

提高曲线轮廓汉字还原质量的一些方法

胡长原 武港山 张福炎

(南京大学计算机系, 210008)

摘要: 本文讨论了曲线轮廓汉字还原输出时的质量问题, 并从还原算法及字库设计等方面探讨了提高曲线轮廓汉字还原质量的一些方法。

关键词: 曲线轮廓汉字, 字形还原算法, 网格适配, 提示技术。

一、引言

汉字字形的描述经历了从笔画式汉字和点阵式汉字到向量轮廓汉字的发展过程, 目前已进入研究和应用高精度曲线轮廓汉字的阶段。曲线轮廓汉字是通过用直线段和曲线段共同构造汉字字形轮廓的方法来描述汉字字形的。汉字曲线轮廓字形的描述多选用三次 Bezier 曲线(以下简称 Bezier 曲线)。原因有二: 一是 Bezier 曲线具有良好的直观性和局部修改性, 方便交互式地修改曲线的形状; 二是 Bezier 曲线与其它曲线一样也具有仿射变换不变性, 这使得对字形作任意放大和缩小变换时, 曲线均能精确地描述字形轮廓。

曲线轮廓汉字字形还原输出时, 首先根据直线和曲线的描述信息和变化数值构造出相应大小的字形轮廓(又称为路径), 然后对轮廓做扫描变换, 生成字形点阵信息。原理上讲, 曲线轮廓字形可以随意地放大和缩小。但在以光栅设备为输出手段的实际应用场合, 精确的轮廓描述最终要变换为离散的光栅点阵。这一过程势必引进舍入误差。对于以较小尺寸(如 40×40 像素阵列)还原的字形来说, 舍入误差会影响到字形轮廓描述的精度, 导致字形还原失真。字形还原的失真现象是多种多样的, 归纳起来有以下几种:

- 笔画宽度不一致。宽度应该相同的笔画, 可能因为舍入作用而造成还原结果宽度不同。另一方面字形设计上笔画宽度的微小差别, 被舍入作用加以夸大, 也会导致笔画宽度的明显不一致(图 1a)。
- 笔锋失真。汉字笔锋形状复杂, 种类繁多, 尺寸又相对较小, 因而更容易产生还原失真。笔锋失真的一个例子是, 相同的笔锋形状还原出不同的点阵效果(图 1b)。
- “断笔”与“粘笔”。当上述失真现象严重时, 会出现字形笔画丢失、断开或者粘合在一起的现象。例如, 当笔画处于两排像素之间时, 若不作特别处理, 这一笔就丢失了。粘笔现象

本文于 1993 年 10 收到。本课题受国家自然科学基金资助。胡长原, 1966 年生, 1988 年、1991 年分别获南京大学计算机系系统结构专业学士学位和计算机应用专业硕士学位, 现为清华大学计算机应用专业博士研究生, 主要从事电子印刷和出版、字库技术等研究。张福炎, 1938 年生, 现为南京大学计算机科学系教授, 南京大学和清华大学计算机系博士生导师, 主要从事计算机系统结构、计算机图形学与 CAD、电子印刷和出版技术、多媒体技术等研究。

正好相反,当两笔画中间的白空处于两排象素之间时,白空还原不出,于是两笔就粘成为一粗笔了(图 1c)。

- 黑白韵律难以保持。一个美观的汉字,字形空间上黑白分布是有一定规律的,如并列等长的横、竖,其间所夹的白空也常常是等宽的,也即这些横、竖笔画是等距离排放的,从而产生黑白韵律的美观。不加控制的舍入作用,却常常破坏了这种和谐的黑白关系,不仅仅是笔画宽度不能保持一致,笔画间的距离关系也不能保持。

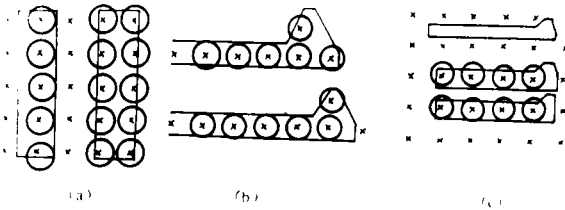


图 1

上述失真现象均是当还原尺寸较小时产生的,因此我们也称之为小字失真现象。小字失真现象不仅损害了字形的美观特性,还可能导致字形难以辨认。这在很大程度上限制了曲线轮廓汉字库的应用范围。因此,必须对曲线轮廓字形的还原失真问题作专门研究。

本文以下从字形还原技术和字库的 hinting 技术两方面探讨解决这一问题的一些方法,所讨论的算法均已实现。

廓字形的还原失真问题作专门研究。

本文以下从字形还原技术和字库的 hinting 技术两方面探讨解决这一问题的一些方法,所讨论的算法均已实现。

二、曲线轮廓汉字还原技术

曲线轮廓汉字的还原技术包括轮廓还原、区域填充和网格适配等具体算法。从提高字形还原质量的目标出发,这些算法均有专门的设计。

2.1 标记填充算法

传统的多边形区域填充算法往往将轮廓边缘上的点均作为区域内点加以着色,使得实际填充区域的面积与理想的填充面积相差较大。例如,一个长和宽分别为 3 和 2 的矩形,其面积为 6,包括边缘的填充算法得到的实际填充面积为 12(如图 2a)。标记填充算法是一个方便硬件实现的高质量填充算法^[1]。该算法的基本原理是,将轮廓线与扫描线的交点的精确值向上取整,对应的象素作一标记。这样就标记了轮廓线与扫描线交点的精确值右边最近一个象素。填充时,自左向右遇上标记的象素就反转着色(如图 2b)。这种算法保证了实际填充面积尽可能逼近理想填充面积。上例改用标记填充算法,实际填充面积仍为 6。

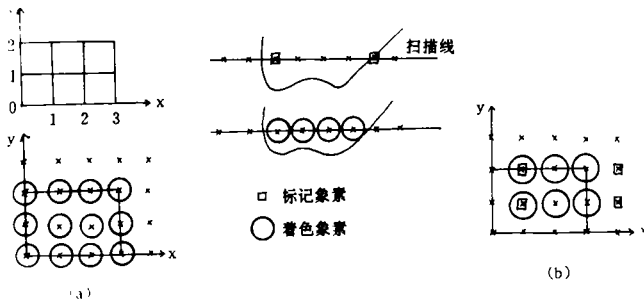


图 2

2.2 采用定点数计算的垂直扫描轮廓还原算法

为实现标记填充算法,需要求出轮廓线与扫描线的所有交点,并且这些交点应是没有取整的精确值。字形缩放变换是通过轮廓上的分段点和曲线控制点作缩放变换实现的。为了减少缩放

过程引入的舍入误差，分段点与控制点的变化结果要以其坐标的精确值表示，而不作取整。这就是实数运算减少误差积累的原理。我们采用定点数计算来保证运算的速度。这一方法是，将整型数的小数点从最末位向左移动 n 位，即扩大了 2^n 倍，再以整数方式运算。通常取 n 为 2 或 3 就足够了。定点数计算减少了由于分段点或控制点舍入取整而给下一步求扫描线与轮廓线精确交点带来的误差。

曲线段首先用直线段去逼近，经过直线逼近，轮廓线与扫描线求交问题就统一成直线段与扫描线的求交问题。即求出 $P_d(x_d, y_d)$ 与 $P_a(x_a, y_a)$ 所连线段与扫描线的所有交点：

$$\{P(x, y) | y \in N \wedge y \in (y_d, y_a), y_d \leq y_a\}$$

垂直扫描变换是一种专为这一问题而设计的算法^[2]，该算法是一种数字微分分析(DDA)方法。

不妨设 $P_d P_a$ 在第一象限， P_d 、 P_a 的坐标均为精确值。 $P_d P_a$ 与扫描线的第 K 个精确交点记为 P_{ik} ，相应的标记点为 P_k (图 3)。第一个交点可由下式求得

$$\begin{cases} y_{i0} = \text{roundup}(y_d) \\ x_{i0} = x_d + \frac{\Delta x}{\Delta y}(y_{i0} - y_d) \end{cases}$$

相应的标记象素为

$$\begin{cases} y_0 = y_{i0} \\ x_0 = \text{roundup}(x_{i0}) \end{cases}$$

式中， $\text{roundup}(x) = \text{trunc}(x + 1) = [x] = [x + 1]$

其余交点 P_{ik} 及相应标记象素 P_k 可以由下式计算：

$$\begin{cases} y_{ik+1} = y_{ik} + 1 \\ x_{ik+1} = x_{ik} + \Delta x / \Delta y \end{cases}$$

$$\begin{cases} y_{k+1} = y_{ik} + 1 \\ x_{k+1} = \text{roundup}(x_{ik} + 1) \end{cases}$$

计算次数，即所求交点数为

$$n = \text{trunc}(y_a) - \text{roundup}(y_d) + 1$$

可以证明，该算法对于 $y_d = y_a$ 以及 P_d 或 P_a 为水平或垂直局部极点的情况均能正确运行。

2.3 网格适配技术(grid-fitting)

采用定点数计算的垂直扫描技术与标记填充算法，已能从填充面积上较为准确地逼近字形轮廓，但对于轮廓中某些复杂的细节部分，仍可能产生退化现象。所谓填充区域的退化是指填充后的图形与轮廓所描述的理想图形产生几何形态上的变化，表现在字形上就是断笔、粘笔等现象(如图 4a)。

出现填充退化现象的一个原因是退化部分的轮廓还原不出相应的象素，解决这一问题的基本原理是对轮廓线作适当修改，使之能套住所要着色的象素的中心。这时就需要对字形轮廓做网格适配。上例的一种可能的网格适配结果见图 4b。

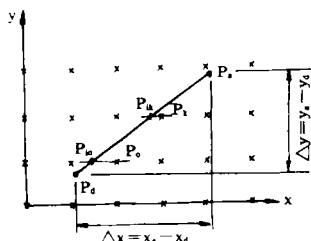


图 3

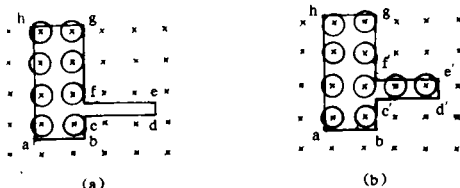


图 4

汉字网格适配技术的具体实现要考虑到汉字小字失真问题的特点和算法效率。汉字笔划细长，横、竖、撇、捺和点等构成了字形骨架。在较低分辨率的设备上，断笔现象是最明显而严重的失真。本文作者针对这一特点，设计并实现了一套用于曲线轮廓汉字还原的网格适配算法，该算法在还原字形时，实时地检测可能的轮廓退化部分并补充着色象素（俗称补象素），从而可以解决断笔现象。

2.4 补象素算法

补象素算法的目的是将汉字的主要笔划，如横、竖、撇、捺在小尺寸还原时可能出现的断笔之处补上着色的象素，在大尺寸还原时，可以不做补象素操作。假设字形轮廓是按逆时针方向定义的，该算法的原理如下：

对于横笔画，在扫描到笔画轮廓线的上边缘时，将上边缘的下邻象素补成填充象素。由于轮廓是按逆时针方向描述的，所以横的上边缘即是轮廓上从右向左画的子路径(如图 5a)。补填的象素不会影响正常的笔画形状。

竖笔画的断笔现象是在边标记填充算法中左右标记象素重合时产生的。一种处理方法是直接在右边缘扩充一个填充象素，以使填充段长度不为零。另一种处理方法考察当前填充段的上下相邻象素，若上下相邻象素均被填充，而该填充段长度为零，则要补填一个象素。

撇、捺在小尺寸还原时除了会产生断笔，还可能出现长度延伸不够的情况。它们的补象素方法与横类似，但是由于轮廓还原时采用的是垂直扫描算法，因此所补的象素与轮廓线与扫描线的交点不是一一对应关系。对于某一个交点，可能要补多个象素(图 5b)。

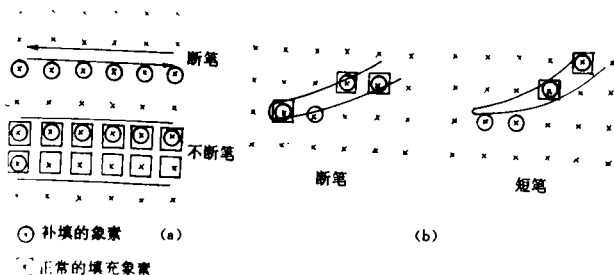


图 5

补象素算法保证了字形还原不断笔，但还难以解决笔画宽度不一致、笔锋不规则以及黑白韵律丢失等问题。要解决这些问题，进一步提高字形还原质量，需要更加精密、高效的网格适配，这时就可使用汉字 hinting 技术。

三、曲线轮廓汉字字形的 hinting 技术

3.1 什么是 hinting 技术

在曲线轮廓字形的轮廓描述数据之外，再增添一些辅助信息(hint)，利用这些辅助信息在还原字形时正确地完成网格适配工作，以解决小字还原失真问题，这就是 hinting 技术。Hinting 技术是提高曲线轮廓汉字还原质量的又一有效途径。前面介绍的补象素算法是通过还原字形时实时地对轮廓的特征作判定从而决定是否补填象素的一种网格适配方法。汉字 Hinting 技术，则是在字库中事先给出了字形轮廓的属性作为辅助信息，如笔画的宽度、笔画之间的相互关系等等。由

这些辅助信息的帮助,网格适配工作就可以准确地得到轮廓的特征,从而可以正确高效地对轮廓做网格适配。理论上讲,Hinting 技术能更好地进行网格适配以提高字形还原质量。

Hinting 技术已成功地应用在西文曲线轮廓字库技术中。代表性的实现方法有两类^[3]。一类如 TrueType,是对轮廓上的分段点以及曲线控制点逐个给出属性,如可否 x/y 方向拉伸,是否笔画宽度参照点,以及与其它点的关系等等。因此,字形设计者实际上已经把如何进行网格适配的说明添加在每个字的描述中。另一类如 Adobe Type 1^[4],这种方法认为 Hinting 技术只要给出一些水平或垂直的“带子”(hstem 与 vstem)。还原字形时,通过对这些水平或垂直的带子在字形网格空间上作优化安排,从而对由这些带子所对应的字形轮廓上的相应部分作网格适配。Type 1 的还原算法相对复杂一些。但是,Type 1 的辅助信息简单而直观,较为容易实现字形辅助信息的自动添加(Auto-hinting)。

3.2 汉字的 Hinting 技术

给汉字轮廓字形添加辅助信息有两个原则。一是信息量不可太大,汉字字数繁多,字形描述的数据量已经很大了,因此要设法控制辅助信息量相对于字形描述数据量的比例。二是辅助信息类型要简单,易于添加。

我们设计了兼容 Type 1 的曲线轮廓汉字库的 hinting 技术,这是因为一方面 Type 1 在国际上已得到广泛的应用,另一方面 Type 1 的 hstem 与 vstem 技术对于以横竖笔画居多的汉字字形也比较合适。对于汉字字形中水平或垂直的笔画段,需要给出其笔画的位置和宽度。这正好是 hstem 或 vstem 的两个参数。这两个参数可以通过对笔画的测量得到。例如 hstem 的两个参数为 y 和 Δy ,其中 y 为水平段的一边的 y 坐标,另一边的 y 坐标为 $y + \Delta y$,所以 $|\Delta y|$ 即为笔画的宽度。用这种方法可以给出笔画的宽度、笔画之间的距离、字形闭包框的大小甚至笔锋的高度与宽度。通过对汉字笔画的分解、分类可以实现辅助信息的自动测量和添加技术(Auto-hinting)^[5]。

带有辅助信息的汉字曲线轮廓字形还原时,首先要对辅助信息作处理。处理的方法是将一个字的所有 hstem 和所有 vstem 分别地在 y 方向和 x 方向作优化安排,也即对由 hstem 和 vstem 所给出的水平或垂直的“带子”作网格适配。然后,再在轮廓还原过程中对轮廓上的分段点和控制点参照相应的 hstem 和 vstem 作调整,从而实现网格适配。这一处理过程把复杂的网格适配工作分为两步完成,第一步是对 hstem 和 vstem 这两种简单的形状作网格适配;第二步才对字形轮廓作网格适配,由于有了 hstem 和 vstem 的参照,对轮廓的网格适配可以在扫描变换之前准确地完成。

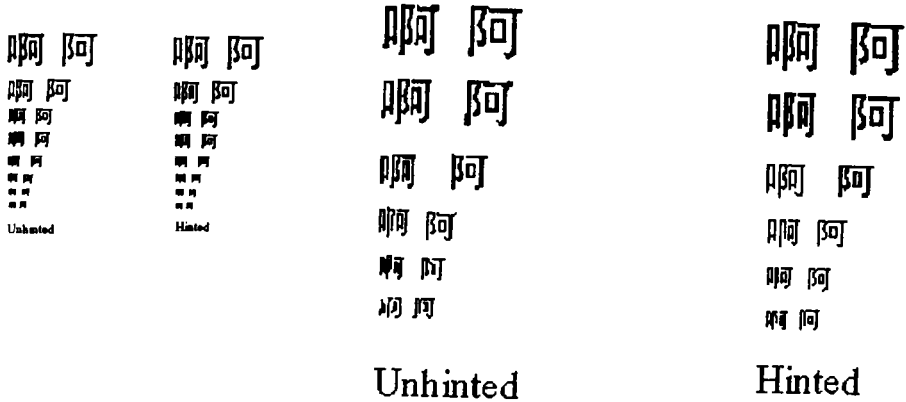
四、结 语

为了提高曲线轮廓汉字字形的还原质量,解决还原时的小字失真现象,从字库设计到还原算法有许多技术可以采用。灵活地结合使用这些技术,就可以有效地改进字形的还原质量。本文介绍的补像素算法和汉字 Hinting 技术是作者设计并实现的两个实用于汉字曲线轮廓字形的网格适配方法。实验表明这两种方法的均能有效地提高轮廓汉字的还原质量。图 6 是在激光印字机上以 300dpi 分辨率输出的实例。

(I) 补像素算法:左下图为补像素前,断笔现象严重;右下图为补像素后。

| | | | |
|----|---------------|----|---------------|
| 44 | 南京大学计算机系第六教研室 | 44 | 南京大学计算机系第六教研室 |
| 38 | 南京大学计算机系第六教研室 | 38 | 南京大学计算机系第六教研室 |
| 32 | 南京大学计算机系第六教研室 | 32 | 南京大学计算机系第六教研室 |
| 28 | 南京大学计算机系第六教研室 | 28 | 南京大学计算机系第六教研室 |
| 24 | 南京大学计算机系第六教研室 | 24 | 南京大学计算机系第六教研室 |
| 20 | 南京大学计算机系第六教研室 | 20 | 南京大学计算机系第六教研室 |
| 18 | 南京大学计算机系第六教研室 | 18 | 南京大学计算机系第六教研室 |
| 16 | 南京大学计算机系第六教研室 | 16 | 南京大学计算机系第六教研室 |

(II) 汉字 Hinting 技术: 下图是以 100dpi 输出的示意图, 即将象素放大为 3×3 倍。



参 考 文 献

- [1] B. D. Ackland & N. H. Weste, The Edge Flag Algorithm—A Fill Method for Raster Scan Display, IEEE Trans. on Computer, Vol. C-30, No. 1, Jan. 1981.
- [2] R. D Hersch, Vertical Scan-Conversion for Filling Purpose, Proceedings CGI'88, 1988.
- [3] R. D Hersch, Font Rasterization; The State of the Art, Tutorial Paper Supported by the EEC DIDOT-COMETT Project, 1991.
- [4] Adobe System Inc, The Type 1 Format Specification, Addison-Wesley, 1990.
- [5] Changyuan Hu & Fuyan Zhang, Automatic Hinting of Chinese Outline Font Based on Stroke Separating Method, Proceedings Pacific Graphics '93, Seoul, Korea, World Scientific Press, 1993.

Some Approaches to Improve the Rasterization Quality of Outline Chinese Characters

Hu Changyuan, Wu Gangshan, and Zhang Fuyan

Nanjing University, 210008

Abstract This paper discusses issues regarding the quality of outline Chinese characters' rasterization. Approaches in both font design and rasterization algorithm which we have practised to help Chinese outline characters achieve high output quality are presented in detail.

Key words: outline Chinese character, rasterization, grid-fitting, hinting.