

# 自动微分在隐式曲线绘制中的应用

寿华好, 何 苹, 缪永伟

SHOU Hua-hao, HE Ping, MIAO Yong-wei

浙江工业大学 理学院, 杭州 310023

College of Science, Zhejiang University of Technology, Hangzhou 310023, China

SHOU Hua-hao, HE Ping, MIAO Yong-wei. Automatic differentiation and its application in implicit curve plotting. *Computer Engineering and Applications*, 2010, 46(1): 150-153.

**Abstract:** Automatic differentiation is a technique to evaluate the derivatives of a function defined by a computer program. This paper combines automatic differentiation technique with Taylor method and applies them into the subdivision based implicit curve plotting algorithm. A comparison of this new method with other two traditional methods is conducted which shows that the new method is more accurate.

**Key words:** automatic differentiation; implicit curve; Taylor method

**摘 要:** 自动微分是用于计算多变量函数的导数和偏导数的一种微分技术, 在给定一个多变量光滑函数值的程序代码后, 可以很容易地利用自动微分来实现有关导数和偏导数的精确计算。将自动微分技术与泰勒方法相结合应用到计算机图形学领域隐式函数曲线绘制的细分算法中, 并与未使用自动微分技术前的隐式曲线绘制方法作比较和分析, 展示了自动微分方法在绘制隐式曲线方面的优势。

**关键词:** 自动微分; 隐式曲线; 泰勒方法

DOI: 10.3778/j.issn.1002-8331.2010.01.046 文章编号: 1002-8331(2010)01-0150-04 文献标识码: A 中图分类号: TP391

## 1 引言

自动微分是一种计算技术, 理论上可以计算任意变量任意阶的导数和偏导数, 并非常容易用计算机编程实现。利用自动微分的易编程、高效率及高计算精度等特点来实现计算机图形学里的隐式曲线绘制, 既可以降低算法的时间复杂性, 又能提高图形绘制的质量。

虽然目前研究自动微分的文献已经非常丰富, 但是把自动微分应用在计算机图形学里的研究尚不多见, 尤其是应用在隐式曲线绘制细分算法中的研究尚未见报道。

绘制隐式曲线的细分算法中最关键一步是估计二元函数在矩形区域内的取值范围, 常用的方法是自然区间法、泰勒展开法等。多变量函数的泰勒表达式展开中需要对导数和偏导数进行计算, 这为自动微分方法与泰勒展开法相结合用于绘制隐式曲线提供了条件。在未使用自动微分以前常规泰勒展开法对二元函数泰勒展开式余项中偏导数取值范围的估计是靠手工完成的, 而现在结合自动微分方法就可以通过计算机程序代码高效、精确地估计出泰勒展开式余项中偏导数取值范围。该文将把应用自动微分的方法与自然区间法及手动的泰勒展开法对绘制曲线的效果进行比较和分析, 以揭示自动微分在隐式曲线绘制中的应用价值。

## 2 自动微分

自动微分方法(AD)<sup>[1-2]</sup>通过分析原程序对象的数据相关性, 是一系列基于链式求导法则的代码转换技术, 通过各种预编译手段, 可以把一个函数值程序代码转换成对应的计算微分的程序代码<sup>[3]</sup>。

自动微分实现的基本出发点是: 一个数据相对独立的程序对象(模式、过程、程序段、数值语句乃至数值表达式), 无论多么复杂, 总可以分解为一系列有限数目的基本函数(如 sin、exp、log)和基本运算操作(加、减、乘、除、乘方)的有序复合; 对所有这些基本函数及基本运算操作, 重复使用链式求导法则, 将得到的中间结果自上而下地做正向积分就可以建立起对应的切线性模式, 而自下而上地做反向积分就可以建立起对应的伴随模式<sup>[4]</sup>。这里提到的正向积分(forward accumulation)和反向积分(backward accumulation)是两种最基本的计算微分方法<sup>[4]</sup>, 正向积分沿着程序运行的自然顺序“自上而下地”计算函数的导数, 而反向积分沿着与程序运行相反的顺序“自下而上地”计算函数的导数<sup>[5]</sup>。

对于一个给定的函数, 首先将其分解为若干个基本操作的组合, 每个基本操作中至多有 2 个操作数, 然后再通过对这些基本操作进行求导, 从而得出目标函数的梯度。

**基金项目:** 教育部留学回国基金; 浙江省自然科学基金(the Natural Science Foundation of Zhejiang Province of China under Grant No.Y107311); 浙江省中青年学科带头人基金。

**作者简介:** 寿华好(1964-), 男, 博士, 教授, 主要研究领域为计算机辅助几何设计与图形学; 何苹(1976-), 女, 硕士生, 主要研究领域为计算机辅助几何设计与图形学; 缪永伟(1971-), 男, 博士, 主要研究领域为计算机辅助几何设计与图形学。

**收稿日期:** 2009-10-15 **修回日期:** 2009-11-16

如对于函数:

$$y = \cos x_1 x_2 + \sin x_1^2 \quad (1)$$

其梯度计算过程可表示如下,首先进行运算分解,令

$$x_1 = x_1, x_2 = x_2$$

$$x_3 = f_3 = x_1 x_2$$

$$x_4 = f_4 = \cos x_3$$

$$x_5 = f_5 = x_1^2 \quad (2)$$

$$x_6 = f_6 = \sin x_5$$

$$y = x_7 = f_7 = x_4 + x_6$$

根据式(2)所示的函数组合,  $y$  对于  $x_1$  的偏导数利用链式法则可得到:

$$\frac{\partial y}{\partial x_1} = \frac{\partial f_7}{\partial x_4} \frac{\partial f_4}{\partial x_3} \frac{\partial f_3}{\partial x_1} + \frac{\partial f_7}{\partial x_6} \frac{\partial f_6}{\partial x_5} \frac{\partial f_5}{\partial x_1} \quad (3)$$

式中  $\partial f_i / \partial x_j$  可利用简单求导法则得到。例如:

$$f_7 = x_4 + x_6, \partial f_7 / \partial x_4 = \partial f_7 / \partial x_6 = 1$$

$$\partial f_4 / \partial x_3 = -\sin x_3, \partial f_3 / \partial x_1 = x_2$$

$$\partial f_6 / \partial x_5 = \cos x_5, \partial f_5 / \partial x_1 = 2x_1$$

这样,利用简单求导规则和链式法则即可求得函数的导数。

以上讨论的是单值函数,计算的一阶导数是函数的梯度向量。对于多值函数,用同样的方法可计算它的 Jacobia 矩阵。在 AD 的实现过程中,通常是先记录目标函数  $y=f(x)$  的执行过程,再按照上述过程进行目标函数的梯度信息计算<sup>[6]</sup>。

所以自动微分可以实现任意变量任意阶导数和偏导数值的计算,使用切线性模式计算雅可比矩阵-向量乘积的计算精度较高;而使用伴随模式在计算函数的梯度时具有理想的计算代价<sup>[7]</sup>。当自变量个数较大时,自动微分所需的计算量比通常方法所需的计算量小很多,可以提高科学计算的效率。与符号微分、差分近似等微分方法相比,自动微分具有代码简练、计算精度高、投入人力少及适用范围广等优点,在科学计算、工程计算及其应用领域中有着广泛的应用。

### 3 隐式曲线绘制的细分算法

平面曲线既可以用参数式  $\{(x(t), y(t)): t \in \mathbf{R}\}$  表示,也可以用隐式  $\{(x, y): f(x, y) = 0\}$  表示。隐式曲线在几何造型特别在 CSG 和在以参数表示的形状的裁剪运算中非常有用。举例来说,它们能够表示三维空间中两张参数曲面的交,或者一张参数曲面对于某一给定视图的轮廓线<sup>[8]</sup>。

关于参数曲线的绘制已有不少非常有效的方法,然而隐式曲线的绘制却并不那么简单。历史上关于隐式曲线的绘制方法大致可以分为两类:一类是连续跟踪法,从曲线上某些起始点开始借助于一定规则连续跟踪绘制曲线。如 TN 法<sup>[9]</sup>,正负法<sup>[10]</sup>等。由于隐式曲线通常比较复杂,它可能包含互不相连的几个部分,而且可能包含奇点(如尖点,切点,自交点等),这类方法最根本的困难在于完整的起始点集的选取,而且需要对奇点进行特别的处理。另外一类是基于细分的方法<sup>[8,11-12]</sup>。通常先考虑整个绘制区域,通过估计函数  $f(x, y)$  在该区域的界,排除无关区域和将有关区域不断细分,进一步不断排除无关区域,直到有关区域到达一个像素的大小为止,如果还是排除不了则把这个像素画出来。该类方法的优点是能可靠地绘制出隐式曲线,不会丢失隐式曲线的任何部分,而且不需要对奇点进行特别的处理。其缺点是如果估计方法过于保守会导致“胖”曲线的出

现,原因是它有可能将不在曲线上的象素由于无法排除而画出来。

隐式曲线绘制的细分算法的基本思想是:先考虑整个绘制区域  $[x, \bar{x}] \times [y, \bar{y}]$ , 通过保守地估计函数  $f(x, y)$  在该区域的界  $[f, \bar{f}]$ , 如果  $0 \notin [f, \bar{f}]$ , 则说明曲线  $f(x, y) = 0$  不通过该区域, 可以将该区域排除, 否则将该区域在其中点处用处于水平方向和垂直方向的两条直线一分为四, 再逐个考虑所产生的四个小区域, 该细分过程可以一直进行下去, 直到所考虑的区域到达一个像素的大小为止, 如果还是排除不了则把这个像素画出来。该算法的程序如下:

PROCEDURE Quadtree ( $\underline{x}, \bar{x}, \underline{y}, \bar{y}$ ):

$[f, \bar{f}] = \text{Bound of } f \text{ on } (\underline{x}, \bar{x}, \underline{y}, \bar{y})$ ;

IF  $f \leq 0 \leq \bar{f}$  THEN

IF  $\bar{x} - \underline{x} \leq \text{Pixel size}$  AND

$\bar{y} - \underline{y} \leq \text{Pixel size}$  THEN

Plot Pixel ( $\underline{x}, \bar{x}, \underline{y}, \bar{y}$ ),

ELSE Subdivide ( $\underline{x}, \bar{x}, \underline{y}, \bar{y}$ ).

PROCEDURE Subdivide ( $\underline{x}, \bar{x}, \underline{y}, \bar{y}$ ):

$\hat{x} = (\underline{x} + \bar{x}) / 2$ ;

$\hat{y} = (\underline{y} + \bar{y}) / 2$ ;

Quadtree( $\underline{x}, \hat{x}, \underline{y}, \hat{y}$ );

Quadtree( $\underline{x}, \hat{x}, \hat{y}, \bar{y}$ );

Quadtree( $\hat{x}, \bar{x}, \underline{y}, \hat{y}$ );

Quadtree( $\hat{x}, \bar{x}, \hat{y}, \bar{y}$ ).

其中  $(\hat{x}, \hat{y})$  是区域  $[\underline{x}, \bar{x}] \times [\underline{y}, \bar{y}]$  的中点。

隐式曲线绘制细分算法最近有非常多的进展,具体可见文献[13]及其中参考文献。

### 4 自动微分结合到隐式曲线绘制细分算法中

隐式曲线绘制的细分算法中最关键的一步是估计函数  $f(x, y)$  在区域  $[\underline{x}, \bar{x}] \times [\underline{y}, \bar{y}]$  上的界  $[f, \bar{f}]$ , 不同的估计方法就绘制的效果和绘制的效率而言有差异。一般情况下估计越精确, 则绘制效果越好, 而要使估计更精确, 往往需要更大的计算量, 从而会降低算法的效率。显然, 效果和效率是一对矛盾。

估计二元函数  $f(x, y)$  在区域  $[\underline{x}, \bar{x}] \times [\underline{y}, \bar{y}]$  上的界  $[f, \bar{f}]$ , 通常的方法有如下几种:

(1) 自然区间法

即估计  $f(x, y)$  在  $[\underline{x}, \bar{x}] \times [\underline{y}, \bar{y}]$  中的取值范围的时候, 分别以区间  $[\underline{x}, \bar{x}]$  和  $[\underline{y}, \bar{y}]$  代替  $f(x, y)$  中的  $x$  和  $y$  作区间运算最终得到一个估值区间  $[f, \bar{f}]$  的方法。用这种方法得到的估值区间  $[f, \bar{f}]$  往往过于保守。

(2) 泰勒展开法

① 零阶泰勒展开

假设  $f(x, y)$  在  $[\underline{x}, \bar{x}] \times [\underline{y}, \bar{y}]$  上具有一阶连续偏导数。为了估计  $f(x, y)$  在  $[\underline{x}, \bar{x}] \times [\underline{y}, \bar{y}]$  的界, 将  $f(x, y)$  在区域  $[\underline{x}, \bar{x}] \times [\underline{y}, \bar{y}]$  的中点  $(x_0, y_0)$  处零阶泰勒展开得:

$$f(x, y) = f(x_0, y_0) + hf'_x(x_0 + \theta h, y_0 + \theta h) + kf'_y(x_0 + \theta h, y_0 + \theta h)$$

其中

$$(x, y) \in [\underline{x}, \bar{x}] \times [\underline{y}, \bar{y}], x_0 = \frac{\underline{x} + \bar{x}}{2}$$

$$y_0 = \frac{y+\bar{y}}{2}, 0 < \theta < 1$$

$$h = x - x_0 \in \left[ -\frac{\bar{x}-x}{2}, \frac{\bar{x}-x}{2} \right] = \frac{\bar{x}-x}{2} [-1, 1]$$

$$k = y - y_0 \in \left[ -\frac{\bar{y}-y}{2}, \frac{\bar{y}-y}{2} \right] = \frac{\bar{y}-y}{2} [-1, 1]$$

如果已知函数  $f(x, y)$  的二阶一阶偏导数  $f_x(x, y), f_y(x, y)$  在区域  $[x, \bar{x}] \times [y, \bar{y}]$  上的二个区间界  $B_x, B_y$  使得  $f_x(x, y) \in B_x, f_y(x, y) \in B_y$ 。并记  $x_1 = (\bar{x}-x)/2, y_1 = (\bar{y}-y)/2$ , 则  $f(x, y)$  在区域  $[x, \bar{x}] \times [y, \bar{y}]$  上的界  $[f, \bar{f}]$  可以用区间运算表示为:

$$[f, \bar{f}] = f(x_0, y_0) + x_1 B_x [-1, 1] + y_1 B_y [-1, 1]$$

该方法的主要缺点是需要手工预先估计函数  $f(x, y)$  的二阶一阶偏导数  $f_x(x, y), f_y(x, y)$  在区域  $[x, \bar{x}] \times [y, \bar{y}]$  上的二个区间界  $B_x, B_y$ 。

②一阶泰勒展开

假设  $f(x, y)$  在  $[x, \bar{x}] \times [y, \bar{y}]$  上具有二阶连续偏导数。为了估计  $f(x, y)$  在  $[x, \bar{x}] \times [y, \bar{y}]$  的界, 也可以将  $f(x, y)$  在区域  $[x, \bar{x}] \times [y, \bar{y}]$  的中点  $(x_0, y_0)$  处一阶泰勒展开得:

$$f(x, y) = f(x_0, y_0) + hf_x(x_0, y_0) + kf_y(x_0, y_0) + \frac{1}{2} h^2 f_{xx}(x_0 + \theta h, y_0 + \theta k) + \frac{1}{2} k^2 f_{yy}(x_0 + \theta h, y_0 + \theta k) + hk f_{xy}(x_0 + \theta h, y_0 + \theta k)$$

其中  $(x, y) \in [x, \bar{x}] \times [y, \bar{y}], x_0 = (x + \bar{x})/2$ 。

$$y_0 = (y + \bar{y})/2, 0 < \theta < 1$$

$$h = x - x_0 \in \left[ -\frac{\bar{x}-x}{2}, \frac{\bar{x}-x}{2} \right] = \frac{\bar{x}-x}{2} [-1, 1]$$

$$k = y - y_0 \in \left[ -\frac{\bar{y}-y}{2}, \frac{\bar{y}-y}{2} \right] = \frac{\bar{y}-y}{2} [-1, 1]$$

如果已知函数  $f(x, y)$  的二阶偏导数  $f_{xx}(x, y), f_{yy}(x, y), f_{xy}(x, y)$  在区域  $[x, \bar{x}] \times [y, \bar{y}]$  上的三个区间界  $B_{xx}, B_{yy}, B_{xy}$  使得  $f_{xx}(x, y) \in B_{xx}, f_{yy}(x, y) \in B_{yy}, f_{xy}(x, y) \in B_{xy}$ 。并记  $x_1 = (\bar{x}-x)/2, y_1 = (\bar{y}-y)/2$ , 则  $f(x, y)$  在区域  $[x, \bar{x}] \times [y, \bar{y}]$  上的界  $[f, \bar{f}]$  可以用区间运算表示为:

$$[f, \bar{f}] = f(x_0, y_0) + x_1 f_x(x_0, y_0) [-1, 1] + y_1 f_y(x_0, y_0) [-1, 1] + \frac{1}{2} x_1^2 B_{xx} [0, 1] + \frac{1}{2} y_1^2 B_{yy} [0, 1] + x_1 y_1 B_{xy} [-1, 1]$$

该方法的主要缺点是需要手工预先估计函数  $f(x, y)$  的二阶偏导数  $f_{xx}(x, y), f_{yy}(x, y), f_{xy}(x, y)$  在区域  $[x, \bar{x}] \times [y, \bar{y}]$  上的三个界  $B_{xx}, B_{yy}, B_{xy}$ 。

(3)自动微分+泰勒展开法

无论是在零阶还是一阶泰勒展开, 泰勒展开法中对  $B_x, B_y, B_{xx}, B_{yy}, B_{xy}$  的区间范围只能手工估计, 这在函数  $f(x, y)$  的表达式比较简单时是可行的, 当  $f(x, y)$  表达式比较复杂的时候手工估计会变得很困难甚至根本不可行。

在对函数泰勒展开的基础上, 可以运用以上所述的自动微分方法使得在计算函数值的同时把偏导数值以及偏导数的界同时计算出来。

在利用自动微分技术计算偏导数的界的时候, 只要将原来变量  $x$  和  $y$  分别换成区间  $[x, \bar{x}]$  和  $[y, \bar{y}]$ , 那么原来自动微分过程中的数值运算就变成了区间运算, 原来的计算结果即偏导数的值就变成了偏导数的界。称这种传统自动微分技术的改进为区间自动微分技术。

与手动的泰勒展开法相比, 运用区间自动微分技术的优势在于不管  $f(x, y)$  有多复杂, 都可以由计算机程序自动高效地估计出  $B_x, B_y, B_{xx}, B_{yy}, B_{xy}$  的区间范围。

5 实验结果

以函数

$$f(x, y) = x^2 + y^2 + \cos 2\pi x + \sin 2\pi y + \sin(2\pi x^2) \cos(2\pi y^2) - 1$$

为例, 得到的实验结果如表 1~2 和图 1~5。该实例取材于 1992 年的 SIGGRAPH 会议论文<sup>[8]</sup>。

表 1 自然区间法和泰勒展开法计算结果

	自然区间	泰勒展开	
		零阶	一阶
CPU	10.415	10.622	10.908
SUB	3 023	2 805	2 695
PIX	3 220	2 846	2 066
AREA	0.954 163	0.958 931	0.970 078

表 2 自动微分+泰勒展开方法计算结果

	自动微分+泰勒展开	
	零阶	一阶
CPU	11.707	19.618
SUB	2 151	1 991
PIX	1 884	1 798
AREA	0.972 809	0.974 121

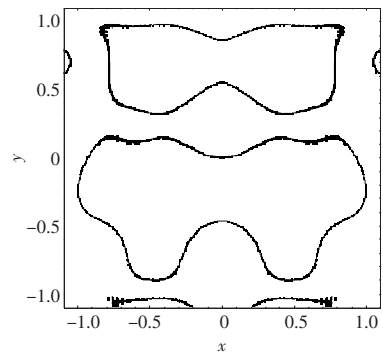


图 1 用自然区间方法绘制的曲线

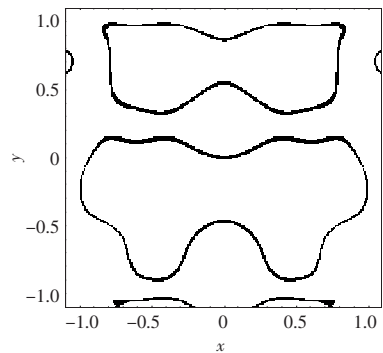


图 2 用手动零阶泰勒展开法绘制的曲线

从实验结果对比上可以看出运用自动微分方法的优势:

(1)CPU(CPU Time Used):从 CPU 的运行时间来看用自然区间方法和泰勒展开法稍微短一些, 但是跟用自动微分+泰勒展开法的差距并不是很大。

(2)SUB(Number of SUBdivisions):即细分的次数。好的方法细分的次数少, 这便于节约时间和空间, 同时还表明算法的收敛速度更快, 很明显, 自动微分+泰勒展开法在这方面表现较好。

(3)PIX(Number of PIXels):即像素数量。像素数量越少, 绘制的图形质量越好; 反之像素数量越多, 绘制的图形质量越差。从这个指标来看也是自动微分+泰勒展开法表现最好。而用自然区间方法和手动泰勒展开方法绘制的图形从图中(图 1~

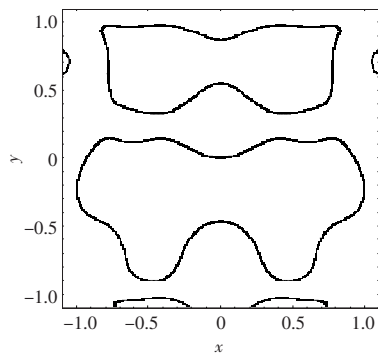


图3 用手动一阶泰勒展开法绘制的曲线

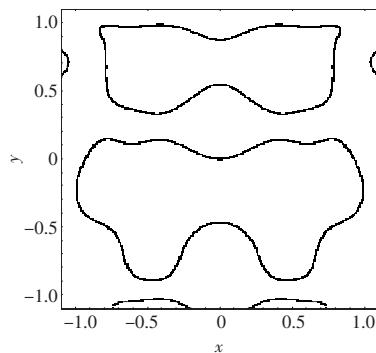


图4 用自动微分+零阶泰勒展开法绘制的曲线

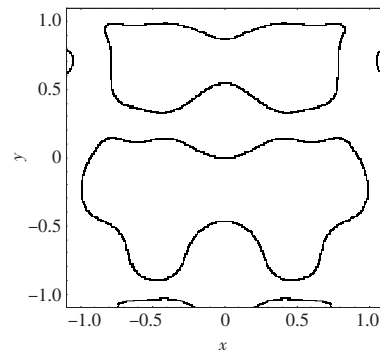


图5 用自动微分+一阶泰勒展开法绘制的曲线

3)明显可以看到所包含的像素太多,图形质量不过关。

(4)AREA(Percentage of AREA Classified):即已探明不包含隐式曲线的空白区域面积所占有的百分比。占有百分比越大的图像质量越好,应用自动微分+泰勒展开法得到的图像质量参数明显较其他两种方法更优。

## 6 结论

从上面的比较分析可以看出:虽然从效率(CPU 占用时间)上看损失了一点,将自动微分与泰勒展开相结合应用到隐式曲线绘制的细分算法中就图形的质量而言有比较明显的优势。由于用自然区间法绘制隐式曲线通常得不到满意的图形,而当表达隐式曲线的函数表达式比较复杂的时候手动泰勒展开法绘制隐式曲线根本不可行,从而用自动微分法绘制隐式曲线的最重要的意义在于使得绘制任意复杂的隐式曲线的高质量图形成为可能。目的是将自动微分应用于计算机图形学领域隐式曲线绘制的细分算法中,自动微分在计算机图形学领域的其他地方仍大有用武之地,理论上讲任何用到导数的地方都可以用自动微分来完成,因此自动微分在计算机图形学中的其他应用,值得进一步探索和研究。

## 参考文献:

[1] Griewank A. On automatic differentiation, mathematical programming: Recent developments and applications[M]. Iri M, Tanabe K. Amsterdam: Kluwer Academic Publishers, 1989: 83-108.

[2] Bartholomew-Biggs M C, Brown S, Christianson B, et al. Automatic differentiation of algorithms[J]. Journal of Computational and Applied Mathematics, 2000, 124(1/2): 171-190.

[3] 陈晓宇,程强,宋金帅.自动微分方法在 XIAMEN 软件优化中的应用[J].数值计算与计算机应用,2009,30(1):21-29.

[4] Griewank A. Evaluating derivatives: Principles and techniques of automatic differentiation[J]. Series Frontiers In Appl Math SIAM, 2000, 19: 15-36.

[5] 张海斌,薛毅.自动微分的基本思想与实现[J].北京工业大学学报,2005,31(5):332-336.

[6] 潘雷,谷良贤,龚春林.改进自动微分方法及其在飞行器气动外形优化中的应用[J].西北工业大学学报,2007,25(3):398-401.

[7] 程强,张海斌,王斌.自动微分的原理和方法[J].计算数学,2009,31(1):1-22.

[8] Snyder J M. Interval analysis for computer graphics [J]. Computer Graphics, 1992, 26(2): 121-130.

[9] 金通洸,沈炎.图形显示和数控绘图的 T-N 方法(1)-曲线的直线逼近[J].浙江大学学报,1979(1):75-88.

[10] 蔡耀志.正负法数控绘图[M].杭州:浙江大学出版社,1990.

[11] Taubin G. Distance approximations for rasterizing implicit curves[J]. ACM Transactions on Graphics, 1994, 13(1): 3-42.

[12] Tupper J. Reliable two-dimensional graphing methods for mathematical formulae with two free variables[C]//ACM SIGGRAPH, 2001: 77-86.

[13] Plantinga S, Vegter G. Isotopic meshing of implicit surfaces[J]. The Visual Computer, 2007, 23(1): 45-58.

(上接 142 页)

脸,最后利用类内特征脸进行 PCA 分析。提高了识别率,大幅度降低了特征空间大小,并且具有较好的鲁棒性。在人脸库中的数目很大或者图片较大时其优点非常明显。

## 参考文献:

[1] Feng G C, Jiang J. Image extraction in DCT domain[J]. IEEE Proc Vis Image Signal Process, 2003, 150(1).

[2] Turk M, Pentland A. Face recognition using eigenfaces [C]//Proceedings of IEEE Conference on Computer Vision and Pattern Recognition. Maui: IEEE, 1991: 586-591.

[3] Turk M, Pentland A. Eigenfaces for recognition[J]. Cognitive Neuroscience, 1991, 3(1): 71-86.

[4] Yang Jian, Zhang D, Alejandro F, et al. Two-dimensional PCA: A new approach to appearance-based face representation and recogni-

tion[J]. IEEE Transactions on PAMI, 2004, 26(1): 131-137.

[5] Wallace G K. The JPEG still picture compression standard[J]. Communication of the ACM, 1991, 34(4): 31-45.

[6] Meng Ji-cheng, Zhang Wen-bin. Volume measure in 2DPCA-based face recognition[J]. Pattern Recognition Letters, 2007, 28: 1203-1208.

[7] Wang Hui-yuan, Leng Yan, Wang Zeng-feng, et al. Application of image correction and bit-plane fusion in generalized PCA based face recognition[J]. Pattern Recognition Letters, 2007, 28(16): 2352-2358.

[8] Wang Peng, Ji Qiang, James L, et al. Modeling and predicting face recognition system performance based on analysis of similarity scores[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2007, 29(4): 665-670.

[9] Gan Jun-ying, Zhou Dang-pei, Li Chun-zhi. A method for improved PCA in face recognition[J]. International Journal of Information Technology, 2005, 11(11).