

# 基于栅格地图的移动机器人完全遍历算法——矩形分解法

田春颖 刘 瑜 冯申坤 朱世强

(浙江大学流体传动及控制国家重点实验室 杭州 310027)

**摘要:** 提出移动机器人的一种新的完全遍历算法: 矩形分解算法。首先通过机器人环境学习建立栅格地图, 对环境中的障碍物实行矩形化建模。而后应用矩形化模型中的关键点将环境分解成为矩形块, 最后在这个分块环境的拓扑图中寻找到一条 Hamilton 路径, 机器人沿此路径即可实现对环境的完全遍历。为处理复杂的局部情况, 又提出基于模板的局部环境处理算法。矩形算法的优点在于机器人可以实现完全自主的复杂环境遍历, 并且可以处理未知障碍, 从而使算法适合于任意非结构化的工作环境。

**关键词:** 矩形分解算法 Hamilton 路径 完全遍历 栅格地图 移动机器人

**中图分类号:** TP24

## 0 前言

完全遍历的路径规划问题要求机器人遍历环境中的所有可达区域。在这个问题上已经提出了多种算法。目前常用的算法有模板算法和分块算法两类。

模板算法是一种利用模板进行遍历的算法, 该算法首先见于参考文献[1], 但该文章中提出的模板算法不能应用于有障碍的环境中, 只对于该文中列举的一些环境具有实用性。参考文献[2]对前面提出的模板算法进行改进, 可以应用于基于地图的有障碍环境并且对环境中的未知障碍有相应的处理策略。但是在一些特殊情况比如遇到存在凹陷的障碍物时, 机器人便会进入无法处理的死循环状态。这种情况出现的主要原因是由于对整个环境缺乏整体的规划。

分块算法是将环境根据障碍物的情况分解为多个分块, 而后在此基础上实现遍历的算法。在设计出对分块进行局部遍历算法的基础上, 完全遍历规划就简化为对各个分块进行依次遍历的问题。目前有两种主要的分块算法: Trapezoidal 分块算法<sup>[3]</sup>和 Boustrophedon 分块算法<sup>[4]</sup>。Trapezoidal 算法将环境分解为梯形块, 在单个分块中机器人通过往返移动来进行遍历。Boustrophedon 算法是 Trapezoidal 算法的改进, 其目的即是减少由于分块过多而造成的重复遍历。

但是 Boustrophedon 算法只减少了机器人在相邻梯形分块之间进行移动时所产生的重复遍历, 机

器人在不相邻分块之间移动时所产生的重复遍历并没有减少。Choset 详细介绍了机器人在不同分块间移动的算法<sup>[5]</sup>。由此算法可知, 如果一个分块有很多个未被遍历的相邻分块, 那么必定会有一个或几个分块被留在后面, 等待机器人遍历其他的分块后再回来将其遍历。在机器人返回的过程中, 必然会有大量的重复遍历产生。

针对重复遍历问题提出矩形分解算法, 利用分块算法的基本思想, 将任意形状障碍进行矩形化建模, 在此基础上构建分块环境拓扑图。此拓扑图中必然存在使各个分块间无重复遍历的 Hamilton 回路。如图 1, Boustrophedon 算法将环境分为 4 个分块, 矩形分解算法将环境分为 3 个分块, 这样不仅进一步减少了分块数目, 而且在分块后的环境拓扑图中可以寻找到一条 Hamilton 路径, 这在 Boustrophedon 算法的环境拓扑图中是不一定能实现的。

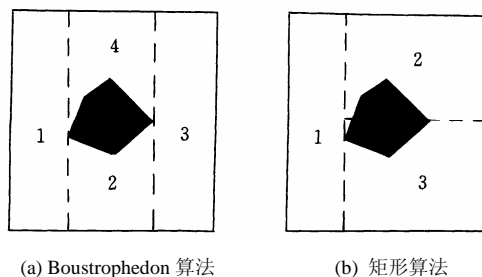


图1 两种算法的分块情况对比

## 1 基于栅格地图的分块建模

### 1.1 栅格地图

栅格地图是由机器人自主学习得到的。每当机

机器人被应用于一个新的环境时, 机器人对环境进行一次自主学习并建立栅格地图。在工作中如果发现地图中未标注的未知障碍即调用模板进行处理并且根据新障碍的情况更新地图。

机器人建立栅格地图的具体算法如图 2 所示。机器人首先从环境原点  $O$  沿  $x$  方向进行移动, 发现障碍即绕过障碍物, 此后继续沿原方向运动, 到达边界后沿  $y$  增大方向移动一个车距返回。重复这个过程, 直到环境学习完毕。当机器人遇到障碍物时, 记录该格点为有障碍, 否则记录为无障碍, 根据机器人的记录情况即可以建立该环境的栅格地图。

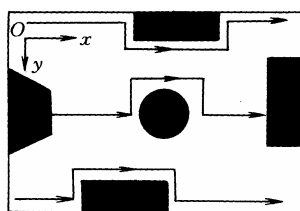


图 2 栅格地图建模过程

## 1.2 矩形化建模

图 3 是对矩形房间遍历学习后建立的栅格地图, 每个格点可以用  $(x, y)$  来表示,  $x$  表示格点所在列数,  $y$  表示格点所在行数。于是, 左上角格点  $(1, 1)$ , 右下角格点  $(20, 10)$ 。其中, 含有障碍物的格点标注为 1, 不含有障碍物的格点标注为 0。可以看到, 该环境中存在两个障碍物。首先, 找到每个障碍物  $x$  值最小的格点, 如果  $x$  值最小的格点不止一个, 则找出这些点中  $y$  值最小的格点, 标记为  $M(x_1, y_1)$ , 而后找出每个障碍物中  $x$  值最大的格点, 如果  $x$  值最大的格点不止一个, 则找出这些点中  $y$  值最大的格点, 记为  $N(x_2, y_2)$ 。这样每个障碍物以其  $M$ 、 $N$  点为对角线虚拟成为一个矩形障碍, 如图 3 中加粗方格线。

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0
0	0	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0
0	0	1	1	1	1	1	1	1	0	0	1	1	1	1	1	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

图 3 栅格地图

## 1.3 环境分块的矩阵表示

按照从左至右, 从上至下的顺序依次对障碍矩形化的栅格地图进行分块。以矩形障碍的  $M$  点为起点, 沿纵向做割线直到边界或矩形障碍或其他的分

割线。以矩形化障碍的  $N$  点为起点, 沿横向做割线直到环境边界或矩形障碍或其他的分割线, 如图 4 所示(用加粗格线表示割线)。

0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
0	0	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0
0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
0	0	1	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0

图 4 环境分块图

其中对分割线所在格点做如下定义。

定义 1: 割线不存在区域  $S_1 = \{x, y | \text{ocup}(x, y) = 0\}$ , 割线存在区域  $S_2 = \{x, y | \text{ocup}(x, y) = 1\}$ 。其中  $\text{ocup}(x, y)$  表示割线占有函数。图 4 中对割线存在的区域进行了直观的标识。

定义 2: 矩形分块  $T = T_1 \cup T_2$ 。其中  $T_1$  表示被割线分离开的独立区域,  $T_2$  表示与独立区域  $T_1$  紧相邻的割线存在区域。所以存在关系  $\sum T_1 = \sum S_1$ ,  $\sum T_2 = \sum S_2$ 。

定义 3: 设  $(x_1, y_1)$ 、 $(x_2, y_2)$  分别表示矩形块左上角和右下角点的坐标, 则矩形分块可以用矩阵表示为:  $A = [x_1 \ x_2 \ y_1 \ y_2]$ 。其中  $(x_1, y_1)$ 、 $(x_2, y_2) \in T$ 。

依据以上定义, 图 3 所示的环境被分割为 5 块, 分别用  $A_1$ 、 $A_2$ 、 $A_3$ 、 $A_4$ 、 $A_5$  表示, 如图 5 所示。相应的矩阵为:  $A_1 = [1 \ 3 \ 1 \ 10]$ ,  $A_2 = [3 \ 12 \ 1 \ 6]$ ,  $A_3 = [12 \ 20 \ 1 \ 8]$ ,  $A_4 = [12 \ 20 \ 8 \ 10]$ ,  $A_5 = [3 \ 12 \ 6 \ 10]$ 。

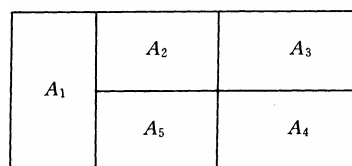


图 5 分块例图

## 2 Hamilton 路径的存在性

### 2.1 矩形分块相邻关系定义

定义 4: 分块环境地图中, 如果任意两个矩形分块之间存在公共边或者公共点, 定义这两个分块相邻。

依照相邻的方位关系分为 4 种相邻方式: 左相邻、右相邻、上相邻和下相邻。没有公共边而只有

公共点的两个分块可认为存在两种相邻方式。以图5为例,图中 $A_2$ 、 $A_5$ 为 $A_1$ 的右相邻块, $A_1$ 为 $A_2$ 、 $A_5$ 的左相邻块, $A_2$ 为 $A_5$ 的上相邻块,而 $A_5$ 则为 $A_2$ 的下相邻块。 $A_2$ 与 $A_4$ 没有公共边只有公共点, $A_2$ 既为 $A_4$ 的上相邻块,也为 $A_4$ 的左相邻块。

## 2.2 分块图中 Hamilton 路径的存在性

图6是图4的矩阵分块表示,其中图6a为分块图。将每个分块用一个点来表示,相邻的点之间通过一条无向边连接,则可以得到无向图6b。可知所需求出的规划路径即为该无向图中的一条 Hamilton 回路。

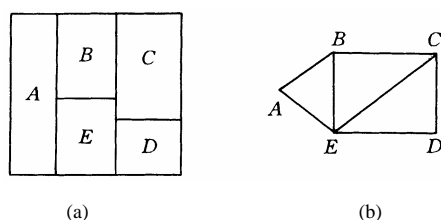


图6 分块环境的图形表示

定理 1: 环境矩形分块无向图中存在且至少存在一条 Hamilton 回路。

对该定理证明如下: 由上述分块过程可知, 环境分块是按照一定顺序逐个分析障碍物后对环境进行分块的过程。障碍物分析完毕, 则环境也分块完毕, 即该分块过程可以理解为对无障碍环境中逐渐添加障碍物的过程。

(1) 若该环境的障碍物为 0, 则无需寻找回路, 这种情况不需考虑。

(2) 当该环境的障碍物个数大于 0 时, 将障碍物按照从左至右, 从上至下的顺序逐个添加至该区域。只存在一个障碍物的分块图和无向图如图 7, 遍历 3 个分块的 Hamilton 回路为  $A-B-C-A$ 。每增加一个障碍会有 3 种情况出现, 下面逐个分析。

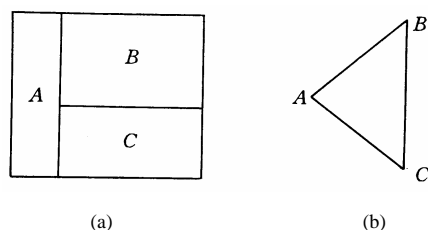


图7 包含一个障碍物的环境分块图

情况 1: 新增的矩形障碍物模型  $O_2$  与前一个矩形障碍模型  $O_1$  存在关系

$$\begin{cases} x_{\min 2} = x_{\min 1} \\ y_{\min 2} > y_{\max 1} \end{cases}$$

式中  $x_{\min 1}$ ,  $x_{\min 2}$ ,  $y_{\min 2}$ ,  $y_{\max 1}$  分别为区域最大、最小

值。那么新分块环境为图 8a 所示, 新的 Hamilton 路径为  $A-B-C-D-A$ 。由于新增加的节点与原来回路中的两个相邻节点都相邻, 也就是用新增加的两条边代替原回路中的一条边, 这样就实现了所有点的遍历。

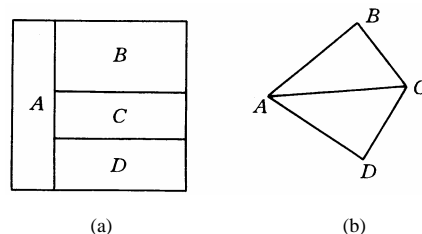


图8 增加障碍物可能情况 1

情况 2: 新增加的障碍分块刚好位于已经存在的水平分割线上。新分块环境为图 9a 所示, 其 Hamilton 回路为  $A-B-D-E-C-A$ 。由于新增加的分块  $D$  与  $B$ 、 $C$  相邻,  $E$  与  $C$ 、 $D$  相邻, 则  $D$ 、 $E$  可以分别增加至  $B$  与  $C$ 、 $C$  与  $D$  之间, 如此则实现了对新增加分块的遍历。

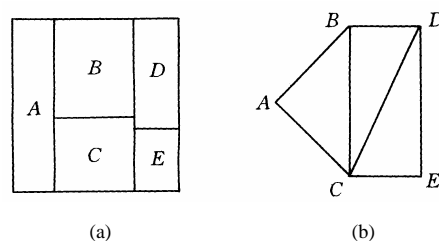


图9 增加障碍物可能情况 2

情况 3: 新增加的障碍物分块没有阻碍任何割线, 增加在原来已存在的独立分块内部, 如图 10 所示。这种情况的遍历与上面情况相同, 新的遍历回路为  $A-B-D-E-C$ 。

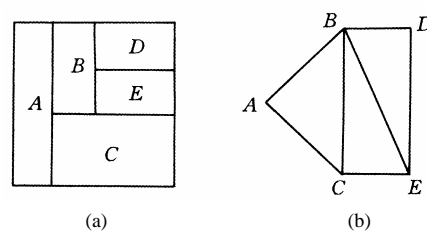


图10 增加障碍物可能情况 3

至此, 对增加一个障碍物的所有可能情况分析完毕。当环境中障碍物继续增加时, 可以用上述方法将新增加的分块加入到原来的回路中。依照上面的矩形分块方法, 无论环境多么复杂, 都在其无向图中存在且至少存在一条 Hamilton 回路, 可以实现无向图中所有结点的遍历。复杂环境下 Boustrophedon 算法虽然可以将结点完全遍历, 但是分块间不一定存在无重复遍历的 Hamilton 路径。

### 3 Hamilton 回路求解算法

#### 3.1 相邻分块的求解算法

任取一个环境进行分块后得到  $n$  个矩阵： $A_1, A_2, \dots, A_n$ ，共同组成一个  $n \times 4$  阶矩阵，记为  $P = [A_1 \ A_2 \ \dots \ A_n]^T$ ，这个矩阵即是该环境所有方块矩阵的组合。现求解其中任一矩阵  $A[x_1 \ x_2 \ y_1 \ y_2]$  的相邻矩阵。首先对  $A$  做变换得变换阵  $A^*$

$$A^* = A \times \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} =$$

$$[x_1 \ x_2 \ y_1 \ y_2] \times \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} = [x_2 \ x_1 \ y_2 \ y_1]$$

(1)

而后代入式(2)

$$U = \left( \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix}_{n \times 1} \times A^* - P \right) \times \begin{bmatrix} f_1(\square) \\ f_2(\square) \\ f_3(\square) \\ f_4(\square) \end{bmatrix} \quad (2)$$

得  $U = [u_1 \ u_2 \ \dots \ u_k \ \dots \ u_k]^T$ 。

式中  $u_k = f_1(x_2 - x_{1k}) + f_2(x_1 - x_{2k}) +$

$$f_3(y_2 - y_{1k}) + f_4(y_1 - y_{2k}) \quad (3)$$

其中矩阵  $[f_1(\square) \ f_2(\square) \ f_3(\square) \ f_4(\square)]^T$  表示需要求解矩阵  $A$  的何种邻接性质。假设需要求解矩阵  $A$  的上相邻阵， $[f_1(\square) \ f_2(\square) \ f_3(\square) \ f_4(\square)]^T$  表示上相邻性质，具体求法如下：对一个矩阵的上相邻矩阵，其约束条件可以表示为  $y_1 = y_{2k}, x_2 \geq x_{1k}$  且  $x_1 \leq x_{2k}$ ，则可以得到

$$f_1(x) = \begin{cases} -1 & x < 0 \\ 0 & x \geq 0 \end{cases} \quad (4)$$

$$f_2(x) = \begin{cases} -1 & x > 0 \\ 0 & x \leq 0 \end{cases} \quad (5)$$

$$f_3(x) = 0 \quad (6)$$

$$f_4(x) = \begin{cases} 1 & x = 0 \\ 0 & x \neq 0 \end{cases} \quad (7)$$

由式(3)知，如果  $u_k = 1$ ，则矩阵  $P$  中第  $k$  个矩阵  $A_k$  与矩阵  $A$  相邻。其他的相邻情况中， $[f_1(\bullet) \ f_2(\bullet) \ f_3(\bullet) \ f_4(\bullet)]^T$  的求法与此相同。首先写出约束条件，而后确定  $[f_1(\bullet) \ f_2(\bullet) \ f_3(\bullet) \ f_4(\bullet)]^T$  的具体值。

#### 3.2 Hamilton 路径的求解算法

该算法应用先入先出队列，规定机器人遍历环境时从参数为  $[1 \ x \ 1 \ y]$  的矩阵开始。首先将该矩阵压入队列顶层，而后按照前面的相邻矩阵求解算法求出该矩阵的所有右相邻矩阵，按顺时针顺序将这些矩阵依次压入队列，这些队列行成了最初的 Hamilton 回路。接下来计算队列内矩阵的所有相邻阵，如果队列中两个相邻矩阵  $A, B$  有共同的相邻阵  $C$ ，且矩阵  $C$  不在队列中，则将其压入队列，置于两个相邻矩阵  $A, B$  之间，由定理 1 的结论可知全部矩阵必然被压入队列，这个队列便是所求的 Hamilton 路径，机器人依此路径可以将所有方块矩阵无重复遍历，问题得解。

### 4 模板遍历算法

#### 4.1 矩形分块

建模得到的矩形分块虽然形状为矩形，但是其真实的空间内部并非为矩形区域，实际上是由障碍和空间共同组成的复杂环境，如图 11 所示。在矩形分块中，只有分块  $A$  是没有任何障碍物，分块  $B, C, D$  内都存在障碍物，分块  $E$  甚至被截成 2 块。如果单纯按照分块分区进行往返遍历，则机器人在分块  $C$  与  $D$  的边界将做两次往返转弯，产生重复遍历，而且分块  $E$  是无法一次性完全遍历的。为有效处理这样的复杂局部环境，同时减少机器人的重复遍历，规定在机器人进行局部遍历时，遇到横行的分割线不做处理，而是以传感器信息为准，直到遇到障碍物才返回。这样，机器人在进行遍历时，就不会在分块  $C$  与  $D$  的边界产生重复遍历，而且在处理分块  $B$  时，会将分块  $E$  的一部分遍历，处理分块  $E$  时也可以实现一次性完全遍历。

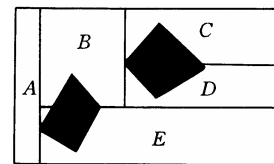


图 11 复杂的局部分块

#### 4.2 局部模板算法

模板算法共引入 5 个模板：TM(Towards marker)，

UT(U turn), SS(Side shift), UTI(U turn interlaced) 和 SCOO(Steer clear of obstacle), 见图 12。

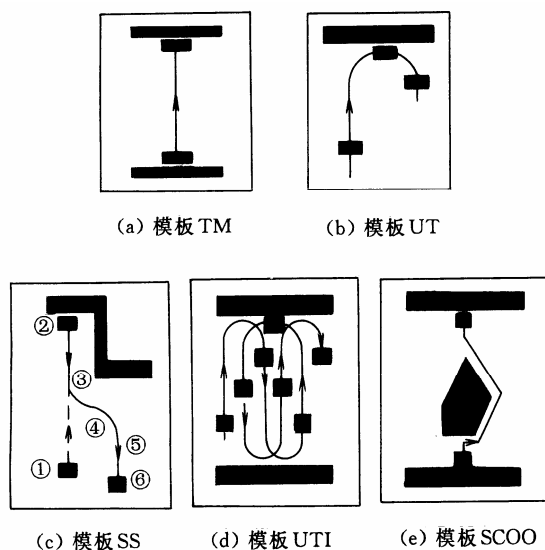


图 12 模板示意图

前 4 个模板的选取可以参见文献[2], 对模板应用功能介绍如下。

(1) 模板 TM: 前行模板。这是最简单的行走规则, 主要完成点对点的无障碍行走。

(2) 模板 UT: U 形转弯模板。该模板应用于最初进入分块时的行走规划, 当分块横向宽度较窄时(横向距离小于 5 倍车距)或者纵向距离很小时(纵向距离小于 5 倍车距)。

(3) 模板 SS: 沿边移动模板。当存在不规则形状或侧面凹焰形状障碍物时采用这种模板。

(4) 模板 UTI: 大半径 U 形交错前行模板, 与 UT 模板一同应用于最初进入分块时的行走规划。该模板在分块横向宽度很宽(横向距离大于 5 倍车距)且纵向距离较大(纵向距离大于 5 倍车距)时使用。使用该模板时由于转弯半径大, 所以机器人不需做太大的变速便可实现转弯, 节省遍历时间, 且减少了在小半径转弯时产生的重复遍历。

(5) 模板 SCOO: 避障模板。在遇到未知障碍时采用此模板进行避障。纵向绕过障碍物后继续进行原定的遍历任务。

在进行遍历时, 首先判断分块的横向距离, 根据其横向距离决定是采用 UT 还是 UTI 模板。之后, 根据其模板算法规则进行遍历。未遇到障碍时采用 TM 模板, 遇到障碍时采用前面决策好的 UT 或 UTI 模板。如果在遍历过程中障碍物存在不规则形状或者侧面凹陷, 则采用 SS 模板和 UT 模板进入凹陷。在遍历块中遇到了未知的障碍, 用 SCOO 模板进行遍历, 并在行走时记录障碍物位置。当一个分块遍

历结束后, 应用 TM 模板实现点对点的移动, 到达下一个分块的未遍历区域。遍历任务完成以后, 根据环境的变化重新进行整体分块规划。

## 5 试验与仿真

通过一个带有 14 个超声传感器的机器人完成环境学习。两个驱动轮单独驱动, 光电编码器实现机器人自主定位, 具体设计见参考文献[6]。对环境遍历学习后得到栅格地图如图 13a。通过 PC 机对图 13a 描述的环境做仿真试验。首先对环境中的障碍矩形化处理, 结果见图 13b。接下来进行环境分块, 见图 13c。由算法可知遍历规划从左边分块开始, 沿顺时针方向依次遍历各个分块。图 14a~f 分别表示对各个分块的遍历结果, 其中图 14c 表示出机器人对于未知障碍的处理过程。

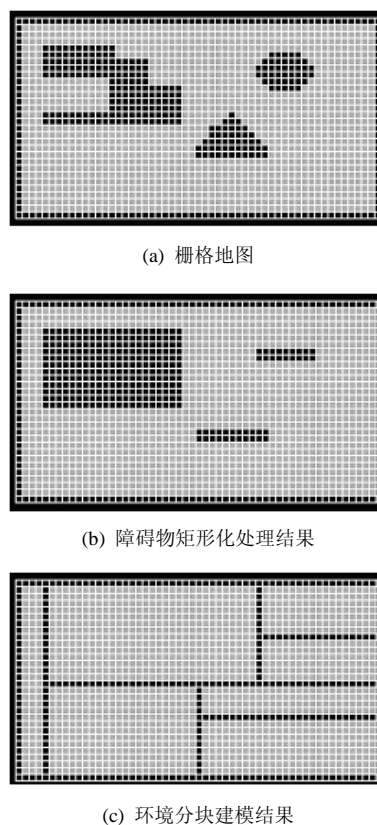


图 13 遍历学习后的栅格地图和分块建模结果

由仿真结果可见, 该算法基本上实现了对环境的完全遍历, 遗漏面积与机器人自身机械结构等原因有关。遍历结果表明机器人在进行相邻块间移动时仍然存在着重复遍历, 但是其重复的情况对比 Boustrophedon 分块算法有了很大改善, 而且对于环境中障碍物存在侧面凹陷的情况也进行了有效处理。虽然在处理完成返回的过程中, 产生了大量的重复遍历, 但是这是无法避免的。此外在遇到未知

障碍物时, 机器人对其进行学习, 没有出现遗漏, 不过 SCOO 模板的调用导致重复遍历现象严重, 这将是下一步的研究重点。

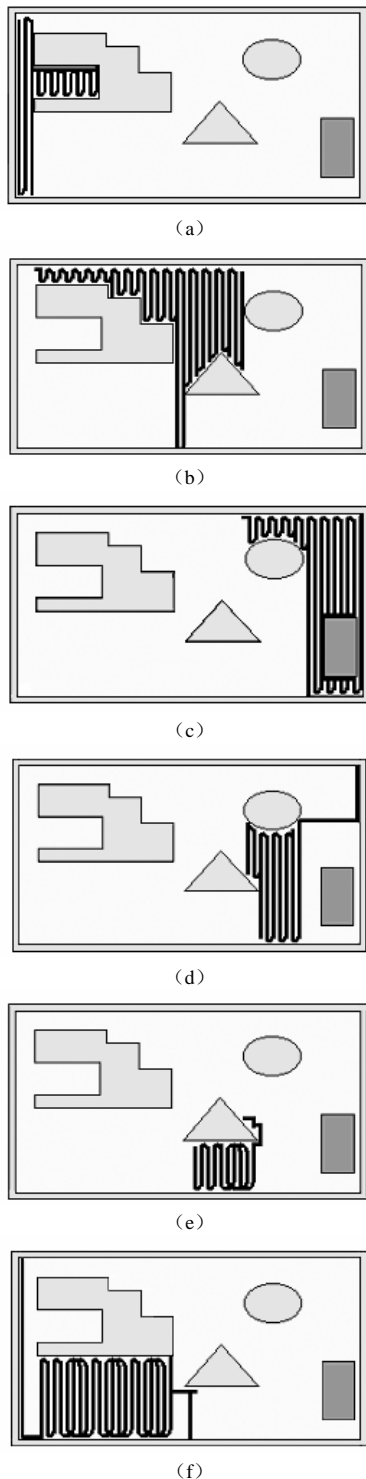


图 14 完全遍历算法仿真结果

## 6 结论

矩形分解算法最主要的特点是进一步减少了分块数目, 并且在环境拓扑图中找到了一条 Hamilton 路径, 并结合局部处理算法实现了对有侧面凹陷的障碍物和未知障碍物的处理。由仿真试验可以看出,

所述及的矩形分解算法对机器人在复杂环境中的自主遍历是非常有效的。

## 参 考 文 献

- 1 Hofner C, Schmidt G. Path planning and guidance techniques for an autonomous mobile cleaning robot. *Robotics and Autonomous Systems*, 1995, 14(2, 3): 199~212
- 2 Neumann R, Vidal H A, Vieira P, et al. Complete coverage path planning and guidance for cleaning robots. In: *Proc. of IEEE Intl. Symposium on Industrial Electronics*, Guimaraes, Portugal: IEEE, 1997: 677~682
- 3 Latombe J C. *Robot Motion Planning*. Boston, MA: Kluwer Academic Publishers, 1991
- 4 Choset H. Coverage of known spaces: the boustrophedon cellular decomposition. *Autonomous Robots*, 2000, 9(3): 247~253
- 5 Choset H, Pignon H. Coverage path planning: the boustrophedon decomposition. In: *Intl. Conf. on Field and Service Robotics*, Canberra, Australia, 1997
- 6 马翔, 朱世强. 智能吸尘器的开发及设计. *计算机应用*, 2000, 26(8): 6~8

## COMPLETE COVERAGE OF KNOWN SPACE——RECTANGULAR DECOMPOSITION

Tian Chunying Liu Yu Feng Shenshen  
Zhu Shiqiang

(State Key Laboratory of Fluid Power Transmission and Control, Zhejiang University, Hangzhou 310027)

**Abstract:** A new cellular decomposition approach, rectangular decomposition, is proposed for the purpose of complete coverage path planning. Firstly, the known grid map is used to build each obstacle into a rectangular model. Secondly, the critical points of each model are used to decompose the environment into rectangular cells. Each cell can be represented as a node in a graph, and then a Hamilton path is found in this graph. Because the environment is very complicated and sometimes there are some unexpected obstacles in the environment, sensors are needed and a local template algorithm is designed. The novelty of the proposed algorithm is that Hamilton path can be found in the topology of the environment and reduce the redundancy produced by the robot when it moves from one cell to the next. A simulation based on a grid map validates this algorithm.

**Key words:** Rectangular decomposition Hamilton path

Complete coverage Grid map Mobile robot

**作者简介:** 田春颖, 男, 1979 年出生, 浙江大学机械电子专业学生。主要研究方向为自主移动机器人。

E-mail: sharecode@zju.edu.cn