arXiv:2602.02389v1 [cs.RO] 2 Feb 2026

面向水下结构协作机器人检测的映射引导的任务发现与分配

Marina Ruediger[1] 和 Ashis G. Banerjee[2]

摘要——**在不具备现有几何形状先验知识的情况下，水下多机器人检查的任务生成可通过分析同步定位与建图（SLAM）数据来实现并优化。通过考虑硬件参数和环境条件，从SLAM网格生成一组任务，并借助预期关键点得分和基于距离的剪枝进行优化。水中试验用于证明该算法的有效性并确定合适的参数。将这些结果与测试环境模型上针对检查覆盖范围的仿真Voronoi分区和往复式覆盖模式进行比较。所提出的任务发现方法的主要优势包括对意外几何形状的适应能力，以及能够在保持覆盖率的同时聚焦于更可能出现缺陷或损伤区域的分布。**

I. 引言

船体与内部空间的自主检查对于评估远洋船舶的安全性与就绪状态至关重要，并用于检测腐蚀、生物污损、裂纹、涂层失效及其他损伤。已有许多工作致力于实现水下检测自动化，但据我们所知，这些都是单机器人系统。现有水下检测工具与方法的综述可见于[1], [2]。然而，诸多挑战依然存在，原因包括高昂的运行成本（昂贵且高度定制的平台）；巨大的作业规模（在开阔水域作业时检测大型船舶上的微小缺陷）；以及系统鲁棒性与韧性不足（机器人硬件故障和自主决策失败）。因此，我们考虑一种低成本协作多机器人系统，并提出一种基于SLAM的任务发现与分配算法，在不具备先验几何知识的情况下从环境网格生成检查任务。该方法旨在应对关键的水下挑战：从机器人感知和/或运动能力的丧失或增加中恢复；极其缓慢且不可靠的通信；以及由浑浊度和稀疏特征导致的有限可见性，这些因素使定位与建图变得复杂。

我们的方法基于一种去中心化多机器人任务分配框架，该框架采用最小消息大小和间歇式通信 [3]。在此基础上，我们扩展了这一框架，加入一种自适应任务发现方法，将检查工作集中在 SLAM 网格生成过程中识别出的几何特征区域。具体而言，我们提出了一种新颖算法，可自动从 SLAM 生成的网格中发现检查任务，供执行船体或充液内部舱罐视觉检测的异构多机器人团队使用。检查过程包括前往由位置与姿态定义的位置并拍摄彩色图像。选定的位置称为任务，而发现过程中可能的位置称为候选检测点（CIP）。

第2节概述水下 SLAM 和任务生成算法的相关背景信息。第3节详细介绍我们的任务发现算法——映射引导的检测任务：发现与分配（Map-TIDAL），包括任务发现过程及其与任务分配的整合。第4节介绍水中测试结果，并与其他任务生成方法进行比较。第5节讨论其优缺点，第6节则给出结论。

II. 背景

A. 水下同步定位与建图

水下同步定位与建图领域已取得诸多进展，相关内容总结见 [4], [5]。由于水下环境极为特殊，许多用于空中或地面同步定位与建图方案的常见传感器无法使用或存在局限。效果良好的视觉传感器包括单目 [6]和双目相机，但需注意颜色偏移、水体透明度 [7], 和图像畸变[8]。声呐系统广泛用于基于测距的同步定位与建图 [9]–[11], ，因为声波在水中的衰减率显著较低，且不受光照条件变化的影响。不过，声呐系统通常价格昂贵。作为替代方案，人们也采用了激光雷达系统，尽管其工作光频与空中应用中的同类系统不同 [12]。

基于内部状态的传感器通常与外部环境传感器融合，以克服外部感知的挑战。多普勒测速仪（DVL）利用声脉冲计算速度，罗盘提供姿态数据，惯性测量单元（IMU）测量线加速度和角加速度，而深度传感器则根据水压计算深度 [4]。通常会融合多个传感器，以提供可靠的解决方案。例如， [13]–[14] 结合了视觉、惯性和声呐数据； [15] 使用视觉、惯性和深度数据； [16]融合了多普勒测速仪、视觉和惯性数据； [17] 融合了声呐、视觉、惯性和水压数据； [18] 则介绍了一个新数据集及相应的解决方案，该方案涵盖立体相机、惯性测量单元、多普勒测速仪和压力传感器数据。

*本研究部分由 ONR 资助，资助编号为 # N00014-21-1-2075。[1]Marina Ruediger 就职于美国华盛顿州西雅图市华盛顿大学机械工程系，邮编 98195marina77@uw.edu[2]Ashis G. Banerjee 就职于美国华盛顿州西雅图市华盛顿大学工业与系统工程系和机械工程系，邮编 98195ashisb@uw.edu

B. 检查覆盖范围

多机器人（或更广义地说，多智能体）覆盖问题广泛存在于诸多领域，包括检查与监视、搜救、农业、制造业、交通管理、执法等。尽管已有许多解决方案，但通常，覆盖问题要么通过详细考量智能体在复杂但已知的几何形状中的视点选择来解决，要么将一个简单的有界区域划分给一组能力已知的智能体。

例如，基于光学参数的测量不确定性用于在 [19] 中生成一组视点。随后，采用增强型快速探索随机树计算能够保持所有测量点完整覆盖率和可见性的最优视点子集。 [20]，提出了一种半自动视点规划器，该规划器利用相机与激光器之间的三维三角测量，根据现有网格模型主动计算视点。 [21] 开发了一种适用于未知二维环境的多机器人往复式覆盖方法，该方法利用探索机器人之间的视距通信。该方法依靠围绕障碍物进行单元划分，使由一组覆盖机器人和最多两组探索机器人组成的异构系统能够实现覆盖。 [22] 提出了一种基于传感器的覆盖路径规划方法，该方法利用基于广义Voronoi图的图对多个机器人的能量容量进行建模。

III. 方法

A. 任务分配

Map-TIDAL 采用一种基于基于共识的捆绑算法（CBBA）的分散式任务分配框架，并针对水下多机器人检查进行了领域特定的修改 [3]。任务分配系统通过分布式拍卖机制运行，每个机器人在保持自主决策能力的同时，通过最少的消息交换与团队进行协调——这对于带宽受限的水下通信环境至关重要。

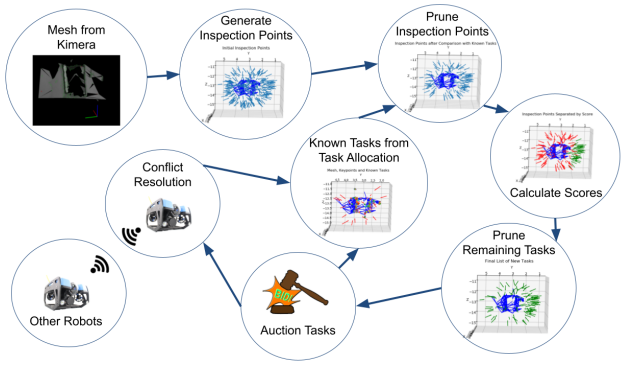


图 1。 任务发现与任务分配流程 。

每个机器人执行一个本地任务管理器，负责维护：(1) 已知任务清单，(2) 竞价历史来自相邻机器人，(3) 通信时序日志，以及 (4) 冲突解决协议。拍卖过程采用基于路径优化的奖励函数，其中将给定任务纳入某一机器人任务包的奖励，按该任务最优插入后的总路径距离与原始路径配置之差计算。

竞价机制采用基于距离的指标，其中每台机器人的出价代表到达相应任务位置所需的累计距离。这种方法在确保收敛性的同时，能够高效地估算完成任务所需的工作量。本地拍卖阶段结束后，出价信息会传播给邻近机器人，以遵循 [23] 中确立的共识协议进行分散式冲突解决。

在 Map-TIDAL 中，将任务分配与 SLAM 和任务发现流程相结合，可实现完全自主运行；新发现的任务会无缝纳入现有分配框架，并在整个机器人网络中高效传达。图 1 展示了流程中任务分配与任务发现步骤的衔接，涵盖从 SLAM 关键帧网格到通信和冲突解决阶段的过程。基于共识的任务共享机制自然形成了对检查区域空间覆盖情况的分布式认知。对现有任务位置的这种共享认知，使任务发现过程中的网格分析更加智能：新生成的候选检测点不仅可以根据其当前的视点覆盖进行评估，还可以根据其对通过集体任务清单识别出的整体覆盖缺口所作的贡献进行评估。

B. 任务发现

在本研究中，我们使用 Kimera-Multi [24], [25] 融合惯性测量单元、双目视觉和多普勒测速仪数据。Kimera-Multi 原本面向空中和地面环境，但也能够处理水下传感器融合。使用经过正确标定的相机，Kimera-Multi 可利用结构规律 [26]，根据极其稀疏的关键点数据计算位置并生成轻量级三维网格。该网格非常适合水下检测，因为水下视觉特征通常只聚集在视觉差异显著的区域，例如藤壶或藻类斑块、划痕或锈蚀区域，或焊缝、海水箱等复杂几何体。这些视觉特征正是检查过程中最需要重点关注的对象。在特征极少的大面积平滑区域，机器人将依靠多普勒测速仪和惯性测量单元进行定位；此时网格的密度会较低，生成的任务也会较少。

Map-TIDAL 流程中的任务发现部分主要分为四个步骤；每当 Kimera 生成关键帧网格 ($\mathcal{M}$) 时，该流程便会启动。

1) **候选检测点生成：** 对于网格 $\mathcal{M}$ 中的每个三角形，沿表面法线方向投影理想长度（L_i），生成候选检测点（CIP），如图2a所示。L_i 根据相机焦距和水

浑浊度进行优化，以最大限度地提高图像质量，同时与船体表面保持安全间距。

2) a) **基于现有任务的剪枝**（当 $T_e \neq \emptyset$时）：每个CIP都会根据双重几何约束，与现有任务 T_e 进行空间筛选：排斥半径（r_{ex}）和排斥角（θ_{ex}）。如果任何现有任务同时位于球形半径 r_{ex} 和角锥 θ_{ex}内，则丢弃该CIP，以确保保留的点能提供足够不同的视点来实现检查覆盖率，如图2b所示。如果 T_e仅包含距任何CIP都在 r_{ex} 之外的点，则改为执行步骤2b。

OR

b) **基于自比较的剪枝**（当 $T_e = \emptyset$时）：如果任何CIP的 r_{ex} 范围内均不存在先前任务，则使用较低的阈值（ r_{ex} 的$p_s = 70\%$和 θ_{ex}）对CIP进行自比较，以剔除候选集中冗余的检查点，避免在特征丰富的区域生成过于密集的任务。

3) **关键点评分：**通过投射相机视场（FOV）角锥并统计其与网格关键点的相交数量，对每个剩余的CIP进行评估，如图2c所示。若某CIP的得分高于局部网格区域内最大可能关键点数的阈值（t）百分比，则将其提升为已确认任务，优先考虑几何结构复杂、可能存在缺陷的区域。

4) **最终空间剪枝：**剩余低于阈值的 CIP 会依据步骤 2a 中的空间排斥准则，与新确认的任务集进行迭代比较。任务按顺序处理，每加入一项任务，都会扩大后续候选项的排斥区域，从而确保最终任务集具有最佳空间分布，如图 2d 所示。

新任务列表确定后，便会发送至 Map-TIDAL 的拍卖阶段；随后，该阶段会将新任务信息发送给处于冲突解决阶段的其他机器人，并更新主任务列表。完整的 Map-TIDAL 流程见算法 1 和图 1。

IV. 实验结果

A. 物理设置与数据采集

为验证 Map-TIDAL，我们在装有盐水的 OSB测试水槽中采集了两组现场数据。我们使用了由两台 BlueROV2 [27]机器人组成的团队，并为其配备 NVIDIA Jetson Xavier AGX 机载处理器、Waterlinked A50 多普勒测速仪（仅安装在目标机器人上）以及 OAK-D立体深度相机。系绳用于安全监控，并在必要时接管手动控制；所有必要的处理和记录均由 Jetson 完成。

在第一次测试中，我们采集了一段短视频和 IMU数据，机器人由人手持并保持稳定位置。随后使用这段视频数据进行离线Kimera处理，以确定初步可行性并开发算法。图 3

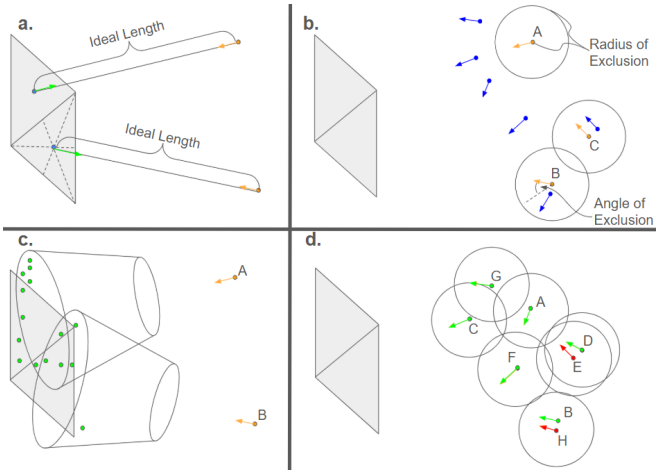


图 2. **a.** 利用质心（蓝色）和法向量（绿色）从网格（灰色）生成检查点（橙色）。**b.**将检查点（橙色）与现有任务（蓝色）进行比较：A 点的排斥半径内没有任务；B 点的包含半径内有一个任务，但其角度超出排斥角；C 点存在一个同时位于排斥半径和排斥角范围内的任务，因此不会被移除。**c.** 根据相机视场计算关键点得分。该得分为检查点（橙色）视场内的关键点（绿色）总数除以关键点数量。**d.**对剩余检查点进行最终剪枝，按字母顺序处理任务：移除任务 E 和 H，其余成为任务。

算法 1任务发现过程

输入： Kimera 网格（$\mathcal{M}$）、现有任务列表（T_e）
参数： L_i, r_{ex}, θ_{ex}, t, FOV
输出： 新任务列表（T_{new}）

▷ ↩ 表示追加到列表末尾
▷ ∅ 表示一个空列表（有序）

从 $\mathcal{M}$ 提取关键点（$\mathcal{K}$）和多边形（$\mathcal{P}$）
$C \leftarrow$ 生成点($\mathcal{P}$) ▷ 步骤1
如果 T_e 为 ∅**则**
  $C \leftarrow$剪枝(C,∅,2) ▷ 步骤 2a
否则
  $C \leftarrow$剪枝(C,T_e,0) ▷ 步骤 2b
$T_{new} \leftarrow \emptyset$
$C_{remains} \leftarrow \emptyset$
$S \leftarrow$ 计算分数($C, \mathcal{K}$) ▷ 步骤 3
对于 i **在** $length(S)$**中执行**
  如果 $S[i] \geq t$ **则**
    $T_{new} \hookleftarrow C[i]$
  else
    $C_{remains} \hookleftarrow C[i]$
$T_{new} \hookleftarrow$ 剪枝($C_{remains}, T_{new}$,1) ▷ 步骤 4
返回 T_{new}

算法 2剪枝

参数： L_i，r_{ex}，θ_{ex}，FOV，p_s

函数 PRUNE($C, C_{com}{}_{p}, n$)
  $precheck \leftarrow 1$ ▷用于标记先前任务是否存在于 r_{ex}
  $C_{pruned} \leftarrow \emptyset$
  $C_{compare} \leftarrow \emptyset$
  如果 $n = 0$ **则** ▷ C 仅与 C_{comp}比较
    $C_{compare} \leftarrow C_{comp}$
  否则，如果 $n = 1$，**则** ▷C 与 C_{comp} 相比，关于自身任务
    $C_{pruned} \leftarrow C_{comp}$
    $C_{compare} \leftarrow C_{comp}$
  否则，如果 $n = 2$，**则** ▷ C 与其自身比较
    $C_{pruned} \hookleftarrow C[0]$
    $C_{compare} \hookleftarrow C[0]$
    $r_{ex} \leftarrow r_{ex} \times p_s$ ▷预检强度
    $\theta_{ex} \leftarrow \theta_{ex} \times p_s$
  对 X **中的** C **执行循环**
    $d \leftarrow \emptyset$ $\alpha \leftarrow \emptyset$
    $d \hookleftarrow distance\ (X,\ Y)$ **对** Y **中的** $C_{compare}$ **执行循环**
    $\alpha \hookleftarrow angle\ (X,\ Y)$ **对** Y **中的** $C_{compare}$ **执行循环**
    $keep \leftarrow 1$ ▷ 布尔值，指示是否保留一个任务
    对于 i **从** 0 **到** $length(d)$ **执行**
      如果 $d[i] < r_{ex}$ **则**
        $precheck \leftarrow 0$
        如果 $\alpha[i] < \theta_{ex}$ **则**
          $keep \leftarrow 0$
          中断
    如果 $keep = 1$ **则**
      $C_{pruned} \hookleftarrow X$
      如果 $n = 1$ **或** $n = 2$ **则**
        $C_{compare} \hookleftarrow X$
  如果 $n = 0$ **且** $precheck = 1$ **则**
    $C_{pruned} \leftarrow$ 剪枝(C, $\emptyset$, 2)
  返回 C_{pruned}

算法 3附加函数

函数 生成点($\mathcal{P}$)
  对于 x **在** $\mathcal{P}$ **中执行**
    $c \leftarrow$ 质心(x)
    $n \leftarrow$ 法向量(x)
    $p \leftarrow c + L_i \times n$
    $\theta \leftarrow -1 \times n$ ▷ 以四元数表示的角度
    $C \hookleftarrow (p, \theta)$
  返回 C

函数 计算分数($C, \mathcal{K}$)
  $S \leftarrow \emptyset$
  对于 X **属于** C，**执行**
    $FOV_t \leftarrow transform\ (FOV,\ X)$
    $T \leftarrow \sum\ (Delaunay\ (FOV_t)\ .find_simplex\ (\mathcal{K}) \geq 0)$
    $S \hookleftarrow T/length\ (\mathcal{K})$
  返回 S

展示了这项初步算法开发在一个包含 326 个三角形的单个网格帧上的结果。初始参数设为 $L_i = 1.5$ m、$r_{ex} = 0.8$ m、$\theta_{ex} = 0.1$ rad 和 $t = 0.4$。带有关键点的左侧图像和网格分别见图 3A 和图 3B；关键点集中在几何和视觉上较为显著的位置，例如梯子脚、窗户和框架的边缘，以及角落处可见的两条边。使用这些数值，并从候选检查点中随机选取初始的 21 个任务后，CIP 数量从 326 个减少至 98 个。通过适当的参数调优，还可以进一步减少数量。

在第二次测试中，允许机器人缓慢上升约 0.5 m，并分析了前 7 个关键帧，以找出适用于该环境的理想参数。预先测试时的观察结果表明，在盐水水箱内水体透明度较高、室内照明良好的情况下，OAK-D 能够提供最准确的点云；当它位于所观测表面约 1 米处时，效果最佳，因此将其设为 L_i 值。

分别改变参数 r_{ex}、θ_{ex}和 t，目标是尽量减少未检查区域和过度覆盖，并将任务数量控制在 15 至 25 个之间。我们观察到，当 θ_{ex} 的值介于 0.25 至 0.45 rad 之间时，任务数量会落在这一范围内；当 r_{ex} 的值介于 1.5 至 2.0 m 之间时，任务数量也会落在这一范围内，且 r_{ex} 和 θ_{ex}的值越低，新增任务的速率越高。阈值对新增任务的速率影响很小，但会显著影响初始任务数量：阈值越高，任务总数越少；而当阈值变化超过 80% 时，生成的任务数量几乎没有变化。

B. 分析

为分析新任务位置的有效覆盖，我们对测试期间使用的测试水箱区域进行了建模。该区域包括一个带嵌入窗的平坦段、两个倾斜段，以及水箱较长侧边的一小段，如图 4 所示。模型不包括水箱底部和水面。我们使用 Open3D 库将模型体素化。

对于已发现任务集中的每个点，都会对 FOV 锥体进行变换，并将锥体内的所有体素标为红色。如果某个体素从多个位置都可见，则每有一个点能看到它，其红色就会加深。通过分析每个体素被看到的次数分布，我们可以比较各参数在其有效范围内不同取值的效果。

表I展示了在不同的 r_{ex}、θ_{ex} 和 t取值下，每个体素的平均视图数、生成的任务数量，以及零次观测的体素所占比例。绿色表示理想值：未检查体素少于5%；对于这些取值，任务数量介于15到25之间，且每个体素的平均视图数低于2.5，也以绿色标示。黄色表示数值超出该范围，但任务数量低于30

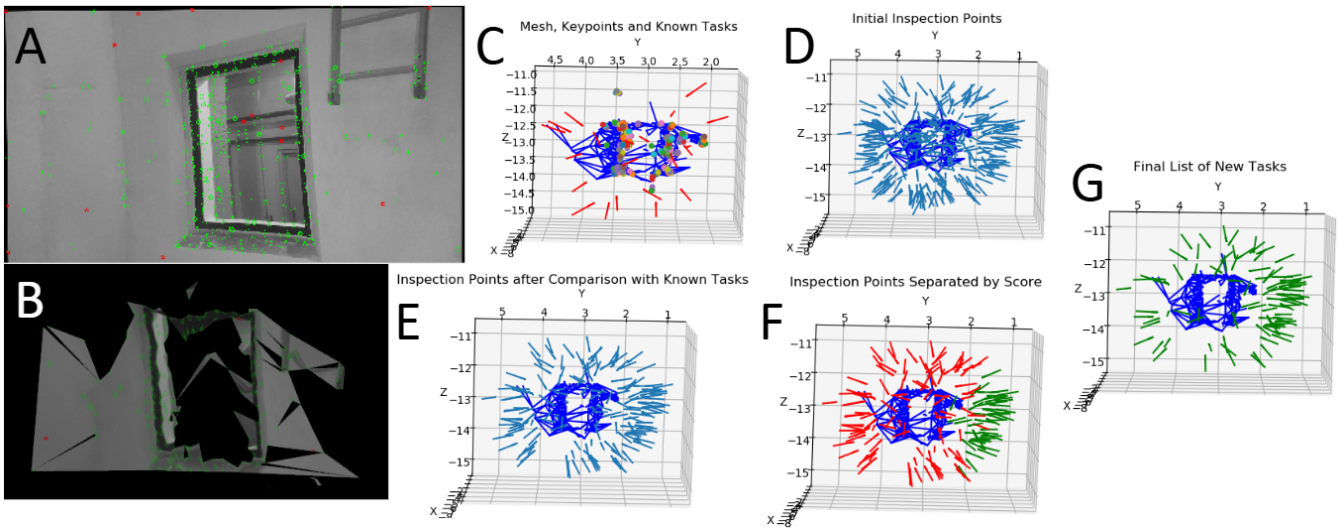


图 3。**A.** 左侧相机图像，其中跟踪到的关键点以绿色叠加显示，未跟踪到的关键点以红色显示。**B.** 以 A. 为纹理的 Kimera 网格。**C.** 初始任务显示为红色箭头，网格显示为蓝色，关键点显示为彩虹色点。**D.** 生成的 326 个 CIP，以青绿色显示。**E.** 第一次剪枝后剩余的 165 个 CIP，以青绿色显示。**F.** 根据关键点得分保留的 49 个任务以绿色显示，其余 116 个 CIP 以红色显示。**G.** 最终剪枝后保留的 98 个任务。

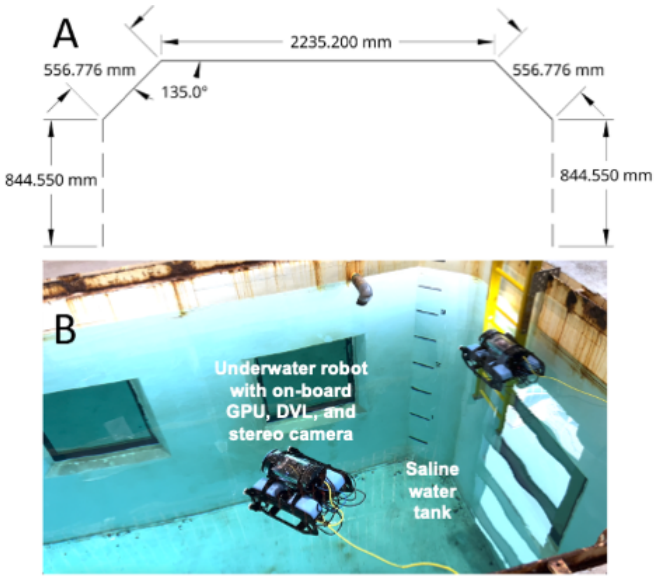


图 4。**A.** 测试水箱的俯视图及其尺寸。实线表示用于为 Voronoi 和扫掠模式方法生成检查点的几何形状，虚线表示用于评估方法有效性的体素化区域边缘。**B.** 测试水箱，其中有两台经过改装的 BlueROV2，主机器人位于中心。

表I 每个体素的平均视角数、任务数量和未检查百分比

t	r_{ex} （米）	θ_{ex} （弧度）					
		0.25			0.3		
		avg	num	%	avg	num	%
70%	1.5	4.26	27	1.49	3.97	25	3.71
	1.6	3.63	24	7.60	3.19	21	8.06
	1.7	2.85	22	4.28	2.30	18	21.16
	1.8	2.71	19	6.18	2.79	19	5.77
75%	1.5	4.05	26	3.05	3.50	23	5.69
	1.6	3.12	22	11.73	2.83	20	9.53
	1.7	2.56	20	3.81	2.38	18	6.79
	1.8	2.48	18	6.84	2.55	18	6.42
80%	1.5	3.26	23	4.06	3.06	21	4.40
	1.6	2.62	22	11.90	2.31	19	8.07
	1.7	4.06	25	3.71	3.15	21	5.13
	1.8	3.78	23	6.35	3.09	20	6.45

Avg 表示每个体素的平均视角数，num 表示使用该组参数生成的任务数量，% 表示未获得任何视角的体素百分比。绿色数值表示最佳性能，黄色数值表示可接受的性能，红色数值表示明显的覆盖过度。

个，或平均值低于3.5；红色表示平均值超过3.5。在 $r_{ex} = 1.7\ m$、 $\theta_{ex} = 0.25\ rad$和 $t = 75\%$的取值下，覆盖率最佳，但其他取值也表现尚可。总体而言，较低的 r_{ex} 和 θ_{ex}取值会带来更多任务和更好的覆盖率，但平均覆盖率也更高。

C. 比较指标

检查质量指标（例如覆盖率百分比和平均值）可以轻松比较，但要找到一个同时展示过度覆盖分布的指标可能是挑战。适度的过度覆盖有时是可取的，尤其是在复杂几何体周围，以确保从各个角度进行观察，从而发现所有缺陷；但过度覆盖过多则表明检查点超出了实际需要，导致时间和能源浪费。那么，怎样才能在确保覆盖率的同时避免过度覆盖？尽管有多种选择，我们选择考察每个体素的视图数与伽马分布的关系。虽然这种函数通常用于描述等待时间变量，但伽马分布具有固有的零下限、左偏特性和指数递减的尾部，因此同样适用于我们的目的。伽马分布的概率密度函数如公式 1 所示。

$$f(x;\alpha,\beta)=\frac{1}{\Gamma(\alpha)\beta^{\alpha}}x^{\alpha-1}e^{-x/\beta} \quad (1)$$

通过选择形状参数 α 和尺度参数 β，我们可以调整分布形状，以满足检查需求：被检查 0 次的体素数量应尽可能少，峰值应位于 1 到 2 之间，而过度覆盖次数较多的体素数量应较少。伽马分布的峰值出现在 $\beta($ $\alpha-1)$，因此，如果设定了期望的峰值，就可以通过任一参数来调整整个分布。将峰值设为 2 并令$\alpha=4$，可得 $\beta=\frac{2}{3}$；通过累积分布函数进行检验可知，这组参数使被检查次数少于 0.5 次的体素占比低于 1%。由于伽马分布是连续的，而视图数是离散整数值，因此将伽马分布数据离散化时，这就是使用整数中心分箱后会被检查 0 次的体素的截止值。

D. 比较

要将其他常见的区域覆盖方法与 MAP-Tidal 的性能进行比较，需要对这些方法有所了解。Voronoi图通过将空间划分为若干区域构成，每个区域包含距离一组点中某一点最近的所有位置。文献 [28] 对用于机器人覆盖的 Voronoi方法进行了综述。用于生成 Voronoi图的点可以根据机器人起始位置、基于各种分布进行随机生成，或通过其他多种方法生成。Voronoi分区可以采用恒定代价函数生成，也可以根据对环境的先验知识，在区域内使用分布式代价函数生成。以起始位置为基础，可以使用 Lloyd算法优化该图：该算法将每个点移向相应 Voronoi划分的质心，并据此重新计算新的分区。在我们的比较中，我们使用 WebSVG 的 Voronoi编辑器 [29]，，它采用一种控制种子分布范围和规则性的种子生成方法，并支持加权分布和避墙。

往复式覆盖模式基于在空间中线性来回扫掠。在 [30],中，扫掠模式的优化发生在一个可由带孔洞（作为禁飞区）的二维空间表示的空间上。Bahnemann等人开发了一种工具，利用广义旅行商问题框架计算优化路径，该框架可在运行时随用户添加额外禁飞区而修改。尽管该实现最初是通过飞行机器人以设定的高度偏移在山坡上搜索地雷来演示的，但它也可应用于其他情形，即三维空间可由带偏移机器人的表面表示的情形，例如我们的检测问题。

对于 Voronoi 和扫掠模式，我们将矩形表面投影到罐体壁面上，距离偏移为 1 m，且角度垂直于罐体表面。所考虑罐体部分的俯视剖面图如图 4 所示。类似于用于参数调整的覆盖率评估，对同一水箱模型进行体素化，并可采用相同的参数来评估这些方法的覆盖率有效性。

图5展示了每种生成方法的最佳版本，图6展示了一张直方图，其中每个体素的视图数与第IV.C部分生成的理想基于伽马分布相叠加。牛耕式模式使用长度超过1 m的区段之间的两个插值点生成，横向重叠为0.1 m。Voronoi划分使用包含避墙的20个点生成。Map-TIDAL使用了 $r_{ex}=1.7$ m、 $\theta_{ex}=0.25$ rad和 $t=75\%$。

从图6中的直方图可以看出，与 Voronoi（17.32% 未检查体素、平均 1.39 个视图、20 个任务）和往复式（8.94% 未检查体素、平均 1.80 个视图、30 个任务）相比，Map-TIDAL 在 20 个任务中实现了 3.81% 的未检查体素，且每个体素平均有 2.56 个视图。Map-TIDAL 与理想值之间的绝对差平均为 0.029，而 Voronoi（0.100）和往复式（0.081）的差异更大。

E. 进一步的实验验证

第三次测试在淡水池中进行，未进行额外的参数调优，因为该水池的能见度与 OSB测试水槽相近。此次测试中，机器人移动得更多，且待检查表面并非与机器人直接垂直，这与机器人沿船体移动时以一定角度观察到的情况相似。

图7展示了一次泳池测试中第一和第二个网格帧的结果。根据测试期间的观察，这组任务似乎在环境中分布良好，但值得注意的是，Kimera 网格的可视化并不完整。网格包含网格中的所有三角形，而可视化仅显示尺寸超过一定阈值的三角形。还可以从之后的网格帧（未展示）中看到，墙壁在水面上的倒影会造成额外的网格不规则性，这表明应开发额外的视觉处理方法，以处理水面附近的区域。

总体而言，第三次测试非常成功，展示了 MAP-TIDAL 算法在几何复杂度各异的不同环境中的功能。

V. 讨论

对图5B中任务发现方法所生成结果里红色最亮区域进行定性分析后，可以清楚地看到视点集中在罐体中央和右上方。这与测试期间安全梯所在的位置相对应，因此，由于该区域的关键点较为集中，预期应从更多角度观察该区域。所给示例中的扫掠和 Voronoi 模式都没有这种集中现象，但通过调整加权参数，两种方法都有望将加权分布纳入任务生成。

Voronoi 和扫掠模式在水下检测场景中的主要局限是，它们需要先获取一个

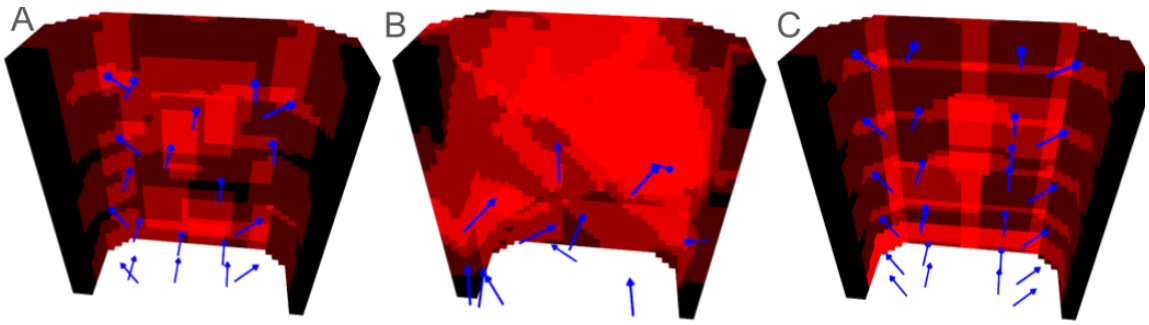


图5。对**A**Voronoi、**B** Map-TIDAL 和**C** 扫掠模式进行的模拟覆盖率分析，其中任务位置以蓝色显示，体素每被查看一次，其颜色就会变得更红。

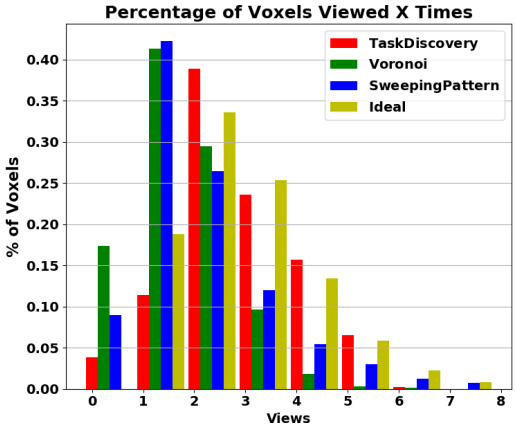


图6。显示不同任务生成方法的每个体素的归一化视图数的直方图，并与离散化理想伽马分布进行比较。

模型，然后才能生成相应的点。扫掠模式工具能够在用户调整空间模型时快速即时地重新计算，但仍然需要用户输入，因此系统无法实现完全自主。Voronoi 模式同样计算迅速，但在开始前需要了解空间的几何边界。改变任一组模式所适用的环境，都会用新的一组任务覆盖现有任务；由于水下通信速度慢且不可靠，这将带来很大挑战。这限制了这两种方法在先验知识极少的动态环境中的有效性，尤其是在对几何形状了解有限的情况下。

所提出的Map-TIDAL方法也并非没有局限性。由于任务发现基于SLAM过程中生成的网格，因此无需预先了解待检查对象的几何形状，但生成任务的质量很大程度上取决于SLAM网格的质量。在浑浊水域中，当没有可见物体时，需要使用多普勒测速仪等额外传感器来维持定位，因为定位中的任何误差都会导致任务位置出现相应误差。此外，在任务发现期间，单个机器人只能利用其在SLAM中观测到的几何形状，因此生成的任务可能会被安排在机器人无法到达的区域。

VI. 结论

未来的工作将研究基于SLAM的任务发现与分配过程中的避碰。未来工作的另一个潜在方向是多模态传感器融合，可通过进一步融合侧扫声呐数据来增强定位和地图生成能力。侧扫声呐数据还能提高Map-TIDAL任务发现的准确性，并用于网格精度验证。虽然Map-TIDAL方法目前侧重于视觉检测数据，但可通过开发适用于其他检测传感器（如超声波测厚传感器）的改进方案，或支持使用刷子、刮刀或探针等其他工具，来扩展相关算法。Map-TIDAL专为水下应用设计，但若将其调整为由飞行无人机组成的系统，也有望用于通信和视觉范围受限的复杂空中环境。

Map-TIDAL 表明，通过生成候选检测点，并根据其关键点得分以及与其他任务的邻近程度对其进行剪枝，可以利用 SLAM 生成的环境网格实现任务发现。随后，可以通过分布式拍卖和冲突解决过程，将这些任务高效地分配给多机器人系统。随着新几何形状的发现，系统会生成新任务并将其分配给系统，从而无需建立待检查对象的模型，并将检查重点放在具有大量视觉显著特征的区域。通过适当的参数调整和准确的 SLAM，Map-TIDAL 能够生成任务位置，在保持适当较低过度检测水平的同时，提供出色的覆盖率。

致谢

我们谨向 Wade Kempf、Tyler Paine、Matthew Craw 和 Martin Renken 致以谢意，感谢他们对我们的指导以及对本研究的支持。我们承认在对本文进行轻度编辑时使用了 NIPRGPT。

参考文献

[1] B. Lin 和 X. Dong，《船体检查：综述》，《*Ocean Engineering*》，第289卷，116281页，2023年。[2] R. Eckholdt Andersen、R. Yding Brogaard 和 E. Boukas，《远程检查技术：面向船舶自主机器人检查的综述》，《*IEEE* 野外机器人汇刊》，第2卷，第1–20页，2025年。

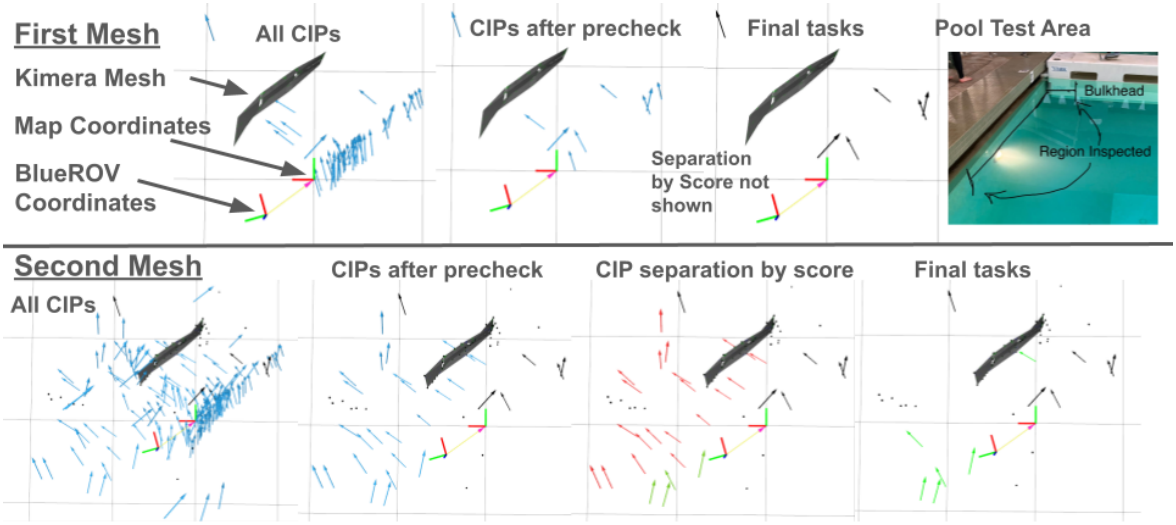


图7. 池测试中前两个网格帧上展示的任务发现步骤。CIP最初着色为蓝色。对于第一个网格，最终任务以黑色显示，并在第二个网格中作为现有任务使用时保持不变。对于第二个网格，在分数分离步骤中，被选中的为绿色，需进行最终剪枝的为红色，最终选择以浅绿色显示。这些浅绿色任务随后与现有任务列表合并，用于下一个网格帧。

[3] M. Ruediger、T. M. Paine 和 A. G. Banerjee，《用于研究船舶船体检测的多机器人系统的水下环境仿真》，载于*IEEE OCEANS*，汉普顿锚地，第1–7页，2022年。[4] X. Wang、X. Fan、P. Shi、J. Ni 和 Z. Zhou，《水下场景关键SLAM技术概述》，《*Remote Sensing*》，第15卷，第10期，2496页，2023年。[5] M. Heshmat、L. Saad Saoud、M. Abujabal、A. Sultan、M. Elmezain、L. Seneviratne 和 I. Hussain，《水下SLAM与深度学习的结合：挑战、多传感器集成及未来方向》，《*Sensors*》，第25卷，第11期，3258页，2025年。[6] F. Hidalgo、C. Kahlefendt 和 T. Br¨aunl，《单目ORB-SLAM在水下场景中的应用》，载于*OCEANS - MTS/IEEE*，第1–4页，2018年。[7] L. M. Hodne、E. Leikvoll、M. Yip、A. L. Teigen、A. Stahl 和 R. Mester，《检测并抑制海雪以实现水下视觉SLAM》，载于*IEEE/CVF*计算机视觉与模式识别会议研讨会，第5097–5105页，2022年。[8] T. Łuczy´nski、M. Pfingsthorn 和 A. Birk，《Pinax模型：平面视窗外壳中水下相机折射校正的准确高效方法》，《*Ocean Engineering*》，第133卷，第9–22页，2017年。[9] E. Westman、A. Hinduja 和 M. Kaess，《面向约束不足地标的成像声呐基于特征的SLAM》，载于*IEEE*国际机器人与自动化会议，第3629–3636页，2018年。[10] J. Wang、F. Chen、Y. Huang、J. McConnell、T. Shan 和 B. Englot，《复杂水下环境自主探索的虚拟地图》，《*IEEE*海洋工程学报》，第47卷，935，2022 4，916 no. pp. – 。[11] N. Loi、Y. Z. Tan、E. W. Goh 和 M. H. Ang，《结构化水下环境中的声呐SLAM》，载于*OCEANS* - 新加坡，第1–10页，2024年。[12] D. C. Estrada、F. R. Dalgleish、C. J. D. Ouden、B. Ramos、Y. Li 和 B. Ouyang，《利用基于生成对抗网络的机器学习技术增强水下激光雷达图像》，《*IEEE*传感器期刊》，第22卷，第5期，第4438–4451页，2022年。[13] J. Zhang、F. Han、D. Han、J. Yang、W. Zhao 和 H. Li，《声呐与视觉惯性系统集成，实现水下环境中的SLAM》，《*IEEE*传感器期刊》，第24卷，第10期，第16792–16804页，2024年。[14] S. Pan、Z. Hong、Z. Hu、X. Xu、W. Lu 和 L. Hu，《RUSSO：针对视觉退化、通过声呐优化实现的鲁棒水下SLAM》，《*IEEE/ASME*机电一体化汇刊》，第1–12页，2025年。[15] S. Ding、T. Zhang、Y. Li、S. Xu 和 M. Lei，《采用混合残差和高效闭环检测的视觉-惯性-深度水下多传感器融合定位》，《*Measurement*》，第238卷，115245页，2024年。[16] S. Xu、K. Zhang 和 S. Wang，《AQUA-SLAM：结合传感器标定的紧耦合水下声学-视觉-惯性SLAM》，《*IEEE*机器人学汇刊》，第41卷，第2785–2803页，2025年。

[17] S. Rahman、A. Quattrini Li和I. Rekleitis，“SVIn2：基于多传感器融合的水下同步定位与建图系统”，《国际机器人研究杂志》，第41卷，第11-12期，第1022–1042页，2022年。[18] S. Xu、J. Scharff Willners、J. Roe、S. Katagiri、T. Luczynski、Y. Petillot和S. Wang，“Tank 数据集：用于同步定位与建图评估的水下多传感器数据集”，《国际机器人研究杂志》，第02783649251364904页，2025年。[19] Y. Liu、W. Zhao、H. Liu、Y. Wang和X. Yue，“面向机器人质量检测的覆盖路径规划及测量不确定性控制”，《*IEEE/ASME*机电一体化汇刊》，第27卷，第5期，第3482–3493页，2022年。[20] A. Mavrinac、X. Chen和J. L. Alarcon-Herrera，“面向主动三角测量三维检测系统的半自动基于模型的视点规划”，《*IEEE/ASME*机电一体化汇刊》，第20卷，811，2015年 2，799no. 第–页。[21] I. Rekleitis、A. P. New、E. S. Rankin和H. Choset，“高效往复式多机器人覆盖：一种算法方法”，《数学与人工智能年刊》，第52卷，第109–142页，2008年。[22] A. Yazici、G. Kirlik、O. Parlaktuna和A. Sipahioglu，“一种考虑能量约束的多机器人基于传感器覆盖的动态路径规划方法”，《*IEEE*控制论汇刊》，第44卷，第3期，第305–314页，2014年。[23] L. Brunet、H.-L. Choi和J. How，“用于鲁棒任务分配的基于共识的分布式拍卖”，《*IEEE*机器人学汇刊》，第25卷，第4期，第912–926页，2009年。[24] A. Rosinol、M. Abate、Y. Chang和L. Carlone，“Kimera：用于实时度量-语义定位与建图的开源库”，载于*IEEE*国际机器人与自动化会议，第1689–1696页，2020年。[25] Y. Tian、Y. Chang、F. Herrera Arias、C. Nieto-Granda、J. P. How和L. Carlone，“Kimera-Multi：面向多机器人系统的鲁棒分布式稠密度量-语义同步定位与建图”，《*IEEE*机器人学汇刊》，第38卷，第4期，第2022–2038页，2022年。[26] A. Rosinol、T. Sattler、M. Pollefeys和L. Carlone，“利用结构规律的增量式视觉惯性三维网格生成”，载于*IEEE*国际机器人与自动化会议，第8220–8226页，2019年。[27] “BlueROV2。” https://bluerobotics.com/store/rov/ bluerov2/，2022年2月。[28] M. Zhou、J. Li、C. Wang、J. Wang和L. Wang，“Voronoi图在多机器人覆盖中的应用：综述”，《海洋科学与工程杂志》，第12卷，第6期：1022，2024年。[29] WebSVG，“Voronoi编辑器。” https://github.com/WebSVG/voronoi。[30] R. B¨ahnemann、N. Lawrance、J. J. Chung、M. Pantic、R. Siegwart和J. Nieto，“将往复式覆盖路径规划重新视为广义旅行商问题”，《现场服务机器人》，第277–290页，2021年。