字稿求精的几个技术问题*

胡永千

(中国科学院软件研究所计算机科学开放实验室 北京 100080)

摘要 自从汉字进入计算机以来,字形技术发展很快. 近几年出现的曲线轮廓字代表了当前计算机汉字字形的最高水平. 它可以应用于很多领域,特别是国际标准的大字符集的推行,其应用前景将更加广阔. 本文着重讨论曲线轮廓字形的几个技术问题. 曲线拟合的好坏直接关系到生成字形的质量;采用多种方法进行拟合,并辅助以其它一些外围处理,可以得到数据量小且精度高的字形. 出版、广告制作、影视等大量地使用形体各异的美术字,曲线轮廓字所具有的独特性质有可能用比较简单的拓扑变换,自动地生成多种规则和不规则几何形状的美术字.

关键词 曲线轮廓,拟合,拓扑变换,最小二乘法.

中图法分类号 TP391

汉字信息处理对汉字字形的精度要求越来越高. 高精度离散型的点阵字形数据量太大,即使有各种压缩方案,数据量仍很可观,而且还带来还原速度问题;放大时,即使经过平滑处理仍留有明显的痕迹. 连续型矢量轮廓字的数据量比点阵字形小多了,但是高倍率放大的质量仍不理想. 连续型的曲线轮廓字代表了当前汉字字形技术的最高水平. 它能够在保证质量的前提下,对字形施行各种变换;同时,连续化的字形数据又有较小的信息冗余度,比矢量轮廓字更加节省存储空间. 本文介绍了汉字字稿求精处理系统 CCRS (Chinese character refinement system)中所采用的某些技术.

1 曲线拟合

一般来说,从离散的点阵信息生成连续化的曲线轮廓数据信息的基本思想是:将轮廓线分段求近似拟合它的低次 Bezier 曲线(以下简称曲线),得到用来描述整个轮廓线的一组连续化数据. 可见生成曲线轮廓字形的关键问题之一就是选择好的拟合方法. 实验表明,单一的拟合算法往往是不完善的. 因为汉字字形的轮廓线形状千差万别,加之外因造成的种种缺陷,使轮廓线上的曲率变化更趋复杂,所以很难找到一种算法能够适合所有情况. 文献[1]中的算法 3.3 介绍了一种拟合算法,它规定 $P_1P_2 // P_0P_3$(如图 1 所示),这个规定保证了经过拟合的轮廓段与原轮廓段有相同的极大值,但不一定有相同的坐标;因此有一部分轮廓段拟合不是很好. 为了提高拟合质量,我们又实验了几种其它的曲线拟合方法. 本文将简单地介

* 作者胡永千,1944 年生,高级工程师,主要研究领域为软件工具,字形技术.
本文通讯联系人:胡永千,北京 100080,中国科学院软件研究所计算机科学开放实验室
本文 1996-12-09 收到修改稿

绍一下最小二乘法和解方程法.

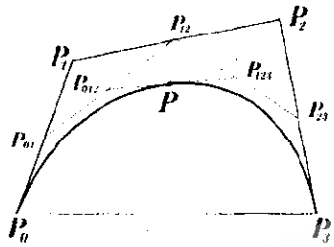

图 1

1.1　最小二乘法[2]

从扫描仪输入的字稿，其轮廓线一般都存在不同程度的缺陷. 但从总体上来看，它们基本是围绕一条中心线内外摆动. 所以，利用最小二乘法拟合曲线比较合适.

由于每条轮廓线由若干首尾相接的曲线组成，这就需要各段轮廓的端点位置保持不变. 直接用最小二乘法时，端点往往会发生偏移，因此采用最小二乘法拟合字形轮廓时，需要附加必要的约束条件，以便使经过拟合以后的轮廓线仍然首尾相接. 假设有某轮廓段初始划分点的一组坐标点 $z_i(i=0,1,2,\ldots,n)$，从三次曲线的参数函数式[3]可以看出端点位置不变就有

$$P_0=P(0)=z_0 \qquad P_3=P(1)=z_n$$

根据最小二乘原理推导，问题最终变为求关于这段轮廓控制导引点 P_1,P_2 的方程组的解：

$$\sum_{i=1}^{n-1}[(1-t_i)^5t_iP_0+3(1-t_i)^4t_i^2P_1+3(1-t_i)^3t_i^3P_2+(1-t^i)^2t_i^4P_3-(1-t_i)^2t_iz_i]=0 \quad (1)$$

$$\sum_{i=1}^{n-1}(1-t_i)^4t_i^2P_0+3(1-t_i)^3t_i^3P_1+3(1-t_i)^2t_i^4P_2+(1-t_i)t_i^5P_3-(1-t_i)t_i^2z_i]=0 \quad (2)$$

用参数式来求解，解方程前先要根据 z_i 近似地确定相应的参数 t_i. 可用如下算法

设有曲线$\overset{\frown}{AB}$，记线段 $z_{i-1}z_i$ 的长度为 $l_i(i=1,2,\ldots,n)$，其中 $z_0=P_0,z_n=P_3$.

① 最简单的近似计算

令 $$\sum\overset{\frown}{AB}=l_1+\ldots+l_n \qquad t_i=\frac{l_1+\ldots+l_i}{\sum\overset{\frown}{AB}}$$

② 较细致一点的近似计算

单凹向三次曲线极大值点的 z_i 所对应的 t_i 和双凹向三次曲线拐点处 z_i 所对应的 t_i 可以近似地取 0.5，将该点称为参数中点 C. 在 $z_i(i=0,\ldots,n)$ 中定出一个参数中点 $t_j=0.5(0<j<n)$ 后，再以此为界限，分别求出$\overset{\frown}{AC}$和$\overset{\frown}{CB}$上点的近似 t_i 值.

令 $$\sum\overset{\frown}{AC}=l_1+\ldots+l_j \qquad \sum\overset{\frown}{CB}=l_{j+1}+\ldots+l_n$$

if$(i<j)$ $$t_i=\frac{l_1+\ldots+l_i}{2\cdot\sum\overset{\frown}{AC}}$$

else $$t_i=0.5+\frac{l_{j+1}+\ldots+l_i}{2\cdot\sum\overset{\frown}{CB}}$$

然后，解(1)、(2)，求出 P_1,P_2. 其中细致一点的近似计算效果要好些. 当初始划分点集的某段链码分布不均匀时，有可能在点比较稀少的部位发生拟合异常. 这种情况一般都是发生在端点附近，使端点处的切线方向在拟合后有很大改变. 可以在点稀少的地方插入一些实验参考点，同时检查端点的切线方向在拟合前后的差.

```
设定误差值δ;
计算B点拟合前的切线方向V_B;
拟合该段轮廓线;
计算B点拟合后的切线方向V'_B;
while(|V'_B−V_B|>δ)
{
    在AB连线间增加实验参考点;
    拟合该段轮廓线;
    计算B点拟合后的切线方向V'_B;
}
合格，继续拟合下一个轮廓段;
```

为了使端点的切线方向也保持不变，还可以再附加一个约束条件，即把 P_1,P_2 限制在 P_0,P_3 的原轮廓段的切线上.

利用轮廓线在 P_0,P_3 处切线的参数式[3]：

$$p_1(t_{01})=P_0+(P'_1-P_0)t_{01} \qquad (t_{01}\text{为任意实数}) \tag{3}$$

$$p_2(t_{32})=P_3+(P'_2-P_3)t_{32} \qquad (t_{32}\text{为任意实数}) \tag{4}$$

其中 P'_1,P'_2 为 P_0,P_3 切线上不与 P_0,P_3 重合的任一点. 用(3)、(4)右边替换(1)、(2)中的 P_1,P_2，将问题变为求关于 t_{01},t_{32} 的方程组的解. 再通过(3)、(4)式求出的 $p_1(t_{01})$ 和 $p_2(t_{32})$ (即 P_1,P_2). 这种方法可以有效地保证两段曲线在它们的连接点是完全平滑的.

1.2 解方程法

我们把根据已知两端点的矢量值和该轮廓段的极大值、建立方程组求解两个控制导引点的方法称作解方程法.

已知：$P_0(x_0,y_0),P_3(x_3,y_3),P(x,y)$(如图1所示)，其中 P 是轮廓段的极大值点. k_0,k_3 为端点 P_0,P_3 的切线斜率；取 m_0,n_0,m_3,n_3，使得 $k_0=n_0/m_0;k_3=n_3/m_3$.

求：$P_1(x_1,y_1),P_2(x_2,y_2)$

解：设

$$x_1=x_0+a_0\times m_0 \tag{5}$$

$$x_2=x_3+a_3\times m_3 \tag{6}$$

$$y_1=y_0+a_0\times n_0 \tag{7}$$

$$y_2=y_3+a_3\times n_3 \tag{8}$$

按 P_0,P_1,P_2,P_3,P 间的关系有

$$x=(x_{012}+x_{123})/2=(x_0+3\times x_1+3\times x_2+x_3)/8$$

$$y=(y_{012}+y_{123})/2=(y_0+3\times y_1+3\times y_2+y_3)/8$$

于是

$$3\times m_0\times a_0+3\times m_3\times a_3=8\times x-4\times(x_0+x_3)$$

$$3\times n_0\times a_0+3\times n_3\times a_3=8\times y-4\times(y_0+y_3)$$

则只需解方程组，求出 a_0,a_3 即

```
if(无解或多解)
{则把P(x,y)作为新的划分点，将该轮廓段一分为二，再分别求解.}
else
{将a_0和a_3代入(5)～(8)式，求出P_1和P_2}
```

该算法的特点是，可以保证拟合以后的轮廓段通过原轮廓段的极大值点.

2　字形还原精度的检验

曲线轮廓字形大大节约了信息存储空间，但是也有一个还原的速度问题. 通常作法是用递归算法计算曲线轮廓段的拱高 h；当 $h<1$ 时，该轮廓段可作为直线段处理；否则，从拱点将该轮廓段分为两段，再分别计算各自的拱高. 为提高效率，可以不用浮点运算和除法，并避免用开方运算. 设有单凹向轮廓段的参数曲线；则两端点间直线段的参数直线的函数式是

$$L(t)=(1-t)P_0+tP_3 \tag{9}$$

当 $t=0.5$ 时到达轮廓段的最高点，可以通过计算向量 $P(0.5)-L(0.5)$ 的长度来判断轮廓段是否可以近似地作为直线段处理.

设　　$h(t)=P(t)-L(t)$

则　　$h(0.5)=\frac{3}{8}(P_1-P_0+P_2-P_3)$

从作图法看(如图 2 所示)，就是

$$\overrightarrow{h(0.5)}=\frac{3}{8}(\overrightarrow{P_0P_1}+\overrightarrow{P_3P_2})$$

用 $\|h(0.5)\|$ 表示 $\overrightarrow{h(0.5)}$ 的长度，当 $\|h(0.5)\|<\varepsilon$ 时，可用 $L(t)$ 代替 $P(t)$（ε 为一给定值，如取 $\varepsilon=\frac{1}{2}$）；这一关系可通过不等式(10)来判断.

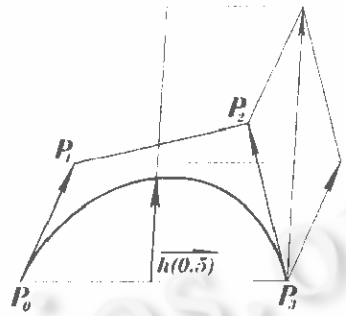

图 2

$$\|h(0.5)\|^2=(\frac{3}{8})^2[(x_1-x_0+x_2-x_3)^2+(y_1-y_0+y_2-y_3)^2]<\frac{1}{4}$$

$$9\cdot[(x_1-x_0+x_2-x_3)^2+(y_1-y_0+y_2-y_3)^2]<16 \tag{10}$$

经测试，还原速度可提高 30～50%.

3　拓扑变换在美术字生成中的应用

在汉字的应用中，大量地使用形体各异的美术字. 采用复杂的拓扑变换技术无疑可以得到极其完美的艺术字体. 现代图形学的发展，对各种复杂的图形图象施行动态的三维立体变换已不是什么困难的事，它已应用于科研、文化、教育以及广告等众多领域；其艺术效果令人赞叹不已. 不过没有必要用复杂的技术去做所有的事情，就如大马拉小车一样. 曲线轮廓字所具有的独特性质，为人们提供了一种可能性；即用比较简单的拓扑变换自动地生成多种多样的规则和不规则形状的美术字. 规则形状的变换不用再说，目前已有一些方法对汉字实现

任意长宽比例，任意大小的矩形、椭圆、梯形等变换，如“中文之星”的艺术字体变换. 对于不规则形状的变换，也可以用一些简单的方法实现. 一种方法是用仿射变换将矩形字体变成任意凸四边形字体；另一种方法是叠加. 采用叠加方法的优点在于以较小代价得到尽可能丰富的结果. 由于本文以最基本的初等函数为变换的基础，计算简单，效率也就比较高. 在(12)式中，当$l>0$时为梯形变换，当$l=0$时就为三角变换，只用了算术运算. 椭圆变换除算术运算外，还用到幂运算，已是最复杂的了. 叠加也非常简单.

设有初等函数$f_1(z), f_2(z), \ldots, f_n(z)$($n$为自然数)，若用$f_i(z)$与$f_j(z)$叠加，套用(11)式即可(其中$1\leqslant i, j\leqslant n$，$z$为某一点).

$$z'=f_i(f_j(z)) \tag{11}$$

几种简单的变换以不同的排列组合相叠加，即可得到不同的结果，非常灵活；若以单一的变换得到这些结果，算法实现要比叠加法复杂. 叠加可以是一种初等函数变换的多次重复，如图3中$\overset{\frown}{A''B}$是AB重复做两次同样的梯形变换的结果；椭圆变换重复两次可得到类似菱形的字体(如图4(c)所示). 叠加也可是两种以上的初等函数变换相叠加；如两次梯形变换以不同的方式叠加能够得到几种不同的非线性结果，原因可用(12)式加以解释：设有直角坐标系(如图3所示)中AB上任一点(x,y)，可用(11)式得到点(x',y')：

$$x'=x \qquad y'=\frac{x+l}{n+l}\cdot y \tag{12}$$

同样，再次利用(11)式又可以从(x',y')得到(x'',y'')；很容易证明点(x'',y'')不在直线段$A''B$上；于是一条平行于X轴的线段经变换成为一条其延长线与X轴相交的线段($AB\rightarrow A'B$)，而不平行于X轴的直线段经变换就成为曲线段($A'B\rightarrow\overset{\frown}{A''B}$). 以字形包围框上下边作为梯形两底边的变换与左右边作为梯形两底边的变换相叠加得到图板扭曲变形的效果(如图4(d)所示)；两次上下底边位置相反的梯形变换相叠加时，根据参数不同可以呈现出炮弹形(如图4(e)所示)纺锤形或鼓形. 又如：以椭圆变换与三角形变换相叠加可以得到水滴形字体(如图4(f)所示)；椭圆变换和梯形变换叠加以后是扇形等；椭圆变换本身也可以近似地作为圆形与矩形变换叠加的结果. 有时当用不同的变换以不同的顺序相叠加时，也可以得到不同的结果；以正弦变换与梯形变换相叠加为例：若以字的包围框的宽度作为正弦曲线的一个周期，那么当先进行正弦变换时，最终该字两端将形成宽窄不同的全周期正弦波形(如图4(g)上所示)；而当先进行梯形变换时，该字的上部将是不足一个周期的波形(如图4(g)下所示). 利用初等函数排列组合可以对字形施行上百种拓扑变换，其中很多是经常会用的；读者可以自己试验各种叠加方法，一定能得到一些有意思的结果(图4(c),(e),(f)为图4(b)变换的结果，图4(d)由图4(a)变换得到).

致谢 本文的工作得到董韫美研究员的关怀和指导，特此致谢.

参考文献

1 卫平，董韫美. 一个将二值黑白图象由离散化为连续的方法. 计算机学报，1989，**12**(**8**)：607～608.
2 清华大学，北京大学《计算方法》编写组. 计算方法. 北京：科学出版社，1974. 51～56.
3 苏步青，刘鼎元. 现代数学丛书：计算几何. 上海：上海科学技术出版社，1981. 102～115.

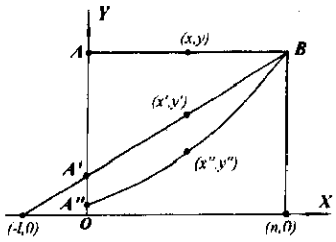

图 3

图 4(a)

图 4(b)

图 4(c)

图 4(d)

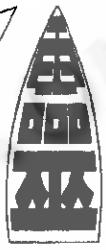

图 4(e)

图 4(f)

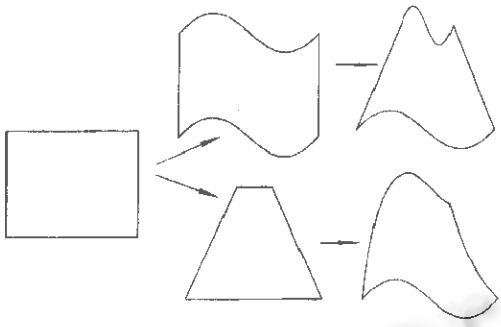

图 4(g)

SOME TECHNIQUES OF CHARACTER REFINEMENT

HU Yongqian

(*Laboratory of Computer Science　Institute of Software　The Chinese Academy of Sciences　Beijing*　100080)

Abstract　This article stresses on the discussion of a few techniques for curve outlined characters refinement. The quality of the fit directly affects the quality of the created shape of the character. By using many fitting ways, one can get a shape of high resolution with only very few data. When forming Chinese characters many artistic-styled characters are often used. Because of the unique characteristics of curved outline of Chinese character, it is possible to use the simpler topological transformation, thus automatically creating artistic-styled characters of regular and irregular shapes.

Key words　Curve outline, fitting, topological transformation, least square method.

Class number　TP391