

三维带符号的欧氏距离变换及其应用

蔺宏伟 王国瑾

(浙江大学计算机图像图形研究所 杭州 310027)
(浙江大学 CAD & CG 国家重点实验室 杭州 310027)

摘 要 距离变换是图像处理中历史悠久的研究课题. 该文将二维带符号的欧氏距离变换推广到三维, 对其进行优化, 分析了它的计算复杂度, 并应用于解决计算机图形学中的两个重要问题: 第一, 将图形对象的三角网格表示转换为它的距离场表示. 即首先将三角网格模型离散为体素表示, 利用三维带符号的距离变换, 将求空间一点到图形对象的最短距离的全局搜索过程, 转化为求这一点到离它最近的特征体素所包含的图形对象部分的局部搜索过程; 第二, 利用类似的思想, 求两张空间曲面之间的最短距离.

关键词 欧氏距离变换; 距离场; 曲面; 最短距离
中图法分类号 TP391

Three Dimensional Signed Euclidean Distance Transform and Its Applications

L IN Hong-Wei WANG Guo-Jin

(Institute of Computer Image and Graphics, Zhejiang University, Hangzhou 310027)
(State Key Laboratory of CAD & CG, Zhejiang University, Hangzhou 310027)

Abstract The researches for distance transform have long history in image processing. In this paper, we extend two dimensional signed distance transform to three dimension, optimize it and analyze its computational complexity. Furthermore, we apply it to computer graphics. Firstly, it can be employed to transform the triangular mesh representation of graphics model into its distance field representation. By the three dimensional signed distance transform on the voxel representation of a graphics model, the global search for the point, which is closest to a given space point and on the graphical model, can be simplified to a local search. It greatly reduces the computational complexity. Secondly, and similarly, it can be employed to calculate the minimum distance between two pieces of surfaces in the space.

Keywords Euclidean distance transform; distance field; surface; minimum distance

1 引 言

给定一幅图像, 图像中的像素被分成特征像素和非特征像素. 特征像素可以是点、边或者物体. 所谓图像上的距离变换, 就是求图像中每一个像素到离它最近的特征像素的距离. 距离变换首先由 Rosen-

feld 与 Pfaltz 提出并给出算法^[1]. 之后, 人们提出了一系列并行和串行算法^[2~7]. 特别地, Danielsson 提出了二维欧几里得距离映射^[2]; 随后, Ye 将 Danielsson 的方法进行了改进, 得到了二维带符号的欧几里得距离变换 (2-Dimension Signed Euclidean Distance Transform, 2-SED T)^[3]; 利用这个方法, 可以得到离每个非特征点最近的特征点 (nearest feature point)

收稿日期: 2002-07-10; 修改稿收到日期: 2002-12-19. 本课题得到国家自然科学基金 (60173034)、国家“九七三”重点基础研究发展规划项目 (2002CB312101) 资助. 蔺宏伟, 男, 1973 年生, 博士研究生, 主要研究领域为计算机辅助几何设计、计算机图形学和反求工程. E-mail: hwlin@css.zju.edu.cn. 王国瑾, 男, 1944 年生, 教授, 博士生导师, 主要研究领域为计算机辅助几何设计、计算机图形学、数字几何处理、离散微分几何与应用逼近论.

的信息;1992年,Leymarie和Levine又对这个算法进行了优化^[4],使运算速度得到提高.但在工程应用中,图像信息也往往以三维形式给出,仅有平面图像理论已经不足以解决新产生的实际问题.有鉴于此,本文将2-SEDT算法推广到三维,给出三维带符号的欧几里得距离变换(简称为3-SEDT),分析了它的计算复杂度,对其进行了优化,并发现了它在计算机图形学中的以下两个重要应用方向.

其一是用于实现图形对象的三角网格表示到其距离场表示的转换.近年来,在计算机图形学领域,图形对象的距离场表示得到广泛应用.比如,在造型方面,距离场不仅适用于物体的形状表示和布尔操作^[8,9],而且可方便地产生等距曲面^[10];在动画方面,利用距离场,无须建立两个物体上顶点的对应关系^[9,11~13]就能实现不同拓扑结构的物体之间的变形(morphing);在绘制方面,距离场既可用于纹理映射^[10],又可用于光线投射的加速算法^[14];另外,距离场在产生层次细节(LOD)模型、精细雕刻等方面也有重要应用^[8].但是,在计算机图形学中,图形对象使用得最广泛的表示形式是三角网格,如何将三角网格有效地转换为距离场表示,还没有得到充分研究.目前仅有少数学者考虑过这个问题,且有关算法并非十分有效.例如Daniel^[11]首先将网格对象离散成体素表示,再分别以网格对象内/外及与其有交的体素为特征体素,两次运用扩展到三维的Chamfer距离变换方法^[6]求得网格对象的距离场.但这个方法的缺点最大是求得的距离场其精度依赖于对网格对象进行体素化的精度,且算法需要确定一个体素在对象内还是在对象外,这并不容易,这样得到的距离场也并不是真正的欧几里得距离场.又如Bradley和Arthur^[10]取空间一点到每一个三角面片的距离最小者作为这点到网格对象的距离.他虽然对这一过程进一步作了优化,但全局搜索点到对象的最短距离其计算量非常大;何况这样求得的距离不带符号,要得到符号,又必须对许多平面片与网格对象求交,其计算量也非常大.正是为了克服以上几种算法的严重局限性,本文利用3-SEDT,给出了将图形对象的三角网格表示有效地转换为距离场表示的一个新方法,将空间一点到图形对象最短距离的全局搜索优化为局部搜索,从而使计算量大大减少.

其二是用于求取两张曲面或者两个物体之间的最短距离.这是涉及到机器人学中的路径规划、路径修改以及数控加工的重要问题.文献^[15]通过递

归地分割两个物体的包围盒,并计算它们之间相距最近的部分来得到两个物体之间的最短距离.而我们则再次应用在本文中创造3-SEDT算法,方便、有效地求出了两个物体表面之间,或者两张曲面之间的最短距离,而不管它们采用显式表示还是隐式表示.

本文清晰地描述了3-SEDT算法的定义和运行过程,并以众多实例表明,这一算法在计算机图形学中具有明显的应用价值.

2 三维带符号的欧氏距离变换

在讲述三维带符号的欧氏距离变换(3-SEDT)之前,为获得一个直观认识,先看一个一维的例子,用一维SEDT算法来求一维图像的距离场.一系列以方框表示且序号为 i 的像素组成一维图像,方框中的数字表示欲求的距离值 f_i .一维图像中特征像素的初始距离值赋为0,非特征像素的初始距离值赋为无穷大(理解为无穷大的数).图1的上半部分是算法中用到的两个模板.待处理的一维图像以向右的方向为正.

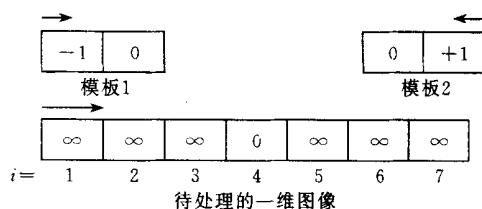


图1 待处理的一维图像及1-SEDT算法中用到的两个模板

算法1-SEDT是这样进行的.首先,用模板1沿待处理的一维图像从左向右扫描一遍,模板作用于取值为0的元素所对应的像素.扫描过程中,第 i ($i=2,3,\dots,7$)个像素的新距离值取为 $\{f_{i-1}-1, f_i+0\}$ 中绝对值最小的一个.然后,用模板2沿待处理的一维图像从右到左扫描一遍,扫描过程中,第 i ($i=6,5,\dots,1$)个像素的新距离值取为 $\{f_i+0, f_{i+1}+1\}$ 中绝对值最小的一个.最后,得到的结果如图2所示.

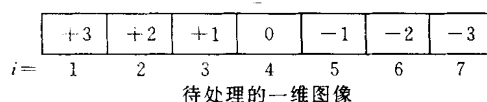


图2 用1-SEDT方法所得到的图1中一维图像的距离场

从图2可以看到,用1-SEDT方法得到的距离场,不仅提供了非特征像素到特征像素的最短距离值,而且还提供了离非特征像素最近的特征像素相对于这个非特征像素的位置.以第6个像素为例,首先,它的距离值非0,说明这是一个非特征像素;其次,它的距离值为-2,说明它离最近的特征像素

的距离值为 2; 最后, 可知离它最近的特征像素是第 $(6 - 2 = 4)$ 个像素。

与 1-SED_T 类似, 二维带符号的欧氏距离变换 (2-SED_T)^[3,4] 是用图 3 中所示的 4 个模板在一幅二维图像上扫描。这些扫描分为两个过程, 一个是由

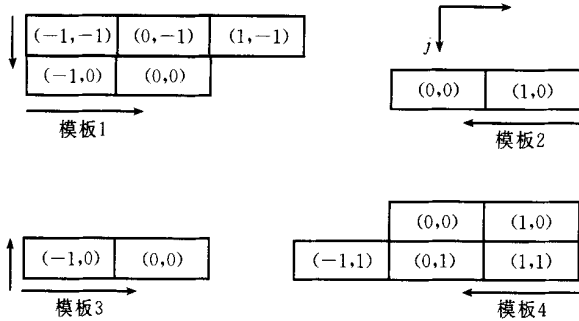


图 3 2-SED_T 的模板

上而下的过程, 对每一行像素, 先用模板 1 从左到右扫描, 再用模板 2 对同一行像素从右到左扫描; 另

一个过程自下而上进行, 对每一行像素, 先用模板 4 从右到左扫描, 然后, 用模板 3 对同一行像素自左至右扫描。下面, 我们将其推广到 3 维带符号欧氏距离变换 (3-SED_T)。

将立方体空间 V 等分离散成体素表示: $\{V_{ijk} | 1 \leq i \leq L, 1 \leq j \leq M, 1 \leq k \leq N\}$, 其中每一个体素都是一个正方体。将体素分为特征体素和非特征体素这样两类。对于特征体素, 它的距离向量赋初值 $f_{ijk} = (0, 0, 0)$, 对于非特征体素, 距离向量赋初值 $f_{ijk} = (u, u, u)$, 其中 u 为足够大的数。

算法 3-SED_T 中用到的模板如图 4 所示。这些模板都在 k -平面上起作用, 它们都以各自赋值为 $(0, 0, 0)$ 的元素为中心, 即模板所对应的运算作用在赋值为 $(0, 0, 0)$ 的元素所对应的体素上。其中, 模板 1 中, 赋值为 $(0, 0, 1)$ 的元素位于赋值为 $(0, 0, 0)$ 的元素的正上方。模板 5 中, 赋值为 $(0, 0, -1)$ 的元素位于赋值为 $(0, 0, 0)$ 的元素的正下方。

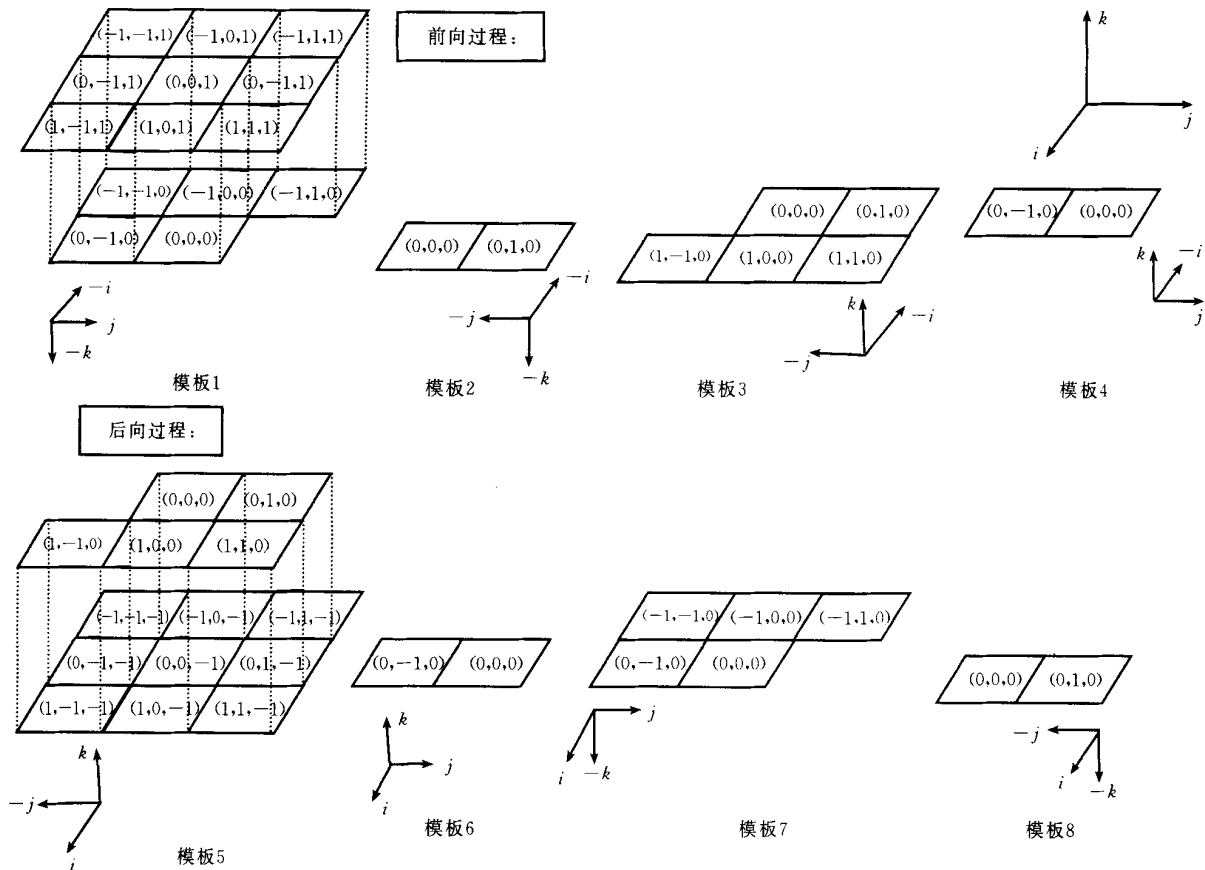


图 4 3-SED_T 算法中所用到的模板 (其中 $-i, -j, -k$ 分别表示 i, j, k 轴的负向)

算法 3-SED_T 分为两个过程, 前向过程和后向过程。前向过程从 $i = L - 1$ 的平面开始, 沿 i 轴的负向, 从前往后, 直到 $i = 2$ 的平面为止。在 $L - 1 \geq i \geq 2$ 的任一平面上, 从 $k = N - 1$ 行开始, 沿着 j 轴

的正向, 用模板 1 对这一行的每一个体素, 即模板中赋值为 $(0, 0, 0)$ 的元素所对应的体素, 作运算 (下面一段将给出运算的定义); 接着, 沿着 j 轴的负向, 用模板 2 对同一行的每一个体素作运算。然后对

$k = N - 2$ 行, $k = N - 3$ 行, ..., 直到 $k = 2$ 行. 随后, 在同一平面上沿着 j 轴的负向, 用模板 3 对 $k = 2$ 行的每一个体素作运算; 接着, 沿着 j 轴的正向, 用模板 4 对同一行作运算. 然后对 $k = 3$ 行, $k = 4$ 行, ..., 直到 $k = N - 1$ 行. 后向过程中, 从 $i = 2$ 的面开始, 沿着 i 轴的正向, 从后往前, 直到 $i = L - 1$ 的面为止. 在 $2 \leq i \leq L - 1$ 的任一平面上, 从 $k = 2$ 行开始, 沿 j 轴的负向, 用模板 5 对这一行作运算; 接着, 沿着 j 轴的正向, 用模板 6 对同一行作运算. 如此逐行进行, 直到 $k = N - 1$ 行. 然后, 沿着 j 轴的正向, 用模板 7 对 $k = N - 1$ 行作运算; 沿着 j 轴的负向, 用模板 8 对同一行作运算. 如此逐行进行, 直到 $k = 2$ 行.

下面以模板 3 为例, 说明它相应的运算的定义. 假设模板 3 对应的运算作用于体素 V_{ijk} 后所得到的新的距离向量为 f_{ijk}^{new} , 则

$$f_{ijk}^{\text{new}} = \begin{cases} f_{ijk} + (0, 0, 0) \\ f_{i,j+1,k} + (0, 1, 0) \\ f_{i+1,j+1,k} + (1, 1, 0) \\ f_{i+1,j,k} + (1, 0, 0) \\ f_{i+1,j-1,k} + (1, -1, 0) \end{cases}$$

中范数最小的那个向量. 其中范数定义为 (x, y, z)
 $= \sqrt{x^2 + y^2 + z^2}$. 其他模板对应的运算与此类似.

实际上, 3-SEDT 算法的运行过程就是一个距离信息的传播过程. 由图 4 可以看出, 模板中元素的赋值, 实际上是这个元素相对于模板中心(即赋值为 $(0, 0, 0)$ 的元素)的相对位移向量. 仍以模板 3 为例, 如果它作用在非特征体素 V_{ijk} 上, 而体素 $V_{i,j+1,k}$ 为特征体素, 它的距离向量为 $f_{i,j+1,k} = (0, 0, 0)$, 则 $\|f_{i,j+1,k} + (0, 1, 0)\| = \|(0, 1, 0)\| = 1$, 若这是范数最小的向量, 那么模板 3 对应的运算作用于体素 V_{ijk} 后所得到的新的距离向量为 $f_{ijk}^{\text{new}} = (0, 1, 0)$, 这说明离体素 V_{ijk} 距离最近的特征体素的下标为 $(i + 0, j + 1, k + 0)$, 即 $V_{i,j+1,k}$. 下一步, 模板 3 作用在体素 $V_{i,j-1,k}$ 上, 即所得到的新的距离向量为

$$f_{i,j-1,k}^{\text{new}} = \begin{cases} f_{i,j-1,k} + (0, 0, 0) \\ f_{ijk} + (0, 1, 0) = (0, 1, 0) + (0, 1, 0) \\ \quad = (0, 2, 0) \\ f_{i+1,j,k} + (1, 1, 0) \\ f_{i+1,j-1,k} + (1, 0, 0) \\ f_{i+1,j-2,k} + (1, -1, 0) \end{cases}$$

中范数最小的向量. 如果 $f_{i,j-1,k}^{\text{new}} = (0, 2, 0)$, 那么距离体素 $V_{i,j-1,k}$ 最近的那个特征体素的下标应当为 $(i + 0, (j - 1) + 2, k + 0)$, 即 $V_{i,j+1,k}$. 于是体素 $V_{i,j-1,k}$ 与 $V_{i,j+1,k}$ 之间的距离信息通过 $f_{ijk} + (0, 1, 0)$ 传播到距离向量 $f_{i,j-1,k}^{\text{new}}$. 依次类推, 经过前向过程和后向过程后, 距离信息传播到每一个体素. 如果体素 V_{ijk} 所对应的距离向量为 f_{ijk} , 那么距离它最近的特征体素的下标为 $(i, j, k) + f_{ijk}$.

下面我们对 3-SEDT 进行优化并分析其时间复杂度. 为简化计算, 我们用范数的平方代替范数作比较. 众所周知, 平方运算的计算量比加减运算大得多. 为减少平方运算, 假设 $f_{ijk} = (f_i, f_j, f_k)$, 考察恒等式

$$\|f_{ijk} + (d_i, d_j, d_k)\|^2 = \|f_{ijk}\|^2 + 2f_{ijk} \cdot (d_i, d_j, d_k) + \|(d_i, d_j, d_k)\|^2,$$

其中 d_i, d_j 和 d_k 分别取 $+1, -1$ 和 0 . 于是, 新的距离向量的范数的平方, 可以通过上式, 由旧的距离向量的范数的平方与相应量的加减运算得到. 再注意到初始化时, 非特征体素的距离向量是相同的. 因而, 在存储距离向量的范数的平方的情况下, 我们仅需要在初始化时计算一次非特征体素的距离向量的范数的平方. 经过这样的优化, 3-SEDT 算法的时间复杂度为 3 次平方运算, $70(L - 2)(N - 2)(M - 2)$ 次移位运算(与 2 相乘), $110(L - 2)(N - 2)(M - 2)$ 次加减运算和 $38(L - 2)(N - 2)(M - 2)$ 次比较运算.

3 图形对象的三角网格表示 向距离场表示的转换

图形对象的距离场表示, 在计算机图形学中起着越来越重要的作用. 但是, 目前在图形学中广泛存在的是图形对象的三角网格表示, 因而, 如何快速地将图形对象的三角网格表示转换为距离场表示成为一个关键问题. 利用 3-SEDT 算法, 我们设计了这种转换的一个自适应算法. 算法的主要思想是这样的: 首先将图形对象离散为体素表示, 并且将这些体素分为特征体素与非特征体素两类; 然后对体素表示的图形对象执行 3-SEDT 算法, 得到它的带符号距离场; 最后递归生成图形对象的自适应采样距离场. 由于已经得到体素表示的图形对象的带符号距离场, 因而, 可以将空间一点到图形对象的最短距离的搜索从全局范围限制到图形对象的一个局部范

围内进行,因而,与已有方法相比,其计算量和存储空间都大为减少。

算法 1(将图形对象的三角网格表示转换为自适应采样距离场表示)。

1. 求出图形对象的包围盒,并将它调整为正方体。然后,把它离散化成体素表示,使得每一个体素内至多含有一个三角网格顶点,其中每一个体素都为一个小正方体。再将体素分为特征体素与非特征体素:包含图形对象的三角网格顶点的体素为特征体素,否则,为非特征体素。

2. 在这个体素表示的正方体包围盒上,执行 3-SEDT 算法,得到离每个非特征体素最近的特征体素的信息。

3. 递归生成自适应采样距离场。初始正方体为包围盒。

3.1 求正方体的 8 个顶点、12 条边的中点、6 个面的中心点以及正方体的中心点到三角网格模型的最短距离。对于每个点:

3.1.1 判断这个点位于哪个体素内,得到离这个体素最近的特征体素 V 。

3.1.2 计算这个点与 V 内所包含的网格顶点邻接的所有三角面片的距离,取最短距离作为这点到图形对象的距离。

3.1.3 判断这个距离的符号。

3.2 按照在正方体的 8 个顶点处所求得距离值,用三线性插值方法求其 12 条边的中点、6 个面的中心点及立方体中心点处的距离值。

如果 3.2 步中由插值得到的值与 3.1 步中得到的相应值的差在一定的误差范围内,则算法停止。否则,将立方体均匀分割成 8 个小正方体,对于每个小正方体,回到 3.1 步重新执行。

下面描述算法 1 的一些关键细节。

(1) 在步骤 3.1.1 中,首先判断空间点位于哪个体素内,记这个体素为 A ,然后根据 A 所对应的距离向量,确定离它最近的特征体素,并求出它们之间的距离 d 。但是,这样只能得到离体素 A 最近的一个特征体素。我们必须进一步考察以体素 A 为中心,以 d 为半径的球的内接正方体和外切正方体之间的区域,从中找出与 A 相距为 d 的全部特征体素。

(2) 在步骤 3.1.2 中,需要计算空间一点 P 到三角片 ABC 的距离。如图 5 所示,首先算出点 P 在

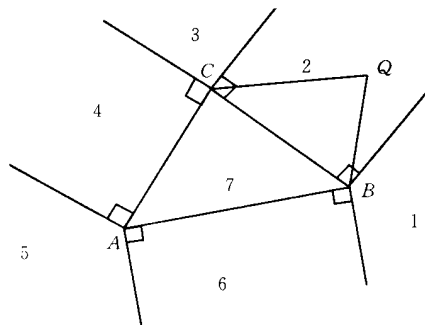


图 5 $\triangle ABC$ 将空间分成 7 个区域

三角面片 ABC 所在平面的垂足 Q ,然后判断点 Q 落在哪个区域中,从而得到三角面片 ABC 上与点 P 距离最近的点。具体做法如下所示:

(i) 计算点 P 到 ABC 所在平面的垂足 Q ,并且计算点 Q 对于 ABC 的面积坐标 (α, β, γ) 。

(ii) 在三角面片 ABC 上寻找距点 P 最近的点 R :

if $(\alpha \geq 0 \ \& \ \beta \geq 0 \ \& \ \gamma \geq 0)$

则点 Q 位于区域 7, R 即为点 Q 。

else if $(\alpha < 0, \text{并且点 } Q \text{ 在直线 } BC \text{ 上的垂足在点 } B \text{ 和 } C \text{ 之间})$

则点 Q 位于区域 2, R 即为点 Q 在线段 BC 上的垂足。

else if $(\beta < 0, \text{并且点 } Q \text{ 在直线 } CA \text{ 上的垂足在点 } C \text{ 和 } A \text{ 之间})$

则点 Q 位于区域 4, R 即为点 Q 在线段 CA 上的垂足。

else if $(\gamma < 0, \text{并且点 } Q \text{ 在直线 } AB \text{ 上的垂足在点 } A \text{ 和 } B \text{ 之间})$

则点 Q 位于区域 6, R 即为点 Q 在线段 AB 上的垂足。

else if (在点 Q 与点 A, B 和 C 三点的距离中,以点 Q 与点 A 的距离最近)

则点 Q 位于区域 5, R 即为点 A 。

else if (在点 Q 与点 A, B 和 C 三点的距离中,以点 Q 与点 B 的距离最近)

则点 Q 位于区域 1, R 即为点 B 。

else if (在点 Q 与点 A, B 和 C 三点的距离中,以点 Q 与点 C 的距离最近)

则点 Q 位于区域 3, R 即为点 C 。

(iii) 计算点 P 到三角面片 ABC 的最短距离。

(3) 在步骤 3.1.3 中,为了判断空间一点 P 到三角面片的最短距离的符号,首先调整图形对象的三角网格表示中所有三角面片的法向,使之朝向图形对象的外侧。然后,设图形对象表面上与点 P 距离最近的点为点 Q ,求出点 Q 处的法向 N ,根据向量 QP 与 N 的点积来判断距离的符号。具体步骤如下所示:

(i) 求图形对象表面上点 Q 处的法向 N 。

如果点 Q 位于某个三角面片内部,则 N 是此三角面片的法向。

如果点 Q 位于三角网格的某条边上,则 N 是这条边邻接的两个三角面片的法向的平均值。

如果点 Q 是三角网格的某个顶点,则 N 是这点邻接的所有三角面片的法向的平均值。

(ii) 求向量 QP 与 N 的点积 dot 。

如果 $dot > 0$,则点 P 在图形对象外部,距离值为正。

如果 $\text{dot} < 0$, 则点 P 在图形对象内部, 距离值为负.

作为考核的实例, 利用算法 1, 我们将图 6 和图 8 中用三角网格表示的图形对象分别变换为距离场表示, 再利用 Marching Cube 算法^[16], 从距离场中抽取出相应的图形对象, 如图 7 和图 9 所示.



图 6 兔子的三角网格表示



图 7 从距离场中抽取出来的兔子



图 8 恐龙的三角网格表示



图 9 从距离场中抽取出来的恐龙

表 1 列出了运行算法 1 将图 6 和图 8 中的图形对象的三角网格表示变换为距离场表示时所得的一些数据. 算法是在下列条件下运行的: CPU: Intel Pentium ,933MHz; 内存: 128MB RAM.

表 1 关于算法 1 的一些数据

对象名称	三角网格表示中的面片数	三角网格表示中的顶点数	三角网格表示变换为距离场表示所用的时间(s)
兔子	67036	33519	52
恐龙	47858	23931	41

4 两张曲面之间的最短距离

引言中已经指出, 求两张曲面 S_1, S_2 之间的最短距离是个重要问题. 这里最短距离为

$$d(S_1, S_2) = \min\{\|x - y\| \mid x \in S_1, y \in S_2\},$$

其中 $\|\cdot\|$ 表示欧几里得范数. 利用 3-SEDT 算法可

以有效地解决这个问题.

由于曲面之间的最短距离是对称的, 即 $d(S_1, S_2) = d(S_2, S_1)$, 我们可以先求出两张曲面中的任意一张曲面的距离场, 然后将另一张曲面嵌入这个距离场, 通过搜索后一张曲面上的点所对应的距离向量, 去找到两张曲面上距离最短的两点, 以及它们之间的最短距离. 计算两张曲面 S_1 和 S_2 之间最短距离的具体步骤如下所示:

(1) 计算包围曲面 S_1 和 S_2 的包围盒 B , 并将其调整为正方体.

(2) 将 B 离散为体素表示, 将与曲面 S_1 相交的体素作为特征体素, 其他体素作为非特征体素, 记特征体素组成的集合为 T_1 , 用 3-SEDT 算法计算曲面 S_1 的距离场.

(3) 记 B 中与曲面 S_2 相交的体素组成的集合为 T_2 , 在 T_2 中搜索与曲面 S_1 距离最短的体素 V_2 , 设其序号为 (i, j, k) , 对应的距离向量为 $f_{ijk} = (f_i, f_j, f_k)$, 则 T_1 中与曲面 S_2 距离最近的体素 V_1 的序号为 $(i + f_i, j + f_j, k + f_k)$.

(4) 对曲面 S_1 和 S_2 包围在体素 V_1 和 V_2 中的部分进行采样, 得到的采样点集分别记为 E_1 和 E_2 . 我们将这两个点集之间距离最短的两个点分别作为这两张曲面上距离最短的点, 它们之间的距离作为这两张曲面之间的最短距离.

下面, 我们详细说明以上方法实现中的几个细节问题. 首先, 按如下方式对曲面被包围在体素中的部分进行采样: 将这部分曲面正交投影到体素的每个面上, 其中, 曲面在体素的相对的两个面上的投影是相同的. 取投影面积最大的两个相对的面, 将每一个面均匀划分成 $n \times n$ 的网格, 得到 n^2 个网格点, 以这两个面的法向为方向向量, 作 n^2 条连接两个面上对应点的线段, 求出它们与曲面的交点, 并取位于体素内的交点作为采样点. 其次, 选取 $(n - 1)\epsilon$ 作为离散包围盒 B 的步长. 这里 ϵ 为给定的绝对误差.

作为考核的实例, 如图 10 所示, 我们用以上方法求两个球面:

$S_1: x^2 + y^2 + z^2 = 9$ 和 $S_2: x^2 + (y - 7)^2 + z^2 = 4$ 之间的最短距离和距离最短的两点, 其中绝对误差 $\epsilon = 10^{-2}$, $n = 10$. 用本文方法计算, 可以得到距离最短的两点(图 10 中两个黑点)为 $(0.0070, 2.9992, 0.0000)$ 和 $(0.0070, 5.0012, 0.0000)$, 最短距离为 2.0020. 显然, 两个球面理论上距离最短的两点分别为 $(0, 3, 0)$ 和 $(0, 5, 0)$, 最短距离为 2. 两者极为吻合.

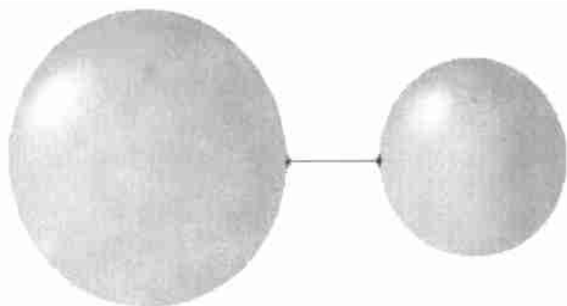


图 10 两个球面上距离最短的两点

5 结 论

在计算机图像处理以及计算机视觉领域,对距离变换的研究已经有比较长的历史. 距离变换作为一种工具,对解决图像处理领域中诸如细化、膨胀的一类问题起了很重要的作用. 在本文中,我们将二维的带符号的距离变换推广到三维,对其进行了优化,并分析了它的计算复杂度. 我们还用它解决了图形学中的两个问题. 其一是将图形对象的三角网格表示转换为它的距离场表示. 首先,将图形对象的包围盒离散成体素表示,并将包含三角网格顶点的体素作为特征体素. 采用 3-SEDT 算法,得到这个体素表示的图形对象的距离场. 这样,对于图形对象包围盒内的任一点,可以得到离它最近的特征体素,它与这个特征体素所包含的三角网格的顶点邻接的三角面片的距离的最小值,即为它到图形对象的距离. 这实际上是将一个最短距离的全局搜索问题转化为一个局部搜索问题,从而大大减少了计算量. 按照类似的思想,我们还利用 3-SEDT 解决了另一个问题——求两个曲面之间的最短距离,也取得了很好的效果.

参 考 文 献

- Rosenfeld A, Pfaltz J. Sequential operations in digital picture processing. *Journal of Association for Computing Machinery*, 1966, 13(4): 471 ~ 494
- Danielsson P. Euclidean distance mapping. *Computer Graphics and Image Processing*, 1980, 14(3): 227 ~ 248
- Ye Q-Z. The signed Euclidean distance transform and its applications. In: *Proceedings of the 9th International Conference of Pattern Recognition*, Rome, Italy, 1988. 495 ~ 499
- Leymarie F, Levine M D. Fast raster scan distance propagation on the discrete rectangular lattice. *CVGIP: Image Understanding*, 1992, 55(1): 84 ~ 94
- Rosenfeld A, Pfaltz J. Distance functions on digital pictures. *Pattern Recognition*, 1968, 1(1): 33 ~ 61
- David W P. Distance transforms: properties and machine vision applications. *CVGIP: Graphical Models and Image Processing*, 1992, 54(1): 56 ~ 74
- Borgefors G. Distance transformations in digital images. *Computer Vision, Graphics, and Image Processing*, 1986, 34(3): 344 ~ 371
- Friskén S F, Perry R N, Rockwood A P, Jones T R. Adaptively sampled distance fields: A general representation of shape for computer graphics. In: *Proceedings of SIGGRAPH 2000 Conference Proceedings*, New Orleans, 2000. 249 ~ 254
- Fang Xiang, Bao Hu-Jun, Heng Pheng-Ann, Wong Tien-Tsin, Peng Qiu-Sheng. Continuous field based free-form surface modeling and morphing. *Computers and Graphics*, 2001, 25(2): 235 ~ 243
- Payne B A, Toga A W. Distance field manipulation of surface models. *IEEE Computer Graphics and Applications*, 1992, 12(1): 65 ~ 71
- Daniel Cohen-Or, Levin D, Solomovici A. Three dimensional distance field metamorphosis. *ACM Transactions on Graphics*, 1998, 17(2): 116 ~ 141
- Daniel Cohen-Or, Levin D, Solomovici A. Contour blending using warp-guided distance field interpolation. In: *Proceedings of Visualization 96*, San Francisco, 1996. 165 ~ 172
- Turk G, O'Brien J. Shape transformation using variational implicit functions. In: *Proceedings of SIGGRAPH '99 Conference Proceedings*, Los Angeles, 1999. 335 ~ 342
- Koning Z A, Viergever M. Acceleration of ray-casting using 3D distance transforms. In: *Proceedings of Visualization in Biomedical Computing 92*, Chapel Hill, North Carolina, 1992. 324 ~ 335
- Johnson D E, Cohen E. A framework for efficient minimum distance computation. In: *Proceedings of IEEE Conference on Robotics and Automation*, Leuven, Belgium, 1998. 3678 ~ 3683
- Lorensen W E, Cline H E. Marching cubes: A high resolution 3D surface construction algorithm. *Computer Graphics*, 1987, 21(4): 163 ~ 169



LIN Hong-Wei, born in 1973, Ph.D. candidate. His research interests include computer aided geometric design, computer graphics and reverse engineering.

WANG Guo-Jin, born in 1944, professor, Ph.D. supervisor. His research interests include computer aided geometric design, computer graphics, digital geometric signal processing, discrete differential geometry and applied approximation theory.