

基于距离变换的多尺度连通骨架算法

丁 颐, 刘文予, 郑宇化

(华中科技大学 电子与信息工程系宽带无线与多媒体系统研究中心, 湖北 武汉 430074)

摘要:传统的基于距离变换的骨架算法不能保证骨架的连通性,需要引入鞍点解决连通问题.该类算法复杂,且不够准确,同时鞍点的定义很难推广到三维,限制了传统算法的发展.本文提出一种新型骨架算法,在图形内根据距离变换的约束,由骨架种子点开始以单像素宽度逐点生长出各骨架分支,逐点生长保证了连通性.实验证明该算法的骨架生长过程是骨架由粗到精的演变过程,能够方便地实现骨架的多尺度控制.

关键词:距离变换;骨架;多尺度;最大圆

中图分类号:TP391 **文献标识码:**A

HIERARCHICAL CONNECTED SKELETONIZATION ALGORITHM BASED ON DISTANCE TRANSFORM

DING Yi, LIU Wen-Yu, ZHENG Yu-Hua

(Department of Electronics & Information Engineering, Huazhong University of Sci. & Tech., Wuhan 430074, China)

Abstract: The traditional skeletonization algorithms based on distance transform can not guarantee the connectivity property, so saddle points should be added to solve the connectivity problem. However, these methods are complex and inaccurate, and saddle points can hardly be extended to 3D case. In this paper a novel method was presented, the skeleton obtained by growing from the skeleton seed with 1 pixel width restricted by distance transform, and the connectivity was assured by the growing process. Experiments show that the growing process is a skeleton evolution from coarse to delicate, and the hierarchical skeleton can be easily achieved.

Key words: distance transform; skeleton; hierarchical; maximal disks

引言

1967 年 Blum 通过烧草模型 (Grassfire Model) 首先给出了骨架的定义^[1],他假设图形边界点同时着火,火源向图形内部各个方向等速燃烧直至熄灭,所有熄灭点的点集构成了该图形的骨架. Blum 还给出了骨架的另一种等效定义^[2],即骨架是所有最大圆 (Maximal Disks) 的圆心集合,最大圆是完全包含在图形内部的圆,并且不被任何其它包含在图形内部的圆所包含.

(1) 细化的方法. 该方法就是模拟烧草的模型,即逐层均匀的剥掉图形的边界,最后剩下最里层已经不能再剥掉 (否则会影响连通性) 的部分就构成了图形的骨架^[3-5].

(2) 基于距离变换的方法. 根据最大圆的定义

可以推导出最大圆必是图形的内切圆,因此考察以某点为圆心作的内切圆,如果该圆不被其它的任何内切圆包含,则此点作的内切圆是最大圆,此点是骨架点.

两种方法在连续域都是完备的,在离散域情况下则各有优缺点. 细化得到的骨架具有良好的拓扑不变性,但是骨架的位置却不准确. 其原因在于离散域里任何点的行进方向最多只能有 8 个 (二维),不能模拟火烧过程中任意的行进方向. 文献 [5] 引入 snake 模型调整骨架的位置,但是提高了复杂度. 基于距离变换的方法在骨架点的准确度上有明显的优势,但连通性却很难保证. 其原因在于离散域内最大圆定义和圆的包含关系很难把握,传统的最大圆判断准则得到的骨架不保证连通性^[6,7],需要另外引入一些鞍点 (Saddle Points) 将不连通的部分连通起

来. 近年来出现了一些改进的距离变换骨架算法^[8,9],但是这些方法主要针对于柱状物体,例如医学应用的肠道导航.

针对这两类算法的局限性,本文提出一种新型骨架算法,该算法利用距离变换信息从骨架种子点以单像素宽度生长出其余的骨架点,该过程保证了骨架的连通性及单像素性,并且可以根据给定权值控制生长精度,实现骨架的多尺度控制.

1 骨架算法

1.1 传统方法的缺点

传统的基于距离变换的方法考察图形内所有的点内切圆,但是如果每个内切圆都要与图形内所有点的内切圆进行比较,复杂度太高.传统的方法大都是比较每个内切圆与其邻域点的内切圆^[6,7],这样做得到的点集是骨架点集的父集,文献[7]对此做了改进,结果仍然不理想,骨架不能保证连通性.文献[7]分析了失去连通性的原因,认为该方法在连续域是完备的,但是应用在离散域,圆的定义(如图1)造成两圆包含关系的连续域判断准则不能直接挪用,这是失去连通性的根本原因.图1中(a)、(b)、(c)、(d)分别代表半径平方值是数字1、2、4、5的离散圆,(e)表示是一个H形的图表及它的骨架,(e)中标出了各像素点的距离变换平方值,图中深色点是骨架点,因为以它们为圆心的内切圆都不能被其它内切圆完全包含,由图可见该骨架是不连通的,且不是单像素宽的.

为了避免传统方法的上述缺陷,本文提出一种由骨架种子点向外生长的方法以保证骨架的连通性和单像素性,并且在生长的过程中利用距离变换信息,保证骨架的准确度.

该算法的步骤如下:

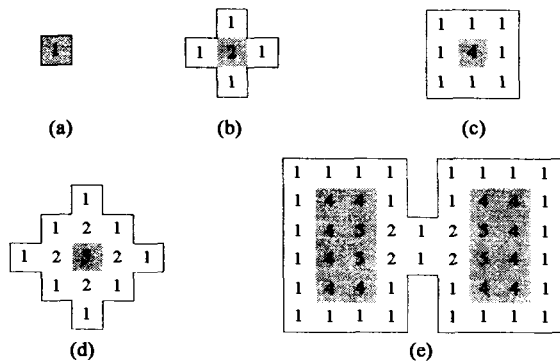


图1 传统的基于距离变换的骨架
Fig.1 The skeleton of the distance based algorithms

(1) 计算距离变换;

(2) 寻找骨架种子点. 该步的目标是找到一个合适的起始骨架点;

(3) 生长骨架. 该步考虑如何从一个已知的种子点经过反复迭代的过程找出其它所有的骨架点.

下面将对这3个步骤做具体的讨论.

1.2 计算距离变换

每个像素或体素的距离变换值是该点到图形边界的最短距离,任何基于距离变换的骨架算法都要首先计算距离变换.计算距离变换通常有两种方法,一种是基于模板的近似方法^[10],另一种是精确方法^[11].前者复杂度低,效率高,但是计算的结果有误差,后者计算结果精确,只是复杂度略高.本文选用文献[11]提出的计算结果精确的距离变换计算方法.

1.3 寻找骨架种子点

理论上骨架种子点的选择十分自由,任何一个骨架点都可以作为种子点.但是选择不同的种子点构造多尺度骨架的效果不同,理想效果是由骨架中间部位发散地各外生长,这样在开始获得较少骨架点的情况下,也能最大程度地表现图形的全局特征.骨架中间部位点的特征是距离变换值相对较大,因此本文选取图形内距离变换值最大的点作为种子点,可以证明该点的内切圆一定不会被其它内切圆所包含,所以其必定是骨架点,符合种子点的基本要求,并且位于骨架的中间部位能够生长出效果满意的多尺度骨架.

1.4 生长骨架

生长骨架的过程是一个迭代过程,本文定义上一轮迭代新产生的骨架点称为下一轮迭代的生长前沿点 f , 每轮迭代的任务就是在各生长前沿点的领域生长出一个或多个本轮的新骨架点.每轮迭代过程分以下3步进行:

(1) 覆盖. 覆盖这一步的任务是以生长前沿点为圆心,相应的距离变换值为半径,覆盖图形.被覆盖的图形部分将被挖去,每轮迭代的操作都会减小图形,直至图形的剩余部分为零,即找到的骨架点已足以覆盖整个图形,整个迭代过程结束.图2显示了上述的操作过程,其中(a)为原始图形,(b)为找到骨架种子点后的第一轮迭代,图中表示了由骨架种子点画圆覆盖后的剩余图形,(c)为第二轮迭代的覆盖结果,此时有3个生长前沿点,(d)为第六轮迭代的覆盖结果,(e)为第十一轮的覆盖结果,此时有4个生长前沿点,(f)为第十四轮的覆盖结果,此时剩余图形为零,迭代

过程结束,此图即骨架的最终结果.

(2) 判断新分支数. 每个生长前沿点都对应图形剩余部分的一个连通区域,经过覆盖后,该连通区域包含的点数将会减少,并且被拆分成了 n 个连通区, n 为大于等于 1 的整数,这 n 个连通区都是此前前沿点应该生长的方向,因此需要产生 n 个分支,继续分别覆盖这 n 个连通区. 如图 2 所示,最初种点对应的是整个图形的一个完整连通区,当第一轮执行覆盖后,如(b)所示,连通区分为 3 块,于是由(b)中的 f_1 应该生长出 3 个新的分支,如(c)所示,第二轮迭代时就已经有 3 个生长前沿点 f_1, f_2, f_3 代表的 3 个分支了,分别对应了图形剩余的 3 个连通区域.

(3) 生长新骨架点. 上一步判断了分支数,这一步考虑如何生长分支,即选择前沿点的哪些邻域作为本轮新骨架点 $f'_i (i=1 \sim n)$,之后开始新一轮的迭代.

选取新骨架点的准则是:选取 8 邻域中距离变换下降最缓慢的点作为骨架点. 需要说明的是,本文考虑的领域点指 8 邻域,背景为 4 连通,这样能反映更加准确平滑的效果.

新骨架点 f' 要求比前沿点 f 距离对应的连通区更近,以便下轮更多地覆盖连通区,在此基础上 f' 的距离变换值要求在局部上有较大值,以保证骨架位于图形的中央. 本算法首先找出 f 的 8 个邻域点中距离连通区最近的一个作为 f' 的初选结果,然后考察 f 的 8 邻域中与 f' 相邻的 2 个点,这 2 个点加上 f' 共覆盖了 f 的 90° 的邻域范围,由于已经确定了大致的方向,精确的新骨架点位置不会超出这个范围,然后计算 f' 及相邻两点相对于 f 的距离变换值的下降程度,其中下降最缓慢的一个是距离变换值局部最大值,选这个点作为 f' 的最终结果. 可以证明初选结果保证了新骨架点朝着未被覆盖的连通区生长,即明确了大体方向,而最终结果在局部上保证了所选择的新骨架点是位于图形中央的,满足骨架的中轴性质. 如图 3 所示,表示的是图 2(d) 第六轮迭代时右上方的分支的生长情况,此时的前沿点 A 坐标

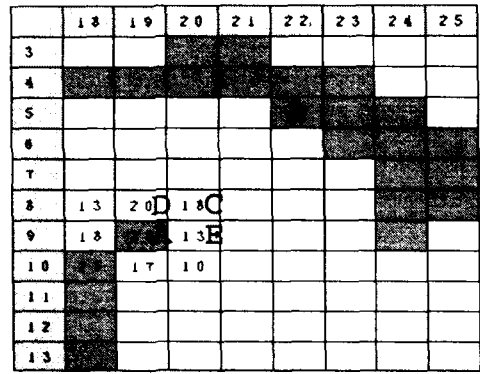


图 3 生长新的骨架点
Fig. 3 The growing process of new skeleton point

为(19,9),此点的距离变换值是 20,其 8 邻域的距离变换值在图中标示出来. 此点右上方的一块连通的深色区就是对应此点的剩余图形连通区,这部分剩余图形的平均位置在 B(22,5),在 A 的 8 个邻域点中,C 是距离 B 最近的点,因此 C 将作为新骨架点的初选点. 接下来考虑 A 的 8 个邻域点中,C 是距离 B 最近的点,因此 C 将作为新骨架点的初选点. 接下来考虑 A 的 8 个邻域中与 C 相邻的两个点 D 与 E,分别计算 C、D、E 三点相对于 A 的距离变换下降梯度, $g_d = (\sqrt{20} - \sqrt{20})/1$, $g_c = (\sqrt{20} - \sqrt{18})/\sqrt{2}$, $g_e = (\sqrt{20} - \sqrt{13})/1$,其中 g_d 值最小,说明 D 点相对于 C 和 E 更接近于图形的对称轴,我们由此选择 D 点作为本轮的新骨架点.

2 骨架的连通性和单像素性保证

传统算法各骨架点的判断都是独立的,不考虑骨架点之间的位置连通关系,因而不保证连通性. 本文算法由起始种子点经迭代逐点生长骨架,每轮新生长的骨架点都是上一轮骨架点的邻域,所以连通性一定能够得到保证.

对于单像素性,同样要考虑骨架点之间的关系. 本文算法每轮迭代过程中对应每个连通区生长一个分支点,各分支就是由这样的单点组成的,因此只要

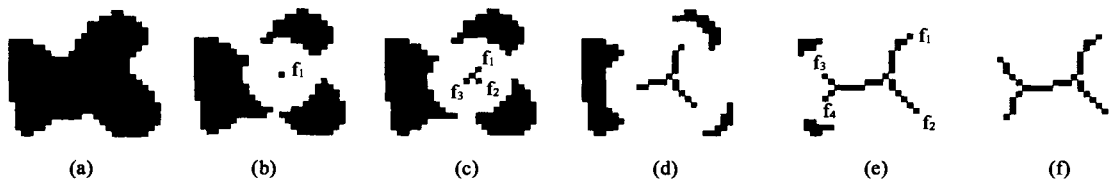


图 2 覆盖的过程
Fig. 2 The process of covering object

不同分支上的点不相邻,同一分支上的点不跳跃相邻,就能保证单像素性.图4说明两种失去单像素性的情况,前者两分支靠在一起,后者分支自身扭曲,针对这2种情况,本文提出了以下2条限制准则:

(1) 每轮迭代由同一个前沿点产生的新骨架点不能相邻

如图4(a)中, f'_1 和 f'_2 都是由 f 产生的,这种情况下, f'_1 与 f'_2 必须被强行间隔开,或者重叠在同一点.这样的限制能够很好地解决(a)中的问题,使得2个位置比较靠近的分支在离散图形中无法完全分离的情况下,先重叠在一起,等到分支距离逐渐扩大,大于2个像素后再分开.

(2) 新产生的骨架点只能与它的前沿骨架点相邻

如图4(b)所示,骨架在向左生长的过程中产生了上下曲回的扰动,这是应该禁止发生的情况.分析骨架点 s_5 ,可发现 s_5 与 s_4, s_3, s_2 都相邻,但只有 s_4 是 s_5 的前沿点,如果加上本条限制, s_5 就不能选择图(b)中的位置.排除此位置后, s_5 若选择到 s_4 左方或左上方的点,都将能表现出令人满意的结果.

实验结果表明,骨架的连通性效果理想,同时由于考虑了骨架点之间的位置关系,经过以上两条准则约束的骨架有效地实现了单像素性.

3 骨架的多尺度控制和识别信息的提取

本文算法十分有利于多尺度骨架的生成,同时也具有良好的抗边界噪声鲁棒性.

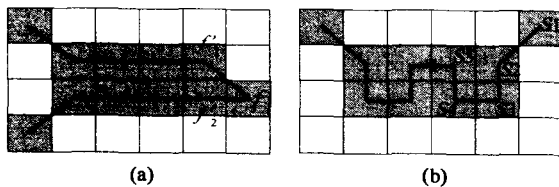


图4 影响单像素性的2种情况
Fig.4 Two cases where the 1-pixel width property lose

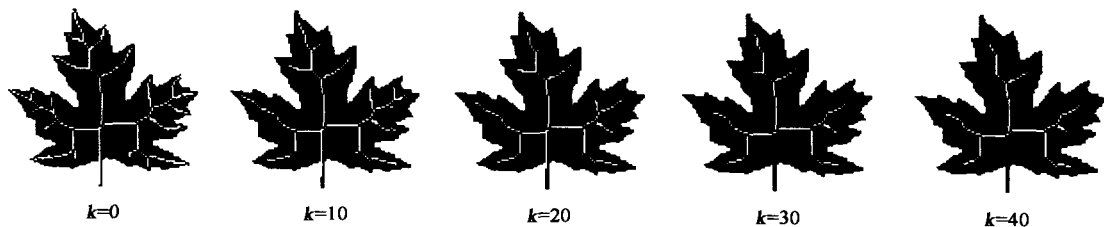


图5 枫叶的多尺度骨架
Fig.5 The hierarchical skeletons of a maple leaf

骨架生长过程的结束条件根据是否有剩余的未被覆盖的连通区决定,每个骨架点被选中时,都会对应一个连通区,而这个连通区范围越大就说明这个骨架点的权重越大,该权重可以作为我们实现多尺度控制的标准.我们规定连通区包含的点数必须大于给定权值 k ,才可以继续生长骨架,否则该骨架分支结束.权值 k 越小,骨架越精细;否则,骨架越粗糙.为了避免噪声的影响,选取一个合适的 k 值就能达到理想的效果了.图5中的图形大小是 100×100 ,从左到右 k 的取值是 0, 10, 20, 30 和 40.

骨架的多尺度控制可以为图形识别所用,实现图形由粗到精不同程度的匹配及分类.同时骨架的单像素及连通性,也有利于匹配算法的操作,除此之外,本算法还可提供下列附加的有利于识别的图形拓扑与形状信息.

(1) 骨架点的距离变换值.图形内所有点的距离变换值都预先计算,包括最后生成的各骨架点.距离变换信息涵盖了图形的形状信息,利用此信息可以实现图形的重建,在图形匹配过程中可以作为形状匹配的参考.

(2) 骨架点的分支数.骨架生长过程中计算了每点的分支数,分支数可以用来判断该骨架点是分叉点,普通点还是终结点,便于物体的描述与识别,在后续的研究中,我们利用骨架点的分支数,可以很方便的实现基于多尺度骨架的物体拓扑分类.

4 算法时间复杂度分析

本骨架算法首先计算距离变换,对于二维 $n \times n$ 图形,文献[11]计算精确距离变换的算法复杂度是 $O(n^3)$.

接下来寻找种子点,此步只需对图形遍历一次即可,计算次数是 n^2 ,复杂度是 $O(n^2)$ 级.

然后由种子点生长骨架,对于每个新骨架点执行三步过程.(1)覆盖过程是找到每个骨架点 i 时,处理它最大圆范围内的图形领域,给该领域内的点

标上覆盖的标示. 此过程的处理次数是 DT_i^2 , 复杂度是 $O(n^2)$. (2) 判断新分支数的过程需检查图形剩余连通区的标识, 更新标识. 本算法采用从连通区内某点深度遍历图形的方法, 处理次数小于图形点数, 复杂度可看为 $O(n^2)$. (3) 生长骨架点. 该过程计算骨架点的 8 邻域点哪一个与剩余连通区最近, 计算次数是与幅度 n 无关的常数. 三过程总的复杂度以其中复杂度最高的计, 为 $O(n^2)$ 级. 因为对于每个骨架点都要执行上述三个过程, 骨架点数是 $O(n)$ 级的, 所以生长骨架点的复杂度以 $O(n^2) \times O(n)$ 计算, 为 $O(n^3)$.

综上, 距离变换复杂度 $O(n^3)$, 寻找种子点复杂度 $O(n^2)$, 生长骨架复杂度 $O(n^3)$, 所以本算法总体复杂度为 $O(n^3)$.

5 实验结果及结论

图 6 给出本文方法得到的部分实验结果及与文献[7]方法(未经去除伪点及连接)的比较. 上一行是本文方法的结果, 下一行是文献[7]的结果. 可以看出本文的方法十分有效的解决了骨架的单像素及连通问题.

文献[7]首先计算最大圆圆心, 其快速算法的复杂度是 $O(n^2)$. 但是由于这样并不能保证连通性和单像素, 还需要经过细化算法去除多余的骨架点, 以及连接算法保证连通. 细化算法的复杂度是 $O(n^3)$, 因此将此算法的复杂度级别又提高到了 $O(n^3)$ 级, 与本算法同级. 相比之下, 本文算法的思路更加直接, 通过分析骨架的基本性质, 一次生长得到具备连通和单像素性的骨架.

本文算法可用于带孔图形, 只需对生长的过程

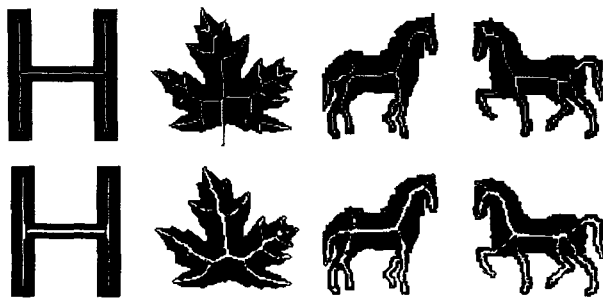


图 6 实验结果

Fig. 6 The experimental results

作细微的改进即可. 因为带孔图形骨架的分支数与剩余的连通区域数是不等的, 有可能几个分支对应一个剩余连通区, 此时新分支数不应对应剩余的连通个数, 而要对应新骨架点画最大圆时与剩余连通区相交的圆弧个数.

本文算法改进了传统的基于距离变换的方法, 解决了骨架连通性的问题, 且保证了单像素宽度, 实现了骨架的多尺度变换, 增加了骨架的附加信息, 这些附加信息对于后面的物体描述和分类非常有用. 相对于细化的方法, 本文的结果更准确. 该方法可以推广到三维, 每轮迭代用球代替圆来覆盖图形, 得到三维图形的线型而非面型三维多尺度骨架.

REFERENCES

- [1] Blum H. *A Transformation for Extracting New Descriptors of Models for the Perception of Speech and Visual Form* [M]. W. Walthen-Dunn, ed., US: MIT Press, Cambridge, 1967.
- [2] Blum H. Biological shape and visual science: Part I [J]. *J. Theoretical Biology*, 1973, **38**(2): 205—287.
- [3] Ma C M, Sonka M. A fully parallel 3D thinning algorithm and its applications [J]. *Computer Vision and Image Understanding*, 1996, **64**(3): 420—433.
- [4] Pudney C. Distance-ordered homotopic thinning: A skeletonization algorithm for 3D digital images [J]. *Computer Vision and Image Understanding*, 1998, **72**(3): 404—413.
- [5] CHE Wu-Jun, YANG Xun-Nina, WANG Guo-Zhao. A dynamic approach to skeletonization [J]. *Journal of Software* (车武军, 杨勋年, 汪国昭. 动态骨架算法. *软件学报*), 2003, **14**(4): 818—823.
- [6] Niblack C W, Gibbon P B, Capson D W. Generating skeletons and centerlines from the distance transform [J]. *CV-GIP: Graphical Models and Image Processing*, 1992, **54**(5): 420—437.
- [7] Ge Y, Fitzpatrick J M. On the generation of skeletons from discrete euclidean distance maps [J]. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 1996, **18**(11): 1055—1066.
- [8] Bitter I, Kaufman A E, Sato M. Penalized-distance volumetric skeleton algorithm [J]. *IEEE Trans. on Visualization and Computer Graphics*, 2001, **7**(3): 195—206.
- [9] Zhou Y, Toga A W. Efficient skeletonization of volumetric objects [J]. *IEEE Trans. on Visualization and Computer Graphics*, 1999, **5**(3): 196—209.
- [10] Svensson S, Borgefors G. Digital distance transform in 3D image using information from neighborhoods up to $5 \times 5 \times 5$ [J]. *Computer Vision and Image Understanding*, 2002, **88**: 24—53.
- [11] Saito T, Toriwaki J. New Algorithm for euclidean distance transformation of an n -dimensional digitized picture with applications [J]. *Pattern Recognition*, 1994, **27**(11): 1551—1565.