



# 自动规划调研报告

## 扫地机器人相关技术学习

第八组：58119131武英豪 58119132李凯新  
58119125蒋卓洋 58119121吕喆行

止于至善





東南大學  
SOUTHEAST UNIVERSITY

# 目录 content

止于至善

## 01 调研背景

## 02 SLAM技术综述

## 03 激光SLAM论文调研

## 04 视觉SLAM论文调研

## 05 规划过程实践

01

## 调研背景

(PPT制作：蒋卓洋)





## 扫地机器人的本质

真正的家电智能化

1. 扫地机器人产生的背景是**人类家务劳动自动化的需求**。
2. 目前大部分家电的使用依然是人类主导的半自动化智能，而**扫地机器人是真正名副其实的智能家电**。
3. 扫地机器人是**动态的**，是目前智能家电中唯一需要不停移动的品类。



感知

理解

决策

执行

## 扫地机器人的本质

### 搭载无人驾驶技术的家庭服务机器人

1. 无人驾驶是模拟人类的驾驶行为，人类的驾驶行为是一个周而复始的“**感知→理解→决策→执行**”过程，无人驾驶功能的实现也是在模拟四个层面。
2. 扫地机器人模拟人的清扫行为，**扫地机器人利用传感器对家庭环境信息进行认知分析、确定行走路线、做出行为决策**，和无人驾驶拥有相同的技术框架，也可以从感知、理解、决策和执行这四个层面去理解。



# 本组对扫地机器人相关自动规划技术的学习思路





理解层和决策层体现扫地机器人的智能化程度。当下扫地机器人的智商集中体现在**路线行走是否有逻辑**以及**能否避开潜在风险物品**，这两个痛点的解决分别对应无人驾驶的**SLAM技术**(Simultaneous Localization And Mapping, 即时定位与地图构建) 和**基于深度学习的视觉物体识别技术**。在扫地机器人的应用中，目前SLAM技术相对成熟，物体识别技术有待突破。



理解层和决策层体现扫地机器人的智能化程度。当下扫地机器人的智商集中体现在**路线行走是否有逻辑**以及**能否避开潜在风险物品**，这两个痛点的解决分别对应无人驾驶的**SLAM技术**(Simultaneous Localization And Mapping, 即时定位与地图构建) 和**基于深度学习的视觉物体识别技术**。在扫地机器人的应用中，目前SLAM技术相对成熟，物体识别技术有待突破。

能否避开潜在风险物品

基于深度学习的视觉物体识别技术

路线行走是否有逻辑

SLAM技术（即时定位与地图构建）



理解层和决策层体现扫地机器人的智能化程度。当下扫地机器人的智商集中体现在**路线行走是否有逻辑**以及**能否避开潜在风险物品**，这两个痛点的解决分别对应无人驾驶的**SLAM技术**(Simultaneous Localization And Mapping, 即时定位与地图构建) 和**基于深度学习的视觉物体识别技术**。在扫地机器人的应用中，目前SLAM技术相对成熟，物体识别技术有待突破。

能否避开潜在风险物品

基于深度学习的视觉物体识别技术

路线行走是否有逻辑

SLAM技术 (即时定位与地图构建)

02

# SLAM技术综述

(PPT制作: 蒋卓洋)

止于至善





## 重要名词解释

LDS

**LDS—Laser Direct Structuring**  
激光直接成型技术

VIO

**visual-inertial odometry**  
视觉惯性里程计

VO

**visual odometry**  
视觉里程计

IMU

**Inertial Measurement Unit**, 即惯性测量单元，用于测量物体三轴姿态角(或角速率)以及加速度。

ORB

ORB 是 Oriented Fast and Rotated Brief 的简称，可以用来对图像中的关键点快速创建特征向量，这些特征向量可以用来识别图像中的对象，其中，Fast 和 Brief 分别是特征检测算法和向量创建算法。



## 点云数据

激光SLAM和基于深度相机的视觉SLAM均是通过直接获取环境中的点云数据。

## SLAM

SLAM英文全称是Simultaneous Localization and Mapping，意为即时定位与地图构建。

它试图解决这样的问题：一个机器人在未知的环境中运动，如何通过对环境的观测确定自身的运动轨迹，同时构建出环境的地图。

## LDS-SLAM

### 基于激光实现的SLAM

通过激光雷达采集周围环境物体信息呈现出的一系列分散的、具有准确角度和距离信息的点(点云)



## vSLAM

### 基于相机等视觉器件实现的SLAM

视觉SLAM可分为单目(Mono)、双目(多目)、RGBD这三类，另还有鱼眼、全景等特殊相机。主要使用两种设计。第一个设计对应于基于滤波器的解决方案：包括基于扩展卡尔曼滤波(EKF-)的算法、基于粒子滤波的方法和基于多状态约束卡尔曼滤波(MSCKF-)的方法。第二种设计利用源自PTAM的并行方法，基于关键帧，使用全局优化集束调整(BA: Bundle Adjustment)方法。

## VI-SLAM

### 视觉-惯性SLAM

结合惯性测量器件(IMU)的视觉SLAM，是现在研究热点之一。



## 动态路线规划是否有逻辑



## SLAM技术 (即时定位与地图构建)

主要方向



- 激光SLAM (LDS SLAM)
- 视觉SLAM (vSLAM)

导航技术	清洁时间	覆盖率	脱困能力	技术研发成本	发展潜力
随机式	较长	较低	低	低	低
LDS SLAM	短	较高	较高	中	中
vSLAM	短	较高	较高	较高	高





## 动态路线规划是否有逻辑



## SLAM技术 (即时定位与地图构建)



### 激光SLAM (LDS SLAM)

主要方向



技术优点与缺点分析

- 优点：算法成熟稳定，建图和定位精度较高；可实现全局定位、回充续扫等功能。
- 缺点：成本相对较高；激光雷达的机械特性导致其易损坏，寿命有限；信息有限，只能提供二维平面的数据，没有其他信息，无法识别障碍物的类型；凸起的结构会增加扫地机器人的厚度，降低通过性



## 动态路线规划是否有逻辑



## SLAM技术 (即时定位与地图构建)

激光SLAM  
(LDS SLAM)

经典算法概览

主要方向



算法	硬件配置	精度	鲁棒性	开源代码
BLAM[80]-[83]	LiDAR	Good	Good	[84]
A-LOAM[76]	LiDAR	Very Good	Very Good	[85]
Cartographer[86]	LiDAR+IMU	Good	Good	[87]
HDL-graph-slam[88]	LiDAR	Good	Good	[89]
LeGO-LOAM[90]	LiDAR	Good	Good	[91]
LOAM-Livox[96][97]	MEMS-LiDAR	Good	Very Good	[98]
LIO-Mapping[92]	LiDAR+IMU+GPS	Excellent	Excellent	[93]
LINS[94]	LiDAR+IMU	Very Good	Very Good	[95]
LIO-SAM[99]	LiDAR+IMU	Very Good	Very Good	[100]
MULLS[101]	LiDAR	Excellent	Good	[102]
ISC-LOAM[103]	LiDAR	Very Good	Good	[104]
FAST_LIO[105]	LiDAR+IMU	Very Good	Very Good	[106]
BALM[107]	LiDAR	Excellent	Good	3D视觉[108]



## 动态路线规划是否有逻辑



## SLAM技术 (即时定位与地图构建)



### 视觉SLAM (vSLAM)

主要方向



技术优点与缺点分析

- 优点：硬件成本低，采集信息丰富，适用范围广，在功能实现上有更多可能性
- 缺点：导航精确度受环境光照、参照物的特征信息以及扫地机器人硬件条件等因素影响，导致定位精度的稳定性不够；过于复杂的环境会导致信息量会暴增，算法复杂度上升，可能影响精准定位



## 动态路线规划是否有逻辑



## SLAM技术 (即时定位与地图构建)



算法	硬件配置	特征类型	精度	鲁棒性	开源代码
PTAM[9]-[11]	Mono	Pyramid SSD	Good	Fair	[12]
LSD-SLAM[13][14]	Mono/Stereo	Direct	Fair	Fair	[15]
SVO[16][17]	Mono/Stereo	Direct	Good	Good	[18]
ORB-SLAM2[19][20]	Mono/Stereo	Descriptor	Very Good	Very Good	[21]
DSO[22]-[24]	Mono/Stereo	Direct	Good	Very Good	[25]
DSM[26]	Mono	Direct	Very Good	Very Good	[27]
MSCKF[28]-[31]	Mono+IMU	Cross	Good	Very Good	[32]
OKVIS[33][34]	Mono/Stereo+IMU	Descriptor	Good	Very Good	[35]
ROVIO[36][37]	Mono+IMU	Direct	Good	Good	[38]
ORB-SLAM-VI[39]	Mono/Stereo+IMU	Descriptor	Very Good	Very Good	-
VINS-Fusion[40][41]	Mono/Stereo+IMU	KLT	Excellent	Excellent	[42]
VI-DSO[43]	Mono+IMU	Direct	Very Good	Excellent	-
BASALT[44]	Stereo+IMU	KLT (LSSD)	Very Good	Excellent	[45]
Kimera[46]	Stereo+IMU	KLT	Good	Very Good	[47]
ORB-SLAM3[48]-[50]	Mono+Stereo+IMU	Descriptor	Excellent	Excellent	[51]
OPEN-VINS[52]	Mono+Stereo+IMU	KLT	Excellent	Excellent	[53]
PL-VIO[54]	Mono+IMU+Line	KLT	Good	Very Good	[55]
PL-SLAM[56]	Mono+Line	Descriptor	Good	Very Good	[57]
SE2CLAM[58]	Mono+Encoder	Descriptor	Excellent	Very Good	[59]
SE2LAM[60]	Mono+Encoder	Descriptor	Excellent	Very Good	[61]

03

# 激光SLAM论文调研

(PPT制作：蒋卓洋)

止于至善





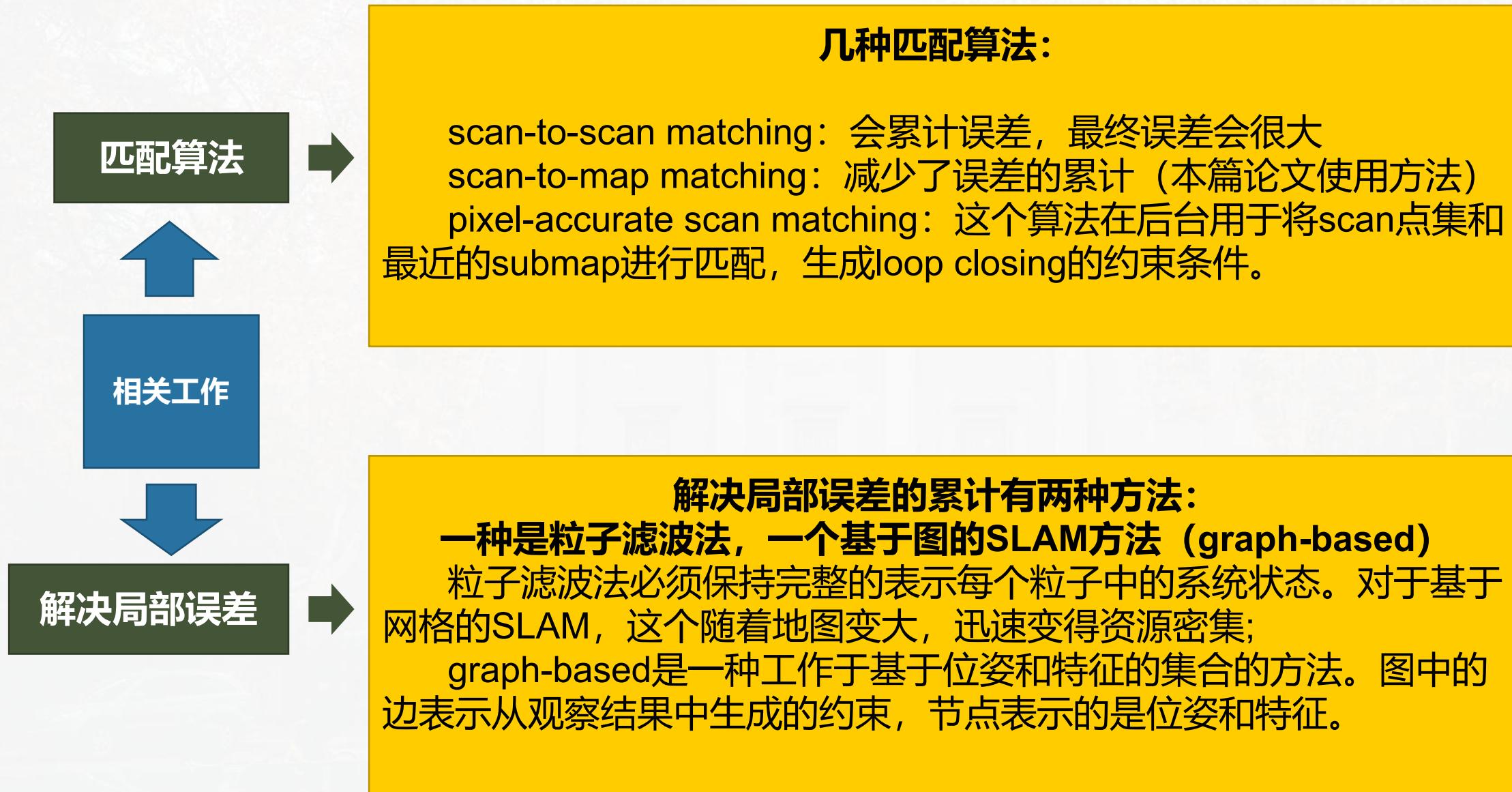
## Real-Time Loop Closure in 2D LIDAR SLAM

Wolfgang Hess<sup>1</sup>, Damon Kohler<sup>1</sup>, Holger Rapp<sup>1</sup>, Daniel Andor<sup>1</sup>



提出问题

人工调查收集数据缓慢，不能准确描述空间。SLAM更加快速和准确。

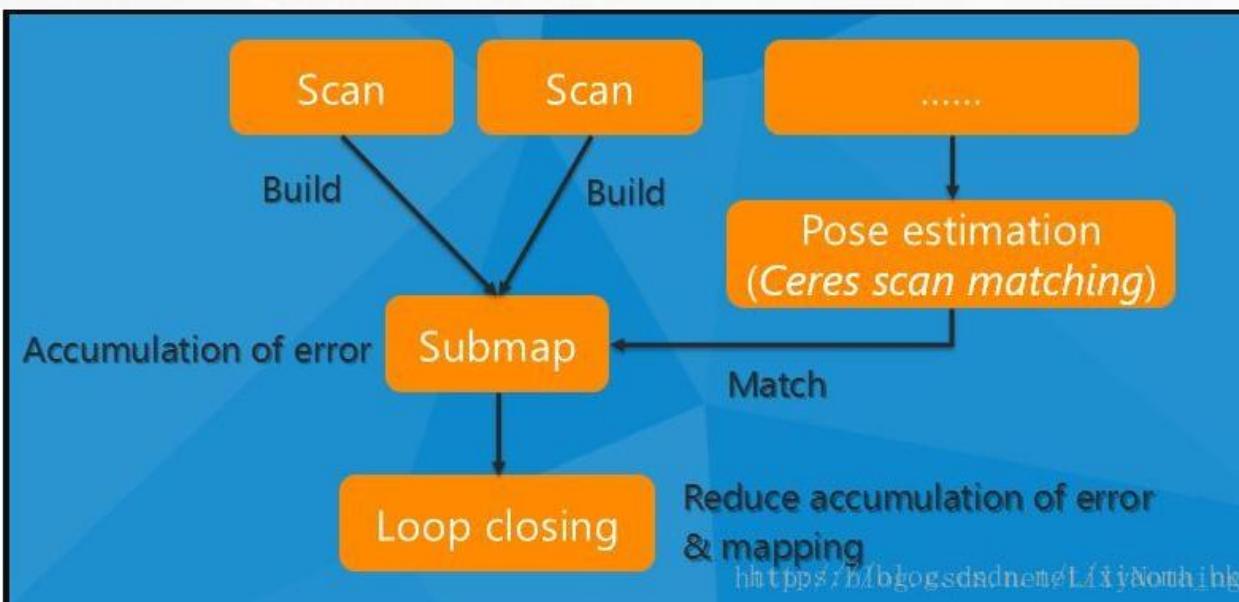


Google  
Cartographer

## 系统概述

## 室内实时绘图的解决方法

基于激光雷达的方法。绘制的是2D图像，分辨率是5cm。激光数据scans会以最佳的位置插入到submap中，这个最佳的位置假设在一段时间内是很准确的。当一个子图构建完成之后，不在有其它的scan插入这个子图当中，这个已有的子图会用来作为loop closing的scan matching。



## 累积误差优化方法

应用回环检测对累积误差进行优化。所有创建完成的submap以及当前的laser scan都会用作回环检测的scan matching。如果当前的scan和所有已创建完成的submap在距离上足够近，则进行回环检测。应用branch and bound(分支定界)优化方法进行优化搜索，减少计算量。

Google  
Cartographer

## 2D SLAM

## 局部二维的实时定位和绘图

系统实现2DSLAM结合了局部的和全局的方法，局部和全局的方法都对LIDAR观测到的位置进行了优化。

局部方法中，每个连续的点集被拿来和整个地图的一部分进行匹配，就是和submap进行匹配。使用一种非线性的优化方法将submap和scan点集联系起来，这也是scan matching的过程。

## Scans

一个scans即激光点云图，包含一个起点和许多的终点。起点称为origin，终点称为scan points。

## Submaps

一个submap是通过几个连续的scans创建而成的，由 $5\text{cm} \times 5\text{cm}$ 大小的概率栅格构造而成。

## Ceres scan matching

每次获得的最新的scan需要插入到submap中最优的位置，使我们scan中的点束的位姿经过转换后落到submap中时，每个点的信度和最高。

Google  
Cartographer

## 2D SLAM

## 局部二维的实时定位和绘图

系统实现2DSLAM结合了局部的和全局的方法，局部和全局的方法都对LIDAR观测到的位置进行了优化。

局部方法中，每个连续的点集被拿来和整个地图的一部分进行匹配，就是和submap进行匹配。使用一种非线性的优化方法将submap和scan点集联系起来，这也是scan matching的过程。

## Scans

一个scans即激光点云图，包含一个起点和许多的终点。起点称为origin，终点称为scan points。

## Submaps

一个submap是通过几个连续的scans创建而成的，由 $5\text{cm} \times 5\text{cm}$ 大小的概率栅格构造而成。

## Ceres scan matching

每次获得的最新的scan需要插入到submap中最优的位置，使我们scan中的点束的位姿经过转换后落到submap中时，每个点的信度和最高。

Google  
Cartographer

## 2D SLAM

## 闭环的实现

一个scan点集被插入submap中的这个相对位置会被存储下来，用于之后的闭环优化。除了这些位置信息，还有的包含scan点集和submap，而且这个submap不再变化的时候，都会被用来作为闭环检测。一个scan matcher一直在后台不断的运行，当一个好的scan match被找到之后，这个相应的位置也会被加入到优化问题中。

## 检测回环

如果当前的scan和所有已创建完成的submap中的某个laser scan的位姿在距离上足够近，那么通过某种 scan match策略就会找到该闭环。这里为了减少计算量，提高实时回环检测的效率，Cartographer应用了branch and bound(分支定界)优化方法进行优化搜索。

## 优化回环

在一个新的scan点集的周围画一个圆，然后不断的修改x, y, 以及角度，采用分支上界法来计算窗口内最佳位置。为了达到一个实际的算法，需要分为以下几步：节点选择，分支，计算上限。

Google  
Cartographer

## 结论

## 实验结果

这项研究已经通过在慕尼黑的德国博物馆进行的实验得到了验证，结果快速而准确，证明了在适度的硬件上实时运行该算法是可能的。

## 总结

这篇论文阐述了一个2D的SLAM的系统，这个系统采用了闭环检测的scan-to-submap matching，同时还有图优化 (graph optimization)。一个submap的创建是使用的是局部的，基于网格的 (grid-based) SLAM方法。在后台，所有的点集与相近的submap的匹配使用的是pixel-accurate scan matching的方法，然后建立闭环检测的约束。这个约束图 (基于submap和scan pose的) 都会周期性的在后台被更改。这个操作是采用GPU加速将已完成的submap和当前的submap进行结合。

04

# 视觉SLAM论文调研

(PPT制作：蒋卓洋)

止于至善



**IMU**

**Inertial Measurement Unit, 惯性测量单元**

**VO**

**visual odometry**  
视觉里程计

**VIO**

**visual-inertial odometry**  
视觉惯性里程计

**vSLAM****基于相机等视觉器件实现的SLAM**

视觉SLAM可分为单目（Mono）、双目（多目）、RGBD这三类，另还有鱼眼、全景等特殊相机。主要使用两种设计：

第一种设计对应于基于滤波器的解决方案：包括基于扩展卡尔曼滤波(EKF-)的算法、基于粒子滤波的方法和基于多状态约束卡尔曼滤波 (MSCKF-) 的方法。

**第二种设计则是利用了源自 PTAM (parallel Tracking and Mapping) 的并行方法，基于关键帧，使用BA的优化方法 (Bundle Adjustment: 本质是一个优化模型，其目的是最小化相机重投影误差。)**

**ORB**

**ORB** 是 Oriented Fast and Rotated Brief 的简称，可以用来对图像中的关键点快速创建特征向量，这些特征向量可以用来识别图像中的对象，其中，Fast 和 Brief 分别是特征检测算法和向量创建算法。

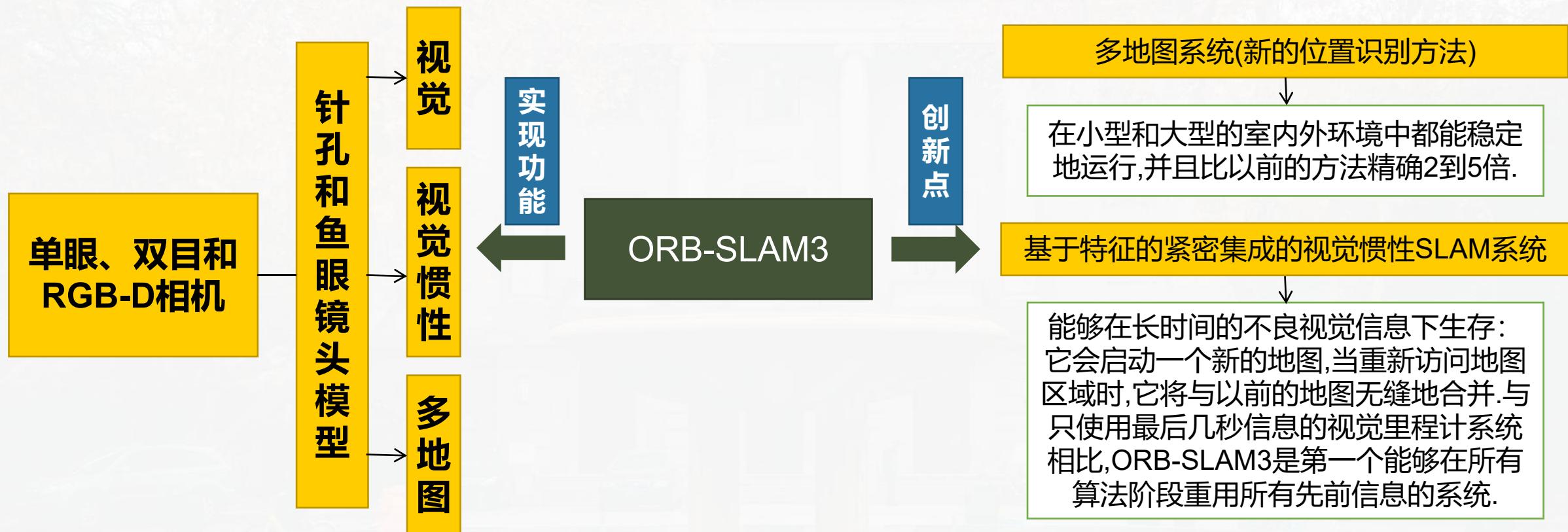
**VI-SLAM****视觉-惯性SLAM**

结合惯性测量器件 (IMU) 的视觉SLAM，是现在研究热点之一。



# ORB-SLAM3: An Accurate Open-Source Library for Visual, Visual-Inertial and Multi-Map SLAM

Carlos Campos\*, Richard Elvira\*, Juan J. Gómez Rodríguez, José M.M. Montiel and Juan D. Tardós





ORB-SLAM3

问题的提出

构造迄今为止最完整和准确的  
视觉、视觉惯性和多地图SLAM系统SLAM算法  
鲁棒性面临  
的主要挑战**数据关联错误，是导致SLAM算法失效的一个主要原因。**

数据关联把观测值和相应状态进行关联。以特征点法视觉SLAM为例，它把每一个特征点与相应的路标点进行关联。

感知混淆 (perceptual aliasing, 感知混淆是指，对于不同的输入，传感器感知到了相同信号的现象) 使得正确的数据关联变得特别困难。感知混淆会使数据关联建立错误的观测-状态 (measurement-state) 匹配关系 (false positive, 假阳性)，这会导致后端优化作出错误的状态估计。

随着最近集成闭环技术的VO系统的出现, VO和SLAM之间的边界更加分散.



ORB-SLAM3



## 问题的提出

随着最近集成闭环技术的VO系统的出现, VO和SLAM之间的边界更加分散.

视觉SLAM的目标是使用传感器来构建环境地图并实时计算姿态

SLAM地图的最大优势是它允许匹配和使用BA先前的观测执行三种类型的数据关联:

短期数据关联: 匹配最近几秒内获得的地图元素

中期数据关联: 匹配离累计漂移仍较小的相机较近的地图元素.

长期数据关联: 使用位置识别技术将观察结果与以前访问过的区域中的元素进行匹配, 而不管累积漂移(环路检测)或跟踪是否丢失(重新定位).

VO系统将注意力放在计算代理的自我运动上,而不是构建地图上.





ORB-SLAM3

问题的提出

构造迄今为止最完整和准确的  
视觉、视觉惯性和多地图SLAM系统

视觉SLAM的目标是使用传感器来构建环境地图并实时计算姿态

VO系统将注意力放在计算代理的自我运动上,而不是构建地图上.

## 实现三种类型的数据关联:

短期

匹配最近几秒内获得的地图元素.这是大多数VO系统使用的唯一数据关联类型,一旦环境元素离开视野,它们就会被遗忘,即使系统在同一区域移动,也会导致持续的估计漂移

中期

匹配离累计漂移仍然较小的相机较近的地图元素.这些可以在BA中以与短期观测相同的方式进行匹配和使用,并允许在系统在地图区域移动时达到零漂移.它们是ORB-SLAM3与具有环路检测的VO系统相比获得更高精度的关键

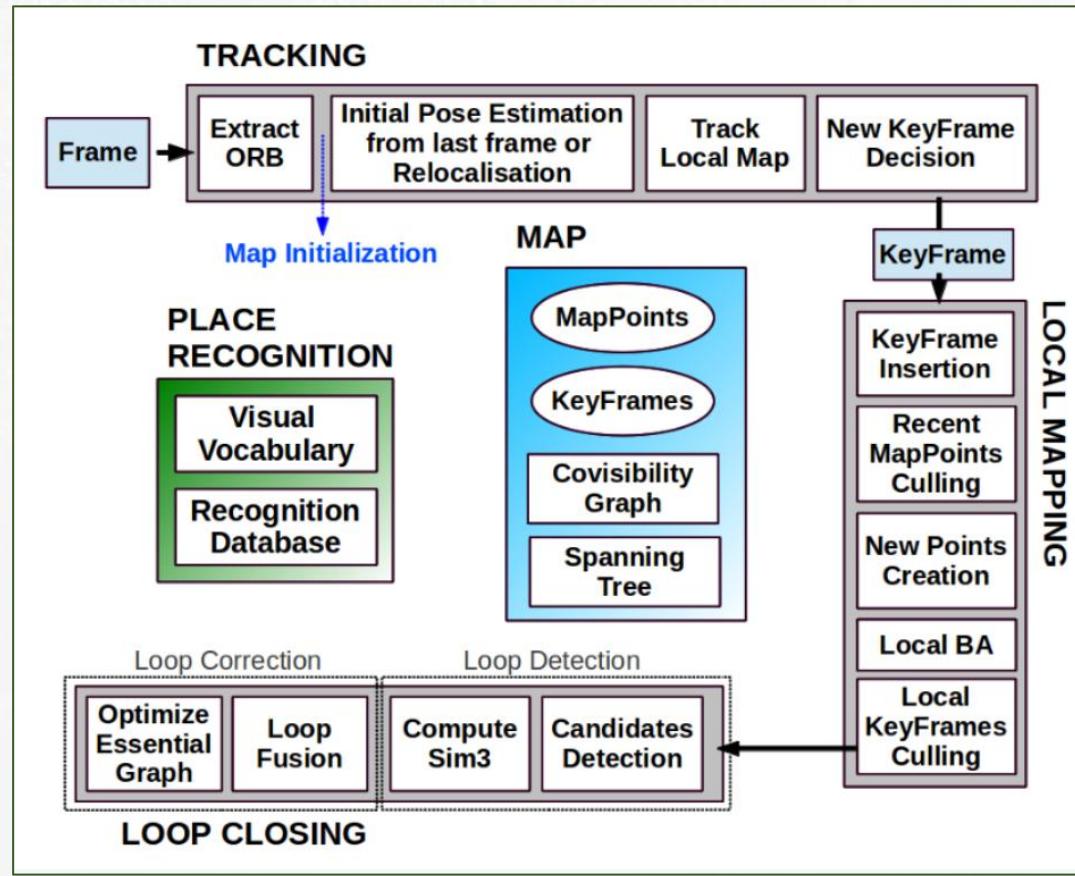
长期

使用位置识别技术将观察结果与以前访问过的区域中的元素进行匹配,而不管累积漂移(环路检测)或跟踪是否丢失(重新定位).长期匹配允许使用姿态图优化或更准确地说,使用BA来重置漂移和校正循环.这是中大型环型环境下SLAM精度的关键。

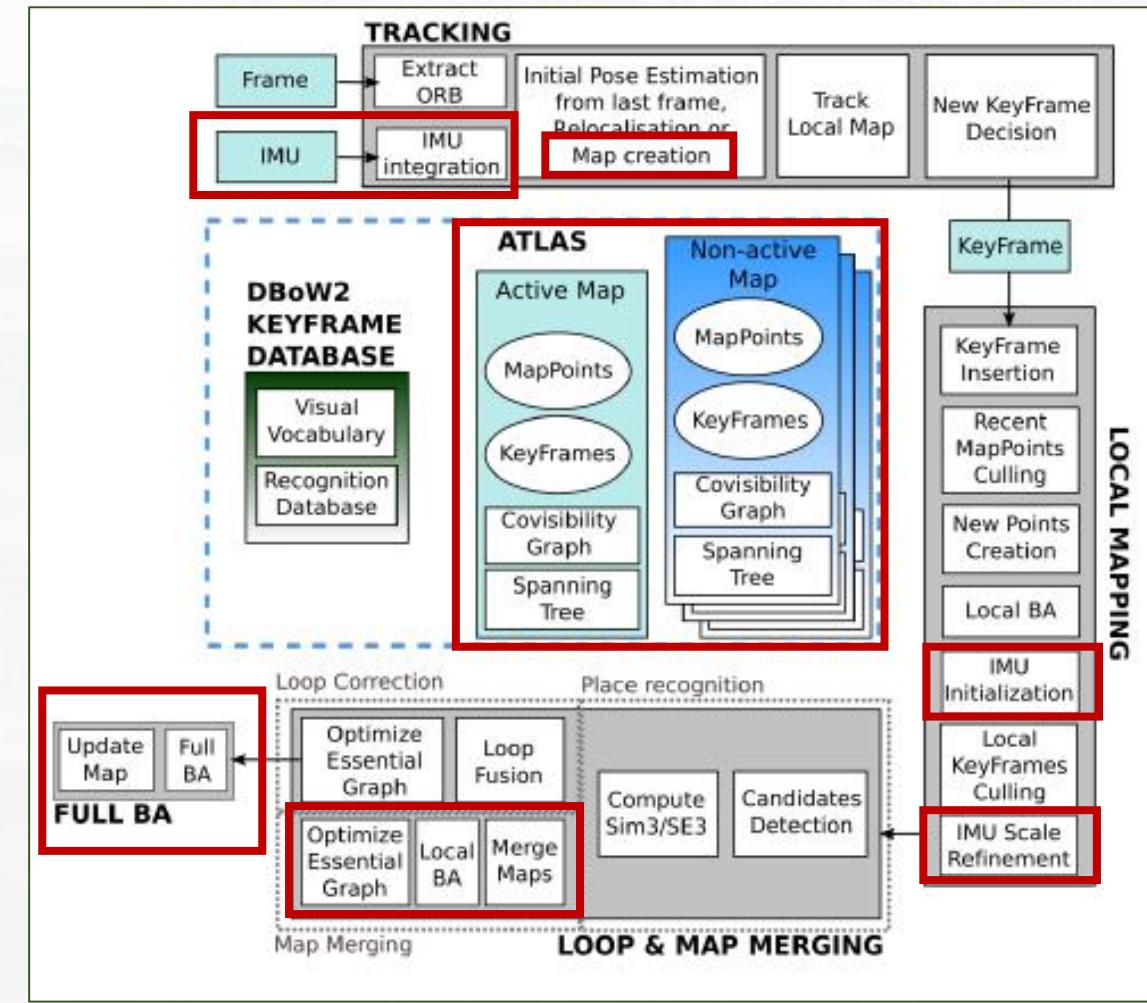


## ORB-SLAM3

## 框架概述



ORB-SLAM2



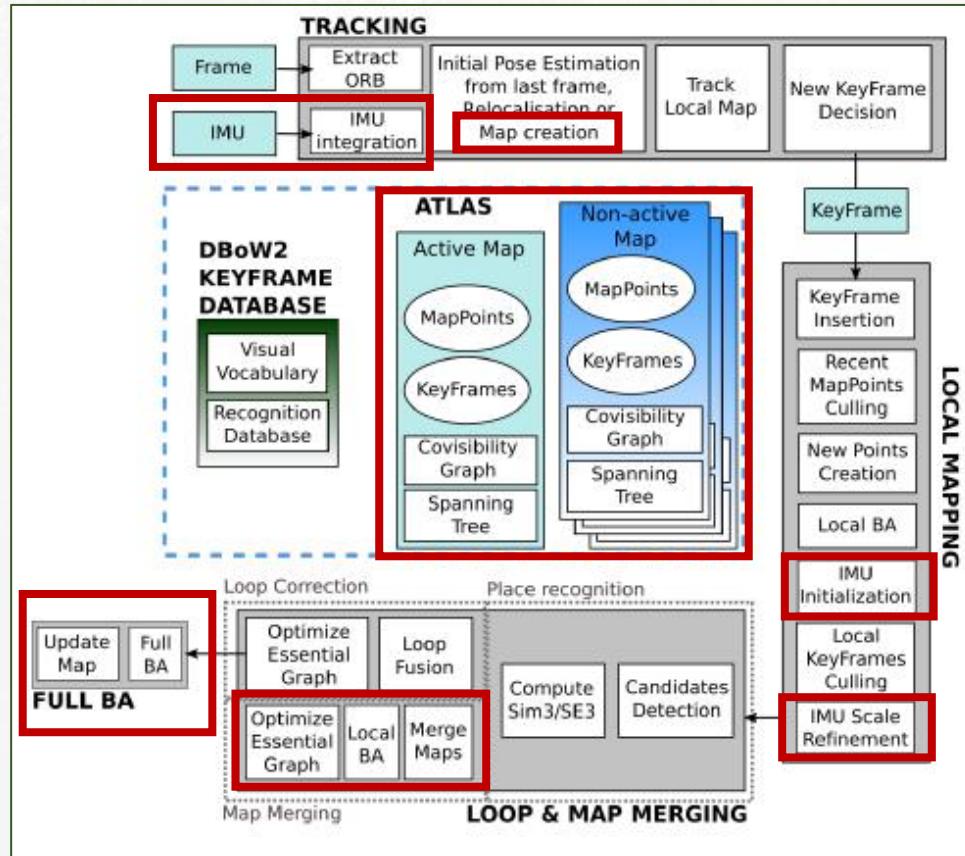
ORB-SLAM3



ORB-SLAM3

框架概述

与ORB-SLAM2相比的新增模块



ORB-SLAM3

## Atlas (地图集)

包括一系列分离的地图组成的多地图表示（包括 active map 和 non-active-map）。动态地图表示当前位于的map，跟踪线程向其中传入帧，由local mapping不断优化以及增加新的关键帧扩大规模。

## Tracking thread (跟踪线程)

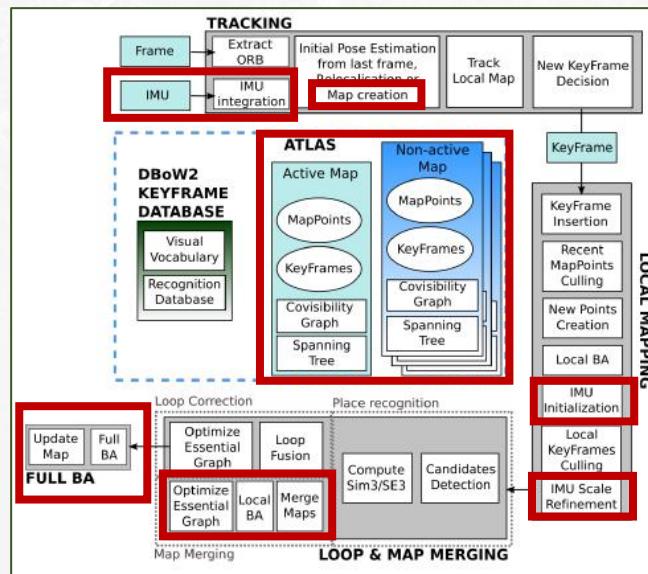
判断当前帧是否为关键帧；实时处理传感器信息并计算当前姿态；最大限度地减少匹配地图特征的(eprojection error)投影误差。地图跟丢时，进行重定位，成功则继续跟踪，失败则在短暂延时后将当前地图存储为非动态地图，同时再新建一个动态地图。



ORB-SLAM3

框架概述

与ORB-SLAM2相比的新增模块



ORB-SLAM3

## Local mapping thread (局部建图线程)

将关键帧和地图点添加到活动地图，删除redundant(多余关键帧)，并使用视觉或视觉惯性BA优化地图。另外此线程也包括惯性情况下，利用MAP估计初始化和优化IMU参数。

## Loop and map merging thread (回环和地图融合线程)

基于关键帧的速度,对地图集中的动态地图和 非动态地图进行相似性度量.如果相似区域处于活动地图,则进行回环校正;若属于不同的地图,则将两个地图进行无缝拼接,并作为活动地图.回环校正时,另开一个线程进行全局BA.



ORB-SLAM3



## 框架概述



## ORB-SLAM3的关键技术

## 相机模型

ORB-SLAM3默认所有系统组件都是针孔摄像机模型,这样通过提取与摄像机模型相关的参数抽象出摄像机模型,同时ORB-SLAM3还提供了鱼眼模型.

## 视觉惯性融合SLAM

ORB-SLAM-VI是第一个真正能够重复使用地图的、视觉惯性SLAM,但是仅限与单目且初始化较慢.ORB-SLAM3建立在ORB-SLAM-VI上,提供了一种快速精准IMU初始化策略.

## 地图合并

## Visual Map Merging (视觉地图合并)

如果位置识别成功,产生了多地图数据关联,在动态地图中的关键帧和地图集中的不同地图中的匹配关键帧之间,使用对齐变换进行地图合并操作.需要确保**MergeMaps**中的信息能被tracking线程及时调用,避免地图重复。包括:

- ① **Welding window assembly** (融合窗口集合)
- ② **Merging maps** (地图合并) ③ **Welding bundle adjustment** (融合BA)
- ④ **Pose-graph optimization** (位姿图优化)

## Visual-Inertial Map Merging (视觉惯性地图合并)

视觉惯性合并算法的步骤与纯视觉情况相似,修改步骤①③,以便更好地利用惯性信息



ORB-SLAM3

实验成果

ORB-SLAM3在EuRoC数据集上的效果

- 在单目和双目中,ORB-SLAM3比ORB-SLAM2更精确,因为更好的位置识别算法可以更早地结束循环,并提供更多的中期匹配.
- 在单目惯性中,ORB-SLAM3的精度是VI-DSO和VINS-Mono的两倍多,显示了中长期数据关联的优势.新颖的快速IMU初始化允许ORB-SLAM3在几秒钟内校准惯性传感器,并从一开始就使用它,能够完成所有EuRoC序列,并获得更好的精度.
- 在双目惯性中,ORB-SLAM3也远远优于OKVIS、VINS-Fusion和Kimera.

Table II: Performance comparison in the EuRoC dataset (RMSE ATE in m., scale error in %). Except where noted, we show results reported by the authors of each system, for all the frames in the trajectory, comparing with the processed GT.

		MH01	MH02	MH03	MH04	MH05	V101	V102	V103	V201	V202	V203	Avg <sup>1</sup>
Monocular	ORB-SLAM <sup>2,3</sup> [4]	0.071	0.067	0.071	0.082	0.060	<b>0.015</b>	0.020	-	<b>0.021</b>	<b>0.018</b>	-	0.047*
	DSO <sup>2,3</sup> [27]	0.046	0.046	0.172	3.810	0.110	0.089	0.107	0.903	0.044	0.132	1.152	0.601
	DSM <sup>2,3</sup> [31]	0.039	0.036	0.055	<b>0.057</b>	0.067	0.095	0.059	0.076	0.056	0.057	<b>0.784</b>	<b>0.126</b>
	ORB-SLAM3 (ours)	<b>0.017</b>	<b>0.017</b>	<b>0.031</b>	0.066	<b>0.044</b>	0.033	<b>0.016</b>	<b>0.037</b>	<b>0.021</b>	0.022	-	0.030*
Stereo	ORB-SLAM2 <sup>2,3</sup> [3]	0.035	<b>0.018</b>	0.028	0.119	0.060	<b>0.035</b>	<b>0.020</b>	<b>0.048</b>	0.037	0.035	-	0.044*
	VINS Fusion <sup>2,3</sup> [44]	0.540	0.460	0.330	0.780	0.500	0.550	0.230	-	0.230	0.200	-	0.424*
	SVO <sup>2,3</sup> [24]	0.040	0.070	0.270	0.170	0.120	0.040	0.040	0.070	0.050	0.090	0.790	0.159
	ORB-SLAM3 (ours)	<b>0.025</b>	0.022	<b>0.027</b>	<b>0.089</b>	<b>0.058</b>	<b>0.035</b>	0.021	0.049	<b>0.032</b>	<b>0.027</b>	<b>0.361</b>	<b>0.068</b>
Monocular Inertial	ORB-SLAM <sup>2,3</sup> VI [4]	0.075	0.084	0.087	0.217	0.082	<b>0.027</b>	0.028	-	<b>0.032</b>	0.041	0.074	0.075*
	scale error <sup>2,3</sup>	0.5	0.8	1.5	3.5	0.5	0.9	0.8	-	0.2	1.4	0.7	1.1*
	VINS Mono <sup>2,3</sup> [7]	0.084	0.105	0.074	0.122	0.147	0.047	0.066	0.180	0.056	0.090	0.244	0.110
	VI-DSO <sup>2,3</sup> [46]	0.062	<b>0.044</b>	0.117	0.132	0.121	0.059	0.067	0.096	0.040	0.062	0.174	0.089
Stereo Inertial	ORB-SLAM3 <sup>2,3</sup> scale error	<b>0.032</b>	0.053	<b>0.033</b>	<b>0.099</b>	<b>0.071</b>	0.043	<b>0.016</b>	<b>0.025</b>	0.041	<b>0.015</b>	<b>0.037</b>	<b>0.042</b>
	(ours)	0.7	1.0	0.3	1.0	0.6	1.5	0.5	1.1	0.5	0.3	0.9	0.8
	OKVIS <sup>2,3</sup> [39]	0.160	0.220	0.240	0.340	0.470	0.090	0.200	0.240	0.130	0.160	0.290	0.230
	VINS Fusion <sup>2,3</sup> [44]	0.166	0.152	0.125	0.280	0.284	0.076	0.069	0.114	0.066	0.091	0.096	0.138
	BASALT <sup>2,3</sup> [47]	0.080	0.060	0.050	0.100	<b>0.080</b>	0.040	0.020	0.030	<b>0.030</b>	0.020	-	0.051*
	Kimera <sup>2,3</sup> [8]	0.080	0.090	0.110	0.150	0.240	0.050	0.110	0.120	0.070	0.100	0.190	0.119
	ORB-SLAM3 <sup>2,3</sup> (ours)	<b>0.037</b>	<b>0.031</b>	<b>0.026</b>	<b>0.059</b>	0.086	<b>0.037</b>	<b>0.014</b>	<b>0.023</b>	0.037	<b>0.014</b>	<b>0.029</b>	<b>0.036</b>
	scale error	0.7	0.2	0.2	0.4	1.0	0.6	0.6	0.6	1.4	0.2	0.8	0.6



ORB-SLAM3

实验成果

ORB-SLAM3的创新总结

## 1.一种单目和双目视觉惯性SLAM系统

扩展至双目惯性SLAM，且ORB-SLAM3单目和双目视觉惯性系统都是极其鲁棒的。

## 2.高召回率位置识别算法

之前的位置识别算法以牺牲召回为代价来提高精度，ORB-SLAM3提出了一种新的位置识别算法，首先检查候选关键帧的几何一致性，然后检查与三个可共视关键帧的局部一致性，这三个关键帧在大多数情况下已经在地图中。这种策略提高了召回率，同时增加了数据关联度，提高了地图精度，但代价是计算成本稍高。

## 3.多地图SLAM系统

第一个完整的多地图SLAM系统，能够处理视觉和视觉惯性系统，且适合单目和双目的配置。地图集可以表示一组不连续的地图，并在其上平滑地应用所有的制图操作：位置识别、相机重新定位、闭环和精确的无缝地图合并。这允许自动使用和组合在不同时间构建的地图，执行增量多会话SLAM。在原版ORB基础上添加了新的地点识别系统，可视化多地图系统及其对公共数据集的评估。

## 4.抽象的相机模型表示

使SLAM与所使用的相机模型无关，并允许通过提供投影、反投影和雅可比函数来添加新模型。同时提供了针孔和鱼眼两种模型实现。



[1] Cadena, Cesar, et al. "Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age." *IEEE Transactions on robotics* 32.6 (2016): 1309-1332.

---

[2] Konolige, Kurt, et al. "A low-cost laser distance sensor." 2008 IEEE international conference on robotics and automation. IEEE, 2008.

---

[3] Hess, Wolfgang, et al. "Real-time loop closure in 2D LIDAR SLAM." 2016 IEEE international conference on robotics and automation (ICRA). IEEE, 2016.

---

[4] Campos, Carlos, et al. "Orb-slam3: An accurate open-source library for visual, visual-inertial, and multimap slam." *IEEE Transactions on Robotics* 37.6 (2021): 1874-1890.

05

# 规划过程实践

(PPT制作：李凯新、武英豪)

止于至善





## 扫地机器人规划过程实践



矩阵模拟随机地图+PDDL规划

(第一次汇报内容)



可视化游戏模拟+PDDL生成规划



更复杂的游戏场景模拟



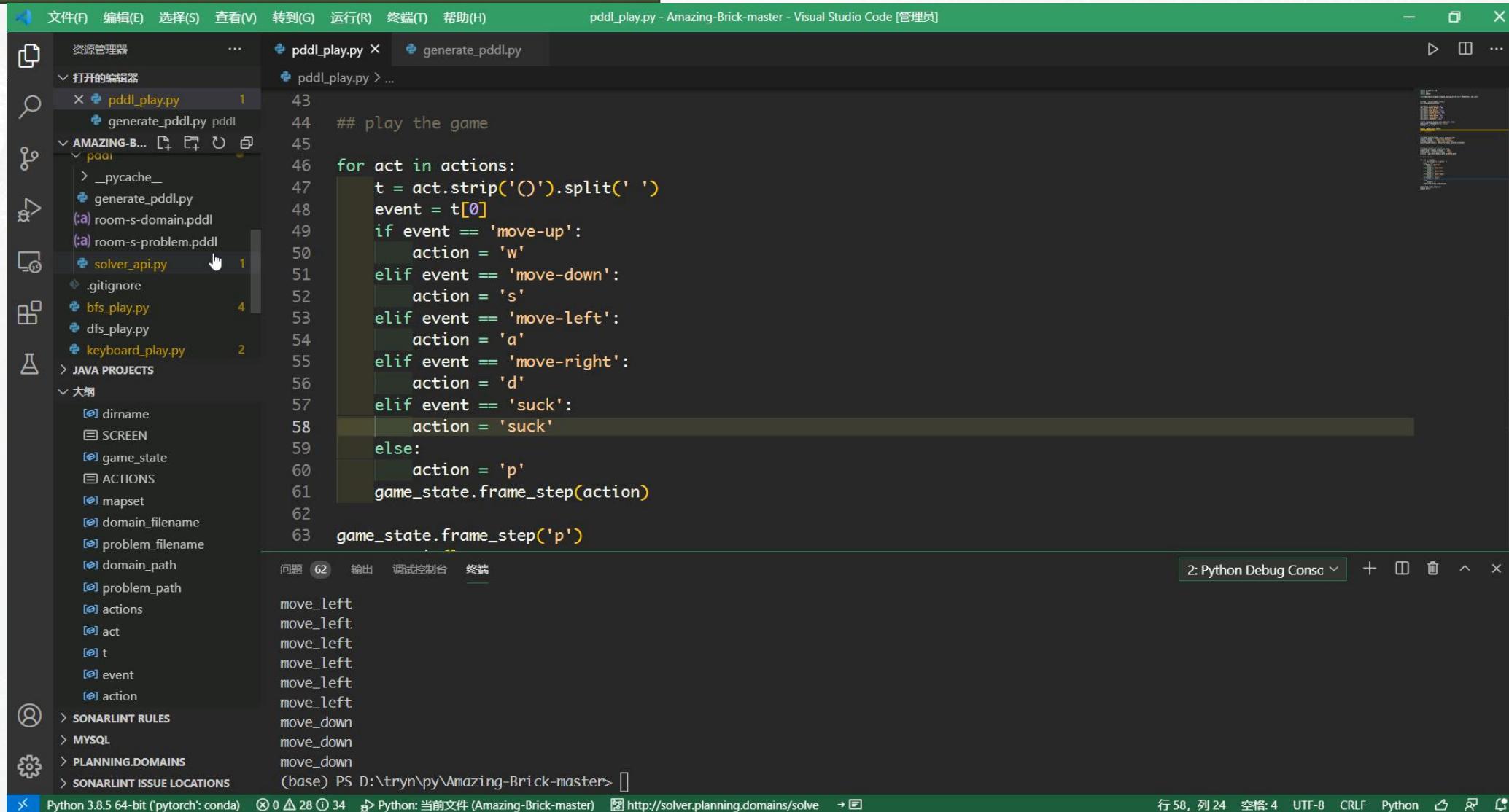
复杂场景自动规划



.....



## 可视化游戏模拟+PDDL生成规划



The screenshot shows a Visual Studio Code interface with the following details:

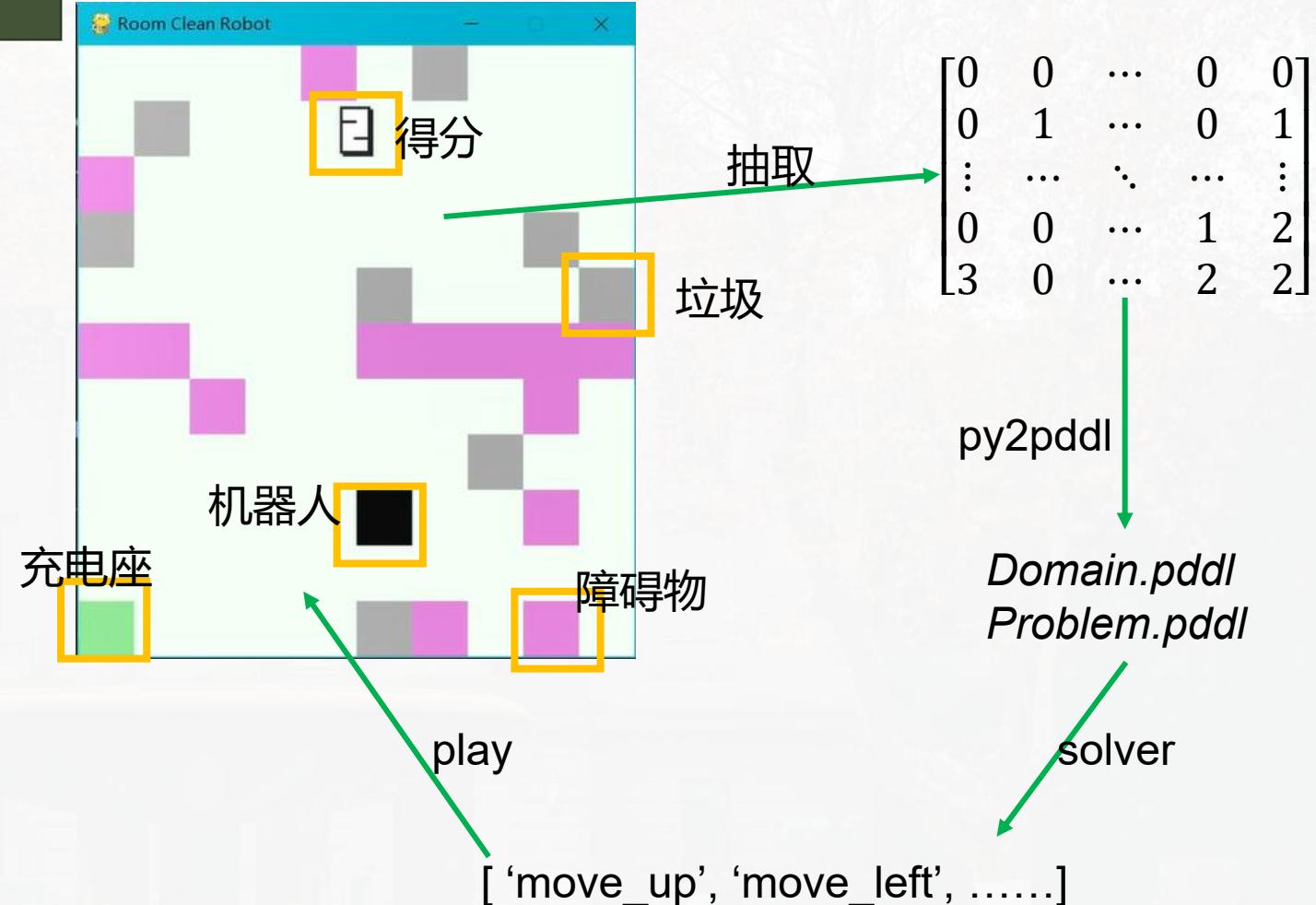
- File Explorer:** Shows files in the project structure:
  - AMAZING-BRICKS-MASTER
  - pddl
  - paai
  - solver\_api.py (selected)
  - bfs\_play.py
  - dfs\_play.py
  - keyboard\_play.py
- Editor:** The file `pddl_play.py` is open, containing Python code for game simulation and PDDL generation. The code uses a dictionary `actions` to map events to actions (w, s, a, d, suck, p) and calls `game_state.frame_step` for each action.
- Terminal:** The terminal shows command-line output for the `game_state.frame_step` function, with repeated calls to `move_left`.
- Status Bar:** Shows the Python version (3.8.5), environment (pytorch: conda), file count (0), line count (28), and other system information.



## 可视化游戏模拟+PDDL生成规划

## 实现细节

- pygame实现界面可视化+游戏逻辑
- 抽取地图表示
- py2pddl脚本生成pddl描述
- pddl solver api获得行动序列

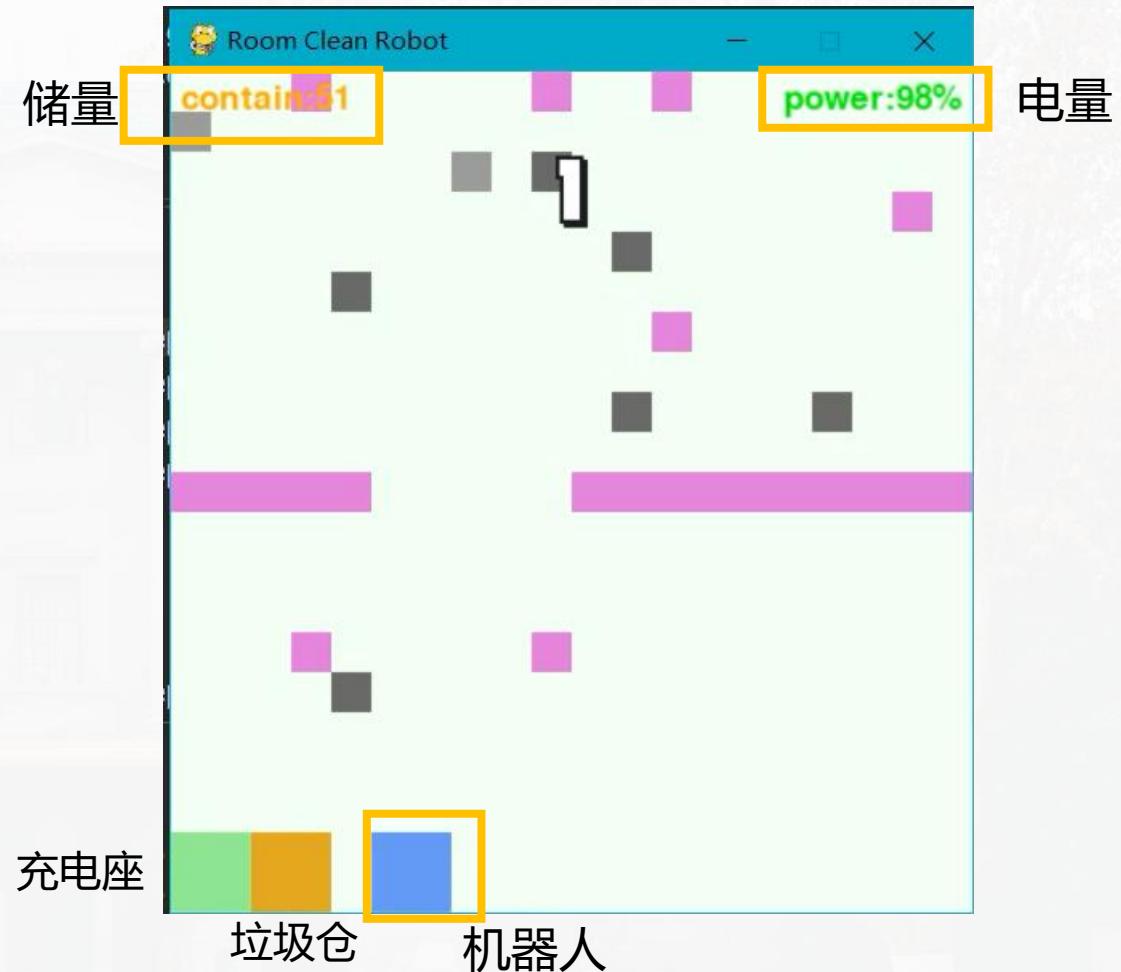




## 游戏场景细化

### 实现细节

- 随机生成垃圾和障碍
- 机器人不同尺寸
- 电量与充电功能
- 垃圾的数量化、倒垃圾功能
- 有限视野范围、障碍遮挡视野



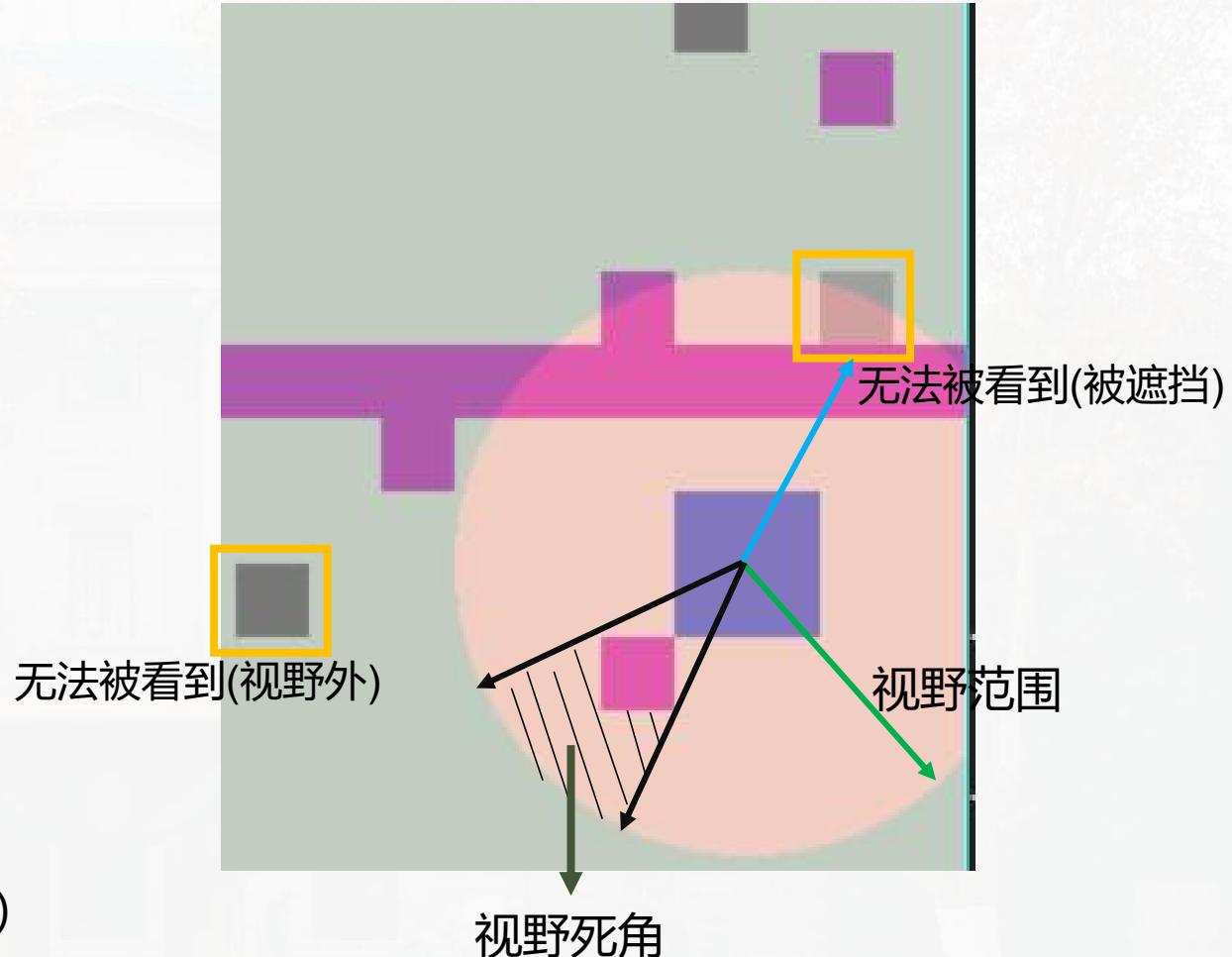


## 游戏场景细化

### 视野范围、视野遮挡

- 设定有限的视野范围  
机器人无法感知到视野外的物体
- 考虑障碍物对视野的遮挡  
基于角度计算遮挡视野，更加贴合实际且精确

(为后续部分可知环境下的自动规划提供基础)





## 复杂场景自动规划

### 实现细节

环境并非完全可知

考虑全局规划+局部规划相结合的方式

- 全局规划：  
使用带启发式的动态优先级生成方法，  
决定每一个阶段的行动
- 局部(单步)规划：  
使用A\*算法进行单次行动的寻路，指哪打哪

任务生成器：  
根据当前信息生成若干带  
优先级的任务

如清理指定点、  
充电、探索未知  
区域、倒垃圾

最高优先级任务

执行任务：  
使用A\*算法，完成主要寻  
路类型的任务

主要为到达  
任务目标地  
点

获得下一次任务



## 复杂场景自动规划

## 结果演示

```
292
293 # 根据记忆地图生成任务并执行
294
295 ## play the game
296 game_state.frame_step()
297 game_state.frame_step()
298 game_state.frame_step()
299 game_state.frame_step()
300
301 auto_clean()
302
303
304 game_state.frame_step()
305 pygame.quit()
306
307
308 # step list1 = auto((15,
```

问题 51 输出 调试控制台 终端

```
move_left
move_left
move_left
move_left
move_left
move_left
(base) PS D:\tryn\py\Amazing-Brick-master> d:; cd 'd:\tryn\py\Amazing-Brick-master'; & 'C:\Anaconda3\envs\pytorch\python.exe' sions\ms-python.python-2021.5.926500501\pythonFiles\lib\python\debugpy\launcher' '56494' '--' 'd:\tryn\py\Amazing-Brick-master'
pygame 2.1.2 (SDL 2.0.18, Python 3.8.5)
Hello from the pygame community. https://www.pygame.org/contribute.html
libpng warning: iCCP: known incorrect sRGB profile
libpng warning: iCCP: known incorrect sRGB profile
```



# 感谢聆听

止于至善

