

# 曲线轮廓汉字的网格适配技术<sup>②</sup>

胡长原 武港山 张福炎

(南京大学计算机科学系)

**【摘要】** 要提高曲线轮廓汉字的还原质量,需要在字形还原时采用网格适配技术。本文介绍了网格适配的基本原理,并在分析汉字字形还原失真现象的基础上,给出了我们自己设计并实现的适合于曲线轮廓汉字特点的二种网格适配方法,即动态的补象素算法和曲线轮廓汉字的 *hinting* 技术。

**关键词:** 曲线轮廓汉字, 网格适配(grid-fitting), 字形扫描变换, *Hinting* 技术。

## 一、引言

随着电子出版、办公自动化和计算机辅助设计等应用领域的飞速发展,人们对汉字字形的输出质量提出了越来越高的要求。曲线轮廓汉字以其存贮开销小,描述精度高并可随意缩放等优点得到了广泛的应用。曲线轮廓汉字采用曲线和直线段描述汉字字形的轮廓。还原时,先将汉字轮廓按要求作缩放变换,再对变换后的轮廓作扫描变换,形成汉字的点阵图象。这一过程从原理上实现了字形的无级缩放,但是由于精确的轮廓描述通过扫描变换形成离散点阵表示时,必须经过舍入,从而导致还原出的汉字字形质量有不同程度的破坏。这种因舍入作用而对还原质量造成的破坏就是曲线轮廓汉字的还原失真现象。

曲线轮廓汉字的还原失真现象有各种表现,归纳起来,较为明显的有以下几种(见图1):

### • 笔画宽度不一致

规则的汉字字体对其笔画宽度有一定的要求,特别是在以较小的光栅点阵还原输出的情况下,横笔和竖笔都应保持一致的宽度。不加控制的舍入作用会使得相同的笔画还原出不同的宽度,从而严重破坏字形的质量。

### • 笔画断、缺

以较小点阵还原汉字时容易发生笔画断开或缺失的现象,例如宋体的横本来就较细,以  $40 \times 40$  以下的点阵还原时经常会缺失。笔画断开的情况常发生在细长的斜笔上,如

①本文1993年10月29日收到

②本文受国家自然科学基金项目资助

撇、捺等。

• 笔锋形状不规则

宋体、黑体系列的汉字具有较规则的笔锋设计，不加控制的舍入作用会使得还原出来的字形笔锋产生畸变。

字形的失真问题还包括黑白韵律失调，楷体、隶书等弧线占多的笔画在局部极点处出现平台或孤立点等等。

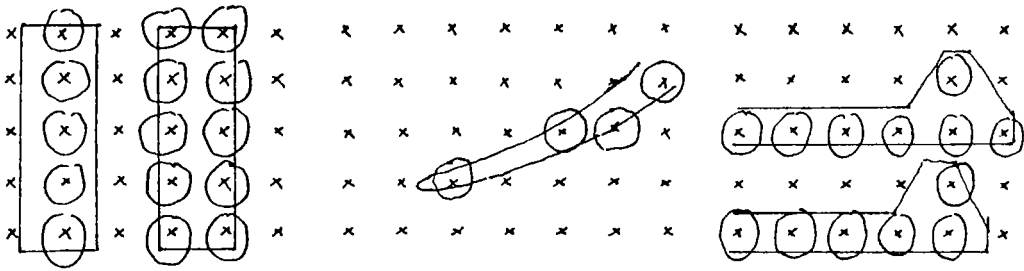


图1 ×象素中心 ⊗还原象素

## 二、网格适配

### 1、网格适配原理

轮廓汉字还原输出时，首先要对轮廓作缩放，然后再对字形轮廓做扫描变换，生成点阵图象。扫描变换算法的原理是当一个象素的中心落在轮廓以内，就将该象素着色。因此，扫描变换得到的字形点阵的实际轮廓是缩放后字形轮廓的舍入，这种舍入作用的随机性是造成字形失真的根本原因。网格适配通过对缩放后的轮廓参照其与象素网格的关系作适当修改，从而有目的地控制舍入作用，减少还原失真现象（见图2）。

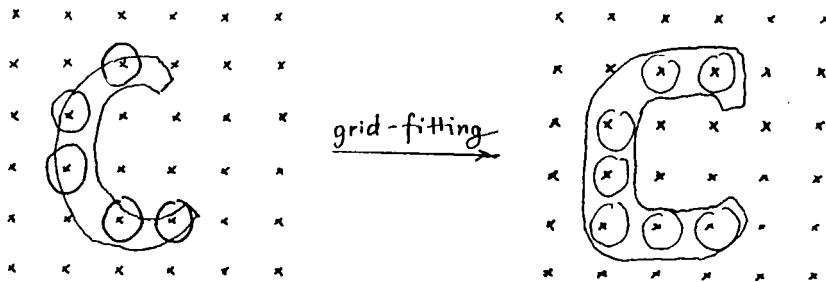


图2

### 2、网格适配的处理方法

网格适配中最基本的技术是相位控制。所谓相位，这里指笔画与象素网的相对位置。在字形定义时，轮廓的垂直或水平边均在象素中心上，经过缩放后，情况就不同了。笔画的轮廓可能出现在两个象素之间的任何位置上，也就是说轮廓的相位可能是任意0-1之间的数。由于轮廓的相位是舍入的直接原因，所以，控制轮廓的相位也就控制了扫描变换的舍入。

设某垂直笔画的两个垂直边之  $x$  坐标分别为  $x_1$ 、 $x_2$ ，可以定义垂直笔画的相位为  $[(x_1+x_2)/2]$  的小数部分。字形定义时， $x_1$ 、 $x_2$  均为整数，故相位只可能是0或0.5。笔画做缩放变换， $x'_1 = x_1 \cdot \text{scale}$ 、 $x'_2 = x_2 \cdot \text{scale}$ 。由于  $\text{scale}$  是任意实数。所以笔画的相位就不能保证是0或0.5，当同样形状（或宽度）的笔画具有不同相位时，其边  $x'_1$ 、 $x'_2$  的舍入就可能使笔画具有不同宽度（见图3），相位控制通过对缩放后的笔画的相位做调整，使之保持为0或0.5（依具体笔画宽度而定），从而得到一致的笔画宽度。

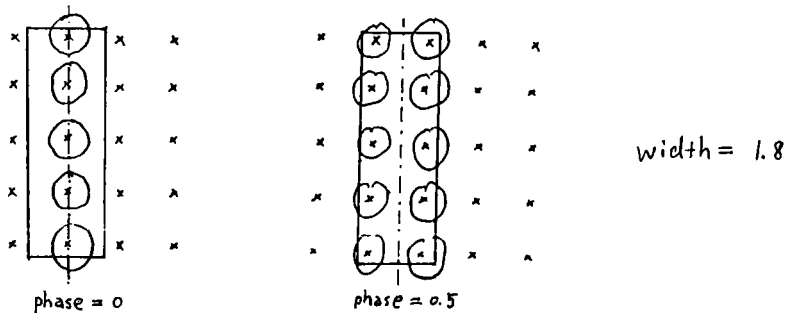


图3

笔画的相位还可以定义为笔画轮廓某一边的相位。这时，相位控制使得笔画的相位总为0，这种方法就是舍入传递的方法[3]。上例中，将  $x'_1$  的相位调整到0，即是对  $x'_1$  做舍入。舍入的误差值加到  $x'_2$  上，再对  $x'_2$  做舍入。由于  $x'_1$  与  $x'_2$  的距离的舍入情况与  $x'_2$  的舍入情况相同，所以舍入传递的相位调整方法也能保证笔画宽度一致。

相位控制的方法可以用来解决曲线极点处的平台或孤立点问题[1]。控制曲线极点处的相位在某一范围，比如  $-1/8$  到  $3/8$ 。就可以改善曲线极点处的还原情况。

网格适配还可以适当地修改轮廓的形状，比如使笔画变粗，而减少断笔现象发生。因为断笔处笔画的水平或垂直宽度已不足1个象素，将断笔处的象素补出，事实上就是对轮廓作了修改。

西文字形为了取得整体上的美观，要求字符的各部分有一致的高度，例如小写字母的高度，大写字母的高度都应保持一致，为此设置了准线来约束字形轮廓在垂直方向上的舍入。西文曲线轮廓字形作网格适配时，需要参照准线的位置来决定对轮廓的修改。

### 三、汉字字形的网格适配

针对曲线轮廓汉字还原失真的现象，汉字字形的网格适配应着重解决如下一些问题：

• 水平、垂直笔画（横、竖）的宽度问题：

横和竖两种笔画在汉字的结构中占有大的比例，它们是组成汉字字形的主要部件。横和竖这两种笔画又相对比较规则，特别是宋体、黑体等，横和竖分别是水平和垂直的，这使得对它们做网格适配也比较容易。这两种笔画的失真现象主要表现为笔画宽度不一致，作网格适配时，要对笔干部分作相位控制以取得笔画宽度的一致。

• 笔锋形状一致性问题：

规则字体的笔锋具有规则的形状。宋体的横具有较大的笔锋，对笔锋的高度应做控制。黑体的横笔锋微小，经常还原为一个像素，网格适配的方法是控制是否允许一个像素笔锋的出现。因为较小的还原点阵下，一个像素的笔锋反而会影响黑体的方头特点。竖的情况与横类似，只是笔锋的形状对字形的影响小些。

• 斜笔画（撇、捺）的断笔问题：

斜笔与网格有一个夹角，失真现象主要是断笔。撇和捺的形状变化多样，可能是渐细的、渐粗的。有时断笔现象发生在细的那一头，因笔尖部分的像素未能还原而使笔画变短，这也是影响字形质量的失真现象。解决断笔或短笔的失真现象，需要使轮廓略变宽，以使得丢失的像素重新落在轮廓边内。前面已经提到，斜笔发生断、短等像素丢失的现象时，该处轮廓的  $x$  方向或  $y$  方向必有一个宽度  $\Delta x$  或  $\Delta y$  小于 1，网格适配可以修改轮廓使轮廓上可能断笔的地方的宽度  $\Delta x$  或  $\Delta y$  均不小于 1（见图 4）。

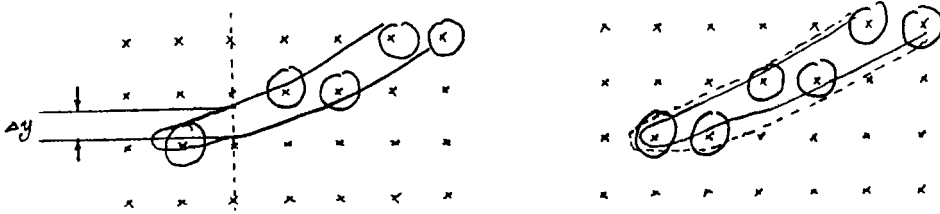


图 4

• 笔画的相互连接关系问题

网格适配要考虑汉字笔画的相互关系。孤立的笔画的网格适配要考虑它与其它笔画的位置关系，例如笔画 A 与 B 之间有一定宽度的空白，作网格适配时，A 与 B 的调整若要保持这一空白不丢失，可以将笔画 A 的调整量加到笔画 B 上去，从而保证 A 与 B 之间的距离不变。相交的笔画做网格适配时，在连接处有时要拉伸或压缩，这时笔画连接处的轮廓应具有“弹性”。有时又要求连接的笔画作同步调整，这时连接处的轮廓就不具有弹性，而应是“刚性”的。图 5 是横和竖连接处的情形。

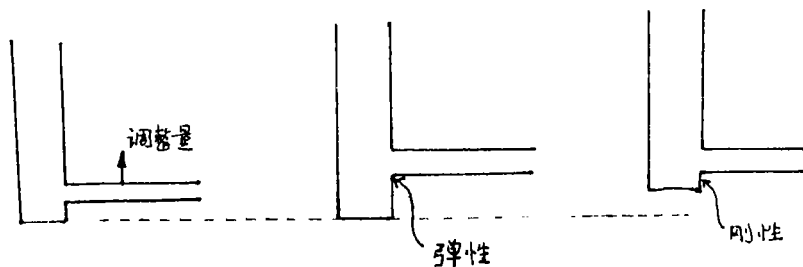


图 5

#### 四、网格适配的动态实现方法

曲线轮廓汉字的网格适配可以在轮廓扫描变换的过程中，实时地对轮廓与网格的关系加以判断而采用相应的适配方法。这种网格适配方法是和字形扫描变换同时完成的，而非事先作用于轮廓，因此是动态的方法。

动态的网格适配重点解决两个问题：一是保持汉字笔画宽度的一致，二是防止笔画断开或丢失。

笔画宽度的一致要有一个标准笔画宽度为参照，扫描变换时，检测轮廓的走向，若是较长的水平或垂直段则要参照标准笔画宽度。方法是，先计算标准笔画宽度在相应还原尺寸下应具有象素宽度  $StdW$ ，再检测当前还原的笔画的象素宽度  $W$ ，若  $|StdW - W| < 1$ ，则取  $StdW$  为当前笔画的宽度，不足宽度的补上一排象素，超过宽度的删去一排象素。由于在某一点阵尺寸以下，笔画的宽度已不可能有太大的变化，所以也可选一个点阵尺寸作为阈值（如 64 点阵），在该尺寸以下还原的字形均取标准笔画宽度。

标准笔画宽度的获得是实时的，在选择某一字体之后，取其中有代表性的一组字，对这组字的横和竖分别作宽度测量。将横的宽度的平均值作为标准横宽( $StdHW$ )，竖的宽度的平均值作为标准竖宽( $StdVW$ )。这种方法有一个局限，它要求字形的横、竖分别是水平和垂直的，对于象楷体、隶书这样的字，笔画宽度就难以动态测量了。

动态地检测断笔并作网格适配的方法，我们称之为补象素方法。该方法和还原算法结合在一起，边还原轮廓，边检测轮廓走向并决定补象素的方法。

定义轮廓方向为逆时针方向。因为曲线最终是以直线段逼近的，可以将轮廓看作是多边形。还原算法在还原轮廓上的一个边  $PQ$  时，补象素算法以着色象素的方法补出可能丢失的象素。该算法描述如下：

$$\Delta x = Q.x - P.x$$

$$\Delta y = Q.y - P.y$$

IF  $|\Delta y| < |\Delta x|$  // 与扫描线夹角在  $(-45, +45)$  范围

IF  $\Delta x < 0$  // 上边

{ 将  $PQ$  与网格垂直线  $\{x = Q.x, \dots, x = P.x\}$

的所有交点的下方 ( $y$  取整) 最近象素着色。

}

ELSE // 与扫描线夹角在  $[45, 135]$  范围

```

IF  $\Delta y > 0$  // 左边
{将 PQ 与网格水平线  $\{y = Q.y, \dots, y = P.y\}$ 
  的所有交点的右方(x 取整)最近象素着色。
}

```

可以验证,补象素的方法不仅可以防止断笔现象,也可以解决撇、捺笔画末端延伸不到位的失真现象。当还原尺寸较大时,字形已不可能出现笔画断、短的失真现象,这时就不必再做补象素操作。补象素算法的一个缺点是要向字形输出缓冲区写大量的象素,这些象素若不是在断、短笔处,都是和扫描变换生成的象素重复的,因而增加了开销。解决这一问题的方法是检测出确切的断、短笔处,但这将增加算法的复杂程度。尽管如此,实验表明上述算法简便易行,对提高字行还原质量有明显的作用(见图6)。

## 五、使用辅助信息 hint 进行网格适配

### 1、什么是 hinting 技术

前面的实时动态网格适配方法,是在字形扫描变换时,通过对轮廓特征的实时检测而完成的。由于轮廓的具体情况千变万化,轮廓特征的实时检测是不能保证完全正确的,这会影响到网格适配的效果。Hinting 技术是另一种网格适配的实现方法,它把网格适配所需要的字形轮廓的特征信息以及网格适配的方法都以辅助信息的方式与字形的轮廓描述数据一起存放。还原算法在获得轮廓数据的同时,也得到了对轮廓做网格适配所需要的信息。Hinting 信息是事先存放在字库中的,因而网格适配可以在扫描变换之前完成。理论上讲,采用 hinting 技术能更好地进行网格适配,提高汉字还原质量。

Hinting 技术已成功地应用在西文的曲线轮廓字库技术中。代表性的实现方法有两类[2],一类以 True Type 为代表,给出轮廓上分段点及控制点的属性,如是否 x/y 方向可拉伸,是否是笔画宽度参照点等等,还要给出该点与其它点的关系甚至网格适配的方法都可以作为辅助信息存放在字库中。另一类是 Adobe Type 1 的 hstem 和 vstem 方法[4]。这种方法认为 hinting 技术只要给出一些水平或垂直的“带子”,网格适配算法将对这些带子做解释,使之在网格平面上作优化安排,从而对它们所对应的字形轮廓做网格适配。Type 1 的 hinting 方法没有 True Type 的灵活,描述能力也不及 True Type 强,因此 Type 1 的网格适配算法相对复杂些。但是,Type 1 的辅助信息内容简单、直观,比较容易在已有的曲线轮廓字形数据上,自动添加 hint 信息。

### 2、遵循 Type 1 格式的曲线轮廓汉字 hinting 技术

给曲线轮廓汉字添加辅助信息有两个原则,一是信息量不可太大,比如,辅助信息不应超过字库容量的 10%—20%;二是辅助信息应相对简单,易于添加,汉字的辅助信息最好是自动添加的,人工添加辅助信息工作量很大。

我们设计并实现了遵循 Type 1 格式的汉字 hinting 方法。之所以选择 hinting 格式,不仅是因为它符合上述两点要求,还因为 Type 1 的 hstem 和 Vstem 方法对于指导以横竖笔划居多的汉字字形的网格适配更为合适。

Type 1 的 hstem 和 vstem 两条 hinting 指令需要两个参数(hstem 为 y,  $\Delta y$ ,vstem 为 x,  $\Delta x$ )以表示水平或垂直的“带子”的位置及宽度。汉字笔画的这两个参数的含义直观,

易于测量。可以用 *hstem* 和 *vstem* 来表示汉字笔画的宽度，笔峰的高度，以及笔画间的相互位置等，笔画的标准宽度也事先存放在字库中。带有这种辅助信息的曲线轮廓汉字的网格适配分作两步完成。第一步是对同时出现的所有 *hstem* 和 *vstem* 分别在 *y* 方向和 *x* 方向上作优化安排，优化安排包括笔画宽度的处理，笔画间关系的处理等；第二步是对轮廓上的分段点和控制点参照 *hstem* 和 *vstem* 作网格适配。这样做保证了字形轮廓的重要部分（横竖）可以由 *hstem* 和 *vstem* 指出并作正确的网格适配，轮廓其它部分的网格适配参照 *hstem* 和 *vstem* 进行，因此可以得到较好的网格适配效果（图 6）。

通过对汉字笔画的分解，根据笔画类型进行自动测量，并将测量结果结合成字符的 *hint*，可以实现汉字字形辅助信息的自动添加[5]。

## 六、结语

解决曲线轮廓汉字还原失真问题的有效方法是对变化后的轮廓做网格适配。汉字的网格适配方法要针对汉字字形的特点综合考虑效果、速度、存储开销以及算法难度等问题，根据不同的应用场合作选择。本文介绍的实时动态的网格适配方法和利用辅助信息的网格适配方法是两个实用的提高汉字还原质量的方法。实验表明，这两种方法都可以有效地解决曲线轮廓汉字的还原失真问题。

## 参考文献

- [1] R.D. Hersch, Character Generation Under Grid Constrains, Computer Graphics, Vol 21, No.4 July 1987.
- [2] R.D Hersch, Introduction to Font Rasterization, Proceedings the International Conference on Raster Imaging and Digital Typography (RIDT). Cambridge University Press, 1989.
- [3] 黄宜华等, 一种高质量汉字向量轮廓压缩与还原算法, Proceedings ICCIP'92, Beijing 1992.
- [4] Adobe System, Type 1 Font Format, Addison-Wesly, 1991.
- [5] Changyuan Hu & Fuyan Chang, Automatic Hinting of Chinese Outling Font Based on Stroke Separating Method, Proceedings Pacific Graphics'93, Seoul, Korea, World Scientific Press, 1993.

(I)补像素算法

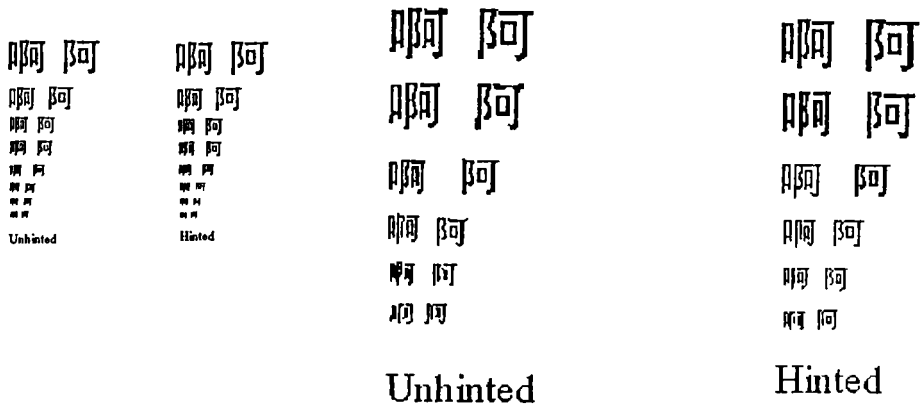
44 南京大学计算机系第六教研室  
 38 南京大学计算机系第六教研室  
 32 南京大学计算机系第六教研室  
 28 南京大学计算机系第六教研室  
 24 南京大学计算机系第六教研室  
 20 南京大学计算机系第六教研室  
 18 南京大学计算机系第六教研室  
 16 南京大学计算机系第六教研室

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  
 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  
 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  
 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

补像素前: 断笔现象严重

补像素后

(II)汉字 Hinting 技术



右边是以 100dpi 输出的示意图, 即将像素放大为 3×3 倍

图 6



# Grid-fitting of Outline Chinese Characters

*Changyuan Hu, Gangshan Wu, Fuyan Zhang*

(Department of Computer Science, Nanjing University, Nanjing 210008, P.R.C)

## **Abstract**

Grid-fitting is the additional process to the rasterization algorithm of outline characters in order to get a better appearance. This paper first introduces the basic principles of grid-fitting, and then, by analysing the trait of Chinese characters, comes to the special grid-fitting methods of outline Chinese characters. This paper also presents two effective experiments, one is the so-called "pixel compensation" algorithm, the other is the hinting method.

## **Key words and phrases:**

outline Chinese character, grid-fitting, character rasterization, hinting method.