一种快速的扫描铅笔稿图细化算法

彭京亮
(北京大学计算机科学技术研究所文字信息处理技术国家重点实验室, 北京 100871)

摘 要 二维动画计算机辅助制作系统中, 在对扫描铅笔稿图进行矢量化后, 可以大大提高描线上色的效率, 并且能够提取线条和闭包, 以便于自动上色和中间帧生成等更高级功能的实现. 扫描铅笔稿图的细化是矢量化的第一步, 在比较了大量现存的图象细化算法后, 提出了一种改进的非迭代线跟踪细化算法. 该算法效率较高, 只需对图象进行一遍扫描加两遍轮廓跟踪, 就能较好地满足二维动画计算机辅助制作系统对细化效率的要求. 该文还对算法的时间复杂性进行了详尽分析, 并提出对于两类图象噪声和冗余分支的消除方法.

关键词 矢量化 细化 线跟踪 骨架
中图法分类号: TP391 **文献标识码**: A **文章编号**: 1006-8961(2000)05-0434-06

An Efficient Algorithm of Thinning Scanned Pencil Drawings

PENG Jing-liang
(*State Key Laboratory for Text Processing, Institute of Computer Science and Technology, Peking University, Beijing* 100871)

Abstract In computer aided 2D cartoon producing systems, the vectorization of scanned pencil drawings can greatly enhance the efficiency of inking and painting, retrieve lines and closures contained in the drawings and therefore lay the basis for implementation of more advanced functions such as autocoloring, inbetweening and so on. Thinning of scanned pencil drawings is the first step of the vectorization process. Having investigated and compared a lot of thinning algorithms, an improved noniterative thinning algorithm based on the idea of line following is presented. The improved algorithm is very efficient, which, requiring only one pass of image scanning and two passes of contour tracing, meets the high efficiency standard in computer aided 2D cartoon producing systems. Time complexity of the algorithm is analyzed in detail, and the method to reduce two kinds of image noises and to delete redundant branches after thinning are given.

Keywords Vectorization, Thinning, Line following, Skeleton

0 引 言

二维动画辅助制作系统中矢量化的对象是扫描铅笔稿图, 由于扫描铅笔稿图一般是灰度图, 因此可以方便地进行黑白二值化. 本文所讨论的细化算法针对黑白二值图的.

关于细化(Thinning)一词的含义, 传统意义上仅指轮廓象素层层删除的算法, 而 Baruch 认为应统指所有的骨架提取算法[1]. 本文中采用 Baruch[1] 关于细化的概念. 关于细化处理方法前人已进行了大量工作, 如 1992 年 Lam, Lee 和 Suen 对各类细化算法曾作了较为详尽的综述[2]. 现存的细化算法大致分为迭代和非迭代算法两类.

在字符识别和工程图矢量化等领域, 细化的传统方法是将轮廓点层层剥除, 仅仅保留单象素宽的骨架线. 这种算法的优点是能较好地保持原图的细节; 缺点是速度慢, 一般需要多次迭代, 对噪声比较敏感, 且细化结果仍然是点阵图, 还需要对其跟踪以得到矢量表示.

大约从 70 年代末 80 年代初, 另一类细化算法开始快速发展. 此类算法不是基于轮廓象素的删除,

收稿日期: 1999-07-27; 改回日期: 1999-11-04

而是基于一定的假设, 而直接提取线条的骨架. 这一类算法的优点是不需要迭代, 因此速度快, 算法结果直接是矢量表示, 且噪敏性低; 缺点是细节保留不够好, 且局部骨架线与线条中轴的一致性可能不够好.

与字符识别、工程图和地图等的矢量化相比, 二维动画辅助制作系统中的矢量化有所不同, 主要表现在以下四方面:

(1) 效率

二维动画辅助制作系统对大量的扫描铅笔稿图经常是以批处理方式进行矢量化, 因而对算法的效率要求是第一位的.

(2) 自动性

由于铅笔稿图中的笔画比较随意, 不像工程图中的笔画那么规则, 也不像字符识别中易于分割, 因此要求细化算法适应性较好, 且能对整图进行自动细化.

(3) 交叉点位置的精确性

在字符识别等领域, 交叉点往往是识别的关键, 并对细化后交叉点位置的精确性要求比较高; 而二维动画辅助制作系统细化则不是主要面向笔画识别, 因此可以不必过分追求交叉点位置的精确性.

(4) 线宽恢复和后期编辑

二维动画辅助制作系统的细化算法要能够便于恢复原图的笔触(即线宽信息), 并且要便于后期的矢量编辑及闭包识别等.

1　线跟踪细化算法

1988 年, Baruch 提出一种细化思想, 即基于线跟踪的细化思想, 同时给出一种细化算法[1], 他认为线跟踪是人类进行图象细化的一种自然方式, 即同时跟踪线条的两侧轮廓, 而两轮廓的中心线即作为线条的骨架, 细化过程一遍完成, 无需迭代.

Chouinard 和 Plamondon 基于 Baruch 的线跟踪细化算法, 提出了一种改进算法[3]; Han 和 Fan 提出了利用两侧轮廓矢量配对来确定中心线的方法[4]; 陆宗骐和张秋萍提出了通过线条检测后进行轮廓跟踪的方法[5]. 所有这些算法都是通过轮廓跟踪决定中心线的方法. Han 和 Fan 以及陆宗骐和张秋萍的算法主要应用于笔画比较规整的工程图识别, 尚不能满足二维动画辅助制作系统对细化算法适应性好的要求. Chouinard 和 Plamondon 的算法(以下简称 C&P 算法)[3]用于字符识别, 且能动态改变跟踪窗口大小, 适应性较好.

C&P 算法首先是利用 Pavlidis 的算法[6]进行轮廓跟踪, 并用 O´Gorman 的算法[7,8]计算轮廓点曲率, 同时将曲率小, 即轮廓线在此转折较大的轮廓点标记为不连续点, 这种不连续点一般位于线端、笔划转折或笔划交汇处; 然后, 选取跟踪起始点, 首先使用方法 1, 即寻找每条轮廓线上曲率最小的点, 若其曲率小于一个阈值(C&P 算法取为 68 度), 则认为此点位于线端, 并将其前后两点作为起始点; 若这种方法失败采用方法 2, 即扫描整图, 即选取宽度最小处的两侧轮廓点作为起始点(图 1). 当图象中含有无线端连通子图(即此连通子图的任何轮廓点都不满足线端判定条件, 如环形) 时, 则要进行多次图象扫描.

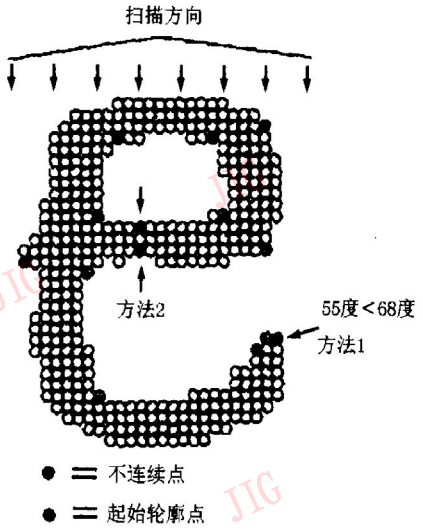


图 1　跟踪起始点的选取

轮廓跟踪是用两个轮廓指针 P_L 和 P_R, 从选定的起始点开始, 沿线的两侧同步移动. 每移动一步, 由 P_L 和 P_R 所指向的两个轮廓点形成一个包围窗口. 同时进行窗口分析, 以确定下一步的动作(图 2): 若窗口内包含线端(图 2a), 则停止当前跟踪; 若窗口包含交叉点(图 2b), 则递归跟踪每一个新的分支; 否则, 继续当前跟踪(图 2c). 然后将窗口的中心作为骨架点, 再连接相邻骨架点形成骨架.

在对图象进行细化的同时, 还对骨架进行分割,

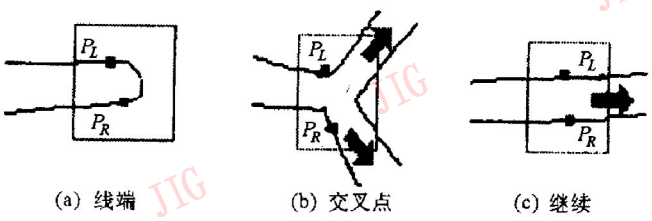


图 2　窗口内线条的不同情况

即对每条骨架段进行标记. 当遇到交叉点时, 若新分支与原分支共线, 则保留原标记, 否则为其分配一个新的标记(图 3). 骨架分割的结果可作为后期字符识别的基础.

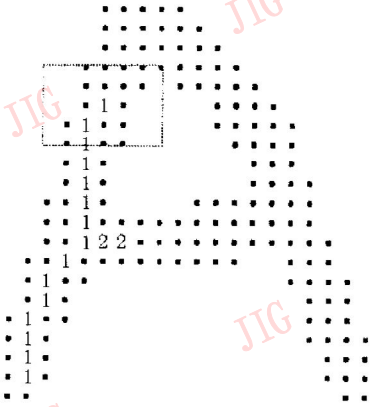


图 3 骨架分割

2 一遍扫描加两遍轮廓跟踪细化

C&P 算法为了选取所有的跟踪起始点, 可能要进行多次图象扫描, 还要进行一遍曲率计算. 一旦选定了跟踪起始点, 细化的过程实际上就是一遍轮廓跟踪的过程. 因此, 可以称 C&P 算法为多次图象扫描加曲率计算, 加轮廓跟踪的细化算法.

本文提出一种改进的线跟踪细化算法, 它沿用了 C&P 算法的窗口定义以及窗口移动和窗口分析的思想. 但对于起始点选取, 提出了新的方法. 该跟踪细化算法的优点: 第一, 只需一遍图象扫描和一遍轮廓跟踪, 而且无需计算曲率; 第二, 可通过轮廓标记的方法防止重复跟踪; 第三, 对交叉点细化采用一种简单可行的方法; 第四, 跟踪起始点选定后的细化过程也是一遍轮廓跟踪的过程. 所以, 可以称之为一遍图象扫描加两遍轮廓跟踪的细化算法.

该算法实现步骤如下:

(1) 跟踪起始点选取

从上到下, 由左到右扫描图象, 每遇到一个未作跟踪标记的轮廓点, 即将其作为一个跟踪起始点, 记入跟踪起始点队列, 并记下当前图象扫描位置; 再从此轮廓点出发, 用轮廓跟踪算法跟踪并标记一条轮廓; 然后从当前图象扫描位置开始继续扫描图象, 并重复上述过程. 这样, 经过一遍图象扫描, 同时进行了一遍轮廓跟踪, 图象中所有轮廓点都被标记, 且每条轮廓线的第一点均被作为跟踪起始点记入队列, 以后的细化就从队列中的跟踪起始点开始.

(2) 窗口定义

用 P_L 和 P_R 作为对角线的两个端点, 可以定义一个矩形. 令

$$D_X = |\overrightarrow{P_L}\,X - \overrightarrow{P_R}\,X|$$
$$D_Y = |\overrightarrow{P_L}\,Y - \overrightarrow{P_R}\,Y|$$
$$X_{\text{Inflation}} = \max\{D_Y, 2\}$$
$$Y_{\text{Inflation}} = \max\{D_X, 2\}$$

矩形的左右边分别外扩 $X_{\text{Inflation}}$, 上下边分别外扩 $Y_{\text{Inflation}}$, 形成包围窗口(图 4). 窗口大小随线宽动态改变, 因而降低了沿线噪声的影响.

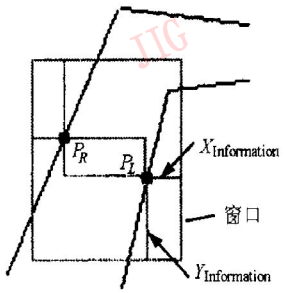


图 4 窗口定义

(3) 窗口移动

定义一个常数 $Pace$, 令指针 P_L 或 P_R 的每次移动最多沿轮廓前进 $Pace$ 个象素.

移动过程中要保持指针 P_L 和 P_R 的同步. 一般线条的转折处其两侧轮廓线长度不等, 为了适应这种情况, 即保持 P_L 和 P_R 的连线与铅笔线条在此处的法向一致, 首先试着将 P_L 和 P_R 各移动 $Pace$ 个象素点得 P'_L 和 P'_R(图 5), 首先计算如图 5 所示的三个平方距离

$$D_1 = (\overrightarrow{P_L}\,X - \overrightarrow{P'_R}\,X)^2 + (\overrightarrow{P_L}\,Y - \overrightarrow{P'_R}\,Y)^2$$
$$D_2 = (\overrightarrow{P'_L}\,X - \overrightarrow{P_R}\,X)^2 + (\overrightarrow{P'_L}\,Y - \overrightarrow{P_R}\,Y)^2$$
$$D_3 = (\overrightarrow{P'_L}\,X - \overrightarrow{P'_R}\,X)^2 + (\overrightarrow{P'_L}\,Y - \overrightarrow{P'_R}\,Y)^2$$

然后求三个距离的最小值, 即移动指针 P_L 和 P_R 到最小距离对应的两个端点(图 5). 在图 5 中,

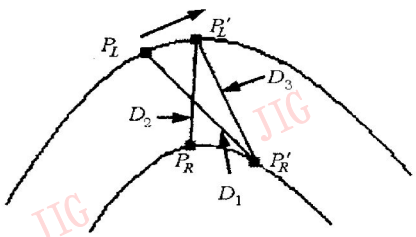


图 5 P_L 和 P_R 的同步

D_2 最小, 故 P_L 移动到 P'_L, P_R 不动.

(4) 窗口分析

窗口分析实际上是一个轮廓跟踪的过程, 即首先在当前窗口内, 用一个指针沿线条轮廓和窗口边, 从 P_R 指向的轮廓点, 逆时针移动到 P_L 指向的轮廓点, 然后统计其间与窗口边相遇的次数. 根据与窗口边相遇的次数, 决定窗口内部是否包含交叉点(图 6).

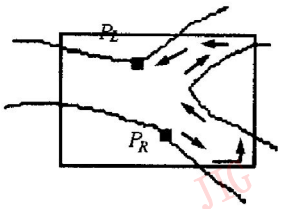


图 6　窗口分析

(5) 轮廓标记

为了防止重复跟踪, 应该标记或删除已跟踪过的部分.

Baruch 提出线条删除的方法[1], 而 Chouinard 和 Plamondon 没有明确地谈及这一问题[3]. 对于线跟踪算法, 笔者认为, 轮廓标记是一种比较自然, 而且高效的方法.

从一个起始标记号开始, 标记指针 PL 和 PR 扫过的轮廓点. 遇到交叉点时, 对交叉点处每一个新的分支分配一个新的标记号. 轮廓标记同时作为骨架标记.

(6) 跟踪终止条件

跟踪终止条件有二(图 7):

① 指针 P_L 和 P_R 相遇, 说明到了线端;

② 指针 P_L 和 P_R 都遇到已标记点, 说明回到某个交叉点处已经标记的分支已构成一个回路.

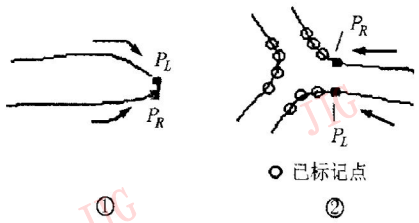


图 7　跟踪终止条件

(7) 交叉点细化

求窗口内每个轮廓片断的中间两个轮廓点, 然后将各中间轮廓点的几何中心作为交叉点. 再将各中间轮廓点分别作为相应分支的跟踪起始点, 并为每个新分支分配不同的标记号, 标记每个新分支的跟踪起始点. 然后从右到左依次递归细化每个新的分支(图 8).

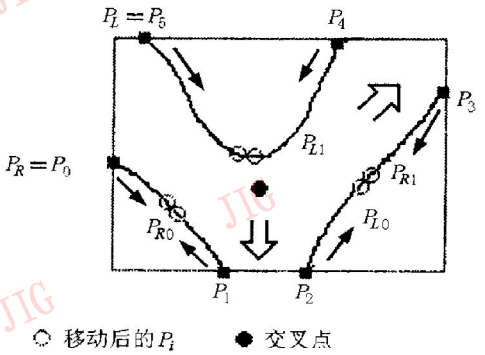


图 8　交叉点细化

3　时间复杂性

设图中轮廓象素点个数为 C, 线条部分面积为 A, 无线端连通子图个数为 M, 图象大小为 N. 以此来计算时间复杂性.

(1) 跟踪起始点选取

C&P 算法需在一遍图象扫描的同时进行轮廓跟踪和计算轮廓点的曲率, 并找出位于线端的跟踪起始点, 其时间代价为 $O(N+C)$. 为寻找无线端连通子图中的跟踪起始点, 还要经过 M 次图象扫描, 其时间代价为 $O(M\times N)$. 因此跟踪起始点选取的总代价为

$$O(K\times N+C),K=M+1$$

而新算法只需在对图象进行一遍图象扫描的同时, 对所有轮廓进行一遍跟踪. 其时间代价为

$$O(N+C)$$

(2) 窗口移动

两种算法中, P_L 和 P_R 都要沿着线条轮廓移动, 因此, 窗口移动的时间代价均为 $O(C)$.

(3) 窗口分析

两种算法都要对指针 P_L 和 P_R 的每一个位置进行窗口分析(图 9).

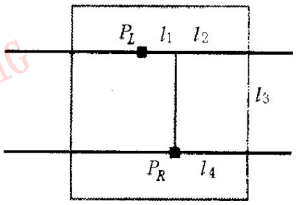


图 9　窗口分析代价

假设线条的总长度为 L, 平均宽度为 W, 轮廓指针每次最多前进 $Pace$ 个象素, 则跟踪过程中至少要形成 $L/Pace$ 个窗口. 每次窗口分析的轮廓时间跟踪代价为

$$WC = l_1 + l_2 + l_3 + l_4$$

由窗口生成方法及 P_L 和 P_R 的同步移动, 有

$$l_2 = l_3 = l_4 = D_Y \approx W$$

$$L_1 = D_X \approx Pace$$

$$WC \approx 3 \times W + Pace$$

所有窗口分析的时间代价为

$$TWC = L/Pace \times WC \approx 3 \times L \times W/Pace + L = 3 \times A/Pace + L = O(A)$$

(4) 总时间代价

C&P 算法:

$$O(K \times N + A + 2 \times C) = O(K \times N) + O(A) + O(C) = O(K \times N), K = M + 1$$

新算法:

$$O(N + A + 2 \times C) = O(N) + O(A) + O(C) = O(N)$$

由于在铅笔稿图中, 线条部分覆盖的象素点和轮廓象素点个数占整幅图象素点个数的比例很小, 算法的主要时间开销在于图象的扫描.

可见, 当图象中无线端连通子图数较多时, 新算法的效率提高比较明显. 目前 C&P 算法主要用于字符识别, 由于图象分割后的单个字符图象较小, 而且字符一般包含线端, 因而 C&P 算法的效率仍然较高; 但对于二维动画辅助制作系统, 扫描铅笔稿图往往比较大, 象素点个数可达几兆, 要对整幅图进行细化, 图象扫描次数的减少意义就显得尤为重大.

4 实验结果及分析

图 10(a) 和图 10(c) 是两幅大小为 591× 407 象素的扫描铅笔稿图, 图 10(b) 和图 10(d) 是利用本文提出的细化算法分别对图 10(a) 和图 10(c) 进行细化的结果. 表 1 中列出一些实验数据. 实验是在 Windows 95 平台和 MSV C++ 5.0 编译环境下进行的.

实验结果及数据表明:

(1) 当图中无线端连通子图(花被上的环形) 较少(图 10(a)) 时, 两种算法效率相近;

(2) 当图中无线端连通子图较多(图 10(c)) 时, 新算法的效率提高比较显著;

(3) 新算法中, 可能会在某些跟踪起始点附近产生冗余分支(如图 10(d) 中环形上的小毛刺).

表 1 实验数据

原图	细化时间(s)	
	新算法	C&P 算法
图 10(a)	0.220	0.257
图 10(c)	0.307	2.503

(a) (b) (c) (d)

图 10 实验结果

5 图象噪声和冗余分支的消除

本算法可以在选取跟踪起始点的同时完成两类图象噪声的消除, 这两类图象噪声我们称之为正噪声和负噪声. 正噪声指孤立的黑色噪声象素团; 负噪声指小的孔洞(图 11). 在选取跟踪起始点进行轮廓跟踪的同时, 统计每条轮廓的象素个数, 若一条轮廓的象素个数小于一个阈值, 则认为其包围的区域为一个噪声. 将轮廓象素点作为多边形顶点, 用多边形填充算法将此轮廓内部的象素取反, 便可消除两类噪/ 声, 即将正噪声删除, 负噪声填充.

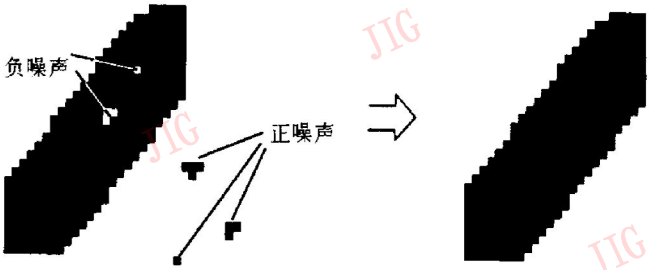
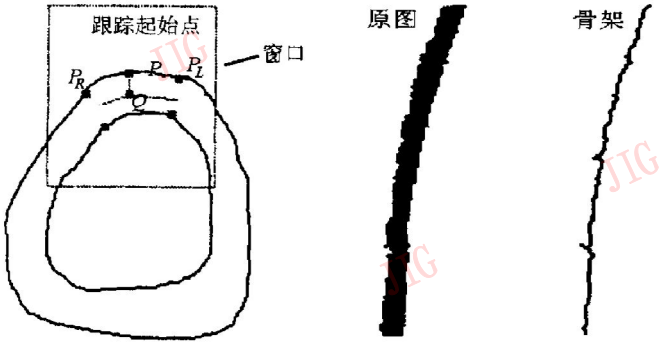


图 11 正负噪声的消除

由于新算法选取每个轮廓中扫描遇到的第一点作为跟踪起始点, 当此点不是线端时, 可能会在开始几步内细化出一个冗余分支(如图 12(a) 中的分支 P_Q); 此外, 由于轮廓噪声的影响, 也可能出现冗余

分支(如图 12(b) 中骨架线上的毛刺). 根据冗余分支长度较小, 且含骨架点数较少的特点, 因此可在细化完成后, 对骨架中的每条分支进行判断, 若此分支首尾两点距离及所包含骨架点数各小于某一阈值, 则此点为冗余分支, 应将其删除.



(a) 起始冗余分支　　(b) 轮廓噪声产生的冗余分支

图 12　冗余分支的消除

6　结　论

(1) 新算法效率较 C&P 算法有较大提高, 效率提高的关键在于减少了图象扫描的次数. 在图象较大, 且含无线端连通子图时, 效率提高尤为明显. 另外, 新算法无需进行曲率计算, 进一步提高了算法效率.

(2) 新算法可以较容易地删除孤立噪声和轮廓噪声.

(3) 新的细化算法可能会产生起始冗余分支, 而且细化后交叉点位置的精确度没有 C&P 算法好. 但起始冗余分支可以很容易地消除, 而且交叉点位置的精确度对于整幅图象矢量化效果的影响十分微小, 可不作为关注的重点.

(4) 将 P_L 和 P_R 所指轮廓点间的距离作为对应骨架点处的线条宽度记录下来, 可以很容易地实现线宽恢复.

(5) 细化后的骨架片断被编号, 可以作为后期闭包识别的基础.

因此, 本文提出的细化算法能较好地满足二维动画辅助制作系统对细化算法适应性好、自动、高效的要求, 便于线宽恢复和闭包提取, 进而便于实现矢量编辑、自动上色、中间帧生成等更高级的功能.

致　谢　工作过程中, 得到郭宗明博士和梁训东博士后的悉心指导, 笔者表示衷心的感谢.

参 考 文 献

1　Baruch O. Line thinning by line following. Pattern Recognition Letters, 1988, 8: 271~ 276.

2　Lam L, Lee S W, Suen C Y. Thinning methodologies-a comprehensive survey. IEEE Trans. Pattern Analysis and Machine Intelligence, 1992, 14(9): 869~ 885.

3　Chouinard C, Plamondon R. Thinning and segmenting handwritten characters by line following. Machine Vision and Applications, 1992, 5: 185~ 197.

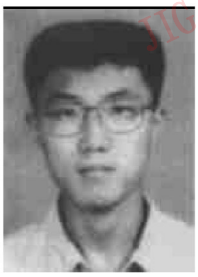
4　Han C C, Fan K C. Skeleton generation of engineering drawings via contour matching. Pattern Recognition, 1994, 27(2): 261~ 275.

5　陆宗骐, 张秋萍. 工程图纸矢量化中的线条轮廓跟踪法. 中国图象图形学报, 1997, 2(12): 878~ 882.

6　Pavlidis T. Algorithms for graphics and image processing. Computer Science Press, 1982, 142~ 148.

7　O′Gorman L. An analysis of feature detectability from curvature estimation. In: Proceedings of Computer Vision and Pattern Recognition Conference. Ann Harbor, 1988, 235~ 240.

8　O′Gorman L. Curvilinear feature detection from curvature estimation. In: Proceedings of the 9th International Conference on Pattern Recognition. Rome, 1988, 1116~ 1119.



彭京亮　1974 年生, 1997 年 7 月毕业于北京大学计算机系软件专业, 现为北京大学计算机系应用技术专业硕士研究生. 主要从事计算机辅助二维动画制作系统中矢量化技术以及视频检索中视频分析技术的研究.