

一种求多边形平移重叠面积最大值的快速算法

刘俊义 王润生

(国防科技大学 ATR 国家重点实验室,长沙 410073)

摘要 设 P 和 Q 是平面上的 2 个简单多边形, $t \in R^2$ 是平面上任意矢量, 多边形 P 与 Q 的平移重叠面积函数定义为 $Ar(t) = Area(P \cap (t+Q))$, 这里 $t+Q$ 表示 Q 平移了 t 后形成的多边形。为快速求解平移重叠面积函数的最大值, 本文提出了一种优化计算策略, 它包括在全局上组合应用遗传算法和最速上升算法快速搜索函数最大值和在局部上利用修正的扫描线算法来快速计算函数值。

关键词 扫描线填充算法, 遗传算法, 多边形, 平移重叠面积

1 概述

目标的形状往往可以由简单多边形来近似表达, 从而目标间的相互关系(如是否相交, 是否相似等)就可由其相对应的多边形之间的相互关系来表征。一般而言, 若 2 个多边形相交, 则根据其相互重叠程度的不同, 我们能得到的信息也不相同。

最近 Mount 等人^[1]研究了下述问题: 在平面上给定 2 个简单多边形, 当一个多边形相对于另一个多边形作平移运动时, 以它们相互重叠部分的面积作为一种测量, 来定量描述这 2 个多边形相交的程度。上述相互重叠部分的面积作为平移矢量的函数(文献中称为多边形的平移重叠面积函数)。他们对该函数的数学性质和计算性质作了详尽的理论分析, 并指出了值得进一步研究的 4 个问题, 但没有给出实验结果。

本文主要针对上述 4 个问题中的第 2 个问题, 即是否可以不必计算平移重叠面积函数在定义域中的全部值, 就可以快速求得该函数的最大值。我们采用下述策略来建立快速算法: 在全局上组合应用遗传算法和最速上升算法快速搜索函数的最大值, 在局部上利用修正的扫描线算法来快速计算函数值。

这个问题有许多重要的应用。例如, 在探测法目

标识别(Object Recognition by Probing)的应用中, 通过放置探针, 测量其是否“击中(hit)”目标来确定目标的形状, 此时, 确定放置探针的最优位置以获得关于目标形状的最大信息量的问题, 其最终的数学模型是求使平移重叠面积函数达到最大值的最优平移矢量^[1]。

2 定义和结论

为叙述问题方便, 给出如下定义: 设 P 和 Q 是 2 个简单多边形, $t = (t_x, t_y)$ 是平面上的任意矢量(我们称之为“位置”), 则 P 与 Q 的平移重叠面积定义为 $Ar(t) = Area(P \cap (t+Q))$, 其 P, Q 中指由其内部和边界组成的点集, $Area(\cdot)$ 指求集合在平面上占有的面积, $t+Q$ 表示 Q 平移了 t 后形成的多边形所成的点集, “ \cap ”为集合求交。

文献[1]对 $Ar(t)$ 函数的数学性质和计算性质作了详细的理论分析, 下面列出几个有用的结论(证明略)。

设 P 和 Q 是 2 个简单多边形, $Ar(t)$ 是 P 和 Q 的平移重叠面积函数, 又令 $S = \{t \in R^2 \mid Ar(t) \neq 0\}$, $T = \{t \in R^2 \mid Ar(t) = 0\}$, 则有:

(1) $S = Int(P) - Int(Q)$, 其中 $Int(\cdot)$ 表示多边形的内集。

- (2) S 是连通的, T 不一定连通。
- (3) $Ar(t)$ 是连续函数。
- (4) 函数 $Ar(t)$ 是 R^2 上的次数不超过 2 的分片多项式 (piecewise polynomial) 函数。

我们的目的是只需计算 $Ar(t)$ 函数在少数位置的值, 就能快速求得函数的最大值 Ar^* 和最大值出现的最优位置 t^* 。

3 局部优化: 修正的扫描线填充算法

局部优化的目标是快速计算某个位置处的平移重叠面积函数值。该问题实质上是多边形如何快速求交的问题, 计算机图形学中关于这方面有许多高效而成熟的算法^[2], 多边形扫描线填充算法即是其中之一, 由于它充分利用了多边形区域的连贯性、扫描线的连贯性及边的连贯性, 从而算法的计算效率很高。

通过对经典的扫描线算法作简单的修正, 便可得出计算 2 个多边形的重叠面积的快速算法。算法的步骤如下:

首先, 将 2 个多边形的端点按逆时针顺序排列, 且计算 2 个多边形的外接矩形, 若这 2 个外接矩形不相交, 则重叠面积为 0, 否则继续。

然后, 为 2 个多边形建立统一的边表。算法所采用的边的数据结构如下:

```
typedef struct tagEDGE{
    int ym;
    long x,dx;
    char flag;
    struct tagEDGE *next;
}EDGE;
```

其中, ym 是多边形某非水平边的上端点的纵坐标, x 是该边下端点的横坐标, dx 是该边斜率的倒数, $flag$ 是新增加的元素 (在经典的扫描线填充算法中不出现), 当该边是上升边时置为 1, 当该边是下降边时置为 -1, $next$ 用来指向与该边的下端点纵坐标相同的其它非水平边。在这部分中, 算法为 2 个多边形的每条非水平边填写上述数据结构, 并进行奇异点处理, 同时将具有相同下端点纵坐标的边链入同一个链表中, 所有的这些链表组成了所谓的“Y 桶”。

最后, 按纵坐标升序地将“Y 桶”中的每个链表依次取出, 将链表中的边按下端点的横坐标升序排列 (若横坐标相同则按 dx 升序排列), 从而形成有

效边表, 并由 AET 指向表头, 然后利用下列程序计算重叠面积。

```
EDGE *AET, *e;
char flag;
int area=0;
...
//对“Y 桶”中的每个由 AET 指向的有效边表计算面积
e=AET; flag=0;
while( e!=NULL ){
    flag+=e->flag;
    if( flag==2 ) area+=(int)e->next->x
        -(int)e->x+1;
    e=e->next;
}
```

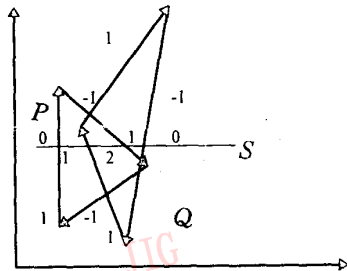


图 1
Fig. 1

图 1 说明程序的合理性。图中显示出 2 个三角形 P 和 Q 相交的情况。水平线 S 为某条扫描线, 其上标注了每段的 $flag$ 值, 可见 $flag=2$ 的段对应为 P 和 Q 相交的部分。三

角形端点顺时针排列, 每条边的 $flag$ 值也标在了图上。

4 全局优化: 遗传算法和最速上升算法

由前面讨论可知, 简单多边形 P 和 Q 的平移重叠面积 $Ar(t)$ 函数具有如下特点: 函数的支集是连通的; 函数是连续的; 函数是以 t 的横轴分量 t_x 和纵轴分量 t_y 为变量、次数不超过 2 的分片多项式曲面。然而, 尽管 $Ar(t)$ 函数有着简单的数学形式, 但是由于多边形的形状和尺寸的任意性, 使得 $Ar(t)$ 函数的形状也复杂多变, 且往往具有多个极大值。

求函数最大值的直接方法是计算函数定义域中的所有 $Ar(t)$ 值, 然后统计出最大值及其出现的位置。然而, 尽管采用扫描线算法可提高计算效率, 但是当多边形的外接矩形较大时, 函数的定义域也很大, 导致计算全部函数值所需的计算量仍然很大。为有效地解决这个问题, 需要研究只计算函数在少数位置的值, 就可以求得函数的最大值的方法。

我们采用了首先由遗传算法搜索函数出现某个近似最大值的位置,然后从这个位置开始利用最速上升算法继续搜索最大值的方法(由于是找最大值,所以我们将最速下降算法中沿负梯度方向搜索改为沿正梯度方向搜索,同时将这样的方法称为“最速上升算法”。

遗传算法是由 Holland^[3]创建的,群体搜索策略和群体中个体之间的信息交换是其两大特点,它的优越性表现为:即使适应度函数很不规则,搜索过程也不容易陷入局部最优,只能是以很大的概率找到全局最优解。但是,遗传算法本身固有的“过早收敛”问题会阻碍最优解的求得,此时收敛到的解并非最优解,而是位于全局最优解附近的近似最优解,如果我们将这个近似解对应的位置作为最速上升算法的起点,则可能很快找到全局最优解,这样既避免了遗传算法的局限性,又避免了最速上升算法易陷入局部极值的局限,而且计算效率有很大的提高。

算法的步骤描述如下:

(1) 在 $Ar(t)$ 函数的定义域中随机取 N 个初始位置 $\{x_i, y_i\}_{i=1}^N$ 作为基因,并对它们进行编码,给定正整数 M, K , 且使 $m=0$ 。

(2) 对上一代基因群进行复制 (Reproduction)、交叉 (Crossover)、变异 (Mutation) 操作,形成新一代,其中可以采用所谓的“精英策略”。

(3) 利用修正的扫描线算法计算新一代中各基因所对应的平移重叠面积,将其作为适应度函数的值,记录新一代中适应度值最大的基因。

(4) 若这个适应度最大的基因在 K 代进化过程中不再发生变化或者遗传总代数 $m > M$, 则认为遗传算法收敛,输出这个基因对应的位置 $\{x_{opt}, y_{opt}\}$, 作为梯度上升算法的初始位置; 否则, $m = m + 1$, 转(2)。

(5) 从 $\{x_{opt}, y_{opt}\}$ 出发,利用 $Ar(t)$ 函数在 x 方向和 y 方向的前向差分近似该点的偏导数,并由此解得梯度,然后沿正梯度方向搜索,记录得到的更优位置,当这个位置在 L 步迭代中不在发生变化,转(6)。

(6) 输出最大值及最优位置,结束。

关于算法的几点说明:

(1) 为采用“精英策略”,一般应将 N 取为奇数,基因进行交叉、变异后,从生成的新一代中选出最“好”的基因,而且这个基因被无条件复制一次,不参加交叉、变异。

(2) 用最速上升算法搜索最优位置时,采取下述方法:平移重叠面积函数可写作 $Ar(t_x, t_y)$, 令 $Ar(t_x, t_y)$ 在 t_x 和 t_y 方向的一阶前向差分分别为:

$$\nabla_x(t_x, t_y) = Ar(t_x + 1, t_y) - Ar(t_x, t_y)$$

$$\nabla_y(t_x, t_y) = Ar(t_x, t_y + 1) - Ar(t_x, t_y)$$

则搜索过程中利用下式:

$$t_x(n + 1) = t_x(n) + \eta(n) \nabla_x(t_x, t_y)$$

$$t_y(n + 1) = t_y(n) + \eta(n) \nabla_y(t_x, t_y)$$

$$\eta(n + 1) = \eta_0 / (n + 1)$$

其中 $\eta(n)$ 为迭代速率, η_0 为初始迭代速率, n 为迭代步数。

(3) 下面考虑该方法的计算量问题。设用修正的扫描线算法求面积所用平均时间为 T , 则遗传算法的计算量为 $2m(N-1)T$, 其中 m 为总进化代数。梯度法的计算量为 $3nT$, 其中 n 为总迭代步数。同时,我们可以将计算得到的函数值保存下来,以便后面搜索过程使用,这样可以进一步减小计算量。

5 试验结果

我们对上述思想作了 2 个试验。

试验一中取的 2 个多边形如图 2 所示,所选的坐标系 x 轴向右 y 轴向下,其中多边形 Q 相对于多边形 P 作平移运动,同时 2 个多边形端点的坐标列在表 1 中:

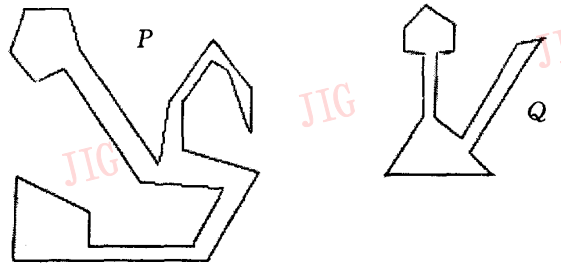


图 2 试验一中的个简单多边形

Fig. 2 Two simple polygons used in experiment 1

表 1 2 个简单多边形的端点数据

Table 1 The vertex data of two simple polygons

	端点 1	端点 2	端点 3	端点 4	端点 5	端点 6	端点 7
多边形 P	63,17	8,44	10,103	27,60	37,125	55,50	111,127
多边形 Q	50,33	34,68	65,95	99,79	93,42		

我们利用修正的扫描线算法计算了 $Ar(t)$ 函数在其定义域内的所有值。图 3 是 $Ar(t)$ 的等值线图

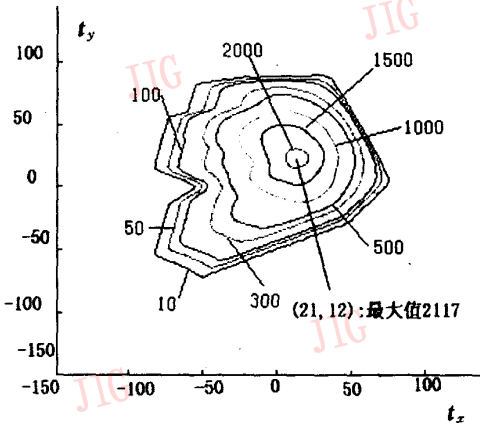


图 3 试验一中平移重叠面积函数的等值线图
Fig. 3 The contour map of area of overlap function in experiment 1

(图中的数字是等值线的值),图 4 是 $Ar(t)$ 的三维曲面图(不同的灰度表示不同的高度)。从图中我们可知函数的在 $(21, 12)$ 取得最大值 2 117,用遗传算法搜索最大值时,取基因个数 $N=5$,交叉概率为 0.6,变异概率为 0.03,最大进化代数 $M=500$,算法收敛最小代数 $K=30$,实验表明经过 56 代的进化,算法已经将函数的最大值找到,当然后面的最速上升算法也没有找到更好的解。

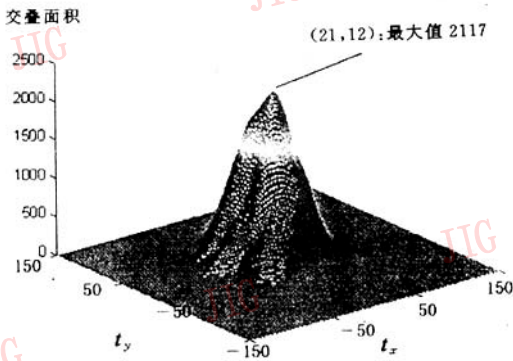


图 4 试验一中重叠面积函数的三维表面图
Fig. 4 The 3D surface map of the overlap-area function in experiment 1

实验 1 中的平移重叠面积函数只有一个极大值,所以搜索较为容易。实验 2 中所考虑的 2 个多边形如图 5 所示,同样,函数 $Ar(t)$ 的等值线图和三维表面图分别由图 6 和图 7 给出。已知函数在 $(3, -46)$ 处取得最大值 1 060,采用与实验一相同的

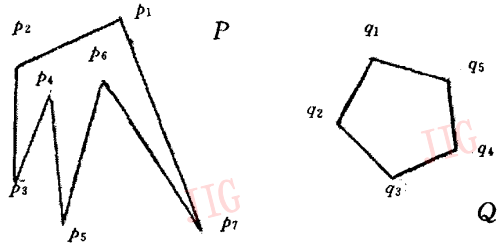


图 5 试验二中的 2 个简单多边形
Fig. 5 Two simple polygons used in experiment 2

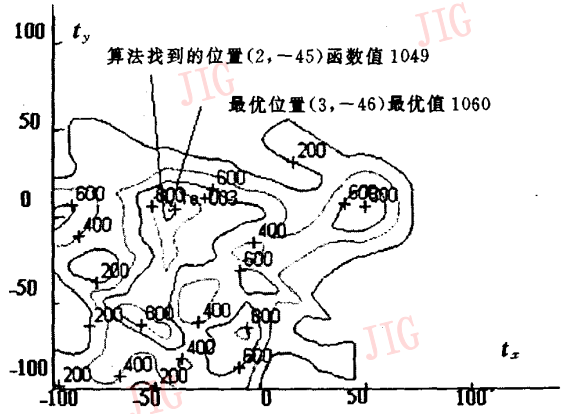


图 6 试验二中平移重叠面积函数的等值线图
Fig. 6 The contour map of area of overlap function in experiment 2

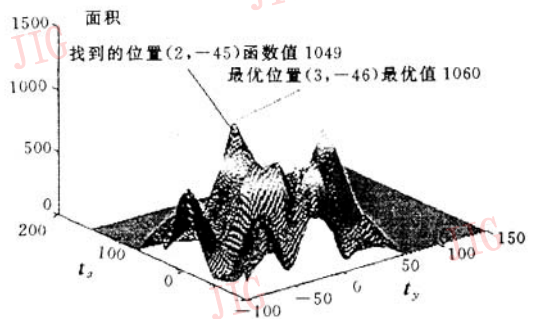


图 7 试验二中重叠面积函数的三维表面图
Fig. 7 The 3D surface map of the overlap-area function in experiment 2

参数,用遗传算法计算 112 步后收敛,得到了较优位置 $(5, -36)$,此时函数值为 903,显然遗传算法此时出现了“过早收敛”,然后由此位置作为初值,取 η_0

为0.4,算法收敛最小步数 $L=10$,用梯度法迭代 21 步求得的近似最优解为 $(2, -45)$,此时的最大值是 1 049。可以看到,此时与最大值点十分接近,这样的精度足够满足工程应用的需要了。

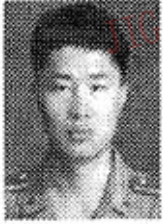
参考文献

1 Mount D M, Silverman R, Wu A Y. On the Area of Overlap of

Translated Polygons, Computer Vision and Image Understanding, 1996, 64(1): 53~61.

2 唐荣锡,汪嘉业,彭群生. 计算机图形学教程. 北京: 科学出版社, 1994.

3 王润生. 图象理解. 长沙: 国防科技大学出版社, 1995.



刘俊义,国防科技大学电子工程学院在读博士生。主要研究方向为图象分析和理解、模式识别和信息融合。

王润生,1964年毕业于军事工程学院电子工程系,现为国防科技大学电子工程学院教授、博士生导师,中国图象图形学会常务理事。主要研究方向为:图象分析与理解、模式识别和信息融合等。

An Efficient Algorithm for Computing the Maximum of Translated Overlap-Area Function of Two Simple Polygons

Liu Junyi, Wang Runsheng

(ATR State Key Lab, National University of Defence Technology Changsha, Hunan 410073)

Abstract Given two polygons P and Q in a plane and a translation vector $t \in R^2$ the area-of-overlap function of P and Q is defined as $Ar(t) = Area(P \cap (t+Q))$, where $t+Q$ denotes the new polygon formed by translating Q with t . In order to rapidly compute the maximum value of the area-of-overlap function, an optimal computing strategy is developed in this paper, which includes both combining genetic algorithm with gradient algorithm for rapidly global searching the maximum value of the function and using modified scan-line algorithm for computing the value of the function efficiently in local range.

Keywords Scan-line fill algorithm, Genetic algorithm, Polygon, Area-of-overlap