

# 平面曲线间 Hausdorff 距离计算

曹利新\*, 董 雷, 曹 京 京

(大连理工大学 机械工程学院, 辽宁 大连 116024)

**摘要:** 为克服传统的针对平面曲线间 Hausdorff 距离 4 种情况需分别求解不同非线性方程组的缺点, 分两个步骤计算平面曲线间的 Hausdorff 距离. 首先将曲线  $A$  进行离散化处理, 并计算各离散点到曲线  $B$  的最小距离, 从中选择若干个距离较大, 且满足曲线  $A$  上相邻点到曲线  $B$  的距离呈“小大小”的点作为近似解; 然后根据各点对处曲线的特点, 判断该点附近可能存在 4 种类型点的哪一种, 建立相应的优化模型并进行局部寻优, 选择优化结果中最大的距离值作为两平面曲线间的单向 Hausdorff 距离. 该法将平面曲线间 Hausdorff 距离的计算转化为点到曲线的最小距离计算, 计算过程简单有效. 两个数值算例验证了该方法的正确性.

**关键词:** Hausdorff 距离; 平面曲线; 最小距离

**中图分类号:** TP391.72      **文献标识码:** A      **doi:** 10.7511/dllgxb201402005

## 0 引 言

几何对象间距离的计算是工程领域中经常用到的一种几何操作, 如几何建模与模式识别中对几何对象的分析与比较<sup>[1]</sup>; 机器人路径规划、CAD/CAM 和触觉仿真中, 对物体间的碰撞干涉检测和反馈力的计算<sup>[2]</sup>. 曲线、曲面逼近中构造优化目标和评价逼近程度<sup>[3]</sup>等众多工程应用领域, 都涉及几何对象间距离的计算. 在不同的距离测度中, 尤以欧几里得最小距离和 Hausdorff 距离受到几何建模、计算几何和计算机图形学等研究领域学者的格外关注.

几何对象间最小距离的计算问题, 就是在两个几何对象间求解对应距离最小的一对点. 当需要比较两个对象间的相似程度时, 则常采用 Hausdorff 距离. Hausdorff 距离是由现代拓扑学的奠基人之一德国数学家 Hausdorff 首先提出<sup>[4]</sup>, 广泛用于衡量两个集合之间差别的度量. 早期的 Hausdorff 距离计算主要出现在模式识别、图像处理领域, 为了对两幅图像进行匹配或比较, 通常对两幅图像首先进行边缘提取、距离变换等预处理; 然后将预处理后的图像像素看作两个集合, 通过对两集合间 Hausdorff 距离的计算, 进而

对两图像间的相似度进行评价; 对其中一幅图像进行仿射变换, 使变换后图像与目标模板图像间的 Hausdorff 距离达到最小, 若该距离小于给定的公差, 则表示两幅图像获得了匹配, 这一方法目前已广泛用于人脸、指纹、字符、笔迹、汽车车牌等的识别<sup>[5-6]</sup>.

与图像领域中 Hausdorff 距离的计算和应用研究形成鲜明对比的是, 针对连续几何对象间的 Hausdorff 距离的计算研究则相对较少, 截至目前, 只有少数学者<sup>[1-3, 7-8]</sup>针对自由曲线或曲面间的 Hausdorff 距离的精确计算进行过研究. 2008 年, Alt 等<sup>[1]</sup>证明了两  $C^1$  连续的平面曲线间单向 Hausdorff 距离  $h(A, B)$  会出现 4 种情况, 即 Hausdorff 距离可能发生在: 曲线  $A$  的一个端点和它在曲线  $B$  上的垂足点之间、曲线  $B$  的一个端点和它在曲线  $A$  上的垂足点之间、双垂足点间、一曲线自身的中线与另一曲线的交点之间, 并给出了 4 种情况分别对应的非线性约束方程组, 最终借助标准的代数方程组求解器获得了平面曲线间的 Hausdorff 距离. 同年, Elber 等<sup>[2]</sup>将平面  $C^1$  连续曲线 Hausdorff 距离出现 4 种情况这一结论推广到空间曲线/曲面间, 并利用作者自己开发的代数方程组求解器获得了相应的 Hausdorff 距

离,同时以平面曲线为例给出了 Hausdorff 距离的几何含义:  $A$  对  $B$  的单向 Hausdorff 距离,可看作是对曲线  $B$  进行等距变换,当曲线  $A$  恰好完全包含在曲线  $B$  的等距线内时,此时的等距距离为  $A$  对  $B$  的单向 Hausdorff 距离. 2010 年,Chen 等<sup>[3]</sup>针对文献[1-2]中非线性方程组求根方法的不足,研究了计算两 B 样条曲线间 Hausdorff 距离的几何裁剪方法,算法给出了 Hausdorff 距离出现在曲线端点的充分条件,利用曲线分割技术和滚动圆裁剪方法,将两曲线的 Hausdorff 距离计算问题转化为点和曲线的最小或最大距离计算问题,从而提高了算法的稳定性和计算效率. 同年, Kim 等<sup>[7]</sup>针对两平面自由曲线提出了一种精确计算 Hausdorff 距离的实时算法. 首先采用  $G^1$  连续的双圆弧,在给定公差下对自由曲线进行逼近;进而对这些圆弧进行距离映射并保存到图形硬件深度缓冲器;并对大部分多余的圆弧段进行裁剪,从而提高了 Hausdorff 距离的计算效率. 2011 年, Bai 等<sup>[8]</sup>提出了计算平面曲线间 Hausdorff 距离的折线方法,该方法首先根据给定容差将连续曲线离散为折线段,进而将曲线间的 Hausdorff 距离计算转化为连续折线段间的 Hausdorff 距离计算,同时针对无用的折线段给出了两种裁剪策略,极大地提高了这一复杂问题的计算效率. 2013 年, Ko<sup>[9]</sup>从理论上深入分析了平面曲线间 Hausdorff 距离计算的复杂性问题.

综合现有文献来看,针对连续曲线或曲面间的 Hausdorff 距离计算问题,近年来受到几何建模、计算几何、计算机图形学等领域学者的关注,同时也是 Hausdorff 距离在工程界获得广泛应用的瓶颈所在. 目前的研究工作存在如下问题: (1) 针对连续几何对象间 Hausdorff 距离计算的研究相对较少,尤其是针对自由曲线或曲面间的 Hausdorff 距离计算更少; (2) 大多学者通过对代数方程组的求解获得 Hausdorff 距离,并将代数方程组的求解当作了一个黑箱或采用标准求解器求解,未给出太多有参考价值的解算方法; (3)  $C^1$  连续平面曲线间的单向 Hausdorff 距离会出现 4 种情况,由于 4 种情况对应的非线性方程组并不相同,需要利用代数方程组求解器对 4 种情况对应的非线性方程组分别进行求解,这将严重影响曲线间 Hausdorff 距离的计算效率.

本文仅讨论平面连续曲线间 Hausdorff 距离的计算问题,算法分两步对平面曲线间的单向

Hausdorff 距离进行计算,最终选择优化结果中的最大值作为两平面曲线间的单向 Hausdorff 距离.

## 1 平面曲线间 Hausdorff 距离对应点的分类

Hausdorff 距离分单向和双向两种,对于两个非空的紧致集合  $A$  与  $B$ ,集合  $A$  中一个对象相对于集合  $B$  中所有对象的最小距离的最大值,称为集合  $A$  到集合  $B$  的单向 Hausdorff 距离,表示为

$$h(A, B) = \max_{a \in A} \min_{b \in B} d(a, b) \quad (1)$$

式中:  $d(a, b)$  为点  $a$  和点  $b$  间的欧几里得距离. 同理,集合  $B$  到集合  $A$  的单向 Hausdorff 距离可表示为

$$h(B, A) = \max_{b \in B} \min_{a \in A} d(a, b) \quad (2)$$

单向 Hausdorff 距离通常用来表示一个集合相对于另一个集合的最大偏差. 两个单向 Hausdorff 距离中的大者称为集合  $A$  与  $B$  的双向 Hausdorff 距离,简称 Hausdorff 距离,用来表示集合  $A, B$  间的最大偏差,亦可用来表示集合  $A$  与  $B$  的相似度. 双向 Hausdorff 距离通常表示为

$$H(A, B) = \max\{h(A, B), h(B, A)\} \quad (3)$$

从 Hausdorff 距离的定义可以看出,最小距离的确定是计算 Hausdorff 距离的基础. 由于双向 Hausdorff 距离与单向 Hausdorff 距离的算法相似,下面仅进行  $h(A, B)$  的计算讨论.

文献[1]对  $C^1$  连续的平面曲线间可能对应单向 Hausdorff 距离的点进行如下分类:

A 类端点: 即曲线  $A$  对曲线  $B$  的单向 Hausdorff 距离  $h(A, B)$ , 出现在曲线  $A$  的一个端点和其在曲线  $B$  上的法向映射点或曲线  $B$  的一个端点处,如图 1(a) 所示.

B 类端点:  $h(A, B)$  出现在曲线  $B$  的一个端点和它在曲线  $A$  上的法向映射点处,如图 1(b) 所示.

双垂足点: 当  $h(A, B)$  的对应点发生在两条曲线内部,且对应点为一一对应时,即出现如图 1(c) 所示的双垂足点情形,因对应点均为垂足点,故称为双垂足点. 根据曲线间距离函数取得极值的必要条件,可以将曲线间双垂足点对应的约束方程表示为

$$\begin{aligned} (\mathbf{r}_A - \mathbf{r}_B) \cdot \mathbf{r}'_A &= 0 \\ (\mathbf{r}_A - \mathbf{r}_B) \cdot \mathbf{r}'_B &= 0 \end{aligned} \quad (4)$$

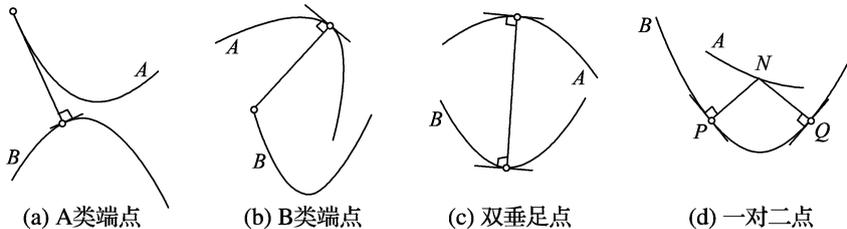


图1 平面曲线间 Hausdorff 距离对应点的分类

Fig. 1 Hausdorff distance between two planar curves and corresponding points

式中： $r_A$ 、 $r_B$  分别表示曲线 A、B 上对应点的矢径； $r'_A$ 、 $r'_B$  分别表示曲线 A、B 在对应点的切矢。上式表明两曲线在对应点处的切线平行，即存在公法线。

一对二点：当  $h(A, B)$  的对应点发生在两条曲线内部，且曲线 A 上一点对应曲线 B 上两点时，则出现如图 1(d) 所示的一对二点情形。一对二点也可以看作是曲线 A 与曲线 B 的中轴线的交点，以及中轴变换圆与曲线 B 的两个切点的总称。一对二点对应的约束方程可表示为

$$\|r_A - r_B(s)\| = \|r_A - r_B(t)\| \quad (5a)$$

$$(r_A - r_B(s)) \cdot r'_B(s) = 0 \quad (5b)$$

$$(r_A - r_B(t)) \cdot r'_B(t) = 0 \quad (5c)$$

式中： $s$ 、 $t$  分别为曲线 B 上两个对应点的曲线参数值；式(5a)表示曲线 A 上的点 N 到曲线 B 上点 P 和点 Q 的距离相等，式 5(b)、(c) 表示向量  $\mathbf{PN}$  和  $\mathbf{QN}$  分别与曲线 B 上 P、Q 两点处的切线垂直。

从平面曲线间单向 Hausdorff 距离对应的 4 种可能点来分析，对于参数曲线，前两种情况均可看作是  $h(A, B)$  的对应点的曲线参数为曲线参数域的边界，只有当曲线为开曲线时才会出现这种情况。当  $h(A, B)$  的对应点均为各自曲线的内部点，若对应点为一一对应时，则为双垂足点情形；若曲线 A 上一点对应曲线 B 上两点时，则为一对二点情形。此时用曲线 B 的等距线包容曲线 A 时，等距线自身必发生自交现象，自交点即为点 N。上面所讨论的 4 种情形均是非退化时的情况，若考虑到退化情况会更复杂。例如，一对二点情形中的 P 点刚好是曲线 B 上的一个曲率极值点时，此时曲线 B 上的对应点退化为一个，但又不同于双垂足点情形。

上面给出了平面曲线间单向 Hausdorff 距离对应的 4 种类型可能点，对于工程中遇到的实际曲线，曲线上局部满足 4 种类型点所对应约束条件的点有多个，通常需要分别计算所有可能点，并从中选取最大距离点作为  $h(A, B)$ 。由于双垂足

点和一对二点所对应的约束方程不同，通常需要对曲线 A 和 B 分别按照约束方程(4)和(5)进行计算，再从中选取最大值作为曲线 A 对 B 的单向 Hausdorff 距离。这样的计算显然效率并不高，因为同样的步骤，如曲线分段、求解约束方程组等，均需要进行两次计算。为此本文下面分两步对平面曲线间的 Hausdorff 距离进行计算。

## 2 曲线间单向 Hausdorff 距离的近似计算

为了计算曲线 A 到曲线 B 的单向 Hausdorff 距离，本文首先对曲线 A 进行离散化处理，获取曲线 A 上的  $n$  个离散点。由于仅需要获得两曲线间  $h(A, B)$  的近似解，最终的计算精度取决于下一节的精确计算，故离散间隔不必太小。对曲线 A 的离散化处理可以简单地按照等参数间隔进行，也可考虑曲线 A 的曲率特点，在曲率较大的位置设置较多的离散点，在曲率较小的位置设置较少的离散点。然后分别计算各个离散点到曲线 B 的最短距离，选取若干个距离较大，且满足曲线 A 上相邻点到曲线 B 的距离呈“小大小”的点，作为曲线间单向 Hausdorff 距离的近似解。这样，平面曲线间单向 Hausdorff 距离的近似计算就转化为点到曲线间最短距离的计算问题。

设平面内一定点  $P$  和一条曲线  $C: r = r(t)$ ,  $t \in [0, 1]$ ,  $t$  为曲线的一般参数。点  $P$  到曲线  $C(t)$  的欧几里得距离可表示为

$$D(t) = \|P - r(t)\| \quad (6)$$

对距离函数式(6)求导并令其为零，可得距离函数取得极值的必要条件为

$$\frac{dD(t)}{dt} = (P - r(t)) \cdot r'(t) = 0 \quad (7)$$

上面距离函数取得极值的必要条件，对应着点  $P$  到曲线  $C$  局部的距离最大值和最小值。该式表明在距离函数取得极值时，定点  $P$  与它在曲线

C 上的对应点的连线必与曲线 C 在该点的切矢正交. 从所有这些极值中选出最小值, 即为点 P 到曲线 C 的全局最小距离. 问题的关键是如何快速、稳定地求解非线性式(7). 该方程的计算方法通常可以分为局部数值迭代和全局计算方法两大类. 局部迭代方法包括常用的牛顿迭代法、二分法、黄金分割法等. 局部迭代方法需要为每个根提供较好的初值点, 且在有些情况下可能产生振荡, 因此很难适用于不便提供初值的多值计算问题. 全局计算方法通常有代数与混合技术、同伦方法和细分方法<sup>[10]</sup>, 其中细分方法由于其在多值问题、稳定性和效率方面的优点, 受到国内外学者的广泛关注. 本文采用细分方法计算点到曲线间的最短距离.

细分方法起源于通过多边形割角 (corner cutting) 获得光滑曲线的朴素思想, Chaikin<sup>[11]</sup> 在 1974 年 Utah 大学举行的 CAGD 国际会议上给出了一种基于割角思想快速生成曲线的算法. 之后 Riesenfeld 和 Forrest 从理论上证明了这种细分极限曲线的数学本质是均匀二次 B 样条曲线, 由此揭开了均匀 B 样条与细分这个新兴理论的联系<sup>[12]</sup>.

此处, 曲线 C 采用 Bezier 形式. 若曲线为 B 样条、NURBS 或其他参数多项式曲线, 均可转化为 Bezier 曲线形式. 设曲线 C 为 n 次 Bezier 曲线, 其表达式为

$$r(t) = \sum_{i=0}^n b_i B_{i,n}(t); t \in [0, 1] \tag{8}$$

$$B_{i,n}(t) = C_n^i (1-t)^{n-i} t^i; i = 0, \dots, n$$

式中: 多项式系数  $b_i$  为控制点;  $B_{i,n}(t)$  为 Bernstein 基函数;  $C_n^i$  为组合数;  $t$  为曲线参数.

曲线 C 关于  $t$  的一阶导数可表示为

$$r'(t) = \frac{dr(t)}{dt} = n \sum_{i=0}^{n-1} (b_{i+1} - b_i) B_{i,n-1}(t); t \in [0, 1] \tag{9}$$

将式(8)、(9)代入式(7), 由文献[10], 可得

$$(P - r(t)) \cdot r'(t) = n \sum_{i=0}^{2n-1} \left( \sum_{j=\max(0, i-n+1)}^{\min(n, i)} \frac{C_n^i C_{n-1}^{i-j}}{C_{2n-1}^j} \times (P - b_i) \cdot (b_{j+1} - b_j) \right) \times B_{i, 2n-1}(t) = 0 \tag{10}$$

式(9)、(10)表明, Bezier 曲线的一阶导数, 以及两个 Bezier 曲线的乘积仍为 Bezier 曲线, 只是该多项式曲线的次数和系数 (即控制点) 发生了变化. 将式(10)简写为

$$f(t) = \sum_{i=0}^m c_i B_{i,m}(t) = 0 \tag{11}$$

式中:  $m = 2n - 1$ ;  $c_i = n \left( \sum_{j=\max(0, i-n+1)}^{\min(n, i)} \frac{C_n^i C_{n-1}^{i-j}}{C_{2n-1}^j} (P - b_i) \cdot (b_{j+1} - b_j) \right)$ . 由于 Bernstein 多项式具有线性精度特性, 式(11)中的参数  $t$  可以表示为系数在  $[0, 1]$  上均匀分布的  $m$  次 Bernstein 多项式的加权和, 即

$$t = \sum_{i=0}^m \frac{i}{m} B_{i,m}(t) \tag{12}$$

以参数  $t$  为水平轴, 函数  $f(t)$  为垂直轴, 则可构造出如下曲线:

$$f(t) = \begin{pmatrix} t \\ f(t) \end{pmatrix} = \sum_{i=0}^m \begin{pmatrix} i/m \\ c_i \end{pmatrix} B_{i,m}(t) \tag{13}$$

式中:  $(i/m \ c_i)^T$  为控制点.

至此, 多项式求根问题即转化为一个 Bezier 曲线与其参数轴求交这样一个几何问题. 本文采用文献[10]中的 PP 算法对式(13)进行求解, 该算法主要依赖于 Bernstein 形式的多变元多项式细分算法和二维点集的凸壳算法, 算法主要包含如下 3 个步骤:

①首先需要构造 Bezier 曲线的凸壳, 并计算凸壳与参数轴的交点;

②利用 de Casteljau 细分算法, 剔除参数域中不含有根的区域;

③若剩余的参数域足够小, 则返回该参数作为一个根. 若经多次剔除不含有根的参数域后, 参数域仍非足够小, 则表明该参数域中包含一个以上的根, 需要利用 de Casteljau 细分算法将该参数域一分为二后重新返回步骤①.

利用上面点到曲线最短距离的算法, 可以获得曲线 A 上各离散点到曲线 B 的最短距离. 选择其中距离较大, 且满足曲线 A 上相邻点到曲线 B 的距离呈“小大小”的点对作为近似解.

### 3 曲线间 Hausdorff 距离的精确计算

将前面获得的若干距离较大, 且满足曲线 A 上相邻点到曲线 B 的距离呈“小大小”的点对作为初始点, 根据初始点附近曲线的形状与位置特点, 构造相应的 Hausdorff 距离的精确算法. 如图 2 所示, 设 P、Q 分别为曲线 A 和 B 上的关于  $h(A, B)$  的一组对应点, 假设前面对曲线 A 的离散化处理遵循数据采样中的采样定理<sup>[13]</sup>, 则精确的

Hausdorff 距离对应点一定会出现在初始解附近,并随着两曲线的形状与位置不同,表现出多种形式:双垂足点、一对二点、A 类端点和 B 类端点,下面分别讨论这 4 种情形.

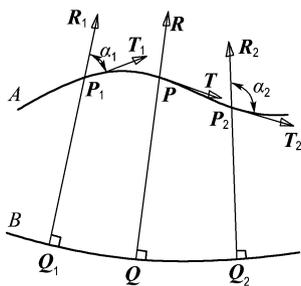


图 2 双垂足点的计算

Fig. 2 Computation of antipodal points

### 3.1 双垂足点的计算

双垂足点包含两曲线间距离的局部最大值和最小值所对应的点对,由于前面的初步计算中仅选取了距离较大的点作为初始点,此处不会出现最小距离所对应的双垂足点.如图 2 所示,点  $P$ 、 $Q$  分别为曲线  $A$ 、 $B$  上近似的 Hausdorff 距离对应点对, $t$ 、 $s$  分别为曲线  $A$ 、 $B$  的参数. $P_1(t_1)$ 、 $P_2(t_2)$  为曲线  $A$  上点  $P$  的左右相邻离散点, $T_1$ 、 $T_2$  为两点处曲线的单位切矢, $Q_1(s_1)$ 、 $Q_2(s_2)$  为它们在曲线  $B$  上的最短距离对应点,且满足“小大小”约束所对应的不等式: $\|P_1Q_1\| \leq \|PQ\| \geq \|P_2Q_2\|$ . $R_1 = P_1 - Q_1$ 、 $R_2 = P_2 - Q_2$  为两曲线上对应点间的矢量, $\alpha_1$ 、 $\alpha_2$  分别为矢量  $R_1$ 、 $R_2$  与矢量  $T_1$ 、 $T_2$  之间的夹角,从图 2 可以看出  $\alpha_1$  和  $\alpha_2$  中一个为锐角,另一个为钝角.若  $Q_1$  和  $Q_2$  两点间的距离  $\|Q_1Q_2\|$  或它们的参数间隔  $\Delta s = |s_1 - s_2|$  小于某个预先设定的常数,则可认为曲线  $A$ 、 $B$  间单向 Hausdorff 距离的对应点为双垂足点.以  $P_1$ 、 $P_2$  两点的曲线参数  $t_1$ 、 $t_2$  为初值,构造函数

$$f_1 = \alpha_1(t_1) - \pi/2; f_2 = \alpha_2(t_2) - \pi/2 \quad (14)$$

由于  $f_1 \cdot f_2 < 0$ ,可通过二分算法来计算两曲线间精确的局部最大距离所对应的双垂足点.

### 3.2 一对二点的计算

参考图 3,若点  $P$ 、 $Q$  分别为曲线  $A$ 、 $B$  上近似的 Hausdorff 距离对应点对,且其邻近离散点满足不等式  $\|P_1Q_1\| \leq \|PQ\| \geq \|P_2Q_2\|$  约束,但  $Q_1$  和  $Q_2$  两点间的距离  $\|Q_1Q_2\|$  或它们的参数间隔  $\Delta s = |s_1 - s_2|$  大于预先设定的常数,则可认为曲线  $A$ 、 $B$  间 Hausdorff 距离的对应点为一对二点.仍

以  $P_1$ 、 $P_2$  两点的曲线参数  $t_1$ 、 $t_2$  为初值,构造函数

$$\begin{aligned} f_1 &= \text{sgn}[\alpha_1(t_1) - \pi/2](d_1^1 - d_1^2) \\ f_2 &= \text{sgn}[\alpha_2(t_2) - \pi/2](d_2^1 - d_2^2) \end{aligned} \quad (15)$$

式中: $\text{sgn}[x]$  为符号函数,根据  $x$  的正负取“+”或“-”; $d_1^1$ 、 $d_1^2$  为点  $P_1$  到曲线  $B$  的最短和次最短距离; $d_2^1$ 、 $d_2^2$  为点  $P_2$  到曲线  $B$  的最短和次最短距离.由于  $f_1 \cdot f_2 < 0$ ,可通过二分算法来计算两曲线间精确的局部最大距离所对应的一对二点.最终精确的 Hausdorff 距离的对应点一定会表现为曲线  $A$  上的一个点  $P$  对应曲线  $B$  上的两个垂足点  $Q_1$ 、 $Q_2$ ,且  $\|PQ_1\| = \|PQ_2\|$ .

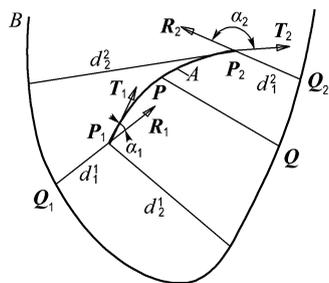


图 3 一对二点的计算

Fig. 3 Computation of intersection point between one curve and self-bisector of other curve

### 3.3 A 类端点与 B 类端点的计算

若曲线  $A$  或  $B$  为开曲线,最终的 Hausdorff 距离可能发生在曲线的端点.为此,可直接利用第 2 章关于点到曲线最短距离的算法,计算开曲线的两个端点到另一曲线的最短距离.至此,利用本章算法可实现平面曲线间 Hausdorff 距离计算中出现的各种形式点的精确计算,与第 2 章算法结合,则可形成平面曲线间 Hausdorff 距离的完整算法.

## 4 数值算例

通过两个算例来验证上述算法的可行性.

**例 1** 曲线  $A$  为 Bezier 开曲线;曲线  $B$  为三次均匀 B 样条表示的封闭曲线,控制点个数为 90.现在要计算曲线  $A$  到  $B$  的单向 Hausdorff 距离  $h(A, B)$ .计算过程分两个步骤:首先进行  $h(A, B)$  的近似计算,将曲线  $A$  按照其参数进行均匀十等分离散化处理,利用文献[14]的算法将曲线  $B$  由 B 样条形式转换为三次分段 Bezier 曲线,然后利用第 2 章的方法计算各离散点到曲线  $B$  的最短距离;从 11 个最短距离中选取距离较

大,且满足“小大小”特点的点,利用第3章的算法进行 Hausdorff 距离的精确计算,计算精度为  $10^{-10}$ . 计算结果如图4所示. 图4(a)给出了曲线A上的11个离散点,以及与第2个离散点对应的曲线B上的14个距离极值点;图4(b)给出了曲线A上11个离散点中与曲线B最小距离较大,且满足“小大小”特征的两个点和曲线B上对应的垂足点;图4(c)表达了经过 Hausdorff 距离的精确计算后对应点的分布,两个距离分别为 66.019 839 mm 和 54.720 453 mm,开曲线A的两端点到曲线B的最短距离分别为 27.011 021 mm 和 13.104 928 mm;对图4(c)中的两个距离以及开曲线的两端点到曲线B的最短距离进行比较,最后的单向 Hausdorff 距离为 66.019 839 mm,对应点如图4(d)所示.

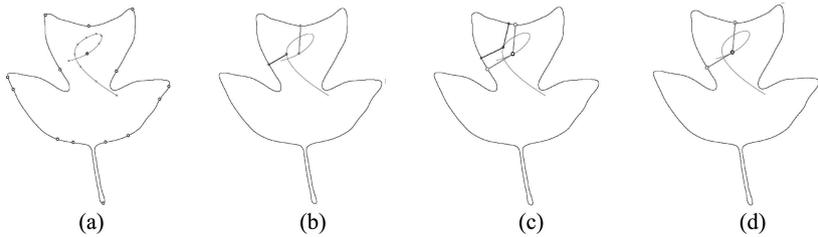


图4 开曲线与封闭曲线间的 Hausdorff 距离计算

Fig. 4 Computation of HD between an open curve and a closed curve

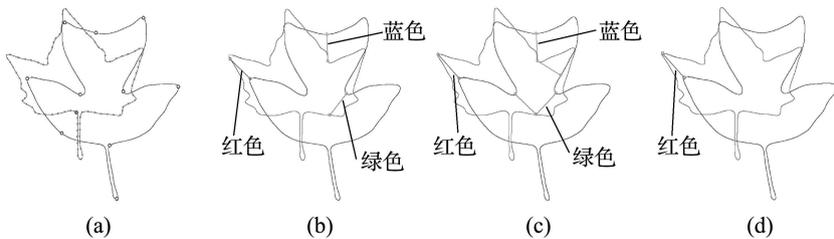


图5 两封闭曲线间的 Hausdorff 距离计算

Fig. 5 Computation of HD between two closed curves

## 5 结 论

(1)本文分两步对平面曲线间单向 Hausdorff 距离进行计算. 第一步将其中一条曲线进行离散处理,并计算各离散点到另一条曲线的最小距离,从中选择若干个距离较大且满足“小大小”特征的点对,作为近似的 Hausdorff 距离对应点;第二步以各近似点对作为初值,依据各点处曲线的特点,建立相应的局部优化模型并求解,从而获得平面曲线间精确的 Hausdorff 距离. 该方法克服了传统的针对4种情况分别求解不同非线性方程组的缺点,简化了计算过程.

**例2** 曲线A、B均为立方均匀B样条表示的封闭曲线,控制点个数均为90,计算精度为  $10^{-10}$ . 现在要计算曲线A到B的单向 Hausdorff 距离  $h(A,B)$ . 计算过程与例1相同,计算结果如图5所示. 图5(a)给出了曲线A上的90个离散点,以及第20个离散点对应的曲线B上的10个距离极值点;图5(b)给出了曲线A上90个离散点中与曲线B最小距离较大,且满足“小大小”特征的前三个点和曲线B上对应的垂足点;图5(c)表达了精确 Hausdorff 距离的计算结果,其中包含一个双垂足点和两个一对二点,3个距离(按红绿蓝顺序)分别为 77.874 235、70.736 729 和 66.611 425 mm;对图5(c)中的3个距离进行比较,最后的单向 Hausdorff 距离为 77.874 235 mm,对应点如图5(d)所示.

(2)通过近似和精确计算单向 Hausdorff 距离的两步算法,将平面曲线间 Hausdorff 距离的计算转化为点到曲线的最小距离计算问题;同时利用细分算法可以快速、稳定地获得点到曲线的最短距离,提高了计算效率和稳定性,数值算例验证了本文所提方法的正确性.

## 参考文献:

- [1] Alt H, Scharf L. Computing the Hausdorff distance between curved objects [J]. *International Journal of Computational Geometry & Applications*, 2008.

- 18(4):307-320.
- [2] Elber G, Grandine T. Hausdorff and minimal distances between parametric freeforms in  $R^2$  and  $R^3$  [J]. **Lecture Notes in Computer Science**, 2008, **4975**:191-204.
- [3] Chen X D, Ma W Y, Xu G, *et al.* Computing the Hausdorff distance between two B-spline curves [J]. **Computer-Aided Design**, 2010, **42**(12):1197-1206.
- [4] Scholz E. Felix Hausdorff and the Hausdorff edition [J/OL]. [2012-12-09]. <http://www.math.uni-wuppertal.de/~scholz/preprints/ENLbioHaus.pdf>.
- [5] Huttenlocher D P, Kedem K. Computing the minimum Hausdorff distance for point sets under translation [C] // **SCG '90 Proceedings of the Sixth Annual Symposium on Computational Geometry**. New York:ACM, 1990:340-349.
- [6] Huttenlocher D P, Klanderman G A, Rucklidge W J. Comparing images using the Hausdorff distance under translation [J]. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, 1993, **15**(9):850-863.
- [7] Kim Y J, Oh Y T, Yoon S H, *et al.* Precise Hausdorff distance computation for planar freeform curves using biarcs and depth buffer [J]. **The Visual Computer**, 2010, **26**(6-8):1007-1016.
- [8] Bai Y B, Yong J H, Liu C Y, *et al.* Polyline approach for approximating Hausdorff distance between planar free-form curves [J]. **Computer-Aided Design**, 2011, **43**(6):687-698.
- [9] Ko K. On the complexity of computing the Hausdorff distance [J]. **Journal of Complexity**, 2013, **29**(3-4):248-262.
- [10] Patrikalakis N M, Maekawa T. **Shape Interrogation for Computer Aided Design and Manufacturing** [M]. New York:Springer, 2001:73-108.
- [11] Chaikin G M. An algorithm for high speed curve generation [J]. **Computer Graphics and Image Processing**, 1974, **3**(4):346-349.
- [12] 李保军. 指数多项式曲线细分重构与插值细分曲面快速计算[D]. 大连:大连理工大学, 2009.  
LI Bao-jun. Construction of exponentials reproducing subdivision schemes and rapid evaluation of interpolatory subdivision surfaces [D]. Dalian: Dalian University of Technology, 2009. (in Chinese)
- [13] Castleman K R. **Digital Image Processing** [M]. New Jersey:Prentice-Hall, 1996.
- [14] Piegl L, Tiller W. **The NURBS Book** [M]. 2nd ed. Berlin:Springer, 1997:142-228.

## Computation of Hausdorff distance between planar curves

CAO Li-xin\*, DONG Lei, CAO Jing-jing

( School of Mechanical Engineering, Dalian University of Technology, Dalian 116024, China )

**Abstract:** The difficulty of computing the Hausdorff distance(HD) between planar curves lies in solving different nonlinear equations for four kinds of special cases which are encountered in this computing process. To simplify the process of calculation, a two-step method for computing the HD between planar curves is proposed. The first step of the method is sampling the curve  $A$ , and calculating the minimum distance between each discrete point on the curve  $A$  and the curve  $B$ . Then, selecting several points from the discrete points as the approximate solutions, which correspond to the larger minimum distance and the minimum distances of each point together with its adjacent two points obeying the 'small-large-small' order. The second step is identifying which case the approximate solution belongs to according to the shape and position of the curves, establishing corresponding optimization model and finding the local optimal solution. Comparing these local optimal solutions, the final directed HD with the largest minimum distance will be acquired. By using this method, the computing process of the HD between planar curves can be converted into the computation of the minimum distance between a point and a curve, which can improve the computational efficiency and stability of the algorithm. The feasibility of the algorithm has been verified by two numerical examples.

**Key words:** Hausdorff distance; planar curves; minimum distance