SHAPIRO L G. Survey: image segmentation[J]. Computer Vision Graphics Processing, 1985, 129: 100-132.

[4] 潘 峰, 刘文予, 朱光喜. MATLAB 在图像处理与研究中的应用[J]. 计算机应用研究, 1999, 16(12), 73-75.