一种凸多边形直径算法*

An Algorithm for Calculating Diameter of Convex Polygon

张显全，刘丽娜，唐振军

Zhang Xianquan，Liu Lina，Tang Zhenjun

（**广西师范大学计算机科学系，广西桂林** 541004）

（Dept. of Comp. Sci.，Guangxi Normal Univ.，Guilin，Guangxi，541004，China）

摘要：在研究凸多边形性质的基础上，构建一种新的凸多边形直径算法。该算法首先计算凸多边形顶点 x 坐标、y 坐标的极值点，然后通过极值点将凸多边形分为几个区域，最后计算这些不同区域中顶点的距离可得凸多边形的直径。该算法简单，运行效率高。

关键词：凸多边形　区域　直径　计算几何

中图法分类号：TP301.6　　**文献标识码**：A　　**文章编号**：1002-7378(2005)04-0199-03

Abstract: The properties of the convex polygon are discussed and a algorithm for the diameter of convex polygon based on its properties is proposed. In the algorithm, the extremum points at the x and y coordinates are calculated, and the convex polygon is divided into several regions by the extremum points. The diameter of convex polygon is obtained by computing of the distances of vertexes in different regions. The present algorithm is simple, effective and applicable.

Key words: convex polygon, region, diameter, computation geometry

计算平面点集的直径问题是计算几何的基本问题之一，在计算机图形学、图像处理、CAD/CAM、模式识别等众多领域中有广泛地应用。对给定平面点集 $P=\{P_1,P_2,\cdots,P_n\}$，若对每对点对进行计算，则需计算 $n(n-1)/2$ 个点对的距离，然后选取这些距离中的最大值即可得直径。该算法简单，但是运算量较大。解决平面点集的直径问题是先求出平面点集的凸壳，它是包含点集的最小凸集，即以点集中部分点为顶点的一个凸多边形，求出凸多边形的直径就是平面点集的直径。计算给定平面点集凸壳算法中，R. L. Graham[1]给出一个确定平面点集凸壳算法：首先确定点集 y 坐标最小的点，计算该点到其他点连线与水平线的夹角，按角的大小进行排序，然后得出凸壳上的点。Preparata 和 Hong[2]把分治技术应用于求解凸壳问题，把点集分成大小近似相等的 2 个子集，递归求出 2 个子集的凸壳，通过计算 2 个凸壳的并求出最终的凸壳。周培德等[3]证明线段集的平面凸包的时间复杂度为 $O(n\log n)$，然后将线段转换成平面简单多边形链，通过简单多边形链计算凸壳。崔国华等[4]研究排序与凸壳之间的内在联系，建立以排序算法为基础利用双动线检测方法确定平面点集凸壳。T. Chen[5]任取点对计算由点对确定直线的斜率，找出斜率的中值，通过中值把平面点集分为 2 个部分，应用递归算法求出凸壳，另一种算法是对点集进行分组，分别求出每组的凸壳，然后通过计算凸多边形的直径并求出整个点集的凸壳。这些计算平面点集凸壳的算法，其时间复杂度均为 $O(n\log n)$，因而平面点集的直径问题可转化为凸多边形的直径问题。

在计算凸多边形的直径算法中，周培德等[6]通过计算夹角序列求凸多边形的直径，夏舒杰等[7]对其进行改进和完善，利用夹角正切值符号和余弦值符号序列来求解凸多边形直径，这些求直径算法的时间复杂度为 $O(n^2)$。本文在研究凸壳性质的基础上，构建一种新的凸多边形直径算法。该算法首先计算凸多边形顶点的 8 个极值点，再根据这些极值点把凸多边形划分为直角三角形区域，计算凸壳的直

收稿日期：2005-06-29

作者简介：张显全(1964-)，男，副教授，主要从事图像处理和计算机图形方面的研究工作。

* 广西自然科学基金(No. 0447035)资助项目。

径时只须通过对不同区域中的顶点进行计算，可得到凸多边形的直径。该算法简单，运行的效率高。

1 凸多边形的性质

凸壳是一凸多边形，对于凸多边形的任意一条边，所有不在该边上的顶点都在该边的同一侧，具有任意2个顶点间的线段都在多边形内部，凸壳的内角小于180°等性质。下面对凸多边形的极值和区域进行研究。

设 $P=\{P_1,P_2,\cdots,P_M\}$ 为凸多边形上的顶点，编号为顺时针方向依次进行；在点集 P 的横坐标最小的集合中，记纵坐标最大的点为 Q_{xY}，纵坐标最小的点为 Q_{xy}；在点集 P 的横坐标最大的集合中，记纵坐标最大的点为 Q_{XY}，纵坐标最小的点为 Q_{Xy}；在点集 Q 的纵坐标最小的集合中，记横坐标最大的点为 Q_{yX}，横坐标最小的点为 Q_{yx}；在点集 Q 的纵坐标最大的集合中，记横坐标最大的点为 Q_{YX}，横坐标最小的点为 Q_{Yx}，第1个下标表示是那一个坐标的极值，第2个坐标是表示在第1个坐标的极值集合中另一个坐标的极值，大写表示极大值，小写表示极小值。对这些点进行定义。

定义1 凸多边形中 $Q_{xy},Q_{xY},Q_{Xy},Q_{XY},Q_{yx},Q_{yX},Q_{Yx},Q_{YX}$ 这8个点称为该凸多边形的极值点。其中，Q_{xy} 与 Q_{xY}、Q_{Xy} 与 Q_{XY}、Q_{yx} 与 Q_{yX} 和 Q_{Yx} 与 Q_{YX} 称为同类极值。

对于这些极值点，过 Q_{yX}、Q_{Yx} 作 y 轴垂线，过 Q_{xy}、Q_{Xy} 作 x 轴垂线，凸多边形的顶点均在矩形内部，通过不同类而相邻的2个极值点的连线把矩形内部分为几个部分，最多有5个部分，如图1(a)；在这5个部分中，对于不是极值点的凸多边形的顶点，他们一定在以不同类而相邻2个极值点间的线段为斜边的直角三角形中，如图1(a)中的区域1、2、3和4；若存在极值相等，可能出现3类情况，第一类如图1(a)，第二类如图1(b)，第三类如图1(c)，图1中(b)和(c)是(a)的特殊情况，最少有3个部分，如图1(c)，在图1(c)中区域2和4除极值点外不含凸多边形的其他顶点。下面对极值点所在的区域进行定义。

定义2 在凸多边形中，在以不同类而相邻2个极值点间线段为斜边的直角三角形中，所有顶点称为同域点集。

对于凸多边形，最多有4个直角三角形区域，凸多边形上的顶点分别在这些区域中，对同域点集中2个顶点间的距离，有如下定理。

定理1 在凸多边形的同域点集中，两极值点间的距离最大。

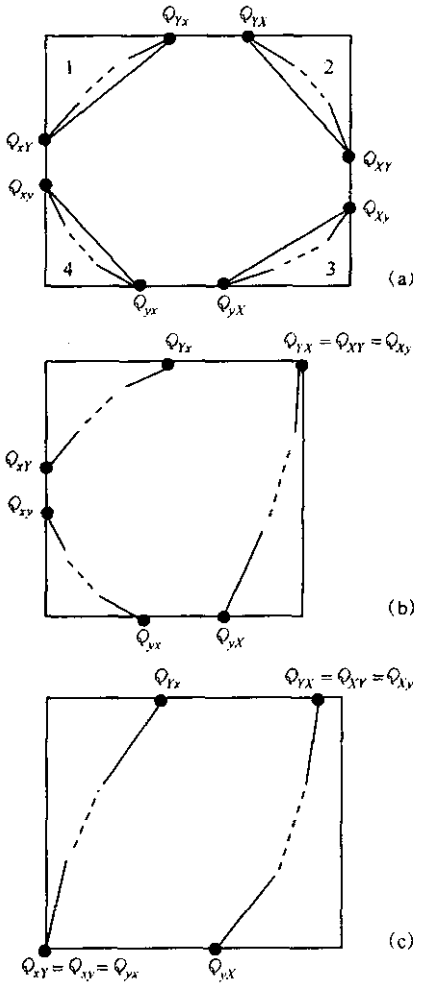

图1 凸壳的极值点和区域

(a)5个区域；(b)4个区域；(c)3个区域；1、2、3、4为区域编号。

证明 图1中，由极值点把凸多边形的顶点分为几个区域，这些区域是由直角三角形组成，极值点间的线段是这些直角三角形区域的斜边，以区域1为例，设直角三角形区域的3个顶点分别为 A、Q_{xY} 和 Q_{Yx}，三角形的斜边为 $Q_{xY}Q_{Yx}$，P_i、P_j 为该同域点集中任意2个顶点，则 P_i、P_j 一定在三角形内部，直线 P_iP_j 与直角三角形交于 B、C 两点(见图2)，根据凸多边形的性质，Q_{xY}、Q_{Yx} 在直线 P_iP_j 的同侧，∴ $|AB|$

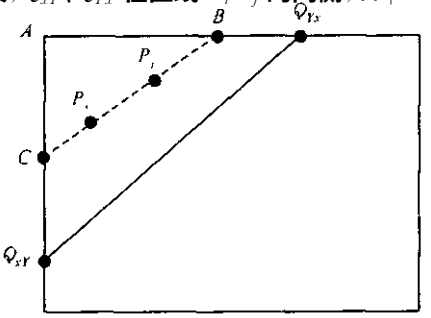

图2 同域点集中顶点的最大距离

$< |AQ_{Yx}|$，$|AC| < |AQ_{xY}|$，因而有 $|BC| < |Q_{xY}Q_{Yx}|$，又 $\because |P_iP_j| < |BC|$，$\therefore |P_iP_j| < |Q_{xY}Q_{Yx}|$，同理可证其他同域点集，定理证毕。

2 凸多边形的直径算法

由于凸多边形的顶点都属于几个同域点集，对于同域点集由上节定理知，2 个极值点间的距离最大，因而在求凸多边形的直径时，无需计算和比较同域点集中的顶点之间的距离，只需计算和比较不属于同一个同域点集之间顶点的距离，求出最大距离就可得出凸多边形的直径。设 4 个同域点集分别为 Q_1、Q_2、Q_3 和 Q_4，首先计算 Q_1 中的每个点到 Q_2、Q_3 和 Q_4 中顶点的最大距离，其次计算 Q_2 中的每个点到 Q_3 和 Q_4 中顶点的最大距离，第三步计算 Q_3 中的每个点到 Q_4 中顶点的最大距离，最后与每个同域点集中 2 个极值点间的距离进行比较可得凸多边形的直径。具体算法如下。

步骤 1：计算凸壳顶点 $P=\{P_1,P_2,\cdots,P_M\}$ 的 8 个极值点 $P_{m_1}=Q_{xY}$、$P_{m_2}=Q_{Yx}$、$P_{m_3}=Q_{YX}$、$P_{m_4}=Q_{XY}$、$P_{m_5}=Q_{Xy}$、$P_{m_6}=Q_{yX}$、$P_{m_7}=Q_{yx}$ 和 $P_{m_8}=Q_{xy}$；求出相临不同类极值间的距离：$d_a=|P_{m_1}P_{m_2}|$、$d_b=|P_{m_3}P_{m_4}|$、$d_c=|P_{m_5}P_{m_6}|$、$d_e=|P_{m_7}P_{m_8}|$，$f=1$；

步骤 2：IF $(f>5)$ 转到步骤 6；//顶点的最大直径计算结束

ELSE $i=m_f$，$j=m_{f+2}$，$d_i=|P_iP_j|$，转到步骤 3；//计算下一三角区域

步骤 3：$j=j+1$，$d=|P_iP_{j+1}|$；

IF $(d>d_i)d_i=d$，转到步骤 4；

步骤 4：IF $(j=m8)i=i+1$，转到步骤 5；//计算下一顶点的最大直径

IF $(j=M)$ $j=0$；//到下标最大的点转到步骤 3；

步骤 5：IF $(i=m_{f+1})f=f+2$；转到步骤 2；//到下一个三角区域

IF $(i>M)i=1$，转到步骤 3；//到下标最大的点

步骤 6：$i=2$，$d=d_i$，转到步骤 7；

步骤 7：IF $(d<d_i)d=d_i$，转到步骤 8；

步骤 8：$i=i+1$；

IF $(i\leqslant M)$，转到步骤 7；

ELSE 转到步骤 9；

步骤 9：$d=\mathrm{Max}\,(d_a,d_b,d_c,d_e,d)$ 为凸多边形的直径，结束。

设 P_i、P_j 两点的坐标为 (x_i,y_i) 和 (x_j,y_j)，两点间的距离为：$d=\sqrt{(x_j-x_i)^2+(y_j-y_i)^2}$，为了提高运算速度，在进行距离计算时，可通过计算 d^2 进行比较，这样不需要进行开方运算，若坐标为整形则可转化为整数运算，运算效率可进一步提高。

3 复杂度分析

在本文的凸多边形直径算法中，计算 8 个极值点的时间复杂度为 $O(n)$，由于对凸多边形每个同域点集中的顶点都需要与其他域中的顶点进行距离计算，因而计算每个顶点与其他域中顶点的最大距离的时间复杂度为 $O(n^2)$，根据每个点的最大距离求直径的时间复杂度为 $O(n)$，所以整个算法的复杂度为 $O(n^2)$，与文献[7]的算法复杂度相同，但是通过极值点把凸多边形上的顶点分为几个同域点集，每个同域点集中任意两点间的距离不用计算，因而减少了运算量，提高了效率。对于平面点集可先求出其凸壳，然后再用本文算法求出直径。由于平面点集凸壳算法的时间复杂度为 $O(n\log n)$，因而用本文算法计算平面点集直径的时间复杂度仍为 $O(n^2)$。

4 结束语

本文对凸多边形和平面点集进行了大量的实验，都能正确求出他们的直径，说明通过极值确定了凸多边形的 4 个区域，根据这一性质计算凸多边形的直径算法简单可行。对每个同域点集中的点不需进行距离计算，直接用 2 个极值点间的距离进行比较，因而减小了计算点集的规模，提高了运算速度，算法有较好的性能，具有较强的实用性。

参考文献：

[1] Graham R L. An efficient algorithm for determine the convex hull of a finite linear set [J]. Information Processing Letters, 1972, 1(1): 132-133.

[2] Preparata F P, Hong S J. Convex hulls of finite sets of points in two and three dimensions [J]. CACM, 1977, 20(2): 87-93.

[3] 周培德. 寻求平面上线段集的凸壳的算法[J]. 工程图学学报, 2003, 23(2): 116-119.

[4] 崔国华，洪 帆，余祥宣. 确定平面点集凸包的一类最优算法[J]. 计算机学报, 1997, 20(4): 330-334.

[5] Chan T. Optimal output-sensitive convex hull algorithms in two and three dimensions [J]. Discrete Comput Geom, 1996, 16(3): 361-368.

[6] 周培德，周忠平. 求凸多边形直径和算法[J]. 工程图学学报, 1996, 16(2): 29-32.

[7] 夏舒杰，陆国栋，谭建荣. 基于夹角序列的凸多边形直径算法[J]. 计算机工程与应用, 2002, 38(22): 65-67.

（责任编辑：邓大玉）