

# 基于矢量斜率的分段线性拟合\*

王明江 唐璞山

(复旦大学电子工程系 上海 200433)

**摘要** 介绍了用统计矢量斜率进行平面数据点分段线性拟合的算法。对于欲拟合的一系列平面数据点,一般是有先后顺序的。首先给出了矢量斜率的定义,然后计算每个平面数据点的统计矢量斜率,根据各点矢量斜率值接近的情况,将数据点分割成组,拟合各组数据形成线段,把各线段首尾连接起来就得到了平面数据点的分段线性拟合。定义的矢量斜率包含大小和方向两方面信息,主值区间为(-4~+4),它在主值区间的变化与角度在(-180°~180°)区间中的变化一一对应,且它们的关系曲线有很好的线性度。使用传统斜率进行分段线性拟合,存在斜率值与角度的关系曲线线性度差、斜率取值有时趋向无穷等问题,这些问题影响了拟合的精度,并限制了算法的使用范围。矢量斜率克服了上述问题,从而提供了拟合曲线的质量,且算法可适用于任意曲线。算法时间复杂度为线性。

**关键词** 特征提取,分段线性拟合,形状分析,轮廓分析,边缘检测。

**中图法分类号** TP391

在图纸图像自动读入及识别、PCB版图图像验证以及集成电路的逆向设计中,所处理的图像物体一般都是由水平、垂直和45°等线段构成的。在这些处理中,最后都需要将提取的物体由图像表示转化成图形表示。一般方法是求出图像物体的边缘或骨骼线,将它们拟合成图形。图像物体的边缘及骨骼线都是由一系列有先后顺序的图像像素坐标数据链构成的,所以将其拟合成图形就相当于将平面上的一系列有先后顺序的数据点拟合成曲线的问题。由图像物体的特征,可以考虑用一些简单的曲线(如线段、圆弧等)来近似表示这些数据点。其中最简单的就是分段线性表示,即将数据点分成若干部分,用线段来表示其中的每一部分。本文提出的算法就是分段地线性拟合平面上的一系列数据点。

Tomex<sup>[1]</sup>,吴立德<sup>[2]</sup>,Hakimi<sup>[3]</sup>和 Hiroshi<sup>[4]</sup>等人都分别提出了用分段线性表示平面中的一系列点的算法。Tomex算法是搜索包含在两条平行切线间的最长曲线段。它的计算复杂度是线性的。其性能有时不理想,而改进时需要显著地增加成本。Hakimi的算法与Tomex的算法基本相似。吴立德基于统计模型的算法,其基本思想是先将数据点进行分组,然后用最小二乘法计算各组的拟合直线的斜率,将斜率发生剧烈变化处定为断点,由此将组合并为段,再计算各段的最小二乘方的直线拟合,即得所求的线。若要求连续,即在各断点处取平均以形成顶点,用这些顶点构成一个连续分段性的逼近。

吴立德的算法思想简单,算法复杂度为线性。但吴立德的算法在实际应用中存在下面几个问题:

(1) 吴立德的算法是建立在  $x_1 < x_2 < \dots < x_i < \dots < x_N$  的条件 ( $x_i$  为给定数据点的  $x$  坐标)之上的。对于闭合曲线需要按上述条件分段,然后进行拟合。对于图1的情况,则会出现  $dy/dx$  为  $\infty$  的情况。

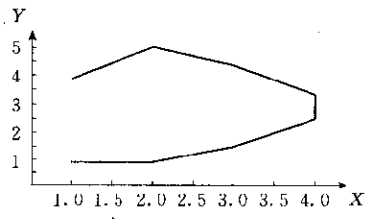


图1

(2) 吴立德的算法是用斜率值  $dy/dx$  求取分割点的。把通过点  $(x, y)$  的曲线切线与  $X$  轴的交角记为  $\alpha$ ,  $\alpha$  的

\* 作者王明江,1968年生,博士生,主要研究领域为图像理解,集成电路逆向设计系统的开发。唐璞山,1934年生,教授,博士生导师,主要研究领域为集成电路布局,布线。

本文通讯联系人:王明江,上海 200433,复旦大学电子工程系 CAD室

本文 1997-11-11 收到原稿,1998-03-03 收到修改稿

变化准确地反映了曲线的变化,当  $\alpha$  较小时,  $dy/dx$  与  $\alpha$  的变化基本成线性,当  $\alpha$  接近  $90^\circ$  时,角度的很小变化会带来  $dy/dx$  的很大变化,故在整个斜率变化范围内,算法的一致性是很差的.

本文提出的基于矢量斜率的分段线性拟合算法,有效地解决了吴立德的算法存在的问题.算法思想简单,可适用于任意图形.

### 1 矢量斜率的定义

我们知道,直线斜率是没有方向信息的,它只是直线与 X 轴交角的正切值,其微分定义为  $dy/dx$ ,取值范围为  $(-\infty \sim +\infty)$ ,直线斜率与角度不是一一对应的,以上这些为计算机程序处理带来一些不便.本文所定义的矢量斜率考虑了直线斜率的特点,是对直线斜率定义的修正和补充.下面给出矢量斜率的定义.

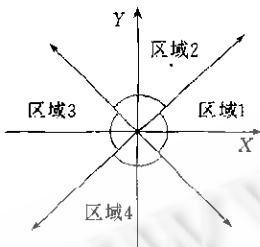


图2 矢量斜率定义区间的划分

参考图 2,我们将平面分割成 4 个区域,在每个区域中定义矢量斜率如下:

在区域 1 中,角度范围  $(-45^\circ \sim 45^\circ)$ ,  $|dy/dx| \leq 1$ ,矢量的斜率值定义为  $(K * 8 + C * (dy/dx))$ ;

在区域 2 中,角度范围  $(45^\circ \sim 135^\circ)$ ,  $|dy/dx| > 1$ ,矢量的斜率值定义为  $(K * 8 + 2 + C * (dx/dy))$ ;

在区域 3 中,角度范围  $(135^\circ \sim 225^\circ)$ ,  $|dy/dx| \leq 1$ ,矢量的斜率值定义为  $(K * 8 + 6 + C * (dy/dx))$ ;

在区域 4 中,角度范围  $(225^\circ \sim 315^\circ)$ ,  $|dy/dx| > 1$ ,矢量的斜率值定义为  $(K * 8 + 6 + C * (dx/dy))$ .

$(K * 8 + 6 + C * (dx/dy))$ .

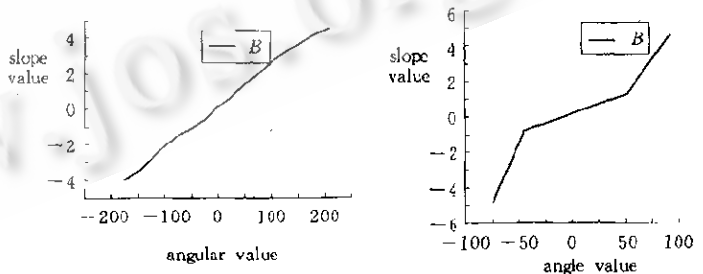
$K, C$  为系数,  $K$  可以取任意整数,  $C$  在 1、3 区域中取值 1,在 2、4 区域中取值 -1.可将矢量斜率定义写成统一的形式:

$$K * 8 + A + C * B, \tag{1}$$

$K, C$  取值与上述相同.  $A$  在区域 1-4 中分别取 0、2、4、6.  $B$  在区域 1、3 中为  $dy/dx$ ,在区域 2、4 中为  $dx/dy$ .

与角度  $(-180^\circ \sim +180^\circ)$  区间定义为主值区间类似,定义  $(-4 \sim +4)$  为矢量斜率的主值区间.

由上述定义可知,矢量斜率值  $V$  与  $V + K * 8$  对应于同一角度值,定义  $V$  与  $V + K * 8$  是等价的,且在本文算法中可进行任意替换,主值区间的矢量斜率值与主值区间的角度是一一对应的,当主值区间的矢量斜率值确定后,对应的主值区间的角度就确定了,从而方向也确定了.另外,当  $|dy/dx| > 1$  时,取  $dx/dy$  值来定义矢量斜率的  $B$  值,从而避免了取值趋向无穷的情况.图 3(a)为矢量斜率与角度的关系,图 3(b)为直线斜率与角度的关系.从图 3 可以看到,矢量斜率与角度(度)的线性度比直线斜率好得多,这为本文算法的计算机处理的一致性和精度的提高带来很大好处.



(a) 矢量斜率与角度的关系曲线 (b) 直线斜率与角度的关系曲线

图3 矢量斜率、直线斜率与角度关系的对比

虽然上述定义在数学形式上比直线斜率定义复杂些,但是由于矢

量斜率的处理主要是由计算机来完成的,所以上述定义并没有带来多大影响,并且由于矢量斜率主值区间为  $(-4 \sim +4)$ ,避免了溢出的发生,所以,从另一方面也简化了程序的处理.

### 2 平面数据点分段线性拟合的算法步骤

本节用矢量斜率来研究平面数据点的分段线性拟合的问题.平面数据点都是按一定的顺序记录和存储的,对于数据中的各点,可以从其前面和后面各取一定数目的点组成一个邻域,用最小二乘法求得数据点矢量斜率

的统计值,取当前点前面数据点的重心位置以及后面数据点的重心位置.由前面重心位置和后面的重心位置可求得数据点局部发展方向,由这一方向和矢量斜率的统计值即可求得当前点的矢量斜率值.求出每个点的斜率值后,将斜率值接近的点聚合到一起,拟合成线段,将线段连接起来就构成了数据点的分段线性表示.下面我们给出具体的算法步骤.

步骤 1. 求各数据点的矢量斜率值.选取一个奇数  $M_1 = 2 \times m + 1 \leq N$ , 对于数据链中的每一个点,抽取其前面的  $m$  个点、后面的  $m$  个点和当前点组成一个邻域集合,记为  $SY_i$ .

$$SY_i = \{(x_{i+j}, y_{i+j}), -m \leq j \leq m\}, \quad 1 \leq i < N, \quad (2)$$

求取下面各式的值

$$f_{xx} = \sum_{j=-m}^m (x_{i+j} - x_i)^2; \quad f_{yy} = \sum_{j=-m}^m (y_{i+j} - y_i)^2;$$

if  $f_{xx} \geq f_{yy}$

$$B_{iy} = \sum_{j=-m}^m j * y_{i+j} / f_{xx}$$

$$x_b = \sum_{j=-m}^i x_j / (m+1); \quad x_e = \sum_{j=i}^{i+m} x_j / (m+1);$$

if  $x_e > x_b$

矢量在区域 1, 其矢量斜率值为  $B_{iy}$ .

else

矢量在区域 3, 其矢量斜率值为  $4 + B_{iy}$ .

else

$$B_{ix} = \sum_{j=-m}^m j * x_{i+j} / f_{yy};$$

$$y_b = \sum_{j=-m}^i y_j / (m+1); \quad y_e = \sum_{j=i}^{i+m} y_j / (m+1);$$

if  $y_e > y_b$

矢量在区域 2, 其矢量斜率值为  $2 - B_{ix}$ .

else

矢量在区域 4, 其矢量斜率值为  $4 - B_{ix}$ .

由矢量斜率定义,通过对求得的矢量斜率值加 8 或减 8 的操作,将其值调整到主值区间内.实际上,  $B_{iy}$  为矢量斜率值定义中的 1、3 区域中的  $B$  值,  $B_{ix}$  为 2、4 区域中的  $B$  值.  $(x_b, y_b)$  为前面的  $(m+1)$  个点的重心位置,  $(x_e, y_e)$  为后面的  $(m+1)$  个点的重心位置.对于数据开始和结尾处的一些点所形成的  $SY_i$  集合,其点的数目小于  $M_1$ , 可不考虑这些点.将  $B_i$  加入到数据链,得到一个新的数据链.

$$S = \{(x_i, y_i, B_i), \quad m \leq i < N - m\}. \quad (3)$$

步骤 2. 根据预先给定的偏差  $E$ , 选取一个门限值  $T > 0$ , 对数据链式(3)中每一个点可求出由其邻近点所形成的子集  $S_j$ , 在这个最大子集中每一个点都满足以下两个条件:

$$(1) |B_j - B_i| < T, \quad B_j \in S_j, \quad j \neq i;$$

(2) 子集中的每一点与当前点相连.

找到数据链式(3)中的最大邻域子集,记为  $S_1$ , 并将其所包含的数据点从数据链中分割出去,然后在剩余的数据链中继续寻找最大的邻域子集,记为  $S_2$ , 将其包含的数据从数据链中分割出去,重复以上操作,直到数据链中的所有数据都被分割出去.这样,我们就获得了一个新的集合

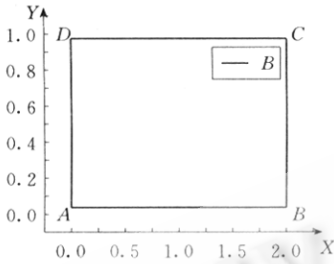
$$\{S_1, S_2, S_3, \dots\} \text{ 且 } S_1 \cup S_2 \cup S_3 \cup \dots = S; \quad S_i \cap S_j = \emptyset, \quad j \neq i. \quad (4)$$

步骤 3. 将链表(2)中的各数据段按路径  $S$  的顺序重新进行排序,删掉点数小于  $3 \times m$  的数据段,对于每个数据段,头、尾各去掉  $m$  个数据.然后对每一个数据段对  $(x, y)$  坐标点用最小二乘法进行线性拟合.将拟合所得的直线按每对相邻直线相交,求得顶点,用起点坐标和终点坐标对相应直线作垂直线,求得交点作为线段的两个端点,将端点和顶点连接起来就构成了分段线性拟合.

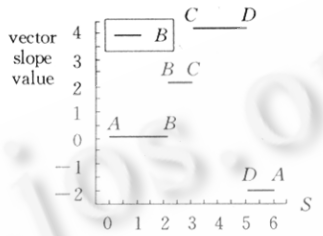
### 3 算法分析

本文算法的基本思想就是通过曲线的斜率值的变化将数据分割成数据段,对每段数据用最小二乘法做( $x, y$ )的线性拟合,连接所得的线段就得到了最后的结果.算法思想简单,容易实现.下面就几种图形来对算法的特点作一些分析.

图 4(a)为以矩形图形形成的函数,图 4(b)为矩形函数各段的矢量斜率与路径  $S$  的关系曲线.从此例可以看到,本算法可应用于传统斜率为零的场合.



(a) 函数1(矩形图形)



(b) 矩形各边的矢量斜率与路径  $S$  的关系曲线(方向 ABCDA)

图4 矩形图形的拟合

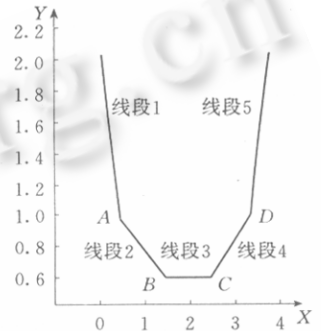


图5 函数2(方向 ABCD)

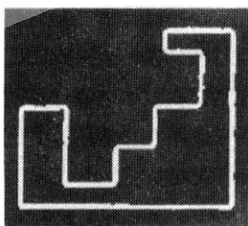
下面使用图 5 所示的函数来分析算法的精度.对图 5 的函数作了噪声水平为 0.01 的实验来比较本文算法与吴立德的算法<sup>[2]</sup>的精度情况.取  $N=200$  个数据点,对给定的数据点( $x_i, y_i$ ),加入干扰  $\epsilon_i$ ,其中  $\{\epsilon_i, 0 \leq i \leq N\}$  是独立的  $N(0, \delta^2)$  分布的随机变量,其中干扰是按直线法方向方向加入的.即  $x'_i = x_i + \epsilon_i * \cos\theta, y'_i = y_i + \epsilon_i * \sin\theta$ .其中  $(x'_i, y'_i)$  为加入干扰后的数据点; $\theta$  为法线与  $x$  轴的交角.取  $\delta=0.01$ ,分别用吴立德的算法和本文提出的算法进行拟合求取顶点,结果如表 1 所示.从表 1 可以看到,本文的算法较之吴立德的算法,精度有很大的改进.

表 1

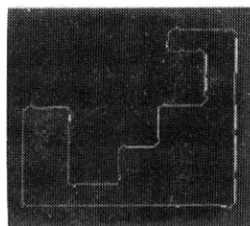
真实值	$x$	0	0.382 6	1.306 4	2.306 4	3.230 2	3.610 8
	$y$	2	1.076 2	0.693 6	0.693 6	1.076 2	2
吴立德的算法 $\delta=0.01$	$x$	0	0.345 6	1.406 4	2.281 3	3.253 6	3.614 1
	$y$	2.045 1	1.159 0	0.681 5	0.680 0	1.101 2	1.973 9
本文的算法 $\delta=0.01$	$x$	0	0.379 1	1.317 9	2.291 4	3.225 0	3.614 1
	$y$	2.003 9	1.071 4	0.695 9	0.687 1	1.068 5	2.004 8

从以上分析中看到,本文算法有以下特点:(1) 使用的斜率值范围是  $(-4 \sim +4)$ . (2) 斜率值与角度  $\alpha$  基本成线性关系. (3) 可应用于任意图形. (4) 算法精度较好.

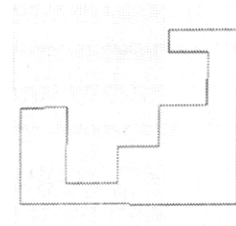
最后给出一个具体实例.它是在开发集成电路图像版图提取系统过程中遇到的一个有源区边缘图像.图 6(a)是一个分离出来的二值有源区边缘图像,图 6(b)是其细化后的图像,图 6(c)为拟合后的多边形图形.



(a) 分离出来的多边形图像



(b) 细化后的图像图形



(c) 拟合图形

图 6 实例

#### 4 参数选取及算法复杂度分析

算法中有两个参数  $M$  和  $T^{[2]}$ , 它们选择的好坏直接影响算法的精度, 非常重要. 参数  $M$  的选取与噪声干扰和运算速度有关.  $M$  选取较大, 在一定程度上可抑制噪声, 但将花费更多的机时; 同时,  $M$  选取较大, 会抑制拐角处点斜率的变化.  $M$  选取较小, 运算速度可提高, 但对噪声的抑制将降低. 使用时可根据具体情况来设定. 一般可在 2、3、4、5 等较小的值中选取.  $T$  的选取与最后的拟合精度有很大的关系.  $T$  选取得小, 精度可提高, 但线段的数目将增加; 而  $T$  较大, 线段的数目虽然可以减少, 但精度却会下降. 所以,  $T$  值的选择可视具体情况来决定.

本算法采用最大邻域聚类的方法进行分簇, 提高了拟合精度, 下面分析本文算法的时间复杂度. 设  $N$  为总的点数, 由第 2 节算法步骤可求得, 步骤 1 的运算复杂度为  $C_1N$ ,  $C_1$  是与  $M$  有关的常数. 在步骤 2 中, 当计算当前点的最大邻域时, 可完成其相邻点的最大邻域的部分搜索. 例如, 假设  $S$  点与  $T$  点相邻, 当  $S$  点向  $T$  点方向搜索时, 它在这一方向所包含的最大邻域点除去  $T$  点之外, 均包含在  $T$  点在这个方向的最大邻域中, 当搜索  $T$  点在这个方向的邻域时, 只需在前一次基础上搜索即可. 这样, 时间复杂度得到大幅度下降. 其时间复杂度可近似为  $C_2N$ ,  $C_2$  为常数. 步骤 3 中的排序和删除可同时进行. 即先找到一个最大集, 然后将剩下点中包含于最大集中的点全部删除, 然后对剩下的点重复上述操作. 其时间复杂度为  $C_3N$ ,  $C_3$  与线段个数有关, 可看作常数. 故此, 总的时间复杂度为  $(C_1+C_2+C_3)N$ , 算法时间复杂度为线性.

#### 5 结论

本文介绍了一种基于矢量斜率和统计模型的平面数据点分段线性拟合算法. 它与以前此类算法相比, 算法的精度有很大的提高, 可适用于任意图形, 算法思想简单, 时间复杂度为线性, 且程序容易实现. 这一算法已被用于我们所设计的集成电路图像版图计算机辅助识别系统中, 并取得了满意的效果.

#### 参考文献

- 1 Arelli C, Massarotti A. On the parallel generation of straight digital lines. *Computer Vision, Graphics and Image Processing*, 1978, 7(1): 67~83
- 2 吴立德. 计算机视觉. 上海: 复旦大学出版社, 1993  
(Wu Li-de. *Computer Vision*. Shanghai: Fudan University Press, 1993)
- 3 Hakimi S L. Fitting polygonal functions to a set of points in the plane. *Graphical Models and Image Processing*, 1991, 53(2): 132~136
- 4 Hiroshi Aoyama, Masahiro Kawagoe. A piecewise linear approximation method preserving visual feature points of original figures. *Graphical Models and Image Processing*, Sept. 1991, 53(5): 435~446

### A Piecewise Linear Approximation Based on Vector Slope

WANG Ming-jiang TANG Pu-shan

(Department of Electronic and Engineering Fudan University Shanghai 200433)

**Abstract** A piecewise linear approximation based on vector slope is presented in this paper. Given a set of planar data points, generally, these points have sequence. The vector slope definition is introduced at first in this paper. The slope of every points in set  $G$  is computed. The points cluster in a group, where vector slope of these points are very near. Fitting every group to a line and linking these lines, a fitting curve can be attained. The vector slope definition has the message of size and direction, basic interval is  $(-4 \sim 4)$ . The varing of vector slope in basic interval with angle  $(-180^\circ \sim 180^\circ)$  is in one-to-one correspondence and the relation curve has good linearity. A piecewise linear approximation based on conventional slope has some problems that the slope value is infinite in some cases and the relation curve with angle has bad linearity. All these very effect the quality of fitting curve and confine the application range. This algorithm overcome these problems, so the authors can obtain a high quality fitting curve and the algorithm can be used in every graph. The time complexity is linear.

**Key words** Feature extraction, piecewise linear approximation, shape analysis, contour analysis, edge detection.