

基于有向边的 Java 手机多边形算法研究

徐 巍, 陈东方

(武汉科技大学 计算机科学与技术学院, 湖北 武汉 430081)

摘 要:手机终端的地图处理是移动位置服务系统(LBS)的核心问题之一。为了在手机及其他资源受限设备上解决填充多边形的问题,提出了基于有向边的顺序分割法进行处理,即为多边形的边定义方向。将多边形外轮廓的方向定义为顺时针;内边(即分割边,下同)被分割出的两个三角形共用,其在相邻的两三角形内方向相反。所以被分割出的三角形的方向都是顺时针的。再按照边的方向和顺序,选择满足条件的分割边将多边形依次分割为若干三角形。Java 手机中的实验结果表明顺序法分割效率高,消耗资源少,因此非常适用于手机等资源受限设备。

关键词:多边形;三角形;J2ME;顺序分割法

中图分类号:TP391

文献标识码:A

文章编号:1673-629X(2008)08-0105-03

Research of Polygon on Java Phone Based on Directed Line

XU Wei, CHEN Dong-fang

(College of Computer Science & Technology, Wuhan University of Science & Technology, Wuhan 430081, China)

Abstract: Mobile map service is an important technology in location based service (LBS). An orderly segmentation method is put forward to fill polygons on resource limited devices in this paper. This method defines direction of polygonal contour as clockwise and directions of inside lines which are shared by two triangles are opposite. Then, it chooses lines which satisfy condition to segment polygon, according to the direction and order of lines. Orderly segmentation method is efficiency and costs less, so it adapts to resource limited devices well.

Key words: polygon; triangle; J2ME; orderly segmentation method

0 引 言

随着 WAP 技术和手机终端的发展,我国手机产业近年保持了高速发展的势头。但针对手机的应用业务需求也在日益增加,现有的手机硬件仍然显得捉襟见肘。执行效率自然也就成为衡量一个手机程序优良与否的关键。

在手机地图服务等手机应用系统中,多边形区域是一种非常常用的区域,而 J2ME 却并没有提供绘制多边形的 API,仅仅提供了绘制直线和填充三角形等 API。由于手机的处理能力、内存、电源等资源有限,研究如何能正确并高效地处理多边形是非常必要的。前人在三角剖分方面提出过一些算法^[1~3],但判定条件较为复杂,都主要运用于 PC 平台上。帅小应^[4]等首次提出了利用二分法在资源受限平台上处理填充多边形,但占用资源多,效率不高。

这种方法不仅占用大量额外空间来存储中间变量,还必须在每次分割后分别判断分割后的两个是否为三角形。随着多边形边数的增加,这两点表现的更为明显。文中提出的顺序分割法,能实现任意凹多边形和凸多边形的分割,解决了占用资源过多和效率不高的问题,可应用于手机地图服务等手机应用服务系统。

1 填充多边形算法

二分法的算法思想是对于任意的填充多边形 $P < P_0, P_1, \dots, P_n > (n \geq 3)$ 选择不相邻的两顶点 P_i 和 P_j , 连接 P_i 和 P_j 将多边形分成两个与 P 不相交的内多边形 P_1 和 P_2 , 再递归分割新生成的 P_1 和 P_2 , 直到 P_1 和 P_2 为三角形为止。

由于二分法每次分割时,是选择的任意不相邻的两顶点,因此每次分割完后都要判断分割出来的是否是三角形。同时,需要很多的额外空间来存储分割出的中间多边形。那么分割一个 n 边形时,需要进行 $2 * (n - 3)$ 次判断,需要存储 $n - 4$ 个中间临时多边形。随着 n 的增加,程序的执行时间和内存的开销都会显著增

收稿日期:2007-11-11

基金项目:湖北省教育科研重点项目(2004D006)

作者简介:徐 巍(1983-),男,硕士研究生,研究方向为图形图像;陈东方,副教授,博士,研究方向为科学计算可视化、图形学与虚拟现实。

大。

对于手机这样的资源受限终端来说,执行效率会降低很多。甚至可能让手机死机。因此,研究如何进行快速有效的分割是必要的。

1.1 顺序分割法

1.1.1 基本概念

定义 1 两线段相交是指两线段交点位于两条线段端点之间;反之,若至少位于一条线段两端点之外,称为不相交。

定义 2 线段在多边形内是指线段两端点间所有点(两端点除外)都位于多边形内。如果试添加的分割线段位于多边形 P 内,且与多边形任意边不相交,则称为正确的分割线段。

定义 3 边的方向是指将每条边看作是由起点指向终点的有向线段,那么多边形便是由有向边首尾连接而成。文中将多边形外轮廓的方向定义为顺时针。多边形的内边(即分割边,下同)被分割出的两个三角形所共用,其在相邻的两三角形内方向相反。所以被分割出的三角形的方向都是顺时针的。

定义 4 边的可用性是指某边是否能在后续分割中使用。因为多边形的外边在分割时只可能被用一次,而内边要被用两次。那么在分割时,若外边被使用一次,内边被使用两次后,则将其视为不可用的。

1.1.2 判定法则

法则 1 点的包含性检验。经过平面上的一点 P_0 向任意方向作一条射线与多边形相交,若交点数为偶数则点 P 一定位于多边形外,否则必位于多边形内。取当前分割边的中点,判定是否为原多边形外的点。若是外点,则试添加的分割线段在原多边形外。

法则 2 相交判定^[5]。分两步判定两条线段是否相交。

第一步:快速排斥实验。

先求出两线段的界限框(包含某线段且四边分别平行于 X 轴和 Y 轴的最小矩形),如图 1 所示。

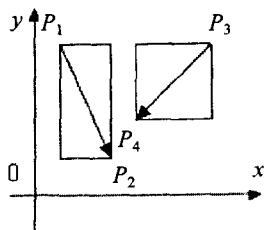


图 1 排斥实验图

如果两线段的界限框不能同时在 X 方向和 Y 方向相交,则这两条线段不可能相交。因此通过快速排斥实验,可以确定两条线段是否有相交的可能。

第二步:跨立实验。

确定线段 P_1P_2 (或 P_3P_4) 是否跨立另一线段 P_3P_4 (或 P_1P_2) 所在的直线。所谓跨立是指某线段的两个端点是否分别处于另一线段所在的直线的两旁,或者是否有其中一个端点处于另一线段所在的直线上。其设计思想是运用叉积的方法来确定。从 P_1 出发向另一线段的两个端点 P_3 和 P_4 引出两条辅助线段 P_1P_3 、 P_1P_4 。

然后计算两个叉积:

由 $(P_3 - P_1) * (P_2 - P_1)$ 确定 P_1P_3 在 P_1P_2 的哪个方向;由 $(P_4 - P_1) * (P_2 - P_1)$ 确定 P_1P_4 在 P_1P_2 的哪个方向。若两个叉积的正负号相同,说明 P_1P_3 、 P_1P_4 同在 P_1P_2 的一边,即 P_3P_4 不能跨立 P_1P_2 所在的直线,如图 2(b) 所示;若两个叉积的正负号相反,说明 P_1P_3 、 P_1P_4 分别在 P_1P_2 的两边,即 P_3P_4 跨立 P_1P_2 所在的直线,如图 2(a) 所示。若任何一个叉积为 0,由于 P_1P_2 和 P_3P_4 已经通过了快速排斥实验,因此 P_3 、 P_4 两点中必有一点处于线段 P_1P_2 所在的直线上,如图 2(c) 所示。

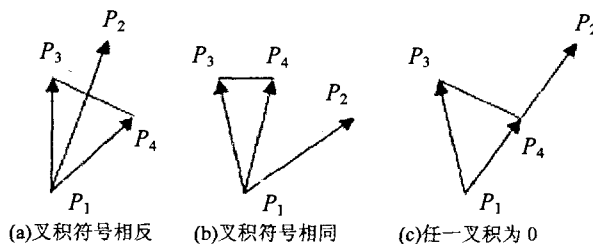


图 2 跨立实验图

两条线段相交当且仅当它们能够通过快速排斥实验并且每条线段能够跨立另一线段所在的直线。

1.1.3 算法描述

任意一个填充 n 边形 P , 其顶点按一定方向有序排列为 $\langle P_0, P_1, \dots, P_{n-1} \rangle$ ($n \geq 3$), 进行如下操作:

① 选择顺序相邻的三个顶点: P_i, P_{i+1}, P_{i+2} (初始 $i = 0$), 使 $P_{i+2}P_i$ 为当前边, 判断该边是否在其他边内 && 是否与其他边(包括新分割边, 下同)都不相交。若否则 $i++$, 重复①; 若是则 $\text{numtriangs}++$ (初始 $\text{numtriangs} = 1$), 将用到的边由可用边集合中删除, 并将当前边加入可用边集合, 转换当前边方向。转②;

② 在可用边集合中寻找其终点等于当前边起点的边, 若有则连接当前边的终点和可用边的起点, 并使该边为当前边, 转③; 若无则寻找其起点等于当前边终点的边, 重复③;

③ 判断该边是否在其他边内 && 是否在其他边都不相交。若否则将当前边复原, 转②; 若是则 $\text{numtriangs}++$, 将用到的边由可用边集合中删除, 并将当前边加入可用边集合, 转换当前边方向。(若 $\text{numtriangs} = n - 2$ 则程序结束), 转②。

2 实验过程

2.1 算法实现

①类说明。程序主类 OrderlySegment 及 Line 类和 Point 类。如图 3 所示。

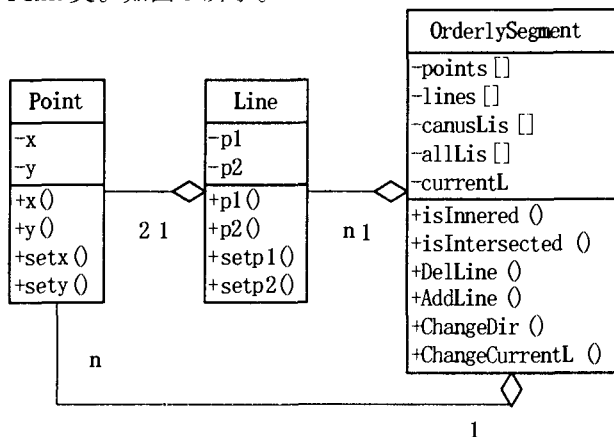


图 3 类图

类图中列出了主要的属性和操作。

②函数说明。Point 类中 $x()$ 、 $y()$ 、 $setx()$ 、 $sety()$ 四个函数用来设定及获取多边形顶点的坐标, Line 类中的函数类似。isInnered() 函数使用法则 1 判断当前边是否在多边形内。在实现时,射线的具体做法是以该边中点为始端做一条与 x 轴平行的线段,线段的末端的 x 坐标为 0, y 坐标与始端的 y 坐标相等。isIntersected() 函数使用法则 2 判断当前边是否和多边形相交。DelLine() 函数将不再可用的边由可用边集合中删除。AddLine() 函数向可用边集合和所有边集合添加当前边。

③数据结构说明。points[] 存放顶点的坐标。lines[] 存放多边形外边。canusLis[] 存放可用边。allLis[] 存放所有边。所有边包括分割边和外边,在判断当前边是否和其他边相交时使用。

2.2 实验结果

根据算法在联想家悦 D2040A 机器上利用 J2DK1.4.2 与 WTK2.2 平台测试了基于有向边的顺序分割法的分割过程和分割效率。

2.2.1 凹多边形测试

凹多边形的顶点按顺时针排列为 $P_0(10,30)$ 、 $P_1(110,30)$ 、 $P_2(110,80)$ 、 $P_3(80,50)$ 、 $P_4(60,70)$ 、 $P_5(60,90)$ 、 $P_6(80,80)$ 、 $P_7(70,110)$ 、 $P_8(10,110)$, 如图 4(a) 所示。分割过程为:

①选择 $P_0P_1P_2$, 取 P_2P_0 为当前边, 不满足条件, $i++$;

②选择 $P_1P_2P_3$, 取 P_3P_1 为当前边, 满足条件, 加入可用边集合, 删除边 P_1P_2 和 P_2P_3 , $numtriangs++$, 使 P_3P_1 转换方向;

③找到终点与 P_1P_3 的起点相等的可用边 P_0P_1 , 连接 P_3P_0 , 使其为当前边, 满足条件, 加入可用边集合, $numtriangs++$, 删除边 P_1P_3 和 P_0P_1 并转换 P_3P_0 方向;

④找到终点与 P_0P_3 的起点相等的可用边 P_8P_0 , 连接 P_3P_8 , 使其为当前边, 不满足条件, 将分割边还原为 P_0P_3 。

同理可找到其他满足条件的分割边。直到 $numtriangs == 7$, 程序结束。最后填充结果如图 4(b) 所示。

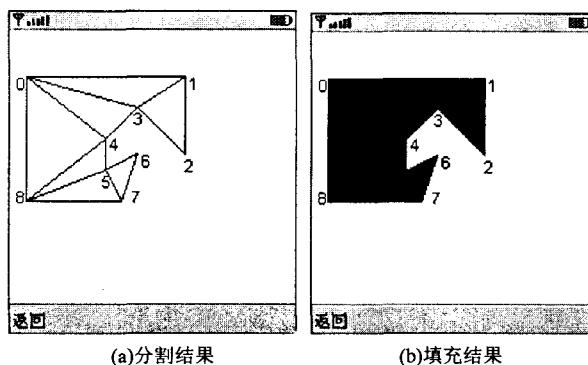


图 4 实验结果

2.2.2 对比测试

与文献[4]中的二分法进行了对比实验。并对两者在时间(单位:ms)和内存消耗(单位:byte)两方面进行了比较。实验结果如表 1 所示。

表 1 对比结果表

边数	5	6	7	8	9	10
顺序法用时	10	20	30	42	50	66
二分法用时	10	20	30	50	60	80
用时差	0	0	0	8	10	14
顺序法内存	50112	50180	52304	52380	52456	52536
二分法内存	51024	53236	53336	53436	53560	53684
内存消耗差	912	3056	1032	1056	1104	1148

2.3 结果分析

测试中显示,凹多边形处理时间为 44ms。使用 WTK Profiler 测得主要操作判断当前边是否在多边形内及是否与其他边都不相交的时间分别占总时间的 22% 和 48%。使用 Nokia S40 DP2.0 SDK1.1 Diagnostics 工具测得内存使用峰值为 53560 字节。同样在 Nokia7270 手机上测试通过。对比测试时,由于二分法必须每次分割后判断分割出的是否是三角形,还需将分割出的临时多边形保存起来,这样随着多边形边数的增加,顺序法的处理时间增量小于二分法的处理时间增量,所消耗的内存也较少。

(下转第 151 页)

5 结束语

以上基于 PC/104 硬件仿真平台的 16 位存储器容错技术,可有效对 RAM 的存储单元进行纠错检错处理,消除 SEU 的影响。

在整个小卫星仿真模拟过程中,该方法得到了多次很好的验证,可以满足当前的要求。同时将 FPGA 引入到整个硬件仿真过程当中,在 modelsim 仿真过程中取得了良好的效果。

随着 FPGA 在航天领域的广泛应用,采用 FPGA 完成整个检错纠错功能也成为一种趋势。当然这种纠错设计也有一些不足之处,它要求在存储器中加入 30% 左右的冗余存储器,同时由于检错纠错需要时间,CPU 不得不在存储读写时插入等待周期,从而一定程度上降低了处理器的性能。

因此,这里考虑在下一步的工作中将该检错容错方法和滞后校验结合起来,允许错误的数据进入处理器。这样,处理器在读取数据时就不需要等待检错纠错的结果,直接从存储器中读取,而数据的校验和纠正滞后于数据的读入。

(上接第 107 页)

3 结束语

基于有向边的顺序法是针对资源受限平台提出的,有目的地选择符合条件的分割边进行分割,避免了分割的盲目性,运行效率高,占用资源少,适合手机、PDA 等资源受限设备。将其应用到地理信息系统手机终端的设计中,能较好地满足程序需要。同时也能应用到其他,如手机游戏地图等,需要填充多边形处理的应用服务中。

参考文献:

- [1] 马小虎,潘志庚,石教英.基于凹凸顶点判定的简单多边形

(上接第 110 页)

4 结束语

由以上算例仿真结果比较,可以看出文中的算法比文献[2]无论是在精度上还是在算法结束 CPU 所用的时间上更加有效。

参考文献:

- [1] Ge Renpu. A filled function method for finding a Global Minimizer of a function several variables [C]//Dundee Biennial conference on numerical analysis. Dundee, Scotland; [s. n.], 1981.

参考文献:

- [1] 余金陪,杨根庆.现场小卫星技术与应用[M].上海:上海科学普及出版社,2004.
- [2] Tang Ming, Bu—Sung, Yeo Chai—Kiat, et al. Dynamic replication algorithms for the multi—tier data grid[J]. Future Generation Computer Systems, 2005, 16(1): 21—39.
- [3] Kent S T. Logic and computer design fundamental[M].北京:电子工业出版社,1998.
- [4] Ciletti M D. VerilogHDL 高级数字设计[M].北京:电子工业出版社,2005.
- [5] Xilinx, Inc. Virtex—II Platform FPGA User Guide[R]. [s. l.]: Xilinx, Inc., 2005.
- [6] Wang J J, Katz R B. SRAM Based Re—programmable FPGA for Space Applications[J]. IEEE Transactions on Nuclear Science, 1999, 18(1): 14—20.
- [7] 胡 谋. 计算机容错技术[M].北京:中国铁道出版社,1998.
- [8] 陆如新.可编程专用集成电路原理设计和应用[M].北京:电子工业出版社,1997.
- [9] 潘 松,王国栋. VHDL 实用教程[M].成都:电子科技大学出版社,2000.

Delaunay 三角剖分[J]. 计算机辅助设计与图形学学报, 1999, 11(1): 1—3.

- [2] 杨 杰.基于凹凸顶点判定的简单多边形的三角剖分[J]. 小型微型计算机系统, 2000, 21(9): 974—975.
- [3] 刘 强,李德仁.基于二叉树思想的任意多边形三角剖分递归算法[J]. 武汉大学学报:信息科学版, 2002(5): 529—533.
- [4] 帅小应,廉东本. Java 手机多边形处理的研究[J]. 小型微型计算机系统, 2006, 27(2): 279—281.
- [5] 吴文虎,王建德. 实用算法的分析与程序设计[M]. 北京:电子工业出版社, 1998: 80—83.

- [2] Liang Y M, Zhang L S, Li M M, et al. A filled function method for global optimization[J]. Journal of Computational and Applied Mathematics, 2007, 205: 16—31.
- [3] Han Qiaoming, Han Jiye. Revised filled function methods for global optimization[J]. Applied Mathematics and Computation, 2001(119): 217—228.
- [4] Yang Y J, Shang Y L. A New Filled Function Method for Unconstrained Global Optimization[J]. Applied Mathematics and Computation, 2006(173): 501—512.
- [5] Zhang L S, NG C K, Li D, et al. A new filled function method for global optimization[J]. Global Optim, 2001, 20: 49—65.