



(12)发明专利

(10)授权公告号 CN 106444773 B

(45)授权公告日 2019.08.09

(21)申请号 201610938679.8

G06Q 10/04(2012.01)

(22)申请日 2016.10.25

(56)对比文件

(65)同一申请的已公布的文献号  
申请公布号 CN 106444773 A

US 8576235 B1,2013.11.05,  
CN 101738195 A,2010.06.16,  
张琦,等.基于简化可视图的环境建模方法.  
《东北大学学报(自然科学版)》.2013,第34卷(第  
10期),

(43)申请公布日 2017.02.22

(73)专利权人 大连理工大学  
地址 116024 辽宁省大连市高新园区凌工  
路2号

Huang H P,Chung S Y.Dynamic  
visibility graph for path planning.《IEEE  
International Conference on Intelligent  
Robots and Systems》.2004,

(72)发明人 吴振宇 聂政 李胜铭 李航  
冯林

审查员 吴垠

(74)专利代理机构 大连星海专利事务所有限公  
司 21208

代理人 王树本

(51)Int.Cl.

G05D 1/02(2006.01)

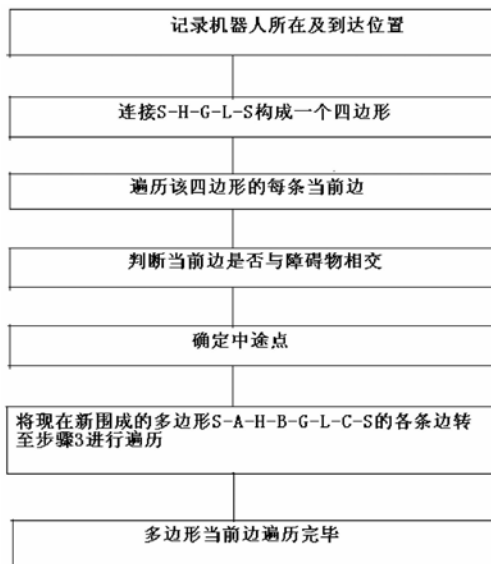
权利要求书1页 说明书5页 附图3页

(54)发明名称

一种基于递归简化可视图的环境建模方法

(57)摘要

本发明涉及一种环境建模方法,一种基于递归简化可视图的环境建模方法,包括以下步骤:  
(1)记录机器人所在及到达位置,(2)连接S-H-G-L-S构成一个四边形(3)遍历该四边形的每条当前边,(4)判断当前边是否与障碍物相交,(5)确定中途点,(6)将现在新围成的多边形S-A-H-B-G-L-C-S的各条边转至步骤3进行遍历,(7)多边形当前边遍历完毕。本发明最后得到一个多边形区域,该区域外的障碍物可以全部简化掉,区域内的才是机器人路径规划需要考虑的障碍物。本发明有效的减少可视线的数量,提高了后续机器人路径规划算法的执行效率。



1. 一种基于递归简化可视图的环境建模方法,包括以下步骤:

步骤1、记录机器人所在及到达位置,将机器人所在的位置记为点S,期望到达的位置记为点G,再将机器人所处的整个地图环境采用计算机进行扫描,并采用封闭多边形对每个障碍物进行包络,保留这些多边形;

步骤2、连接S-H-G-L-S构成一个四边形,连接点S和点G,线段SG称为机器人穿越线,此时穿越线SG会与多个障碍物多边形相交,记录这些相交的障碍物多边形2、7、8,随后在穿越线SG的两侧,分别找到一个点,该点必需满足以下两个条件:其一,是这些相交的障碍物多边形2、7、8中的一个顶点,其二,在穿越线SG的一侧范围内该点距离穿越线最远,并分别记为点H和点L,连接S-H-G-L-S构成一个四边形,其特征在于:

步骤3、遍历该四边形的每条当前边,即当前边SH、HG、GL、LS,每条当前边的端点分别为当前边起点和当前边终点,遍历到当前边HG时,当前边起点就是点H,当前边终点就是点G;

步骤4、判断当前边是否与障碍物相交,若不相交,即当前边GL,则将该当前边保存至活动区域集中,然后返回至步骤3继续遍历后续的当前边,当遍历完成时则跳转至步骤7;若相交,即当前边SH、HG、LS,则跳转至步骤5;

步骤5、确定中途点,在与当前边相交的所有障碍物中,取这些障碍物中离穿越线SG最远且与当前边同一侧的顶点,记为中途点A、B、C,分别连接当前边起点和中途点,中途点和当前边终点,形成2条新的当前边;原来的当前边SH由当前边起点S连接中途点A、再由中途点A连接当前边终点H形成由SA与AH两条边组成现在的当前边;原来的当前边HG由当前边起点H连接中途点B、再由中途点B连接当前边终点G形成由HB与BG两条边组成现在的当前边;原来的当前边LS由当前边起点L连接中途点C、再由中途点C连接当前边终点S形成由LC与CS两条边组成现在的当前边;

步骤6、将现在新围成的多边形S-A-H-B-G-L-C-S的各条边转至步骤3进行遍历;

步骤7、多边形当前边遍历完毕,活动区域集中所有的边所围成的多边形即为输出结果,活动区域外的障碍物多边形1、3、10、11不需要考虑,而活动区域内的障碍物多边形2、4、5、6、7、8、9就是机器人在路径规划时需要考虑的。

## 一种基于递归简化可视图的环境建模方法

### 技术领域

[0001] 本发明涉及一种环境建模方法,更具体地说,涉及一种基于递归简化可视图的环境建模方法。

### 背景技术

[0002] 移动机器人在一个存在多个障碍物的环境下移动时,就需要找出一条可以避开所有障碍物的由起点到终点的最优路径,比如路径最短,耗时最少,能耗最低。为了解决个问题,移动机器人在进行路径规划时,一般需要先对环境进行建模,已有的环境建模主要有:栅格图、切线图、Voronoi图、概率图和可视图等。

[0003] 基于可视图的建模方法将环境中的障碍物通过凸多边形包络表示,多边形的顶点连同移动机器人的起点和终点一起被看成质点,每个点的坐标在全局坐标系中作为已知,将起点和终点以及障碍物多边形的顶点进行可视化判断:如果两点连线组成的线段不与任何障碍物多边形相交,则称两点可视,该线段称为可视线。由于可视图具有简单直观的特性,基于可视图的建模方法成为了众多学者关注的热点。

[0004] 目前可视图简化的主要方法有两种,一个是Huang提出的动态可视图方法,它可以精简部分“冗余”的障碍物;另一个是改进的动态可视图法,由张琦提出,称之为简化可视图法,该方法通过设计必要的障碍物规则,可以更大限度的剔除无关障碍物。然而简化可视图法存在这样一个问题,即在复杂环境下可能出现错误简化障碍物的问题。

### 发明内容

[0005] 为了克服现有技术中存在的不足,本发明目的是提供一种基于递归简化可视图的环境建模方法。通过本发明提供的方法,最后可以得到一个多边形区域,该区域外的障碍物可以全部简化掉,区域内的才是机器人路径规划需要考虑的障碍物。本发明有效的减少可视线的数量,提高了后续机器人路径规划算法的执行效率。

[0006] 为了实现上述发明目的,解决现有技术中所存在的问题,本发明采取的技术方案是:一种基于递归简化可视图的环境建模方法,包括以下步骤:

[0007] 步骤1、记录机器人所在及到达位置,将机器人所在的位置记为点S,期望到达的位置记为点G,再将机器人所处的整个地图环境采用计算机进行扫描,并采用封闭多边形对每个障碍物进行包络,保留这些多边形;

[0008] 步骤2、连接S-H-G-L-S构成一个四边形,连接点S和点G,线段SG称为机器人穿越线,此时穿越线SG会与多个障碍物多边形相交,记录这些相交的障碍物多边形2、7、8,随后在穿越线SG的两侧,分别找到一个点,该点必需满足以下两个条件:其一,是这些相交的障碍物多边形2、7、8中的一个顶点,其二,在穿越线SG的一侧范围内该点距离穿越线最远,并分别记为点H和点L,连接S-H-G-L-S构成一个四边形;

[0009] 步骤3、遍历该四边形的每条当前边,即当前边SH、HG、GL、LS,每条当前边的端点分别为当前边起点和当前边终点,例如遍历到当前边HG时,当前边起点就是点H,当前边终点

就是点G；

[0010] 步骤4、判断当前边是否与障碍物相交，若不相交，例如当前边GL，则将该当前边保存至活动区域集中，然后返回至步骤3继续遍历后续的前边，当遍历完成时则跳转至步骤7；若相交，例如当前边SH、HG、LS，则跳转至步骤5；

[0011] 步骤5、确定中途点，在与当前边相交的所有障碍物中，取这些障碍物中离穿越线SG最远且与当前边同一侧的顶点，记为中途点A、B、C，分别连接当前边起点和中途点，中途点和当前边终点，形成2条新的当前边；例如原来的当前边SH由当前边起点S连接中途点A、再由中途点A连接当前边终点H形成由SA与AH两条边组成现在的当前边；例如原来的当前边HG由当前边起点H连接中途点B、再由中途点B连接当前边终点G形成由HB与BG两条边组成现在的当前边；例如原来的当前边LS由当前边起点L连接中途点C、再由中途点C连接当前边终点S 形成由LC与CS两条边组成现在的当前边；

[0012] 步骤6、将现在新围成的多边形S-A-H-B-G-L-C-S的各条边转至步骤3进行遍历；

[0013] 步骤7、多边形当前边遍历完毕，活动区域集中所有的边所围成的多边形即为输出结果，活动区域外的障碍物多边形1、3、10、11不需要考虑，而活动区域内的障碍物多边形2、4、5、6、7、8、9就是机器人在路径规划时需要考虑的。

[0014] 本发明有益效果是：一种基于递归简化可视图的环境建模方法，包括以下步骤：(1) 记录机器人所在及到达位置，(2) 连接S-H-G-L-S构成一个四边形(3) 遍历该四边形的每条当前边，(4) 判断当前边是否与障碍物相交，(5) 确定中途点，(6) 将现在新围成的多边形S-A-H-B-G-L-C-S的各条边转至步骤3进行遍历，(7) 多边形当前边遍历完毕。与已有技术相比，本发明最后得到一个多边形区域，该区域外的障碍物可以全部简化掉，区域内的才是机器人路径规划需要考虑的障碍物。本发明有效的减少可视线的数量，提高了后续机器人路径规划算法的执行效率。

## 附图说明

[0015] 图1为本发明方法步骤流程图。

[0016] 图2为本发明方法中步骤2所生成的可视图。

[0017] 图3为本发明方法递归的第1层生成的可视图。

[0018] 图4为本发明方法递归的第2层生成的可视图。

[0019] 图5为本发明方法递归的第3层生成的可视图。

## 具体实施方式

[0020] 下面结合附图对本发明作进一步说明。

[0021] 如图1所示，一种基于递归简化可视图的环境建模方法，包括以下步骤：

[0022] 步骤1、记录机器人所在及到达位置，将机器人所在的位置记为点S，期望到达的位置记为点G，再将机器人所处的整个地图环境采用计算机进行扫描，并采用封闭多边形对每个障碍物进行包络，保留这些多边形；

[0023] 步骤2、连接S-H-G-L-S构成一个四边形，连接点S和点G，线段SG称为机器人穿越线，此时穿越线SG会与多个障碍物多边形相交，记录这些相交的障碍物多边形2、7、8，随后在穿越线SG的两侧，分别找到一个点，该点必需满足以下两个条件：其一，是这些相交的障

碍物多边形2、7、8中的一个顶点,其二,在穿越线SG的一侧范围内该点距离穿越线最远,并分别记为点H和点L,连接S-H-G-L-S构成一个四边形;

[0024] 步骤3、遍历该四边形的每条当前边,即当前边SH、HG、GL、LS,每条当前边的端点分别为当前边起点和当前边终点,例如遍历到当前边HG时,当前边起点就是点H,当前边终点就是点G;

[0025] 步骤4、判断当前边是否与障碍物相交,若不相交,例如当前边GL,则将该当前边保存至活动区域集中,然后返回至步骤3继续遍历后续在当前边,当遍历完成时则跳转至步骤7;若相交,例如当前边SH、HG、LS,则跳转至步骤5;

[0026] 步骤5、确定中途点,在与当前边相交的所有障碍物中,取这些障碍物中离穿越线SG最远且与当前边同一侧的顶点,记为中途点A、B、C,分别连接当前边起点和中途点,中途点和当前边终点,形成2条新的当前边;例如原来的当前边SH由当前边起点S连接中途点A、再由中途点A连接当前边终点H形成由SA与AH两条边组成现在的当前边:例如原来的当前边HG由当前边起点H连接中途点B、再由中途点B连接当前边终点G形成由HB与BG两条边组成现在的当前边;例如原来的当前边LS由当前边起点L连接中途点C、再由中途点C连接当前边终点S形成由LC与CS两条边组成现在的当前边;

[0027] 步骤6、将现在新围成的多边形S-A-H-B-G-L-C-S的各条边转至步骤3进行遍历;

[0028] 步骤7、多边形当前边遍历完毕,活动区域集中所有的边所围成的多边形即为输出结果,活动区域外的障碍物多边形1、3、10、11不需要考虑,而活动区域内的障碍物多边形2、4、5、6、7、8、9就是机器人在路径规划时需要考虑的。

[0029] 具体工作过程如下:

[0030] 本发明采用的伪代码如表1所示,其中包含了主函数和递归核函数。

[0031] 表1

[0032]

主函数	<pre> line[] getActiveRegion (point start,point goal){     line line_sg = getLine (start,goal);     obs[] obs_pass = getObsPass (line_sg);     line[] activeRegion = null;     if (obs_pass.length == 0){         activeRegion = [line_sg];     }else{         point high = getHighPoint (line_sg,obs_pass);         point low = getLowPoint (line_sg,obs_pass);         line[] lineList_sh = getLineList (start,high);         line[] lineList_hg = getLineList (high,goal);         line[] lineList_sl = getLineList (start,low);         line[] lineList_lg = getLineList (low,goal);         activeRegion = lineList_sh + lineList_hg + lineList_sl + lineList_lg;     }     return activeRegion; } </pre>
递归核函数	<pre> line[] getLineList (point start, point goal){     line line_sg = getLine (start,goal);     obs[] obs_pass = getObsPass (line_sg);     if (obs_pass.length == 0){         return [line_sg];     }else{         point midPoint = getMidPoint (line_sg,obs_pass);         line[] lineList_sm = getLineList (start,midPoint);         line[] lineList_mg = getLineList (midPoint,goal);         return lineList_sm + lineList_mg;     } } </pre>

[0033] 主函数是本发明的程序入口,通过给定的起点S和终点G,由函数getLine得到穿越线line\_sg,再由函数getObsPass得到与穿越线相交的障碍物集合obs\_pass,以函数getHighPoint和getLowPoint计算出在line\_sg两侧,距离line\_sg最远的且被line\_sg穿过的障碍物顶点high(点H)和low(点L),连接S-H-G-L-S,即可得到图2中的可视图,黑色色块为障碍物多边形,虚线为当前的活动区域边界。然后,通过调用递归核函数,可以分别算出S-H-G-L-S的各条边演化出的活动区域的边界线,其中getMidPoint函数是求出距离穿越线最远的一侧属于被line\_sg所穿过的障碍物的顶点。递归第1层形成的可视图如图3所示,递

归第2层形成的可视图如图4所示,递归第3层形成的可视图如图5所示。最终,程序递归至第三层就结束,活动区域的边界线保存在集合activeRegion中,形成的最终可视图如图5所示。

[0034] 图5中,活动区域外的障碍物(1、3、10、11)在后续路径规划中将会被简化掉,不需要考虑,而活动区域内的障碍物(2、4、5、6、7、8、9)就是机器人在路径规划时需要考虑的。

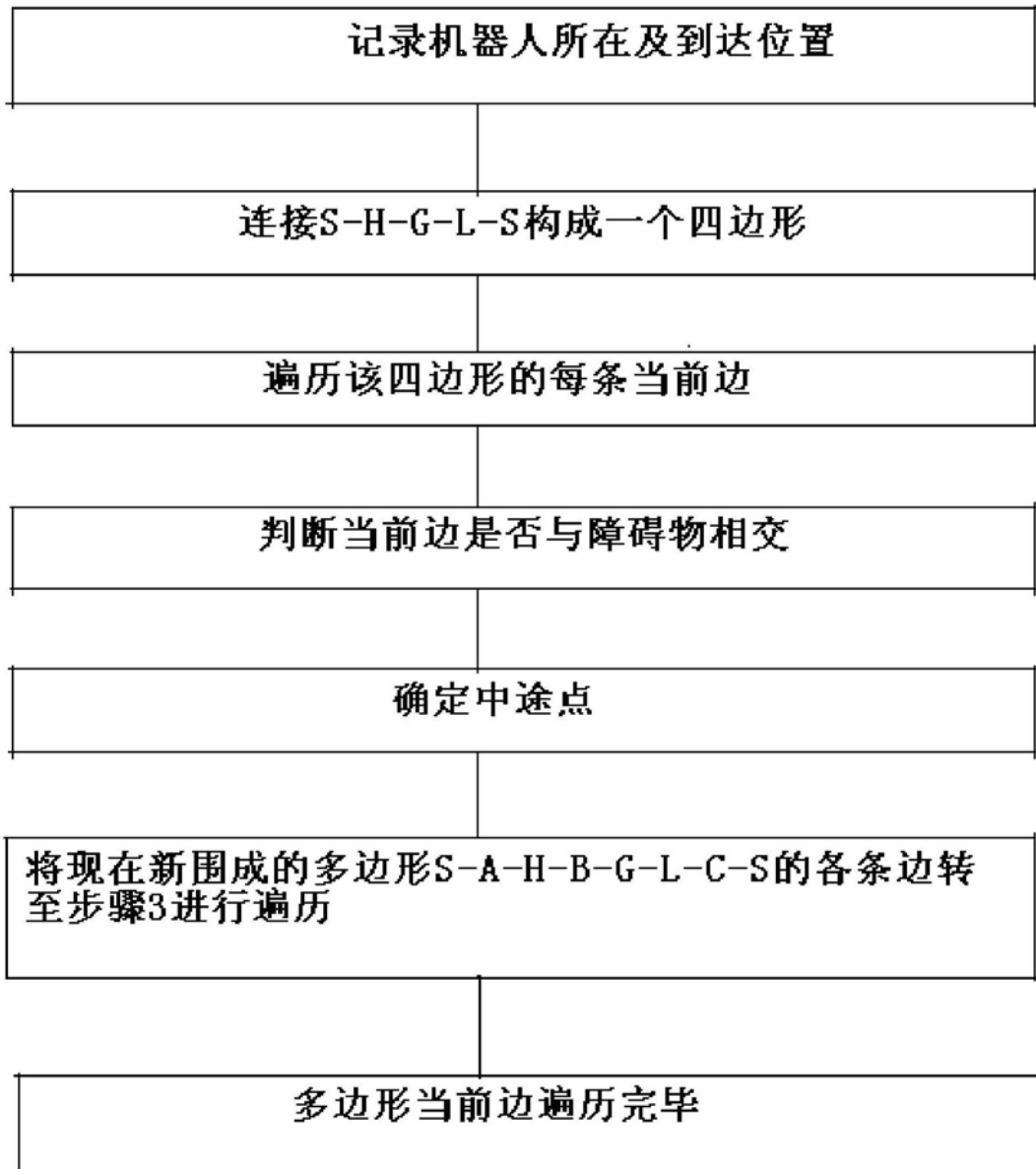


图1

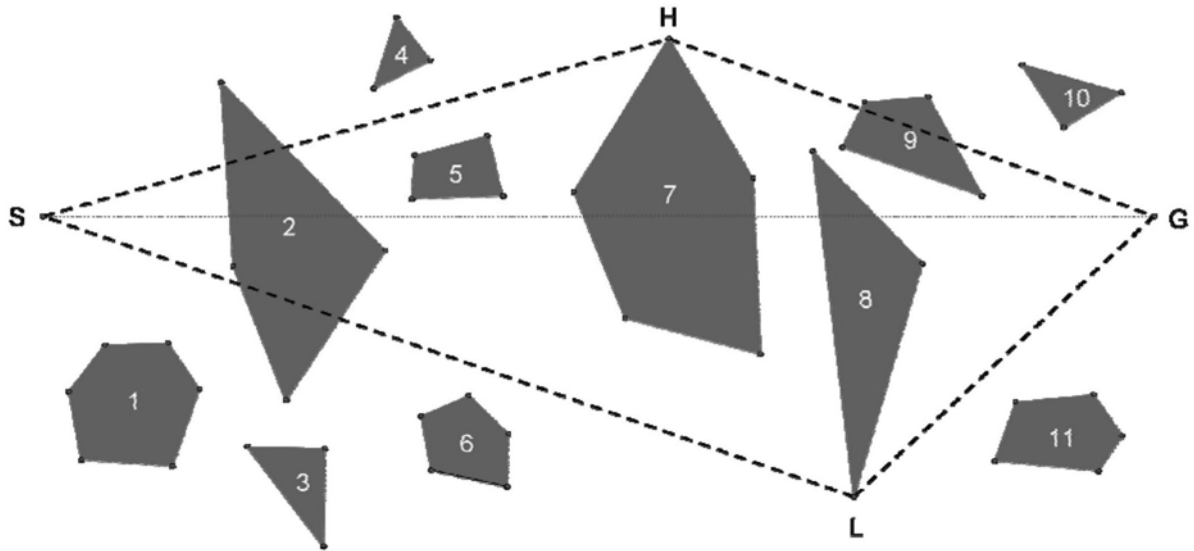


图2

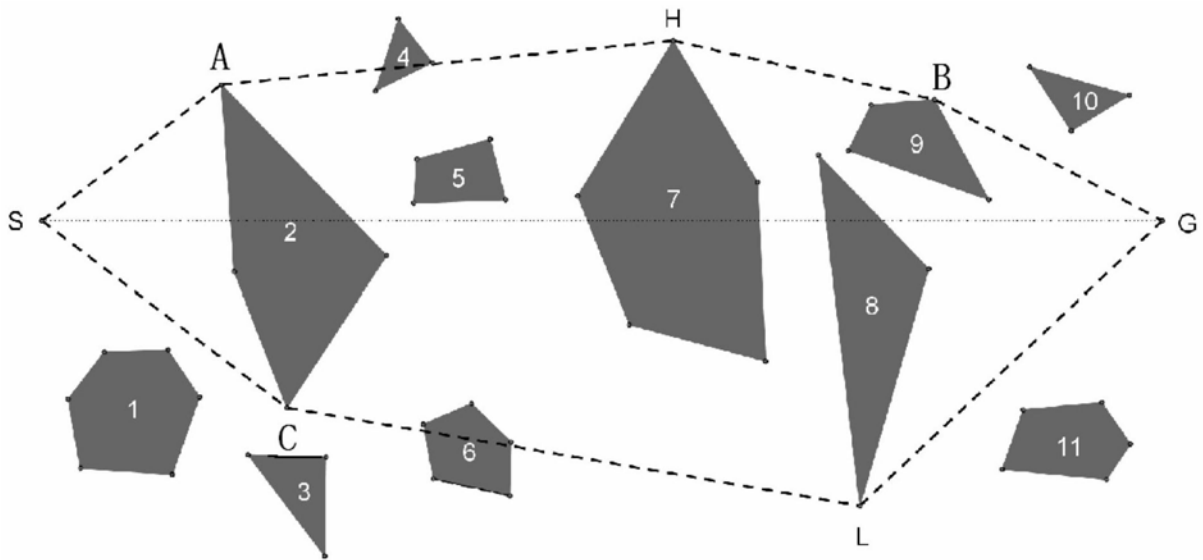


图3

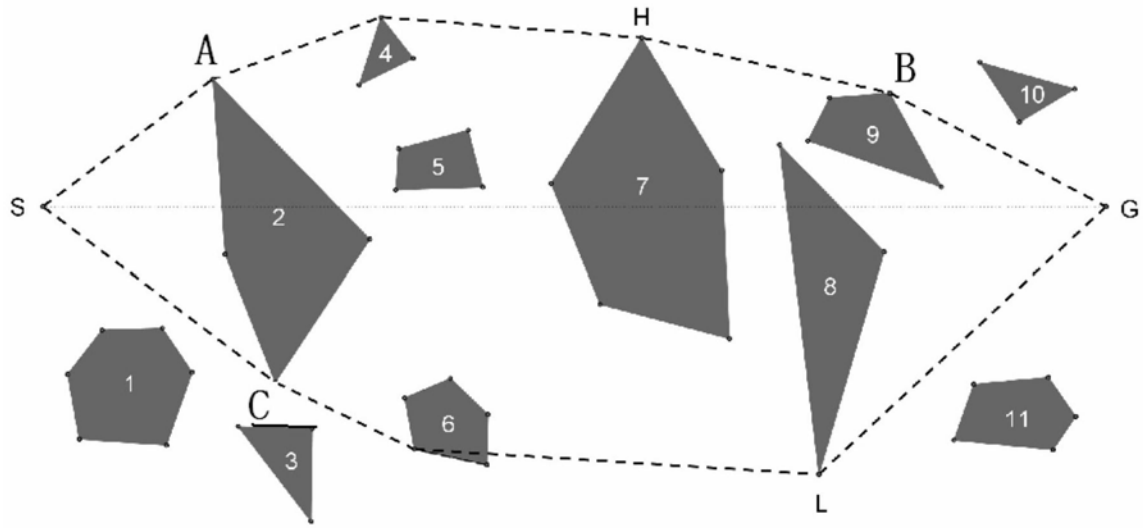


图4

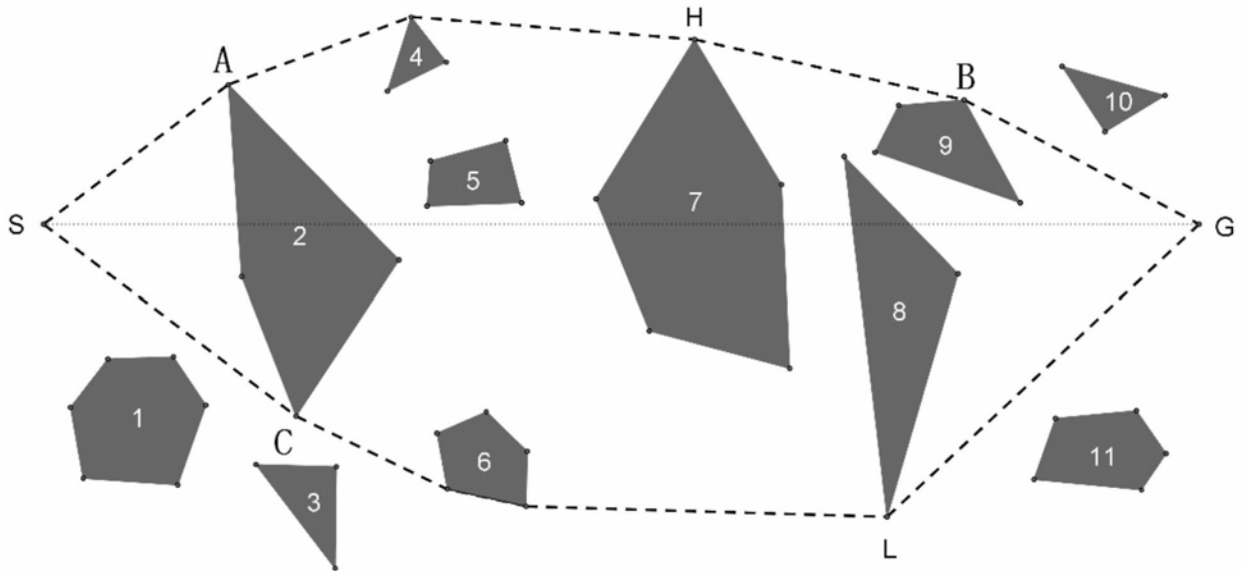


图5