

DOI: 10.19650/j.cnki.cjsi.J2210641

# 基于霍夫曼行程编码的 SLAM 算法\*

邵锦江, 蔡体菁

(东南大学仪器科学与工程学院 南京 210096)

**摘要:**针对基于粒子滤波的 SLAM 算法生成的栅格地图存储效率不高的问题,提出一种基于霍夫曼行程编码的 SLAM 算法,以解决原有栅格地图的编码冗余和空间冗余问题。该算法在基于粒子滤波的 SLAM 算法基础上采用霍夫曼行程编码的地图表示方式,根据不同的应用场景,设计出两种霍夫曼行程编码地图的存储方式,当栅格地图的规模较小时,使用定长编码;当栅格地图的规模比较大时,例如大型商场环境,考虑使用变长编码,进一步增广了该地图表示方式的应用范围。仿真和真实场景实验表明,在一定条件下基于霍夫曼行程编码的 SLAM 算法最多能够减少 94.8% 的内存消耗,证明了该算法的可行性与有效性。

**关键词:** SLAM 算法; 粒子滤波; 霍夫曼行程编码; 地图处理

**中图分类号:** TH702 TP242 **文献标识码:** A **国家标准学科分类代码:** 520.50

## SLAM algorithm based on Hoffman run length coding

Shao Jinjiang, Cai Tijing

(School of Instrument Science and Engineering, Southeast University, Nanjing 210096, China)

**Abstract:** The raster map generated by the particle filter-based SLAM algorithm has the problem of low storage efficiency. To address this issue, a SLAM algorithm based on Huffman run-length coding is proposed. The coding redundancy and space redundancy problems of the original raster map are solved. Based on the particle filter-based SLAM algorithm, the algorithm uses Huffman run-length coding map representation. According to different application scenarios, two storage methods of Huffman run-length coding maps are proposed. When the scale of the grid map is small, the fixed-length coding is used. When the scale of the raster map is relatively large, such as a large shopping mall environment, the variable-length coding is utilized, which further expands the application range of the map representation. Simulation and real scene experiments show that, under certain conditions, the SLAM algorithm based on Huffman run-length coding can reduce memory consumption by up to 94.8%, which proves the feasibility and effectiveness of the algorithm.

**Keywords:** SLAM algorithm; particle filter; Hoffman run length coding; map processing

## 0 引言

同时定位与地图创建(simultaneous localization and mapping, SLAM)问题最早由 Smith 等<sup>[1]</sup>提出。这个问题可以被描述为:机器人在地图未知的场景中进行运动,在运动时根据传感器信息和一些数据处理算法对自身定位,同时获取环境信息对周围环境建立地图。研究初期,基于扩展卡尔曼滤波器的 SLAM 算法被广泛使用<sup>[2]</sup>。它将机器人的环境数据和位姿数据融合成新的用于计算的数据,并

使用扩展的卡尔曼滤波器(extend Kalman filter, EKF)估计最小误差值,从而获得最优估计。在移动机器人的 SLAM 算法中,基于粒子滤波的 SLAM 算法<sup>[3]</sup>相对成熟。例如,Grissetti 等<sup>[4]</sup>为了减少粒子过滤器的计算量,提出了 Ra-Blackwellized 粒子过滤器,大大减少了粒子过滤器的运行时间,使其可以实时应用,在此基础上,蒙特卡洛等人,提出了 Fast-SLAM 算法<sup>[5]</sup>,该算法结合了概率方法和卡尔曼方法,并降低了计算的复杂性。借助自适应采样方法,Gyrill Stachniss 和 Giorgio Grissetti 等人有效地解决了减少了粒子数以提高粒子滤波算法有效性的问题<sup>[6]</sup>。

收稿日期:2022-10-29 Received Date: 2022-10-29

\* 基金项目:国家重点研发基金(2016YFC0303006)项目资助

地图的表示形式是 SLAM 算法的研究热点之一,若采用激光雷达传感器,最容易得出的是点云地图,在点云地图的基础上进一步处理,根据使用目的,将从点云地图中提取出的特征聚在一起可以构成使用特征进行定位的特征地图,还可以构成用于导航避障和路径规划的占据栅格地图以及压缩性能更好的八叉树地图<sup>[7]</sup>。特征地图<sup>[8]</sup>是一种符合人类认知习惯的地图,它使用点线面等几何特征来表示环境,而不存储所有环境数据。对于包含有高层语义信息的语义地图,李秀智等<sup>[9]</sup>提出了一个移动机器人实时语义地图构建系统。Hornung 等<sup>[10]</sup>提出一种基于八叉树的地图存储模型,在此基础上中国科学技术大学的陈赢峰提出了一种基于四叉树的大规模 SLAM 算法<sup>[11]</sup>。Olson 等<sup>[12]</sup>构建了一种二维占据栅格地图,用 0~1 的概率表示占据状态,这种占据栅格地图能用 2 维 8 bit 图像表示,即把 0~1 的占据概率映射到 0~255 的像素灰度值,这种表示方法相对比较简单,但是所需内存较大,需要加以改进才有利于实际工程应用。本文着力于地图处理研究,提出一种基于霍夫曼行程编码的 SLAM 算法,在不损失地图信息的前提下极大的提高了地图的存储效率。

## 1 理论基础

### 1.1 系统整体框架

基于霍夫曼行程编码的 SLAM 系统包含两个模块,分别为霍夫曼行程编码地图生成模块和地图处理算法模块,系统整体框架如图 1 所示。整个系统建立在基于粒子滤波的 SLAM 算法的基础上,其中霍夫曼行程编码地图生成模块通过本文所述算法生成霍夫曼行程编码地图,地图处理模块是为了扩大本文所述算法的应用范围,包含两种地图编码方式和一种地图分辨率降低算法。

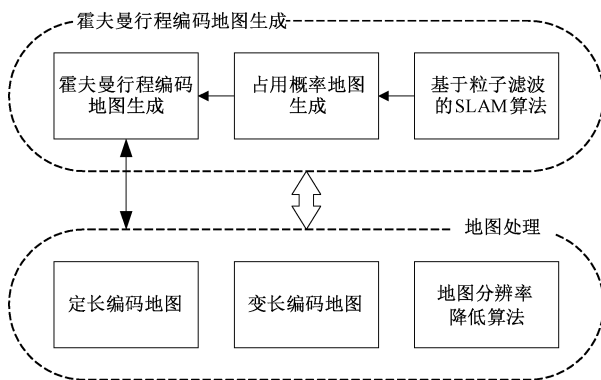


图1 系统整体框架  
Fig. 1 Overall framework of the system

### 1.2 机器人里程计模型

本文以 TurtleBot3 Burger 机器人(模型如图 2)为例建立机器人里程计模型<sup>[13]</sup>,里程计数据来源于机器人两差动轮上的编码器。

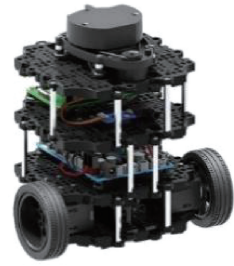


图2 TB3 Burger  
Fig. 2 TB3 Burger

用  $x$  和  $y$  表示机器人的坐标,用  $\theta$  表示机器人的方位角也即机器人的朝向,那么机器人的位可以用以下向量描述:

$$\mathbf{x}_k = (x_k, y_k, \theta_k)^T \tag{1}$$

设机器人移动前的位姿为  $(x_k, y_k, \theta_k)^T$ ,移动后的为  $(x_{k+1}, y_{k+1}, \theta_{k+1})^T$ ,则一次机器人的运动可以描述为如图 3 所示的模型。设机器人在时刻  $T_e$  内的平移速度为  $v_k$ ,旋转速度为  $\omega_k$ ,那么利用上一时刻机器人位姿计算下一时刻机器人位姿公式为:

$$\begin{pmatrix} \Delta s \\ \Delta \theta \end{pmatrix} = \begin{bmatrix} v_k T_e \\ \omega_k T_e \end{bmatrix} \tag{2}$$

$$\begin{pmatrix} x_{k+1} \\ y_{k+1} \\ \theta_{k+1} \end{pmatrix} = \begin{bmatrix} x_k + \Delta s \cdot \cos(\theta_k + \Delta\theta/2) \\ y_k + \Delta s \cdot \sin(\theta_k + \Delta\theta/2) \\ \theta_k + \Delta\theta \end{bmatrix} \tag{3}$$

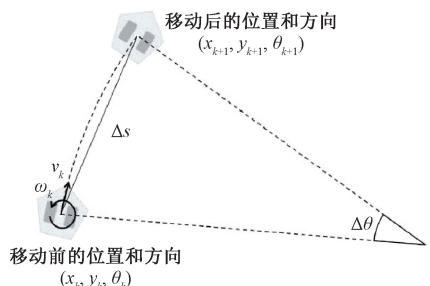


图3 里程计模型  
Fig. 3 Odometer model

### 1.3 激光雷达传感器观测模型

系统传感器采用激光雷达传感器<sup>[14]</sup>,其观测数据模型如图 4 所示,观测数据可以表示成:

$$\mathbf{Z}_i = \{L_i^1, L_i^2, \dots, L_i^n\} \tag{4}$$

其中,  $L_i^i$  表示第  $i$  束激光反射位置在机器人坐标系中的位置, $n$  个激光束组成一簇激光数据,将距离数据

返回激光雷达传感器,同时可以得到观测点相对于激光雷达传感器的坐标。再通过坐标转换转化可求得环境坐标系中观测点的坐标,以及该坐标处的区域状态。图中白色区域表示没有障碍物,黑色区域表示该区域有障碍物,灰色区域表示该区域未被观测到。

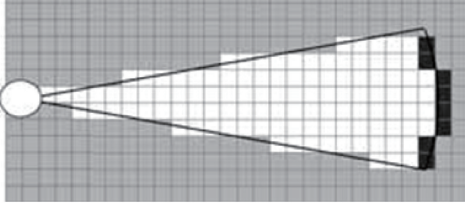


图 4 二维激光雷达观测模型

Fig. 4 The 2D-Lidar observation model

### 1.4 SLAM 问题概率模型

SLAM 问题的建模如图 5 所示,图中三角形代表的机器人在环境中移动,在不同位置收集传感器数据,然后根据这些数据融合出完整的地图模型。

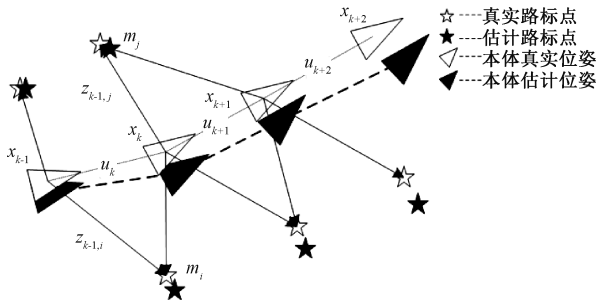


图 5 SLAM 问题观测模型

Fig. 5 The SLAM problem observation model

从概率学的观点,可以将图 5 中的 SLAM 问题表示为<sup>[12]</sup>:

$$p(\mathbf{x}_{1:t}, \mathbf{M} | \mathbf{Z}_{1:t}, \mathbf{u}_{1:t}) \quad (5)$$

$$\mathbf{M} = \{m_1, m_2, \dots, m_n\} \quad (6)$$

$$\mathbf{u}_{1:t} = \{u_1, u_2, \dots, u_t\}$$

$$\mathbf{Z}_{1:t} = \{\mathbf{Z}_1, \mathbf{Z}_2, \dots, \mathbf{Z}_t\} \quad (7)$$

$$\mathbf{Z}_k = \{z_{k,1}, z_{k,2}, \dots, z_{k,j}\}$$

其中,  $\mathbf{M}$  表示环境地图的所有路标  $m_n$  的集合;  $\mathbf{Z}_{1:t}$  为机器人在  $1 \sim t$  时刻内的观察数据,第  $k$  时刻机器人的观察数据为  $\mathbf{Z}_k$ , 其中  $z_{k,j}$  为第  $k$  时刻机器人观察到的第  $j$  个路标;  $\mathbf{u}_{1:t}$  为机器人在  $1 \sim t$  时刻接受到的控制指令;  $\mathbf{x}_{1:t}$  为机器人的位姿。

SLAM 问题的解决方法是贝叶斯估计,给定  $t - 1$  时刻概率分布  $p(\mathbf{x}_{t-1}, \mathbf{M} | \mathbf{Z}_{1:t-1}, \mathbf{u}_{1:t-1})$  作为先验估计,控制指令  $u_t$  和观测数据  $z_t$  为后验概率,利用贝叶斯更新就能

得到  $t$  时刻的状态估计值,分为以下两个步骤:

1) 状态预测

$$p(\mathbf{x}_t, \mathbf{M} | \mathbf{Z}_{1:t-1}, \mathbf{u}_{1:t}) =$$

$$\int p(\mathbf{x}_t | \mathbf{x}_{t-1}, u_t) \times p(\mathbf{x}_{t-1}, \mathbf{M} | \mathbf{Z}_{1:t-1}, \mathbf{u}_{1:t-1}) d\mathbf{x}_{t-1} \quad (8)$$

2) 状态更新

$$p(\mathbf{x}_t, \mathbf{M} | \mathbf{Z}_{1:t}, \mathbf{u}_{1:t}) =$$

$$\frac{p(\mathbf{Z}_t | \mathbf{x}_t, \mathbf{M}) \times p(\mathbf{x}_t, \mathbf{M} | \mathbf{Z}_{1:t-1}, \mathbf{u}_{1:t})}{p(\mathbf{Z}_t | \mathbf{Z}_{1:t-1}, \mathbf{u}_{1:t})} \quad (9)$$

利用以上两式进行贝叶斯递归运算,理论上可以解决 SLAM 问题。其中:

$$p(\mathbf{x}_t | \mathbf{x}_{t-1}, u_t), p(\mathbf{Z}_t | \mathbf{x}_t, \mathbf{M}) \quad (10)$$

分别表示状态转移概率与测量概率。

## 2 基于霍夫曼行程编码的 SLAM 算法

### 2.1 关键技术

1) 基于粒子滤波的 SLAM 算法

基于粒子滤波的 SLAM 算法主要有重要性采样、粒子重要性因子计算、重采样和环境地图更新 4 个步骤。Fast-SLAM<sup>[5]</sup> 算法就是使用粒子滤波算法估计机器人路径。Fast-SLAM 算法的核心思想是:在已知每一时刻机器人位姿的条件下,机器人路径与地图环境无关,对于每一个粒子,单个地图的误差是条件独立的。因此,地图构建问题可以分解为很多单独的问题,地图中的每个特征对应一个问题,那么 Fast-SLAM 问题将表示为:

$$p(\mathbf{x}_{1:t}, \mathbf{M} | \mathbf{Z}_{1:t}, \mathbf{u}_{1:t}) = p(\mathbf{x}_{1:t} | \mathbf{Z}_{1:t}, \mathbf{u}_{1:t}) \times$$

$$p(\mathbf{M} | \mathbf{x}_{1:t}, \mathbf{Z}_{1:t}) = p(\mathbf{x}_{1:t} | \mathbf{Z}_{1:t}, \mathbf{u}_{1:t}) \times \prod_{i=1}^n p(m_i | \mathbf{x}_{1:t}, \mathbf{Z}_{1:t}) \quad (11)$$

其中, Fast-SLAM 算法使用粒子滤波器计算机器人路径的后验概率:

$$p(\mathbf{x}_{1:t} | \mathbf{Z}_{1:t}, \mathbf{u}_{1:t}) \quad (12)$$

使用单独的位置后验概率:

$$p(m_i | \mathbf{x}_{1:t}, \mathbf{Z}_{1:t}) \quad (13)$$

表示地图中的每个特征,即每个路标点  $i (i = 1, 2, \dots, n)$  的特征,使用这种因式分解的方式将 SLAM 问题分解为路径和地图的  $n+1$  个后验概率的乘积。因为每个都代表机器人运动的轨迹,都能基于不同数据关联,所以在每个粒子的基础上估计数据关联变量成为可能,因此粒子滤波算法可以用简单而强大的机制来解决 SLAM 中的数据关联问题。

2) 基于霍夫曼行程编码的地图表示方式

上一节的最后提到基于粒子滤波的 SLAM 算法对内存资源的消耗过大,其直观表现为创建地图的数据量大,而基于粒子滤波的 SLAM 算法所构建的地图编码形式一

一般为如图6所示的栅格地图。栅格地图存在着大量的数据冗余,主要体现在两个方面;编码冗余,栅格地图为8比特灰度图像,可以表示256个灰度级,然而用于导航的地图中只需要3个灰度级,分别代表该区域为自由区域,占用区域和未知区域;空间冗余,栅格图像中,自由区域和未知区域通常大片的出现,也就是说这些区域的像素点是空间相关的,在相关的像素表示中,信息被没有必要的重复了。解决这两个问题的方法是利用霍夫曼编码和行程编码<sup>[15]</sup>来压缩栅格地图。

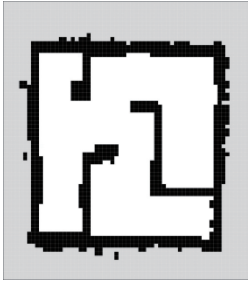


图6 栅格地图示例

Fig. 6 Raster map example

以栅格地图6为例,对其进行霍夫曼行程编码,过程如下:

(1)统计图像中各个灰度值的像素数量,通过像素数量计算归一化概率。如表1所示。

表1 像素概率统计表

Table 1 Pixel probability statistics table

灰度值	像素数	概率值
0	825	0.19
205	1754	0.40
254	1761	0.41

(2)根据概率从大到小的顺序,将像素灰度值排序,按照霍夫曼编码流程对每个像素进行编码;将原灰度与对应编码值之间的形成查找表。

(3)根据如表2查找表对整个图像中的每个像素进行编码。通过计算可知,经过霍夫曼编码后的平均码长为:

$$L_{avg} = 0.41 \times 1 + 0.4 \times 2 + 0.19 \times 2 = 1.59 \text{ bit/pixel} \quad (14)$$

表2 像素霍夫曼编码查找表

Table 2 Pixel Huffman coding lookup table

原始信源			信源化简	
灰度	概率	编码	概率	编码
254	0.41	0	0.41	0
205	0.4	11	0.59	1
0	0.19	10		

(4)对霍夫曼编码后的图像的每一行,按照像素的行程进行行程编码,行程编码后得到图7所示的数据结构。图中只取了图6中的几行像素,其中圆圈节点表示未知区域,空心正方形节点表示自由区域,实心正方形节点表示被占用区域,图形节点中的数字代表像素数,因此每个图形代表的节点占用两个存储空间,一个用来存储区域类型,另一个用来存储像素数。原本图像中存储一行像素点需要62个存储空间,而使用霍夫曼行程编码后,理论上—行数据最少只需要两个存储空间,存储效率大大提升。

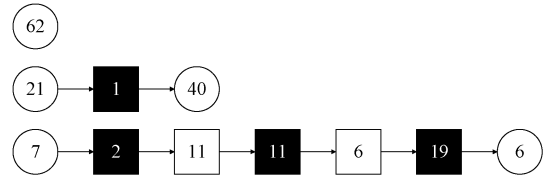


图7 霍夫曼行程编码地图表示

Fig. 7 Huffman run-length coding map representation

通过霍夫曼行程编码,可以将栅格地图无损的转化为霍夫曼行程编码地图,因为编码的每个过程都可以进行无损解码,所以该方法对栅格地图的信息保存的很完整。计算原地图、霍夫曼编码之后和霍夫曼行程编码地图所需的比特数量:

$$\text{原地图比特数} = 62 \times 70 \times 8 = 34\,720 \text{ bit}$$

$$\text{霍夫曼编码后比特数} = 1\,761 \times 1 + 1\,754 \times 2 + 825 \times 2 = 6\,919 \text{ bit}$$

$$\text{霍夫曼行程编码比特数} = \sum((\text{每一行数据行程编码比特数})) = 1\,996 \text{ bit}$$

$$\text{压缩率} = 34\,720 \div 1\,996 = 17.3948 \quad (15)$$

可知该栅格地图转化为霍夫曼行程编码地图后,被压缩了将近17.4倍,大大节省了内存资源。现在,本文可以将霍夫曼行程编码地图描述为:

(1)霍夫曼行程编码地图是一种数据结构,由3种带有数值的节点表示地图信息;

(2)每个节点代表一个实际空间区域,其中圆圈节点表示未知区域,空心正方形节点表示自由区域,实心正方形节点表示被占用区域,图形节点中的数字代表像素数;每个图形代表的节点占用两个存储空间,一个用来存储区域类型,另一个用来存储像素数;

(3)存储区域类型的空间中的数据是通过霍夫曼编码得到的,存储像素数的空间中的数据是通过行程编码得到的。

## 2.2 占用概率地图生成算法

本文所述的SLAM算法是建立在2.1节中所述的基于粒子滤波的SLAM算法上的,粒子滤波算法将SLAM问题中机器人位姿估计和地图构建两个问题相互独立,

其中机器人位姿估计问题可以使用本文所述的里程计模型解决,而地图构建问题常采用生成占用概率地图的方式解决。地图的占用概率表示为:

$$bel_i(m_i) = p(m_i | Z_{1:i}) \quad (16)$$

其中,  $bel_i(m_i)$  表示第  $m_i$  个区域被占用的概率,此概率的取值在 0~1 之间,可以通过前述激光数据反演观测概率,从而更新占用地图,详见《概率机器人》<sup>[16]</sup> 第 9 章。结合前述里程计模型、激光雷达传感器模型和粒子滤波算法,本文将占用概率地图更新算法描述如下:通过遥控机器人在室内移动,以遍历整个室内环境,在这个过程中,机器人定通过里程计模型进行定位,并利用激光雷达进行数据观测,观测数据经过观测模型和粒子滤波算法更新占用地图。

### 2.3 霍夫曼行程编码地图生成与应用

占用概率地图要使用霍夫曼行程编码地图表示必须先转化为栅格地图形式,占用概率地图中每个节点存储 0~1 的概率值,而栅格地图中每个像素存储 0~255 的灰度值,像素点灰度值与节点概率值之间的转化关系为:

$$V(m_i) = \lfloor 255 \times (1 - p(m_i)) \rfloor \quad (17)$$

其中“ $\lfloor \cdot \rfloor$ ”表示向下取整数。如此,概率占用地图在形式上变成了栅格地图,但是仍不满足霍夫曼行程编码的两个条件:编码冗余和空间冗余。原因在于由于概率值的分散性以及随机噪声,导致转化后的灰度图像灰度值可能分布在 0~255 这 256 个灰度级上,因此需要对灰度级设置阀门,将灰度分成黑白灰分别来表示占用、空闲和未知区域,设置下限阀门为  $V_{\min}$ ,上限阀门为  $V_{\max}$ ,用公式表示为:

$$V(m_i) = \begin{cases} 0, & V(m_i) \leq V_{\min} \\ 254, & V(m_i) \geq V_{\max} \\ 205, & \text{其他} \end{cases} \quad (18)$$

通过式(17)与(18),将占用概率地图转化为了只有 3 种灰度级的栅格地图,之后栅格地图就可以使用本文所述霍夫曼行程编码转化为霍夫曼行程编码地图。

实际应用中,在不同的环境中通常需要不同分辨率的地图来满足不同的应用场景,使用霍夫曼行程编码地图可以很容易的得到不同分辨率的地图,不需要重新进行地图构建,具体算法如下:

1) 将栅格地图用二元函数  $f(x,y)$  表示,函数值代表第  $x$  行第  $y$  个像素的灰度值。设原栅格地图为  $f(x,y)$  有  $Raw$  行  $Col$  列,分辨率降低的栅格地图  $g(x,y)$  有  $r$  行  $c$  列则有:

$$r = \lfloor Raw/2 \rfloor, c = \lfloor Col/2 \rfloor \quad (19)$$

2) 遍历图像计算分辨率降低后的各点像素值。取相邻 4 个像素点中的灰度最大值作为新生成图像中的像素

值。计算过程用公式表示为:

$$w\left(\frac{x+1}{2}, \frac{y+1}{2}\right) = \min(1 \cdot f(x+i, y+j)) \quad (20)$$

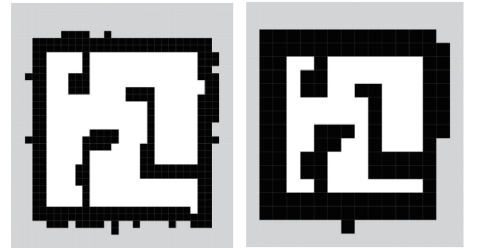
$$\left\{ \begin{array}{l} (x, y, i, j) \mid x = 2k + 1, x \leq Raw; \\ y = 2l + 1, y \leq Col; i, j \in \{0, 1\}; k, l \in N^+ \end{array} \right\}$$

3) 得到分辨率降低后的栅格地图  $g(x,y)$ , 其像素总数减少了 75%:

$$g(x,y) = w(x,y) \quad (21)$$

$$\{(x,y) \mid 1 \leq x \leq r, 1 \leq y \leq c\}$$

对图 6 进行一次和两次分辨率降低操作得到的栅格地图如图 8 所示,从图中可以看出该分辨率降低算法的效果是显著的。



(a) 分辨率降低一次 (a) Resolution reduced once  
(b) 分辨率降低两次 (b) Resolution reduced twice

图 8 分辨率降低后的栅格图

Fig. 8 Raster image with reduced resolution

### 2.4 霍夫曼行程编码地图存储方式

在实际问题中,地图的存储方式也是重要的问题,霍夫曼行程编码地图存储是将地图配置信息以及编码后的节点数据存储到储存设备中,并在有需要时无损恢复数据。其中霍夫曼行程编码地图的配置信息包含地图的行数与列数、霍夫曼编码查找表以及行程编码的数据格式,根据实际情形,本文为霍夫曼行程编码地图的数据存储格式设计了两算法,分别为定长编码算法和变长编码算法。

若采用定长编码策略,取最小的正整数  $i$  满足:

$$Length \leq (2^i + 2^{i-1}) \quad (22)$$

其中,地图的列数为  $Length$ ,则得到定长编码的比特数  $P=i$ 。再对整个图像进行霍夫曼编码,并生成地图配置信息。地图配置信息包含地图的行数与列数,霍夫曼编码查找表,该地图编码策略为定长编码策略,定长编码比特数  $P$ ,接下来对图像中的每一行进行行程编码,行程长度存储为  $P$  比特二进制数。

若采用变长编码策略,则取最小的正整数  $i$  满足:

$$Length \leq 2^{2^i} \quad (23)$$

对整个图像进行霍夫曼编码,并生成地图配置信息。地图配置信息包含地图的行数与列数,霍夫曼编码查找表,该地图编码策略为变长编码策略,变长编码标志位比

特数  $Q=i$ 。其中  $Q$  比特二进制标志位存储行程长度所需的二进制位宽  $w$ , 行程长度存储为  $w$  比特二进制数据。

再遍历图像的每一行, 对图像中的每一行采用行程编码, 编码格式为(区域类型,  $Q$  比特标志位, 行程长度), 区域类型对应编码由霍夫曼编码得到。

使用一定的编码方法将地图配置信息转换为二进制数据即可得到霍夫曼行程编码地图。

应用时根据地图的规模 and 实际需求采用不同的编码策略, 当栅格地图的规模较小时, 使用定长编码, 即用一个固定比特数的内存空间存储行程长度; 当栅格地图的规模比较大时, 例如商场环境, 地图规模可达万行万列级别, 则可以考虑使用变长编码。表 3 作为以上两种编码算法的一个示例, 其中霍夫曼编码查找表见表 2。

表 3 变长编码数据存储格式表

Table 3 Variable length encoding data storage format table

行程编码方式	数据节点表示	内存中二进制数据
定长编码 6 bit	(0, 28)	10011110
	(205, 90)	11111111111011010
变长编码标志位 4 bit	(0, 5, 28)	100101111100
	(205, 14, 16 383)	1111101111111100000

以上两种编码形式都可以有效地减少环境地图对内存资源的依赖, 在实际问题中, 还可以根据需求选择合适的存储方式。

### 3 实验结果及分析

为了验证基于霍夫曼行程编码的 SLAM 算法, 本文分别在仿真环境和真实环境中进行了 SLAM 实验。仿真实验基于机器人操作系统(robot operating system, ROS)和 Gazebo 仿真软件, 验证了算法的存储效率; 真实环境选取了一处居民住房, 实验验证了本文算法的实际可行性。此外本文还根据实际环境设计了多分辨率地图生成实验, 进一步验证了算法的实用性。

#### 3.1 探究霍夫曼行程编码地图的存储效率仿真实验

霍夫曼行程编码效率的影响因素主要为两个, 一为编码冗余, 一为空间冗余。本文利用两个闸门将图像灰度值划分为 3 个固定值, 所以图像的编码冗余基本是一个固定值, 几乎不受影响。而空间冗余则与环境地图的杂乱程度密切相关, 如果环境中有着许多的状态相同的区域, 则霍夫曼行程编码地图有着更高的存储效率。因为相同的区域状态使得单个像素的行程变长, 行程编码的存储效率提高了。为了探究不同环境中霍夫曼行程编

码地图的存储效率, 本文设计了一个可改变环境杂乱程度的仿真场景, 并根据此场景建立不同的霍夫曼行程编码地图。如图 9 所示。

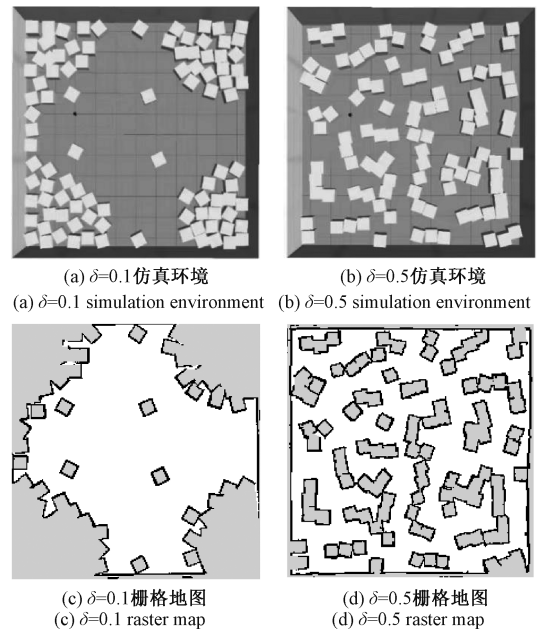


图 9 封闭室内环境及其栅格地图

Fig. 9 Enclosed indoor environment and its raster map

在  $11\text{ m} \times 11\text{ m}$  的封闭室内环境中随机生成 100 个边长为  $0.6\text{ m}$  的立方块, 将方块生成时距离墙的方差除以边长定义为环境的杂乱程度  $\delta$ , 随着  $\delta$  增大, 方块会以更大概率出现在封闭环境中的任意位置, 从而使得环境变得更加杂乱。图 9(a) 和 (b) 分别表示  $\delta=0.1$  和  $\delta=0.5$  的情形。首先分别对两者构建栅格地图如图 9(c) 和 (d) 所示, 再根据栅格地图生成霍夫曼行程编码地图。其中栅格地图 9(c) 有 222 行 220 列, 栅格地图 9(d) 有 224 行 223 列, 这个微小的差异是 SLAM 算法存在漂移造成的。实验结果如表 4 所示。

表 4 环境杂乱程度与所需内存关系表

Table 4 Table of the relationship between environmental clutter and required memory

编码形式	$\delta=0.1$		$\delta=0.5$	
	所需比特	减少消耗/%	所需比特	减少消耗/%
原栅格地图	390 720	0.00	399 616	0.00
霍夫曼编码	71 508	81.70	71 468	82.11
霍夫曼行程编码	20 052	94.86	48 755	87.80

该仿真实验表明在环境不复杂的条件下, 本文提出的基于霍夫曼行程编码的 SLAM 算法能够最多减少 94.8% 的内存消耗。进一步分析可知, 霍夫曼编码效率

基本不受环境杂乱程度影响,霍夫曼行程编码效率与环境杂乱程度呈负相关性,推理可知,在环境极度杂乱无章的情形下,霍夫曼行程编码效率甚至要低于仅进行一步的霍夫曼编码效率,也即消耗更多的内存空间,那么在这种情况下可以考虑将栅格地图只进行一步霍夫曼编码。

### 3.2 真实环境地图建立实验

真实实验场景在一处普通居民住房(示意图如图10所示)内,其大小为 $8\text{ m}\times 14\text{ m}$ ,其中摆放了一些常用的家具,比如床、沙发、柜子、椅子等。



图10 室内环境示意图

Fig. 10 Diagram of indoor environment

使用图2所示机器人 TurtleBot3 Burger 进行地图构建,用遥控器控制机器的运动,让其遍历整个室内环境,并收集激光雷达传感器的数据,使用本文提出的SLAM算法进行地图构建,构建的栅格地图如图11所示,其分辨率为 $0.05\text{ m}/\text{pixel}$ 。



图11 SLAM算法建立的栅格地图

Fig. 11 Raster map created by the SLAM algorithm

可以观察到图11中存在大量编码冗余和空间冗余,通过本文所述算法将栅格地图转化为霍夫曼行程编码地图,编码格式为7 bit 定长编码。原栅格地图需要存储空间为 $255\times 158\times 8=322\ 320\text{ bit}$ ,转化为霍夫曼行程编码地图后,所需存储空间仅为 $18\ 639\text{ bit}$ ,将原栅格地图压缩了17.3倍,减少内存消耗94.2%,符合仿真实验探究的结果。实验充分说明了本文所使用的SLAM算法的实际可行性。

### 3.3 多分辨率地图生成实验

如图12所示,利用2.3节中不同分辨率地图生成算法获取图11不同分辨率的栅格地图,其分辨率分别为 $0.05$ 、 $0.1$ 、 $0.2$ 和 $0.4\text{ m}/\text{pixel}$ 。随着分辨率的降低,可以观察到栅格地图中的白色部分慢慢减少,这是因为分辨率降低算法采取了灰度最小值,这是为了最大程度避免室内机器人在导航时发生碰撞。降低地图分辨率的意义在于:在导航全局路径规划算法中,地图的分辨率越低,路径规划速度越快;可以看到在分辨率为 $0.1\text{ m}/\text{pixel}$ 的栅格地图中,地图特征保留完好,地图内存减少75%,在实际中有着更好的应用。

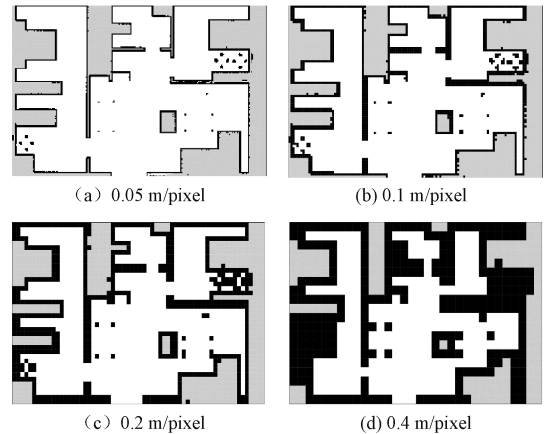


图12 不同分辨率的栅格地图

Fig. 12 Raster maps at different resolutions

## 4 结论

本文针对基于粒子滤波的SLAM算法生成的栅格地图存储效率不高的问题,提出了基于霍夫曼行程编码地图表示的SLAM算法,该方法能够有效地节省SLAM地图所需的内存空间。本文首先阐述了室内机器人概率模型;其次研究了霍夫曼行程编码地图生成方式;最后研究了霍夫曼行程编码地图的编码存储方式,当栅格地图的规模较小时,使用定长编码;当栅格地图的规模比较大时,考虑使用变长编码。在第3节中设计仿真实验和真实环境中的实验,证明了本文提出的算法存储的高效性与应用的广泛性。但本文提出的算法在应用方面还值得进一步的研究,例如提升地图访问效率算法的研究,考虑进一步提升算法实时性的一种方法是采用二维滑窗的方式对霍夫曼行程编码地图进行访问。

### 参考文献

- [1] SMITH R C. CHEESSEMAN P. On the representation and estimation of spatial uncertainty [J]. The International Journal of Robotics Research, 1986, 5(4): 56-68.

- [ 2 ] ZHANG J H, GONG D W, ZHANG Y. A niching PSO-based multi-robot cooperation method for localizing odor sources[J]. *Neurocomputing*, 2014(123): 308-317.
- [ 3 ] THRUN S. Particle filters in robotics [ C ]. Morgan Kaufmann Publisher Inc, 2002;511-518.
- [ 4 ] GRISETTI G, STACHNISS C, BURGARD W. Improved techniques for grid mapping with rao-blackwellized particle filter[J]. *IEEE Transactions on Robotics*, 2007, 23(1):34-46.
- [ 5 ] FAIGL J, KULICH M, VONÁSEK V, et al. An application of the self-organizing map in the non-euclidean traveling salesman problem[J]. *Neurocomputing*, 2011, 74(5): 671-679.
- [ 6 ] MCFETRIDGE L, IBRAHIM M Y. A new methodology of mobile robot navigation: The agoraphilic algorithm [ J ]. *Robotics and Computer-Integrated Manufacturing*, 2009, 25(3): 545-551.
- [ 7 ] 周治国, 曹江微, 邸顺帆. 3D 激光雷达 SLAM 算法综述[J]. *仪器仪表学报*, 2021, 42(9): 13-27.
- ZHOU ZH G, CAO J W, DI SH F. Overview of 3D lidar SLAM algorithms [ J ]. *Chinese Journal of Scientific Instrument*, 2021, 42(9):13-27.
- [ 8 ] DISSANAYAKE M W M G, NEWMAN P, CLARK S, et al. A solution to the simultaneous localization and map building (SLAM) problem [ J ]. *IEEE Transactions on Robotics and Automation*, 2001, 17(3): 229-241.
- [ 9 ] 李秀智, 李尚宇, 贾松敏, 等. 实时的移动机器人语义地图构建系统 [ J ]. *仪器仪表学报*, 2017, 38(11): 2769-2778.
- LI X ZH, LI SH Y, JIA S M, et al. System of real time mobile robot semantic map building [ J ]. *Chinese Journal of Scientific Instrument*, 2017, 38(11):2769-2778.
- [ 10 ] HORNUNG A, WURM K M, BENNEWITZ M, et al. OctoMap: An efficient probabilistic 3D mapping framework based on octrees [ J ]. *Autonomous Robots*, 2013, 34(3): 189-206.
- [ 11 ] 陈赢峰. 大规模复杂场景下室内服务机器人导航的研究 [ D ]. 合肥: 中国科学技术大学, 2017.
- CHEN Y F. Service robot navigation in large, dynamic and complex indoor environments [ D ]. Hefei: University of Science and Technology of China, 2017.
- [ 12 ] OLSON E B. Real-time correlative scan matching [ C ]. *International Conference on Robotics and Automation. IEEE*, 2009: 4387-4393.
- [ 13 ] YOONSEOK P, HANCHEOL C, RYUWOON J, et al. Ros robot programming [ M ]. Korea: ROBOTIS Co., Ltd., 2017: 2-10, 313-359.
- [ 14 ] WIJK O, CHRISTENSEN H I. Triangulation-based fusion of sonar data with application in robot pose tracking [ J ]. *IEEE Transactions on Robotics and Automation*, 2000, 16(6): 740-752.
- [ 15 ] RAFAEL C G, RICHARD E W. Digital image processing, third edition [ M ]. UA: Pearson Education, Inc, 2010.
- [ 16 ] THRUN S, BURGARD W, FOX D. Probabilistic robotics [ M ]. MIT Press, 2005.

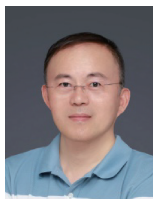
### 作者简介



邵锦江, 2020 年于武汉大学获得学士学位, 现为东南大学硕士研究生, 主要研究方向为惯性导航与组合导航。

E-mail: shaojinjiang@126.com

**Shao Jinjiang** received his B. Sc. degree from Wuhan University in 2020. He is currently a master student at Southeast University. His main research interests include inertial navigation and integrated navigation.



蔡体菁 (通信作者), 1983 年于北京大学获得学士学位, 1985 年于北京大学获得硕士学位, 1997 年于莫斯科大学获得博士学位, 现为东南大学教授, 主要研究方向为光电精密仪器与组合导航。

E-mail: tj\_cai@seu.edu.cn

**Cai Tijing** (Corresponding author) received his B. Sc. degree and M. Sc. degree both from Peking University in 1983 and 1985, and received his Ph. D. degree from Moscow State University in 1997. He is currently a professor at Southeast University. His main research interests include optoelectronic precision instrument and integrated navigation.