

基于 Voxel 编码的曲面重建

王明福¹ 周 勇²

(深圳职业技术学院软件工程系 深圳 518055)¹

(诺思罗普-格鲁曼信息系统公司网络通信部 美国加利福尼亚 90746)²

摘 要 一个完整的基于轮廓的曲面重建方法必须建立轮廓对应、解决分支和三角面片的构建。然而大多数已有的曲面重建算法只能解决问题的某些方面,从而导致这些算法不能有效地运用到复杂曲面重建,比如用磁共振获取的人大脑序列切片等盘旋且多分支凸包数据。提出了一个基于 Voxel(像素)编码技术的曲面重建算法,该算法能以一种全自动的方式处理带有空洞的复杂多分支曲面。首先将两相邻断层轮廓投影到定位于中间的一个辅助平面上,求得其差区域,然后根据差区域的不同情形进行分组。对每组轮廓,从对应的差邻域中提取骨架,并用骨架来度量两轮廓的不相似量,对不相似的进行剪支分解,从而使不相似的、复杂的轮廓转换为简单且相似的骨架轮廓,最后完成三角片构建。重建曲面由二维流体三角面片组成,且仅经过切片上的输入廓线。算法已用手工数据和复杂人脑皮层的磁共振数据进行了仿真测试,检验了算法的有效性。

关键词 轮廓,曲面重建,多分支,骨架,像素编码,轮廓骨架匹配

Voxel-coding for Surface Reconstruction from Contours

WANG Ming-fu¹ ZHOU Yong²

(Department of Software Engineering, Shenzhen Polytechnic, Shenzhen 518055, China)¹

(Northrop Grumman Information Systems, CA 90746, U. S. A.)²

Abstract A complete contour-based reconstruction method must establish correspondence, solve branching problems, and construct tiles. Most modern reconstruction algorithms typically address only one or two of these problems. Therefore, their applications do not achieve complete solutions with complicated objects, such as the considerably convoluted and highly branched cortex of the human brain extracted from Magnetic Resonance Imaging (MRI) data. This paper presented an efficient Voxel-coding algorithm; which can handle complicated many-to-many branching and holes in a fully automatic and systematic way. First the contours from adjacent slices are projected onto an intermediate plane. And then be divided into groups on the basis of their difference regions. For each group of contours, skeletons are extracted from the corresponding region. These skeletons are used to measure contour dissimilarity and to decompose dissimilar or complicated branching contours into simple and similar contour-to-skeleton pairs. Reconstructed surfaces are 2D manifold triangle meshes which pass only input contours along slices. The algorithm has been tested using both hand-made data and real complex human cortical MRI data, demonstrating its efficiency.

Keywords Contour, Surface reconstruction, Branching, Skeleton, Voxel-coding, Contour skeleton match

基于轮廓的三维物体的表面重建一直是国内外研究热点,并取得了一系列研究成果^[1,2,4,6,8,15,19,28]。由于有很多自动化或手工技术获得一系列能够提供物体截面轮廓及其内部结构完整信息的断层图像切片和数据,同时在有些应用领域,轮廓是物体唯一的有效信息,因此对基于轮廓的三维物体的表面重建的研究具有广泛的应用领域,尤其在医学工程方面。

然而,现有的重建技术只解决了简单分支问题^[1,2,17,19],不能有效地运用到复杂曲面,比如用磁共振获取的人大脑序列切片数据,从而限制了该技术的应用发展。为此,本文提出了一个基于 Voxel 编码技术的曲面重建算法,能处理带有空洞的多分支嵌套复杂曲面,对源于序列断层切片轮廓的曲面

重建提供了一个完整的解决方案。首先将两相邻断层轮廓投影到定位于中间的一个辅助平面上,求得其差区域,然后按差区域的不同情形进行分组。对每组轮廓在差区域中提取骨架,并用骨架来度量轮廓的不相似量。对不相似的进行剪支分解,使不相似的复杂轮廓转换为简单且相似的骨架轮廓。算法具有如下特点:

① Voxel 编码为解决对应、分支和三角面片的构建等难题提供了一个完整的解决方案,并以全自动的方式处理带有空洞的复杂多分支曲面。

② Voxel 编码依据在差区域中的空洞个数,可将复杂轮廓分为单空洞和多空洞情形。而不像传统方法根据轮廓数,

到稿日期:2008-10-28 返修日期:2009-01-14 本文受粤港关键领域重点突破项目(2005A11304003)和深圳市科技基金(05KJcd004)资助。

王明福(1956—),男,硕士,副教授,主要研究方向为计算机图形学、算法设计与分析、医学图像处理, E-mail: wmf@oa.szpt.net; 周 勇(1963—),男,博士,主要研究方向为计算机图形学、几何造型、计算机视觉、医学图像处理和应。

将其分为一对一、多对一和多对多。Voxel 编码可直接对单空洞情形进行三角化,并能将多空洞转化为单空洞。

③ Voxel 编码提供一个标准和方法,用来度量轮廓的不相似量,并将不相似的转化为相似的。

④ Voxel 编码能处理任意复杂情形,不需要对输入数据进行预处理。在对应和分支过程中,计算复杂度与轮廓采样无关。

⑤ 在将二维流体三角面片重建曲面时,只需要顺序输入物体断层轮廓。提供了一个可伸缩三角面片输出,在不同采样点不需要增加计算复杂性的前提下确保在低采样轮廓下的三角面片的简化。

文中第 1 节回顾了相关工作;第 2 节是基本概念和算法概述;第 3 节是 Voxel 编码技术;第 4 节是算法执行,包括轮廓的分组对应(4.1 节)、D-区域的估计(4.2 节)、骨架提取(4.3 节)、轮廓与骨架的匹配(4.4 节)和轮廓与骨架的三角化(4.5 节);第 5 节是算法仿真实验;最后是结论和进一步工作。

1 相关工作

一个完整的基于轮廓的曲面重建方法必须建立轮廓对应、解决分支和三角面片的构建。对两相邻序列切片轮廓,对应首先确定哪个轮廓被连接到相邻断层中的哪条轮廓,对应关系建立后,分支则决定该切片上的轮廓如何被连接到另一切片上的一个或多个轮廓。通常需把轮廓分成段,建立段对间的关联,最后从轮廓或轮廓段对构筑三角面片,使得三角面片之顶点跨越在两轮廓线上,而内部边由两三角形公共。

早期基于轮廓的曲面重建只限于一对一的轮廓上建立三角面片,即每个断层上只有一条轮廓^[8,15]。这些研究分为两类:优化和启发式。优化方法通常是将三角面片的构建转化为图的搜索;轮廓的三角化等价于在一定的准则下从图中寻找最优的路径。Keppel^[15]和 Fuchs^[8]是最早的框架开拓者,Keppel 利用最大体素标准,而 Fuchs 使用最小面积标准。随后 Wang 等人提出了弛松优化^[22]、Kehtarnavaz 等人的最小代价路径^[14]、Shinagawa 和 Kunii 的花蕊同长模型^[21],以及 Hong 的最小面积准则^[12]。启发式(Heuristic)方法则应用局部优化规则,主要基于不同断层轮廓上点间的局部距离。Christiansen 和 Sederberg^[6]首先提出最小距离方法,他人的启发式方法则是利用最小边长^[7]、曲面最小张量准则^[9]。

建立对应的普通方法是估计两相邻断层轮廓的重叠区域,把重叠超出给定阈值的轮廓划分为同一组^[9,7,22],对重叠的估计通常采用逼近方法。当轮廓的采样太稀疏,这些方法将导致失真。对此,Meyers 等人^[16]提出了最小生存树,将每一条轮廓与邻接断层上的所有轮廓相连构成树,使距离代价与边相关联。

对分支问题,大多数解决方案是通过建立组合轮廓,将多分支轮廓转化为一对一。组合轮廓是通过在靠近分支点引入一个新中间顶点来实现,然后将轮廓重新标号^[6],然而这也仅仅解决一对二的简单分支。为了解决一对多的分支,Shanty^[20]提出修改组合轮廓方法。后来 Ekoule 等人^[7]对该方法进行了改进,将组合轮廓由局部轮廓和包含在轮廓内的螃蟹状的最小凸壳组成。后来 Meyers 等人^[16]开始对峡谷三角化进行了改进,峡谷三角非常像走廊无法与任何组合轮廓匹配。组合轮廓的不足之处就是没有考虑轮廓的拓扑结构。对简单

分支问题来说,这些研究基本达到了满意结果。但明显不足之处在于组合轮廓只解决一对多,且相当主观。

Boissonnat^[4]采用不同的策略,把对邻近断层轮廓的三角化问题当作是沿轮廓的多面体与三角面片的交,其中多面体通过简化 3D-Delaunay 三角方法获得。为了与断层轮廓一致,必须在轮廓上添加新的顶点。为了处理复杂分支和不相似轮廓,Geiger^[10]对这一思想进行了扩展。

Yoshikazu^[18]首次提出利用轮廓差区域中的骨架,作为解决分支问题的中间结构。Meyers^[17]分析了用骨架作为组合轮廓,细分为子轮廓,最后在骨架与子轮廓间生存三角面片。用传统的 Voronoi 图表方法,从轮廓投影的差区域生存骨架,从中间轴中反复切除长而尖的边:顶点的度小于 2,以及上面的任何边支节,但该方法的限制是轮廓必须有容纳关系,否则轮廓必须变换。

骨架被用作中间几何结构^[17,19],在每一轮廓投影的差区域内,Oliva 递归创建角平分边,然后与相关子轮廓构筑三角面片,直到剩余区域是一简单或者在 Z 方向邻接断层的距离小于指定的阈值。理论上,Oliva's^[19]方法能处理相当不规则或稀少采样轮廓。但在角平分边网格生存之前,必须经过严格的预处理:轮廓投影放大和优化定向使得差区域被限制在一个外轮廓和一个或多个同一容器的内轮廓,如同 Meyers 要求一样^[17]。在医学应用方面,提取的轮廓往往非常小而且锯齿不规则,所以必须对轮廓几何进行修改,操作冗长乏味。由于在递归构筑三角面片是独立处理差区域,γ 蛇在沿着边对应子区域细分时,结果的三角面片可能包含空气。

Shantz^[20]首先讨论嵌套轮廓问题(即带空洞),而 Zyda 等人进行了实验。两个很有价值方法是由 Barequet 等人^[2]和 Bajaj 等人提出的^[1]。他们都采取类似的两步策略:首先是匹配,并在相似轮廓部分进行三角面片的构筑;然后直接三角化剩下的部分^[2],或利用裂缝的中轴对它们三角化^[1]。两者的共同特点是他们能处理任意复杂轮廓,比如多对多、互相嵌套,其差异是 Barequet 用顶点标识来探测轮廓匹配,而 Bajaj 等人采用优化断层最短距离。Barequet 等人改进了他们的方法,考虑了 3D 裂缝的三角化,从而生存光滑的外部边界。

2 基本概念和算法

2.1 术语定义

定义 1 两相邻断层的所有轮廓在一个中间平面 Π 上的垂直投影,除去重叠的内部区域,定义为差区域(简称 D-区域)。其中轮廓投影属差区域, Π 是一个与两断层等距且平行。 Π 上连接的重叠区域叫空洞。

图 1 显示了几种轮廓情形以及对应的 D-区域和空洞,且图 1(1)~(5)分别有 1, 2, 1, 4 和 3 个空洞,在 D-区域内的空洞数表明了轮廓的拓扑结构差异,决定轮廓的对应关系。不同于传统轮廓复杂度分类方法(一对一、一对多和多对多),把最多含有一个空洞归为简单情形,因此图 1(2)属复杂情形。

定义 2 从 D-区域中提取的骨架是一个图,由主骨架和微型骨架组成,其中主骨架是包围空洞的封闭圆或者有一个端点与圆相连。而微型骨架最多只有一个端点与主骨架相连,另一端点以 D-区域的尖点终止,如图 2(2)所示,段 AF 和 DE 是微型骨架,其余是主骨架。主骨架将 D-区域分成若干部分,每一部分由轮廓线和几段主骨架包围,称之为 C-片(因

为它的形状像字母 C), 显然 C-片无主骨架或轮廓经过其内部。

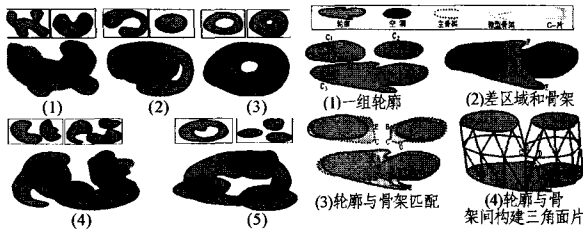


图1 两相邻断层面对、差区域和空洞 图2 算法示意图

定义3 轮廓上点 P 到主骨架的最近点的几何距离定义为该轮廓点的不相似量。在一组的轮廓中, 如果所有轮廓点的最大不相似量小于给定的阈值, 就称它们是几何相似。

2.2 算法概述

类似于传统方法, 在基于序列断层轮廓曲面重建中, 对两相邻面片轮廓处理算法描述如下。

(1) 轮廓分组对应: 精确计算所有轮廓的 D-区域, 依照 D-区域的连通性和空洞将轮廓进行分组, 使得每一组有一个连通的 D-区域分量。不妨仍用 D-区域概念来表示连接分量(如图 2(1)所示), 除非特别说明。如果一组轮廓在 D-区域中没生存空洞, 就不需进行建立公共三角网格而各自三角化, 否则将进行分支处理。

(2) 分支: 按如下步骤处理。①从 D-区域提取骨架; ②估计轮廓点的不相似量, 剪裁小于给定阈值的微型骨架, 如图 2(2)中的 AF; ③进行轮廓骨架匹配, 这有两个功能: 一是对每条轮廓, 通过寻找一条由主骨架构成的闭合曲线, 将复杂轮廓转换为简单情形, 即使轮廓与闭合曲线(称匹配骨架)构成只含一个空洞的简单的情形; 第二个功能是通过添加微型骨架, 将简单但不相似情形转化为相似。骨架的结构确定了每条骨架与两条轮廓(或轮廓段)相匹配, 虽然分别描述, 但在实践中是同时进行的, 如图 2(3)所示。

(3) 构建三角面片: 对匹配好的轮廓骨架对, 通过建立最短路径来完成三角网格的生存, 如图 2(4)所示。

利用这种处理策略, 必须解决如下问题: D-区域的计算、骨架的提取、不相似量估计、骨架与轮廓的匹配, 以及骨架与轮廓间的三角网格的建立, 而 Voxel 编码技术提供了系统的解决方案。

3 Voxel 编码技术

在像素化的 2 维区域 R 中, 以 $S(R)$ 的子集) 中的点为起点(称种子点), 按指定编码算法(或度量“ $n_e - n_o$ ”), 累加取最小运算规则, 递归对像素点进行繁殖编码叫 Voxel-编码。周勇等人首次提出了二维 Voxel 编码概念^[24], 后来扩充到三维^[25]。

Voxel-编码的一个明显的特性就是最短路径的提取, 称之为 SPE 算法^[25]。如图 3(1)所示, 设 B 和 A 是 R 中两像素点, 以 A 为种子点, 在 R 内使用“ $n_e - n_o$ ”度量进行 Voxel-编码, 直到 B 点。然后从 B 点出发, 反向递归搜索最小编码值的邻近点, 可得到 R 内的一条从 B 到 A 的路径, 如图 3(2)所示。

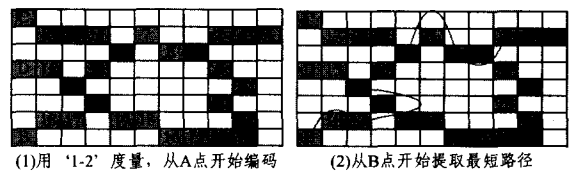


图3 Voxel-编码和最短路径示意

3.1 相关术语

定义4 在像素化的 2 维区域 R 中, 把共顶点的两像素叫 V-连通(或 V-邻域); 共网格边的两像素叫 E-连通(或 E-邻域)。把 V-连通或 E-连通的两像素称是连通的。

定义5 一组像素序列称之为路径。如果序列中任相邻两像素是连通的, 而非相邻像素是不连通的。

定义6 一个像素集是连通的, 如果其中任两像素点都有一条路径, 否则是不连通的。一个 E-连通的像素集是连通的, 如果对任两点都存在一像素点序列, 满足相邻点是 E-连通的。

定义7 在递归繁殖编码中, 限于 E-连通(不对 V-连通的像素)的 Voxel-编码称 E-编码。为区别起见, 原来的两维像素编码叫 EV-编码。

定义8 一条轮廓线将断层分为 3 个区域: 内部、边界和外部, 对应区域的像素点分别定义为内点、边界点和外部点(简称 I-像素、B-像素和 O-像素)。

定义9 轮廓的 E-邻域意指它的 I-像素(或 O-像素)对轮廓 B-像素是 E-连通的。空洞可定义为被闭合路径包围的 E-连通集, 空洞在重叠区域内是 E-连通的。一个区域的 E-轮廓是指在这个轮廓中有至少一个点是该区域的 E-邻域。

3.2 性质定理

定理1 轮廓 C 可表示为一条闭合的路径, 并具有如下性质: (1) 被 C 包围的内点像素集是一个 E-连通区域。(2) 一个内点像素不可能与外点像素是 E-邻域。(3) 设 P 是 C 上的点, 且有最小 y 坐标值, $P_f(x_f, y_f)$ 和 $P_b(x_b, y_b)$ 分别是 C 上位于点 P 的前后邻近点。如果 $x_b < x_f$, 则轮廓 C 的定向是反时针, 否则是顺时针的。(4) 邻近 C 的内点可由 C 上的像素点确定。

证明: 反设内点像素集不是 E-连通区域, 那么必存在两个分离的且被 C 包围的 E-连通区域, 即 C 上存在两非邻近像素点, 但至少是 V-连通的, 这与路径定义矛盾, 于是定理 1(1) 成立。此外, 由于 C 是闭合的, 内点像素不能与外点像素共享一条边, 从而定理 1(2) 得证。

对于定理 1(3), 考虑中心定位在 P 点的 3×3 模板, 如图 4 所示。首先 P_f 或 P_b 都不可能在模板的最下一行, 因为没有一点在 P 点之下。其次, P_f 或 P_b 也不可能在 P 之上(标有“*”的点), 因为 P_f 与 P_b 是不连通的。同样也不可能在同一列, 因此, P_b 和 P_f 分别在第 1 和第 3 列。如果 $x_b < x_f$, 则 P_b 在第 1 列, 而 P_f 在第 3 列, 这意味着 C 从左至右通过模板, 且左边有一内点。根据定义 P 上面的那点是一内点, 所以, C 是反时针, 否则就是顺时针的。

最后来证定理 1(4), 设 P 是 C 上任一点, $P_f(x_f, y_f)$, $P_b(x_b, y_b)$ 分别是 C 上 P 的前后两相邻点。下面来证明所有与 P 相邻的内点都能确定。根据定义 P_f, P_b 是邻近 P 但互不相连通的两点, 于是 P_f, P_b 满足 $|x_b - x_f| = 2$ 或 $|y_b - y_f| = 2$ 。假设 $x_f - x_b = 2$, 则 P_b 在第 1 列, P_f 在第 3 列。换言之, C

从左至右通过模板,并把模板像素分成上下两部分。如果 C 是反时针的话,上部分的点是内点,而下部分是外点,否则相反。其它情形可类似分析。

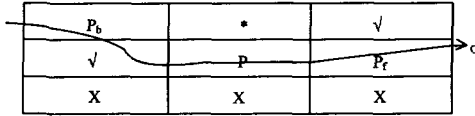


图4 定位在点 P 的 3×3 模板,点 P_f 和 P_b 不能在标有 'X' 或 '*' 的位置,但能定位在标有 '√' 的位置

定理 2 设 R 是一个 E-连通区域,那么在 R 内使用 E-编码可得到 R 的边界集 B ,其中 B 与 R 是 E-邻近但不属于 R 。

定理 2 的证明是一种直接求解的过程。首先在 R 内任选一点为种子进行 E-编码。在递归的每一步,只对已做了标识的像素点的 E-邻近点进行检测。如果它属于 R 就标记,否则必属于 B 。如果属于 R ,就以此为下一递归的种子点,直到 R 中的所有像素被标记,因为 R 是 E-连通的,于是所有像素都可以访问到,同时也就得到了所有的边界点。

实际上就导出了一个轮廓填充算法。对此,只要将种子点作为轮廓的 E-邻近的内点(根据定理 1(4)),所以定理 2 也可用于决定对应、轮廓与骨架的匹配,从而实现轮廓的三角化。

引理 1 设 R 是一连通像素点集, S 是它的任一非空子集,将 S 作为种子点集,利用“ $n_e - n_o$ ”度量对 R 执行 EV-编码。假设 C_1, C_2 是利用 SPE 算法提取的两条最短路径,其中起点是 R 内任意两点,终点在 S 中,那么 C_1, C_2 不可能相互交叉。如果它们有一公共点,则两路径其余所有点都相同。

证明:记 P_i, P_{i+1} 是 C_1 上任两邻近像素点,如果 C_2 与 C_1 在 P_i 和 P_{i+1} 之间相交,那么 P_i 与 P_{i+1} 必是 V-连通的,在 C_2 上必存在一点 Q_j 与 P_i 和 P_{i+1} 都是 E-邻近的。在 C_2 上 Q_j 的下一点 Q 就不可能与 P_i 和 P_{i+1} 是 E-邻近的。根据 SPE 算法, P_{i+1} 的两个 E-邻域点之编码值大于 P_{i+1} 点的值,否则 P_i 的下一点就不是 P_{i+1} 而是它的 E-邻域点,所以 Q_j 的下一点不能是 Q 而是 P_{i+1} ,因为 P_{i+1} 是 Q_j 的邻域点,并比 Q 的编码值小,所以,两条路径永不相交。如果它们有一公共点,则两路径其余所有点都相同。

下面的讨论都局限在两轮廓线 $C_1 = \{P_i | 0 \leq i \leq m\}, C_2 = \{Q_j | 0 \leq j \leq n\}$, D-区域记为 R 。假设 R 是非空的,则 R 是由一些子区域组成。如果 C_1, C_2 不相交, R 就仅含有一个子区域(即两轮廓嵌套)。否则,每一子区域至少由两轮廓段包围(参看图 1(1)、图 1(2)),轮廓段就是邻近交点部分。

引理 2 如果轮廓 C_1, C_2 构成只有一个空洞的 D-区域 R ,那么 R 的每一个子区域分别由 C_1 和 C_2 的两条轮廓线段包围。

证明:反设 S 是被有序轮廓线段 $\{c_{11}, c_{21}, \dots, c_{1k}, c_{2k} | k > 2\}$ 包围的子区域,其中 c_{ji} 是 $c_j (j=1,2)$ 的段。首先 k 不可能是 3,因为这些线段来自两轮廓,并且同一轮廓的任两段不会在末端相遇,于是 $k \geq 4$ 。根据 D-区域的定义,假设 S 属于 C_1 的内部区域、 C_2 的外部区域,如图 5 所示。每一段 $c_{1i} (0 \leq i \leq k)$ 连接两空洞。不失一般性,考察具有端点 C 和 D 的段 c_{12} , A, B 和 E 是邻接线段的端点(注意, A 和 E 有可能是同一点),假设在 c_1 上连接 c_{11} 和 c_{12} 的段是 c_{11} ,在 C_1 上连接 c_{12} 和 E 的段是 c_{12} ,那么 c_{21} 与 c_{11} 就构成一个空洞(两轮廓

的内点区域), c_{22} 与 c_{12} 构成另一个空洞。这与 R 只含一个空洞矛盾,于是结论成立。

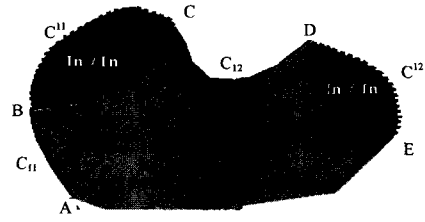


图5 由两轮廓 C_1, C_2 构成的 D-区域中的某一子区域 S

图 1(1)显示仅有一个空洞的 D-区域,图 1(2)含 2 个空洞。我们企图在 C_1 和 C_2 之间建立一个映射。为此,以 C_2 为种子点集,进行 EV-编码。而后在 C_1 上任意点 P_i 利用 SPE 算法提取最短路径,路径上终点必定是 C_2 上的 Q_{ji} ,即 Q_{ji} 可近似当作在 C_2 上 P_i 的最近点。这样就建立了 C_1 到 C_2 的一个映射。对 C_1 上每一点,在 C_2 上有一个映像点且具有如下性质。

定理 3 设 C_1, C_2 是具有同样定向的两条轮廓线, R 是它们生存的 D-区域,且有一个空洞,那么从 C_1 到 C_2 的映射是同序的,即假设 C_1 上的任两点 P_n, P_m 分别映射到 C_2 上两点 Q_n 和 Q_m ,如果 $n < m$,则有 $j_n < j_m$ 。

证明:如果 C_1 与 C_2 不相交,即其中一个轮廓完全包含另一轮廓。根据引理 1,结论成立。现在假设有交点,即 R 由于子区域组成,显然在交点相遇,其交点本身就构成一个映射对,按照引理 2,每两两线段包围一个子区域,这些两两线段定向一致:从同一交点出发,终止于另一交点。根据引理 1,对每一个子区域结论成立,从而推出对所有轮廓结论成立。

4 算法执行

4.1 轮廓的分组对应

给定两相邻断层 Π_i 和 Π_{i+1} 的轮廓,同一断层上的轮廓不交叉但可能嵌套。如果一条轮廓完全被另一轮廓封闭,则它们是顺时针的,否则是反时针定向的。然后,一个断层内的所有像素初始化特定的标识,区别内点、边界点和外点。为此,首先所有边界点初始化为 B-像素,其余的为 O-像素。接下来对内区域进行 E-编码,这是一个轮廓填充过程,以邻近轮廓的内点为种子点集(定理 1(3))。在非边界区域,进行编码扩充繁殖,直到不再有非边界像素要编码。根据定理 2,利用 E-编码重新初始化所有内像点,而不会跨越边界(定理 1(2))。图 1 显示几对邻接断层,请注意被嵌套轮廓(如图 1(3)和图 1(5)所示)包围的区域初始化为外部区域。

轮廓对应的解依赖于 D-区域内空洞的结构。对中间断层 Π_m 内任一像素点 P ,根据 P_1 和 P_2 进行分类,其中 P_1, P_2 分别来自于 Π_i 和 Π_{i+1} 并且与 P 有同样的 x, y 坐标。 P 在 D-区域内,如果是以下情况之一:①(I, O), ②(O, D), ③(B, *) 和 ④(*, B),其中 $(m, n) (m, n=I, O, B)$ 意指 P_1 是 m -像素和 P_2 是 n -像素(在它们自己的面片内), '*' 意指这些类型任何一种。如果 P 是 (I, D) 类型,则 P 在空洞内(称 H-像素)。图 1 显示邻近面片以及在不同情况下的 D-区域。

理论上,重叠区域超过给定阈值的轮廓对才有对应。先讨论两种情况:嵌套与不嵌套。如果在同一断层轮廓间没有嵌套,可得出明显的有用结论:任一空洞最多被两个轮廓包围

(因为同一断层的轮廓不相交);在嵌套情形,空洞可以被任意多个轮廓所包围,特别是作为两个断层的嵌套轮廓。无论那种情况,如果一个空洞被多于两个或两个互相交叉的轮廓包围,则所有包围这个空洞的轮廓划分为同一组,否则(只被一个,或被两个嵌套但不交叉,如图 1(3)所示)这两轮廓不能放在同一组,各自需要进一步分组对应。

轮廓意指它们在中间平面上的垂直投影。为方便起见,仍然用轮廓而不用投影这个名称。以上的分组对应算法可用如下伪代码描述:

```
ContourGrouping()
{ For each hole h
  { //N is the number of contours retrieved
    Retrive_Econtour(h, &N);
    If (N>2 or (N==2 并且轮廓相互交叉))
      收集所有 N 条轮廓为同一组;
    Else for each retrieved contour c
      { //I is a set of I-pixels E-adjacent to c
        Retrive_Econtour(I, &M);
        收集所有 M 条轮廓为同一组;
      }
    }
  For each contour c without collected into the above groups
  { Retrive_Econtour(I, &M);
    收集所有 M 条轮廓为同一组;
  }
  Group_Merging();
}
```

其中在函数 $Retrive_Econtour(h, \&N)$ 中,用 H-像素作为输入参数,用 E-编码操作访问所有 E-连通的 H-像素(构成空洞),同时探测所有 E-轮廓,并输出它们。这要求所有轮廓首先被标记。在探测过程中,如果一个 E-区域不是一个 H-像素而是 B-像素,那么它的值表示那轮廓包围空洞。同样的函数被调用,在两个严格限制情况下:(a) 仅找回一条或 2 条互不相交 E-轮廓;(b) 还有些轮廓未收集,其中函数 $Retrive_Econtour(I, \&M)$ 与轮廓为 E-连通的, I-像素为输入参数,同样采用 E-编码,但此时 D-区域中与同一子集是 E-邻近的轮廓,作为 c 收集到同一组。

$ContourGrouping()$ 是对每条轮廓逐步搜索处理进行。所有 E-连通的像素标记,一旦 H-像素找到, $Retrive_Econtour()$ 就找回所有邻近并标记同样值(空洞的索引)。调用完成后,另一个未标记的空洞的 H-像素被搜索。如对该轮廓找不到 H-像素,就表明还有些轮廓未收集,再次调用 $Retrive_Econtour()$ 。然后转向另一轮廓。一旦所有轮廓被处理后,就调用 $Group_Merging()$ 函数来完成分组合并。

算法实际上提供了一个简单而有效的寻找所有包围一未知区域轮廓的解。这一技术是 Voxel 编码技术的另一特性,称匹配,得到的轮廓称匹配轮廓。

大多数已有的对应的方法^[7,16,22] 都有局限性:计算复杂和方法笨拙。而 Voxel-编码技术使问题变得简单,用很少的计算量就能精确计算轮廓重叠区域的面积。

4.2 D-区域的估计

D-区域的估计算法分两步完成:(1) 计算个别轮廓包围的子区域面积;(2) 求集合的并和交。首先一条轮廓可理解为图像的投影,并将投影平面分为内点、边界和外点。下面来

讨论怎样填充内点区域,理论上可运用一个典型多边形区域的填充算法完成。但对不规则外轮廓线包围组成的复杂点集,其逼近的多边形是由大量的点组成,从而会降低算法效率。

事实上,轮廓是一条封闭路径。根据定理 1(2),任何外部像素点不可能与任何内点是 E-邻近的,于是利用 Voxel-编码能区别这些区域。根据定理 1(3)、1(4),选定一内点为种子点,进行编码探测内部区域。在繁殖编码过程中,只有已访问过的 E-邻域被选定作进一步繁殖编码,这样处理确保繁殖编码不会周游从内部到外部的轮廓(定理 1(1))。得到所有轮廓的内部区域后,再将内部像素投影到中间平面,仅覆盖一次的像素就构成 D-区域(当然 D-区域可能由多个分离的子区域组成)。

D-区域是由一系列轮廓边(图 1 中的实线)所包围的非空两维点集。但边的端点分别是不同断层的两条轮廓的交点。尽管来自同一面片的个别轮廓是分离的,但它们的投影构成一个连接的区域,是由于轮廓的连接点投影来自其它面片。如果轮廓投影相交(图 1,除(3)外),R 将被轮廓的重叠区域(图 1 中的黑色区域)或外部区域分成不同的子区域,所有子区域仅在轮廓交叉点相遇,每一子区域必须是由几条来自不同断层的轮廓边所包围。

如果不相交,所有边就是整个轮廓投影,D-区域就是位于闭合区域与所有嵌套在较大区域之间部分(如图 1(3)所示)。图 1(1)和(2)显示 1-1 轮廓连接,大多数已存在的重构方法将这些认为是简单的情形,但图 1(2)并不是。

4.3 骨架提取

从 D-区域中提取骨架,包括主骨架和微型骨架。骨架提取算法有如下特点:① 探测物体空洞;② 便于修剪不必要的骨架段;③ 生存由中心线组成的骨架;④ 与轮廓的复杂性和退化无关。

基于 Voxel-编码的方法^[25] 可直接应用于整个 D-区域,原来用于三维复杂物体的骨架提取,且具有以上所列特性,与轮廓复杂度无关。骨架由连接中心线组成,开始作为最短路径提取,然后再向中心轴移动。

应用该方法到二维 D-区域提取中心线集,并分为两类:第一类有或者包围空洞或连接空洞,叫主骨架;其它的的中心线组成另一类,叫微型骨架。所有骨架构成图。它们的结点是中心线的端点。如果只有两条中心线在结点相遇,就删除该结点,与它相连作为一条中心线。连接两个结点的中心线叫骨架段。所以,一个结点或者连接一条或者多于两条骨架。

相互连接的微型骨架构成一棵树,树根是树与主骨架的唯一连接点,叶结点是 D-区域的边界。如图 6 所示,同一颜色的曲线段表示一条骨架,中间图的左下边是唯一的一条微型骨架。图 7 显示一条复杂的微型骨架树,注意骨架在轮廓交点没有分开(split),从而避免了细小骨架的生存。



图 6 从图 1(2)、1(4)和 1(5)中提取的骨架

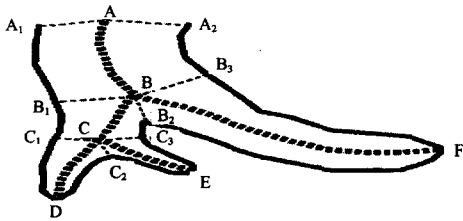


图7 由图6生存的一条退化的复杂微型骨架树

骨架提取后,引入不相似量概念,据此对骨架进行修剪。设 P 是轮廓线上的一点,那么点 P 的不相似量可用如下函数估计:

$$E_{dis}(P) = \begin{cases} 0, & \text{if } d_{avg} = 0 \text{ or } d(P) \leq d_{avg} \\ (d(P) - d_{avg}) / d_{avg}, & \text{otherwise} \end{cases}$$

其中 $d(P)$ 是在 D -区域内,连接 P 和主骨架上最近点的最短路径长度(一般不是直线长度); d_{avg} 是 $d(P)$ 在所有轮廓点上的平均值。根据该公式,微型骨架上远离主骨架的端点有最大的不相似量。该值将作为微型骨架不相似量,为简化三角面片,将对低于阈值的微型骨架进行修剪。

对 d_{avg} 的计算,采用如下近似求解方法:以 P 为种子点,在 D -区域内使用“2-3”度量进行 EV-编码。然后搜索主骨架上的点之最小编码值,应用 SPE 算法提取连接 P 到主骨架上的最近点之最短路径,路径长度即为 $d(P)$ 。对所有轮廓点求出 $d(P)$,取平均即为 d_{avg} 。

设 e 是带有叶节点 P 的微型骨架,如果它的不相似量 $E_{dis}(P)$ 低于用户给定的阈值,就把它剪裁。值得注意的是,在骨架被剪裁之前,相连的微型骨架段不能修剪。如图7所示,在 CE 剪裁之前, DB 不能剪裁。因此,微型骨架修剪是一个递归处理过程。用这种方法,我们能有效地修剪掉大量细小的分支骨架,从而大大简化三角面片。

4.4 轮廓与骨架的匹配

轮廓与骨架的匹配就是对每一条轮廓寻找一条由主骨架构成的闭合曲线,以及相关联的微型骨架。理论上如下结论。

定理4 设 D 是由一组轮廓 C 相关联的 D -区域, S 是从 D 中提取的骨架,那么对 C 中的每一轮廓 c ,在 S 中,有一主骨架集,使之构成一闭合的曲线。

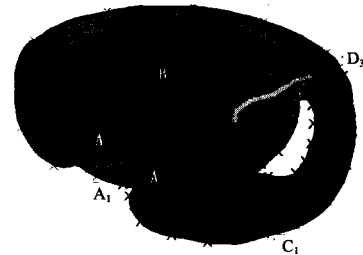
证明:根据 D -区域定义, D 由一些子区域组成。这些子区域包含空洞,并只在轮廓的交点处相连接。主骨架把子区域细分为 n 个 C -片,其中 n 是包围子区域的轮廓段数。每一 C -片仅对一条轮廓段是 E -连通的,所以一轮廓段通过共享同一 C -片而关联几条主骨架(如图6所示)。于是对每一条轮廓,收集与它相关的所有主骨架。下面来证明这些骨架构成一条闭合曲线。事实上,在每一 C -片内,相关的主骨架是连通的,而所有相关的 C -片在 C 与其它轮廓的交点相连接。因为 c 是闭合的,所以这些主骨架也就构成一条闭合的曲线。

下面来讨论轮廓与骨架的匹配,首先对 D -区域重新初始化:轮廓线上所有像素标记为 B -像素;主骨架上的所有像素指定主骨架的标识(暂时忽略微型骨架);空洞内的像素当作 O -像素; D -区域内其余像素当作内点像素。接下来进行 E -编码:从 c 上选择种子点(非主骨架或其它轮廓的交点),繁殖区域限定在 C -片内,因为 C -片是 E -连通的,这样编码处理能访

问 C -片。根据定理2,匹配所有相关的主骨架。再依据定理1(2)和 E -编码定义,繁殖既不能跨越轮廓也不能跨越骨架。

定理4建立了轮廓 c 与主骨架构成的闭合曲线 c' 之匹配关系,于是与 c 相关的所有 C -片就构成了 c 与 c' 的 D -区域。该 D -区域通常含一个空洞。换言之,定理4将复杂轮廓分解为简单情形。在这一处理过程中,我们得到了轮廓间的匹配,而不是子区域间的匹配,从而避免了对细小或退化的子区域计算处理。

定理4也提供了 c' 到 c 的几何映射:对 c' 上点 P ,映像点是 c 上的最近点。在大多数情况下,映像点是在 c 上的投影点。实现方法是:以 P 的 E -邻近点为种子,用 SPE 算法求得的路径端点作为映像点。值得注意的是,在使用 SPE 算法时不需重新计算编码域,因为在轮廓与骨架匹配时编码已生存。图8给出了图1(2)的映射示例,骨架的映射端点被描绘,并列出了映射段。



$A_2 B_3 \rightarrow AB \langle - A_3 B_2$
$B_2 C_2 \rightarrow BC \langle - B_1 C_3$
$C_1 D_3 \rightarrow CD \langle - C_2 D_2$
$D_3 A_2 \rightarrow DA \langle - D_1 A_1$
$A_3 C_1 \rightarrow AC \langle - A_1 C_3$
$B_3 D_2 \rightarrow BD \langle - B_1 D_1$

轮廓 \rightarrow 骨架 \leftarrow 轮廓

图8 图1(2)中轮廓与骨架的映射,每一轮廓段匹配唯一的骨架,用彩色标有‘X’是它的骨架匹配点。每一骨架映射了两次。

特别注意 c' 上的一条主骨架 s 可能被关联同一轮廓的两个 C -片共享(如图8中的骨架 AB 和 CD),即 s 上的轮廓点从不同的 C -片,在相同的 E -编码(相对 c),已经达到了两次。此时,不同的 E -邻域导致不同的路径,从而不同的映射。换言之, s 映射到两个轮廓段。为区别这两个映射, P 的 E -邻域被分为内部和外部(相对于 s)(定理1(3)),从而相对同样的类型的 E -邻域就构造 s 的一个映射。

对不同骨架公共端点,如图8中的 AB 和 BD ,为保证 B 的映像点都是同一点 B_3 ,在 SPE 算法中,用于提取路径的 E -区域必须相同。为此,只要选取第一次处理骨架使用的 E -邻域,并以此作为后面公共段的 E -邻域。

此外,这个映射满足定理3,只要注意到每一个映射只在单个 C -片内进行,并且所有包围 C -片的匹配主骨架能合并为一段。所以一个 C -片实际上分别由 c 和 c' 上的两段包围:合并骨架和轮廓段。接下来,对其它所有轮廓采取同样的处理方式。最后每一条主骨架匹配了两次,生存两个映像点集,使每一轮廓被映射一次。

在匹配和映射过程中,微型骨架具有类似的性质。因为每组微型骨架构造了一骨架树,不是以圆包围空洞,这些微型骨架在同一 C -片内,在匹配时,跟主骨架一样, E -编码能达到

但不能跨越它们,所以同一轮廓的不同段与微型骨架匹配一次。与处理主骨架方式一样,它的映射也能计算出来,同一轮廓有映射两次。在一骨架树中,图7显示了映射。

通常一组轮廓仅有两个轮廓,它们在几何上相似,即所有微型骨架被剪裁了,D-区域包含一个空洞。这时可以放弃主骨架,可直接利用定理3计算映射。骨架与轮廓间三角面片的构建就是基于这些映射关系。

4.5 轮廓与骨架的三角化

先考虑轮廓段 C_1 与之相匹配的骨架 C_2 间的三角化。根据定理1和定理4,问题可描述如下:

设 $C_1 = \{Q_i | 0 \leq i \leq n\}$, $C_2 = \{P_i | 0 \leq i \leq m\}$, 映射 $\eta: P_i \rightarrow Q_{j_i}$ ($0 \leq i \leq m$), 满足:

(1) $j_0 = 0, j_m = n$, 即端点映射到端点;

(2) $d(P_i, Q_{j_i}) \leq d(P_i, Q_{j_k})$ ($0 \leq k \leq m, k \neq i$); 即原像与像构成最短连线;

(3) 如果 $i_1 < i_2$, 则 $j_{i_1} \leq j_{i_2}$, 即连接映射点对的线段互不相交。

三角化构建过程如下: 首先轮廓段上两相邻点的线段做三角形的边, 其次 P_i 与 Q_{j_i} ($0 \leq i \leq m$) 间的线段被当作三角形的边; 最后在邻近点 P_i, P_{i+1} ($0 \leq i \leq m-1$) 与它们的映射点 $Q_{j_i}, Q_{j_{i+1}}$ 三角化; 如果 $j_{i+1} = j_i + 1$, 即 Q_{j_i} 与 $Q_{j_{i+1}}$ 是邻近点, 就对四边形 $P_i, P_{i+1}, Q_{j_{i+1}}, Q_{j_i}$ 直接三角化, 否则 C_2 上的 Q_{j_i} 与 $Q_{j_{i+1}}$ 间的点按简单的索引顺序分别连接 P_i 或 P_{i+1} 。

将独立生存的三角片连接起来, 就构成轮廓与它们匹配的骨架的所有三角片。除了断层内的轮廓, 自然不需要添加额外的线段。

显然三角化遵循局部最小距离规则, 但我们并没有直接计算轮廓点间的距离, 而是利用 SPE 算法。该方法非常漂亮, 有两个原因: 原因之一是三角化是在轮廓与匹配骨架之间, 满足用户指定的误差; 第二, SPE 算法与编码度量无关, 因此在最短距离计算时, 可用更精确的度量^[5]。

现在来分析主骨架和微型骨架对构筑的影响。从轮廓与骨架的匹配得知, 主骨架决定邻近断层中轮廓的三角化的拓扑结构, 没有微型骨架, 能生存正确的拓扑三角构筑。在这种情形, 原来连接到骨架树的微型骨架的所有轮廓点, 都被连接到树的根, 所以生存的三角形能被伸长、扭曲, 甚至沿极不相似段互相交叉, 所以作为过渡, 微型骨架的加入能改进构筑几何质量。

图9显示了两个直接连接的构筑结果。图9(左)是图1(2)的构筑结果, 没有构造骨架; 而图9(右)是图1(4)的构筑结果, 没有考虑微型骨架。为了对比, 其详细结果如图10所示。



图9 图1(2)中没有骨架生存的三角面片和图1(4)没有微型骨架

由于轮廓和骨架均表示为像素点序列, 轮廓通过采样输入。对骨架上的每一采样点, 其映像点可能不是轮廓的采样点, 此时取最邻近采样点作为其映像点。因此, 可生存不同级

别细节的三角面片, 从而实现对输出三角面片密度的控制。

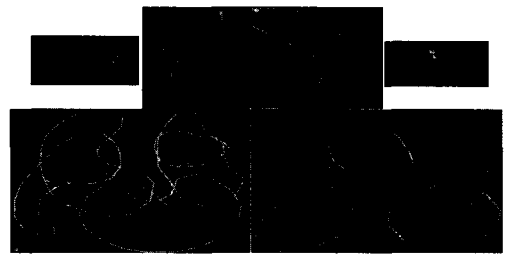


图10 图1构建的三角网格曲面

5 算法仿真实验

首先设计一个数据生存工具, 能产生极其复杂的邻近断层数据。文中给出的案例数据就是用这种方法产生的。为方便对比, 如图6所示, 所有面片均用透明效果显示, 相关的骨架用同一颜色绘画, 使2维D-区域到3维面的对应表示更加清楚。

图10是由图1中两邻近断层数据构建的三角网格曲面, 注意图10(2)、图10(4)和图9的差异。图11是一人体大脑皮层的磁共振数据, 阈值不同的5种重建仿真。所有轮廓都是用文献[26]中介绍的编码技术提取的, 未经任何预处理。表1列出了相关的MRI数据尺寸大小、阈值、轮廓数、分支数、重构三角面片数, 以及时间消耗(不含轮廓提取时间, 在SGI Power Onyx-R10000 CPU运行测试)。结果表明阈值的改变对轮廓的拓扑结构和生存的三角面片有影响: 当阈值增加时, 轮廓数和分支也随之增加, 轮廓将会具有任意复杂, 重建的人体大脑皮层表面变得逐渐皱缩并向内扭曲, 接近真实。

表1 算法测试数据与结果

	Size (threshold)	Contours	Branching	Interval (in voxel)	Triangles	Time (sec)
图11(1)	512×512×146 (50, 0)	150	1	4.5	54516	38
图11(2)	512×512×146 (70, 0)	183	18	4.5	59213	75
图11(3)	512×512×146 (90, 0)	546	76	4.5	87037	298
图11(4)	512×512×146 (110, 0)	980	130	4.5	111731	501
图11(5)	512×512×146 (130, 0)	666	124	4.5	66192	251

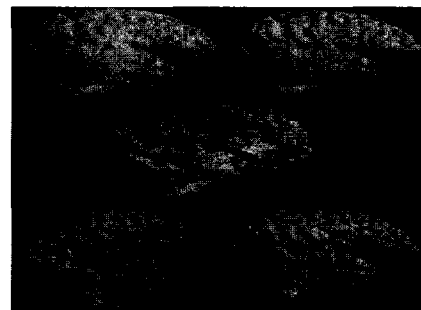


图11 大脑皮层表面1到5分别对应阈值50, 70, 90, 110和130

为了用不同细节的三角形进行曲面重建, 设计两个采样参数: 轮廓点区间长度和邻近断层的区间长度。因为提取的轮廓用路径表示, 所以轮廓点采样是等距的。如果两邻近断

层不含分支轮廓,而且它们的D-区域的面积与整个轮廓投影之百分比率低于用户指定的阈值,则第二个断层可以删除忽略。按照这样的策略,如果物体的拓扑变化低于用户给定的误差的话,可以减少很多中间轮廓,使问题简化。

实验仿真表明,Voxel-编码技术不仅提供了一个有效的快速重建算法,而且简化了输出三角面片,不需付出额外的计算代价。

结束语 本文提出了一个基于Voxel-编码的轮廓线曲面重建算法,能以一种完全自动的方式处理带有空洞的复杂多分支曲面。算法在相邻断层的差区域中提取骨架,利用这些骨架来度量两切片上的轮廓是否相似,对不相似的进行剪支分解,使不相似的、复杂的轮廓转换为简单且相似的骨架与轮廓对。

Voxel-编码技术有几个明显的特性:物体(或轮廓)的拓扑结构可以任意复杂;生存的三角形面片有很好的可控性分布;从算法上来说,它可直接执行,计算速度快,与轮廓的复杂度无关,无需复杂的处理过程,大大降低了计算复杂度。

为了改进三角形面片的质量,可设计一种方法,根据几何分布,在三角形面片中交互性地添加微形骨架。

本文提出的重建算法是文献[25]中用Voxel编码技术提取复杂三维物体的骨架的扩展,可用于各类数据或散点数据的三维物体重建。因为Voxel-编码技术是与空间点的距离相关联的,我们深信,在离散体环境中,对解决许多与距离相关的问题,它将得到更广泛的应用。

参 考 文 献

- [1] Bajaj C L, Coyle E J, Lin K. Arbitrary Topology Shape Reconstruction from Planar Cross Section[J]. Graphical Models and Image Processing, 1996, 58(6): 524-543
- [2] Barequet G, Sharir M. Piecewise-Linear Interpolation between Polygonal Slices[J]. Computer Vision and Image Understanding, 1996, 63(2): 251-272
- [3] Barequet G, Shapiro D, Tal A. History Consideration in Reconstructing Polyhedral Surfaces from Parallel Slices[C]// IEEE Visualization'96. 1996: 140-156
- [4] Boissonnat J D. Shape Reconstruction from Planar Cross Sections[J]. Computer Vision, Graphics, and Image Processing, 1988, 44: 1-29
- [5] Borgefors G. Distance Transformations on Digital Images[J]. Computer Vision Graphics Image Processing, 1986, 34: 344-371
- [6] Christiansen H N, Sederberg T W. Conversion of Complex Contour Line Definition into Polygonal Element Mozaic[J]. Computer Graphics, 1978, 12(3): 187-192
- [7] Ekoule A B, Peyrin F C, Odet C L. A Triangulation Algorithm from Arbitrary Shaped Multiple Planar Contours[J]. ACM Transactions on Graphics, 1991, 10(2): 182-199
- [8] Fuchs H, Kedem Z M, Uselton S P. Optimal Surface Reconstruction from Planar Contours[J]. Communications of ACM, 1997, 20(10): 693-702
- [9] Ganapathy S, Dennehy T G. A New General Triangulation Method for Planar Contours[J]. Computer Graphics (SIGGRAPH'82), 1982, 16(3): 69-75
- [10] Geiger B. Three-Dimensional Modeling of Human Organs and Its Application to Diagnosis and Surgical Planning[R]. 2105, INRIA, France, 1993
- [11] Gitlin C, O'Rourke J, Subramanian V. On Reconstructing Polyhedra from Parallel Slices[J]. International Journal of Computational Geometry and Applications, 1996, 6(1): 103-122
- [12] Hong M, Seberberg T W, Klimaszewski K S, et al. Triangulation of Branching Contours Using Area Minimization[J]. International Journal of Computational Geometry and Applications, 1998, 8(4): 389-406
- [13] Jones M W, Chen M. A New Approach to the Construction of Surface from Contour Data[C]// Computer Graphics Forum (EUROGRAPHICS'94). 1994: c-75-c-84
- [14] Kehtarnavaz N, Simar L R, De Figueiredo R J P. Asyntactic/Semantic Techniques for Surface Reconstruction from Cross-Sectional Contours[J]. Computer Vision, Graphics, Image Processing, 1988, 42: 399-409
- [15] Keppel E. Approximating Complex Surfaces by Triangulation of Contour Lines[J]. IBM J. Research Development, 1975, 19: 2-11
- [16] Meyers D, Skinner S, Sloan K R. Surface from Contour: The Correspondence and Branching Problem[J]. ACM Transactions on Graphics, 1992, 11(3): 228-258
- [17] Meyers D. Reconstruction of Surfaces from Planar Contours[D]. University of Washington, 1994
- [18] Yoshikazu O. Three-dimensional Reconstruction of Pore Geometry from Serial Sections-Image Algebraic Approach[C]// Pug R, Harbaugh J W, eds. Computer Graphics in Geology, Lecture Notes in Geology vol. 41, Springer, 1992: 63-76
- [19] Oliva J M, Mperrin, Coquillart S. 3D Reconstruction of Complex Polyhedral Shapes from Contours Using a Simplified Generalized Voronoi Diagram[J]. Computer Graphics Forum (EUROGRAPHICS'96), 1996, 15(3): c-397-c-408
- [20] Shantz M. Surface Definition for Branching, Contour-Defined Objects[J]. Computer Graphics, 1981, 15(2): 242-271
- [21] Shinagawa Y, Kunii T L. The homotopy Model: A Generalized Model for Smooth Surface Generation from Cross Sectional Data[J]. The Visual Computer, 1991, 7: 72-86
- [22] Wang Y F, Aggarwal J K. Surface Reconstruction and Representation of 3D Scenes[J]. Pattern Recognition, 1986, 19(3): 197-207
- [23] Zhou Y, Kaufman A, Toga A W. 3D Skeleton and Centerline Generation Based on An Approximate Minimum Distance Field[J]. The Visual Computer, 1998, 14(7): 303-314
- [24] Zhou Y, Toga A W. Extracting and Representing the Cortical Sulci[J]. IEEE Computer Graphics and Applications, 1999, 19(3): 49-55
- [25] Zhou Y, Toga A W. Efficient Skeletonization of Volumetric Objects[J]. IEEE Transactions on Visualization and Computer Graphics, 1999: 196-209
- [26] Zhou Y, Toga A W. Turning Unorganized Points into Contours[C]// Pacific Graphics'2000. Hong Kong, IEEE Computer Society, October 2000: 243-252
- [27] Zyda M J, Jones A R, Hogan P G. Surface Construction from Planar Contours[J]. Computers and Graphics, 1987, 11: 393-408
- [28] Nilsson O, Breen D, Museth K. Surface Reconstruction Via Contour Metamorphosis: An Eulerian Approach with Lagrangian Particle Tracking[C]// Proceedings of IEEE Visualization. 2005: 407-414