

Autonomous Exploration and Mapping System Using Heterogeneous UAVs and UGVs in GPS-Denied Environments

Hailong Qin, Zehui Meng , Wei Meng , Xudong Chen, Hao Sun, Feng Lin, and Marcelo H. Ang

Abstract—In this paper, we present a novel integrated vehicular system using collaborative unmanned aerial vehicles (UAVs) and unmanned ground vehicles (UGVs) for autonomous exploration, mapping, and navigation in GPS-denied 3-dimensional (3-D) unknown environments. The system implements a novel two-layered exploration strategy to decompose the perception task into a coarse exploration layer and a fine mapping layer. The coarse exploration makes use of a UGV to carry out fast autonomous exploration and active 2.5D simultaneous localization and mapping (SLAM) to generate a coarse environment model, which serves as a navigation reference for subsequent complementary 3-D fine mapping conducted by a UAV. The two layers share a novel optimized exploration path planning and navigation framework, which provides optimal exploration paths and integrates the collaborative exploration and mapping efforts through an OctoMap-based volumetric motion planning interface. The proposed system provides an efficient pipeline of fast environment perception taking advantages of the agility of the UAVs as well as the powerful computation resource aboard UGVs, also allowing assistive local perception with augmented object information when necessary. The effectiveness of our system is verified by both simulations and experiments, which demonstrate its capability of implementing heterogeneous UAV and UGV collaborative exploration and structural reconstruction of the environments through active SLAM, providing optimized perception for navigation tasks.

Index Terms—Autonomous exploration, active SLAM, view planning, navigation, heterogeneous vehicular systems, collaborative system.

Manuscript received March 24, 2018; revised October 10, 2018; accepted December 17, 2018. Date of publication January 1, 2019; date of current version February 12, 2019. This work was supported by the National Natural Science Foundation of China under Grants NSFC 61803105 and U1701264. The review of this paper was coordinated by Dr. A. Chatterjee. (*Corresponding author: Wei Meng*)

H. Qin was with the Temasek Laboratories, National University of Singapore, Singapore 119077. He is now with Roadstar.ai, Shenzhen 518000, China (e-mail: qinhalong722@gmail.com).

Z. Meng is with the Future Urban Mobility Group, Singapore-MIT Alliance, Singapore 11758 (e-mail: zehui@smart.mit.edu).

W. Meng is with the School of Automation, Guangdong University of Technology, Guangzhou 510006, China, and also with the Guangdong Province Key Laboratory of Intelligent Decision and Cooperative Control, Guangzhou 510006, China (e-mail: meng0025@ntu.edu.sg).

X. Chen was with the Temasek Laboratories, National University of Singapore, Singapore 119077. He is now with the Carnegie Mellon University, Pittsburgh, PA 15213 USA (e-mail: xudongc@andrew.cmu.edu).

F. Lin is with the Temasek Laboratories, National University of Singapore, Singapore 119077 (e-mail: linfeng@nus.edu.sg).

H. Sun and M. H. Ang are with the Department of Mechanical Engineering, National University of Singapore, Singapore 119077 (e-mail: a0132560@u.nus.edu; mpeangh@nus.edu.sg).

Digital Object Identifier 10.1109/TVT.2018.2890416

I. INTRODUCTION

AUTONOMOUS robots such as unmanned aerial vehicles (UAVs) and unmanned ground vehicles (UGVs) are becoming more and more popular in various fields of applications. One of the most demanding capability of the autonomous vehicles is being able to navigate and execute tasks autonomously in complex environments, especially in GPS-denied environments.

Many of the current vehicle systems are able to perform self navigation and manipulation in a given environment. As a prerequisite, the autonomous vehicles must be provided with a well constructed environment model. However, some challenging applications require the vehicles to operate in situations where no environment model is available beforehand. Such challenges call for the capabilities of autonomous perception of the environments to support desired navigation and manipulation tasks. In this work, we focus on the problem of autonomous exploration and reconstruction (mapping) for operation in GPS-denied and fully unknown 3D environments.

In the literature, some of the current systems require partial environment information to be known *a priori*, which is utilized to plan the exploration path [1]–[3]. However, the frameworks become unfeasible in fully unknown environments. Other works propose active SLAM strategies requiring no *a priori* environment information [4]–[6]. However, the associated applications are very limited either because their greedy exploration approaches frequently leads to non-optimal solutions or due to lack of generalization to complex 3D scenarios. An exploration and mapping strategy combining the advantages of existing works is thereby in demand.

Many applications employ multiple vehicles to conduct cooperative tasks [7]–[10]. The collaboration within multi-agent systems (MAS), either homogeneous or heterogeneous, can significantly enhance the overall mobility, flexibility and task-specific performance as the information is exchanged within the system to decompose the task with reasonable distribution of efforts from individual vehicles. Especially, the demand for heterogeneous vehicles collaboration in a variety of harsh environments has dramatically increased since last decade [11]–[14], catching up the growth of homogeneous MAS. To the best of our knowledge, there is no work has been published on autonomous exploration and mapping using heterogeneous UAVs and UGVs in GPS-denied cluttered environments, especially for 3D sensing applications.

In this paper, we present an autonomous exploration and mapping system using a heterogeneous UAV and UGV team. The system is built with depth-based SLAM modules, an integrative exploration path planning and navigation framework, and a collaborative perception scheme, for complex GPS-denied unknown environment exploration. For the depth-based localization and mapping modules, we implement a three-step paradigm adaptable to most point-cloud compatible sensors, including our rotating laser-based module which provides a robust localization and 360-degree full dense maps. The exploration path planning and navigation framework proposes a two-stage optimized view planner to generate optimal exploration paths and a volumetric navigation interface for instantaneous motion planning. Using the proposed collaborative perception scheme, our suggested framework demonstrates the capability using the heterogeneous UAV-UGV system to autonomously explore and reconstruct the unknown environments without pre-planned trajectories.

The remaining contents of the paper are organized as follows. Section II gives a brief introduction to related works. Section III presents the hardware description. In Section IV, the localization and mapping strategy proposed in our work is addressed. Section V elaborates the exploration planning and navigation approach. Section VI briefly describes the collaborative perception scheme utilized in the cooperative exploration framework. Section VII presents the simulation and experiment results. Section VIII concludes the work.

II. RELATED WORKS

During the past decades, MAS, consisting of heterogeneous vehicles, has been extensively studied, but few were tested in realistic environments [7], [8], [13], [15]. As proposed by Paull *et al.* [13], an independent agent with ontology in a MAS should at least include perception and planning modules to understand the surroundings and make decision correspondingly.

A. Perception

In GPS-denied environments, the problem of autonomous navigation is generally referred as SLAM that helps the vehicle determine its position and build maps in *a priori* unknown environment simultaneously. Based on the perception sensors, the SLAM can be mainly divided into two major methodologies, namely vision based SLAM and light detection and ranging (LIDAR) based SLAM. Recent achievements of the SLAM enable vehicles to navigate autonomously in GPS-denied environments. However, the SLAM still has a lot of unsolved problems in practical applications. For instance, the computational cost is quite expensive for vision-based SLAM since a large amount of information need to be extracted and processed from images [16], [17]. Recently, Forster *et al.* [18], [19] proposed a semi-direct visual odometry (SVO) pipeline which is based on the existing tracking and mapping multi-threads framework [20] using a monocular camera. The algorithm flow of the vision-based SLAM generally includes initialization, feature tracking and alignment as well as optimization using bundle adjustment or other optimization methods, e.g., to optimize the nonlinear error functions in the several variants of SLAM and bundle

adjustment problems, graph-based framework has been widely utilized [21]. In the initialization phase, corners are detected and further tracked using optical flow [22] technique. Meanwhile, a feature based map is built to maintain the estimated scale. In the alignment phase, coming new observed features are aligned with previous features using the consistency of photometric. In addition, keyframes, which have enough aligned features and new observed features, are continuously inserted into the map for optimization. Through these delicate processes, the ego-motion can be estimated and a map can be constructed. Similarly, the vision-based SLAM using stereo cameras follows the similar pipeline [23]–[25], except that the initial map is constructed by left and right images with spatial alignment.

Different from cameras, depth sensors such as LIDAR can provide non-scale depth information, however, the energy consumption and sensor weight are both high. Therefore, the most of the existing works use a fixed light-weight 2D LIDAR to obtain a planar motion estimation. Other height estimation sensors such as barometer [26] are usually fused with the planar motion estimation module to achieve autonomous navigation for UAVs. To achieve 6-DOF LIDAR-based pose estimation alone, it is usually required to use a 3D LIDAR or a rotating 2D LIDAR. The iterative closest point (ICP) has been extensively studied for autonomous vehicles related localization and mapping issues [27]. One of the well-known issues of the depth-based SLAM is the data distortion which is caused by unknown ego-motion of vehicles. This phenomenon widely exists in both the 2D and the 3D LIDAR cases [28]. Few works have succeed in solving the data distortion using the LIDAR sensor alone, while additional sensors, e.g. cameras [29], are used to compensate the unknown ego-motion between the successive laser measurements. In this work, we will try to tackle a similar issue on point-cloud based modules with associated feature based motion estimation.

B. Planning

The UAV-UGV cooperative framework proposed in this work is essentially an exploratory MAS with two main categories of issues to be solved:

1) *View Planning*: Recently, the automatic exploration task based on optimized view planning becomes a key issue for vehicles in large-scale GPS-denied environments. Different from traditional waypoints based navigation, vehicles need to determine the path according to the information from each view and to optimize the overall paths. Due to the limited sensing range of onboard sensors, the optimized view sequence is also subject to the accuracy of localization and the quality of reconstruction. Blaer and Allen [30] proposed a two distinct stage data acquisition approach. At first, a 2D map is given to obtain an optimized path in terms of view coverage. After that, a 3D laser is utilized to construct a voxel-based occupancy map and compute the next view incrementally. To deal with the increasing localization errors, Vasquez-Gomez *et al.* [31] proposed a volumetric next-best-view planning algorithm for the purpose of object reconstruction. The view sequence is determined from a set of view candidates to maximize the potential information and minimize the localization uncertainty. More recently, a

genetic algorithm based optimized view planning was proposed by Martin *et al.* [32]. The time-consuming space exploration to guarantee the view coverage becomes more effective because of the optimization process using the genetic algorithms. The effectiveness and accuracy were well demonstrated in the simulated environments. Similarly, the active SLAM in [4] also provided dynamic view planning to improve the quality of reconstruction. However, the most of the existing view planning methods only work to perform a 2D perception and localization which can be of a great disadvantage for the acquisition of view information. View planning combined with a 3D perception in large and complex environments has not yet to be demonstrated.

2) *Cooperative Exploration*: Every vehicle in the heterogeneous team should have certain capability of achieving state estimation, planning and decision making. Especially in the scenario such as exploring an unknown environment, the self-exploration capability of each vehicle is critical to the whole mission. In [33], a cooperative routing for UAVs and UGVs was proposed. The proposed exploration policy travels between predefined targets to optimize the Euclidean traveling distance. However, *a prior* knowledge of the environment is mandatory to this framework which usually does not exist in practice. More recently, Li *et al.* [34] deployed multiple UAVs to build map and identify obstacles using vision sensors. The identified map was further utilized for UGV's path planning. However, the proposed hybrid path planning only provided a limited 2D planar trajectory which does not fully exploit the potentialities of the UAVs. Sometimes UAVs and UGVs teams are preferred to reach for targets in different scenarios. For instance, in [35], a switched control architecture between UAVs and UGVs was proposed to track a moving target. In this work, the UAVs and the UGVs were deployed in groups to efficiently detect and track moving targets in certain regions. The vehicles in the framework were modeled as point masses without differentiating their respective motion constraints and also localization approach was not provided. Similarly, in [36] and [37], probabilistic-based path planning algorithms were proposed for UAVs and UGVs to cooperatively track moving targets. These works well utilize a hierarchical hybrid system to coordinate individual vehicles in searching task, but unfortunately the navigation was either limited in 2D space or did not generalize well among the heterogeneous vehicles.

The aim of our work is to promote the research on 3D exploration in GPS-denied environments using heterogeneous vehicle teams. In this paper, an integrative exploration path planning and navigation framework with optimized perception schemes is proposed for UAV-UGV cooperative exploration to promote the coverage and efficiency of the exploration and mapping for 3D unknown environments.

III. HARDWARE DESCRIPTION

Our proposed MAS consists of a flexible micro UAV and a high-payload autonomous UGV. The platforms are customized to be holonomic w.r.t 3D and planar workspaces (omnidirectional motion with regard to their respective SE3 and SE2 configuration spaces). The UAV is a quadrotor platform which is

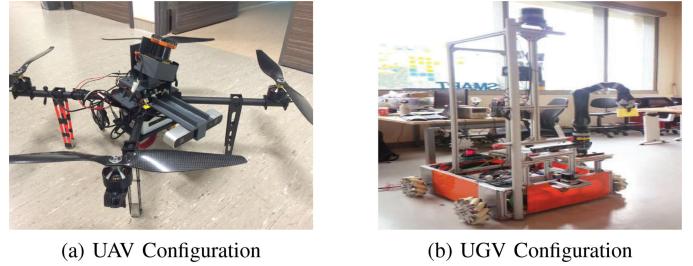


Fig. 1. UAV and UGV platforms with onboard sensors.

made of carbon fiber. The onboard sensors, including inertial measurement unit and magnetic compass, are used to provide basic maneuverability for safe pilot. A high level processor using Intel-NUC is installed for onboard algorithm processing. The UGV is made of Mecanum wheels while driven by 4 independent servo motors. The micro controllers communicate directly with the onboard processor through Ethernet connection. The communication between UAV and UGV is based on a wireless network connection through Secure Shell (SSH) [38]. The customized rotary LIDAR device is rigidly mounted on top of the UAV platform and a stereo camera (ZED) is mounted below the LIDAR to provide fine point cloud. A 3D LIDAR is installed on top of the UGV frame with an RGB-D sensor (KINECT) mounted below. The configuration of our system is shown in Fig. 1.

IV. FEATURE BASED LOCALIZATION AND MAPPING

In this section, we present the depth based SLAM module employed in this work. Basically, we utilize point-cloud based localization and mapping pipelines so as to improve the robustness of our system across textured and non-textured environments.

For all point-cloud compatible sensors, including our rotating LIDAR module, we implement a dedicated feature-based localization and mapping paradigm which follows three steps: 1) feature extraction 2) feature association 3) motion estimation and mapping. The three steps can be customized towards consecutive frames with respect to sensor models. The details are as follows.

A. Feature Extraction

As Abdul mentioned in [39], point cloud feature extraction in real-world environments is practically difficult because the environments contain large amounts of data, noise and outliers. In order to extract reliable features, outlier rejection and robust fitting generally are required. In a 2D laser scanning, the corresponding feature points are usually chosen according to the change of the scanning points in the local neighborhood. In our work, we propose to apply a Euclidean distance clustering step, followed by feature points selection to represent adjacent attributes, edges and flat points. Moreover, instead of focusing on traditional specific forms of the feature points, we insist on employing a general point-based mapping to compensate for motion estimation errors, for which we hereby adopt a

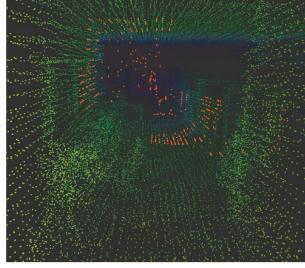


Fig. 2. Front view of feature points (in red) in one rotation.

high-standard feature points selection process with novel customization in our applications.

In our work, for a feature point p in a single scan l of the k -th rotation, a relative variation ratio $R_{k,l,p}$ is defined to distinguish an edge point $e_{k,l,i}$ (larger than a certain variation ratio) from a flat point $f_{k,l,i}$ (smaller than a certain variation ratio). Fig. 2 shows extracted feature after one rotation in a typical corridor area.

$$R_{k,l,p} = \frac{l_{avg,p}}{\|X_{k,l,p}^L\|}, \quad (1)$$

where $X_{k,l,p}^L$ denotes the coordinates of the feature point p , $l_{avg,p}$ is the average length of the edges incident to a point p in a single scan l of the k -th rotation within a local neighborhood, i.e.,

$$l_{avg,p} = \frac{1}{N} \sum_{q \in N(i)} \|X_{k,l,p}^L - X_{k,l,q}^L\|, \quad (2)$$

where N stands for total number of points adjacent to p , and q stands for a neighboring point of p .

B. Feature Association

Our proposed feature association procedure follows the similar concept of vision-based SLAM which performs to minimize the feature projection errors. However, there is no explicit geometric relationship among laser points. To tackle this issue, in our work, the geometric relationships corresponding to both the point-to-line and the point-to-plane strategies are constructed respectively to associate points in different rotations.

- The point-to-line correspondence is formed by using an edge point p in the k -th rotation and its nearest neighbor point m in the $(k+1)$ -th rotation. In addition, a neighboring point of m in consecutive scan is searched and denoted as n . The point-to-line distance can be obtained using following equation:

$$d_e = \frac{|v_l \times v_e|}{|v_l|}, \quad (3)$$

where v_l stands for $(X_{k+1,l,m}^L - X_{k+1,l,n}^L)$ and v_e stands for $(X_{k,l,p}^L - X_{k+1,l,m}^L)$ as shown in Fig. 2(a).

- The point-to-plane correspondence is formed by a flat point q in the k -th rotation to its neighboring point r in the $(k+1)$ -th rotation along the normal direction. Therefore,

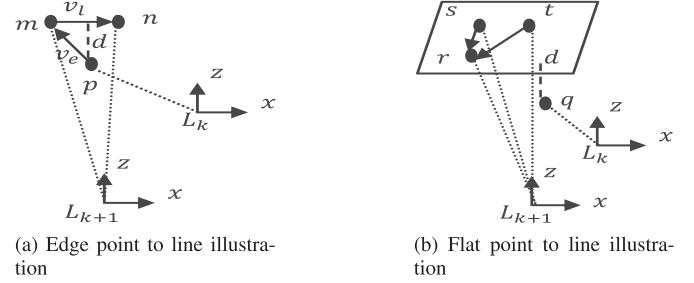


Fig. 3. Edge/flat point to line illustration.

the point-to-plane distance can be calculated as

$$d_p = n \cdot v_p, \quad (4)$$

where v_p stands for $(X_{k,l,q}^L - X_{k,l,r}^L)$ as shown in Fig. 2(b) and n is the unit normal direction determined by non-collinear neighbouring points of r, s and t :

$$n = \frac{v_{p1} \times v_{p2}}{|v_{p1} \times v_{p2}|}, \quad (5)$$

where v_{p1} is $(X_{k+1,l,s}^L - X_{k+1,l,r}^L)$ and v_{p2} is $(X_{k+1,l,t}^L - X_{k+1,l,r}^L)$.

C. Motion Estimation and Mapping

Translation and rotation of a 3D point p in $(k+1)$ th rotation $X_{k+1,l,p}^L$ with respect to the last laser frame k is expressed by

$$X_{k,l,p}^L = \hat{X}_{k+1,l,p}^L = R_{k+1}^k X_{k+1,l,p}^L + P_{k+1}^k, \quad (6)$$

where

$$R_{k+1}^k = \begin{bmatrix} 1 & 0 & 0 \\ 0 & c\alpha & -s\alpha \\ 0 & s\alpha & c\alpha \end{bmatrix} \begin{bmatrix} c\beta & 0 & s\beta \\ 0 & 1 & 0 \\ -s\beta & 0 & c\beta \end{bmatrix} \begin{bmatrix} c\gamma & -s\gamma & 0 \\ s\gamma & c\gamma & 0 \\ 0 & 0 & 1 \end{bmatrix},$$

$$P_{k+1}^k = \begin{bmatrix} p_x \\ p_y \\ p_z \end{bmatrix},$$

α, β and γ represent the UAV body Euler angles correspondingly. Based on the transformation relationship $T_{k+1}^k = (R_{k+1}^k, P_{k+1}^k)$ stated above, combining the feature point correspondence, a nonlinear function $w(T_{k+1}^k) = d$ can be formed to express (3) and (4). The square error function can be established as

$$S = (w(T_{k+1}^k))^2 \quad (7)$$

To minimize the objective function, i.e., $S = 0$, Trust-Region-Reflective method (TRR) [40] is adopted. The corresponding update step $\sigma_i = (T_{k+1}^k)_{i+1} - (T_{k+1}^k)_i$ can be obtained by solving

$$\min_{\sigma \in N} \psi_i(\sigma_i), \quad (8)$$

where $\psi_i(\sigma) = g^T \sigma + \frac{1}{2} \sigma^T H \sigma$ and g and H are the gradient and Hessian, respectively of w evaluated at $(T_{k+1}^{k+1})_i$. $\Delta_k > 0$ is

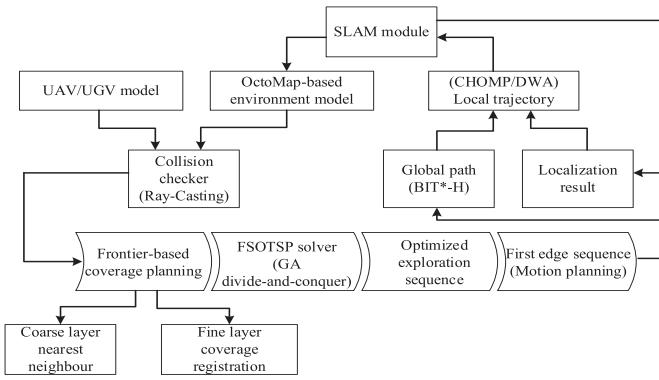


Fig. 4. The overall system architecture of the proposed framework.

Algorithm 1: Optimized Next View Planning.

- 1 Take the first scan and initialize the volumetric map
- 2 Perform coverage planning to obtain minimum viewpoints
- 3 Calculate information gain and associated cost matrix
- 4 Prune the viewpoints if necessary
- 5 Solve FSOTSP for optimal exploration sequence
- 6 Extract the first edge of the exploration sequence and engage the volumetric navigation interface
- 7 Compute valid motion plan and execute
- 8 **while** $InExecution \cap !EndExploration$ **do**
- 9 Take more scans and update the environment model
- 10 Repeat step 2-7

the trust region radius. The acceptable ratio is determined by

$$r_k = \frac{(\psi(0) - \psi_i(\sigma_i))}{(w(T_{k+1}^k) - w(T_{k+1}^k + \sigma_i))}, \quad (9)$$

which is used to decide the iteration of trial step σ_i .

V. EXPLORATION PATH PLANNING AND NAVIGATION

In this section, we present a novel framework which contains a view planning algorithm as well as a volumetric map based motion planning interface, for generating the corresponding exploration paths.

The proposed two-stage optimized next-view planning module is a customization from our recent work [41]. In this work, we further explore the SLAM results and extend the active mapping function. In addition, online view planning and motion planning are performed to further explore the unknown environments, by considering the sensor constraints. The overview of the system architecture is shown in Fig. 4. The proposed optimized next view planning algorithm is summarized in Algorithm 1.

An initial scan is performed when the system is instantiated in an unknown environment while the map is updated accordingly. Each occupancy grid would be registered as free, occupied or unknown, based on the information acquired. Based on the instantaneous map, a set of viewpoints would be generated by the planner using our proposed frontier-based coverage sampler. The objective of this proposed sampler is to have a com-

prehensive coverage of the boundary of the explored and unknown area. In the meanwhile, visualized area of the samples is marked so that there is no extensive overlay among the existing viewpoints and predicted observation of succeeding samples. With the aid of dynamically updated environmental model, in every iteration, viewpoint candidates are trimmed. To achieve globally optimized path to navigate all the viewpoints with the shortest distance and minimum back-and-forth motion, a fixed-start open travelling salesman problem (FSOTSP) solver is employed which heuristically calculates an optimal open sequence of the exploration path. The first edge of the sequence of exploration is extracted and sent for path planning through volumetric navigation interface. Collision checking is then conducted using ray-casting with a specified bounding box, regarding the 3D map built from the SLAM. External sampling based motion planning library in conjunction with local trajectory optimization is utilized to authenticate valid local motion plans. To gather new information and update the environment model, such steps are iteratively executed.

The following subsections explicitly illustrate the details of our optimized next view planning scheme.

A. Heuristic Volumetric Information Gain Model

The proposed next view planning algorithm is based on a volumetric information gain model, which is closely related to the concepts of entropy [5]. In this work, we simplify the implementation with a measure to reflect the possible augmentations of knowledge. Based on sensor characteristics, the information gain is estimated from a visibility propagation with ray-casting.

A simplified indirect indicator of the information gain is proposed to avoid computing practically intractable closed-form solutions. A heuristic augmentation of knowledge, i.e., \hat{g} based on the occupancy probability as well as visibility-based propagation model is built to reflect deviation of uncertainty of the occupancy probability through traveling indeterminate areas.

$$\hat{g}(p(o_i|x_{view}, z)) = p(o_i|x_{view}, z)p(v_i|o_{0:i-1}, x_{view}, z), \quad (10)$$

where $p(o_i|x_{view}, z)$ and $p(v_i|o_{0:i-1}, x_{view}, z)$ are the current occupancy probabilities as registered in the instantaneous map and the visibility probability of o_i measured from the viewpoint x_{view} , respectively.

The visibility transition could be propagated as a function of the complement of occupancy probability distribution, i.e.,

$$p(v_i|o_{0:i-1}, x_{view}) = \prod_{j=0}^{i-1} (1 - p(o_j|x_{view})), \quad (11)$$

which is in line with the fact that the lower occupancy probability indicates the smaller chances of occlusion and also the higher visibility. In this case, the free and occupied cells have small values while the unknown cells have greater ones, which is in a parabolic shape between [0, 1].

The total information gain at x_{view} is then calculated by

$$E[\hat{G}_{x_{view}}] = \int_z p(z|x_{view}) \sum_{o_i \in C(z)} \hat{g}(p(o_i|x_{view})) dz, \quad (12)$$

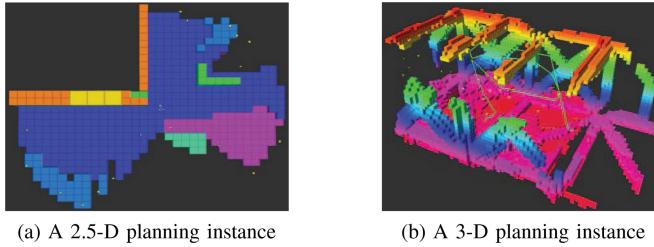


Fig. 5. Frontier-based coverage planning instances. Viewpoints highlighted as yellow arrow markers. OctoMap colors correspond to height variation.

Algorithm 2: Coverage Planner ($X_{viewpoints}$).

```

1 while  $\text{!ViewpointsSetFull} \cap \text{!CompleteCoverage}$  do
2    $x_1 \leftarrow \text{UniformSample}(\mathcal{U}(X))$ 
3    $x_2 \leftarrow \text{NeighbourhoodSample}(x_1, r)$ 
4   if  $\text{IsFrontier}(x_1, x_2) \cap \text{IsVisible}(x_1, x_2)$  then
5      $x_{\text{found}} \leftarrow \text{TheValidOf}(x_1, x_2)$ 
6     if  $\text{CoarseExplorationLayer}$  then
7       if
8          $\text{!NearestNeighbour}(X_{viewpoints}, x_{\text{found}}, R)$ 
9         then
10         $X_{viewpoints} \leftarrow \{x_{\text{found}}\}$ 
11         $\text{GainCalculation}(x_{\text{found}})$ 
12      else
13        if  $\text{!IsCovered}(x_{\text{found}})$  then
14           $X_{viewpoints} \leftarrow \{x_{\text{found}}\}$ 
15           $\text{GainCalculation}(x_{\text{found}})$ 
16           $\text{RegisterCoverage}(x_{\text{found}})$ 
17
18 return  $X_{viewpoints}$ 

```

where z represents the measurements in the volumetric map and $C(z)$ is a set of cells covered by the sensor.

B. Frontier-Based Coverage Planning

For the purpose of a complete exploration, we propose a frontier-based coverage planner for a minimum set of viewpoints planning as described in Algorithm 2, which is necessary for exploring the boundaries of explored and unexplored 3D spaces. Two planning instances of 2.5D and 3D scenarios are shown in Fig. 5.

Since our system contains a UAV and a UGV which are with different kinematic constraints, slight customization has been applied to adapt the planner to the different exploration and mapping layers accordingly. For instance, the viewpoints sampling is converted from 3D (SE3) space to 2D (SE2) space to allow instant navigation of the UGV. However, the information gain model is kept in 3D throughout. Here we briefly describe the underlying core algorithm and explain its customized implementation in this work. Firstly, the viewpoints are randomly sampled from the frontier boundary regions using uniform sampling and neighbourhood sampling, i.e., UniformSample (which randomly draws a uniform sample x_1 from the space X)

Algorithm 3: $\text{PruneSamples}(X_{samples})$.

```

1 for  $i = 0$  to  $i < \text{Size}(X_{samples})$  do
2    $\text{RecalculateGain}(x_i)$ 
3    $X_{samples} \leftarrow \{x \in X_{samples} | x.gain < \text{Threshold}\}$ 

```

and NeighbourSample (which samples a random sample x_2 in the neighbourhood of x_1 with a radius r) in the Algorithm 2, followed by frontier check and visibility check. The function IsFrontier returns true if the pair of input samples has one being *free* and the other being *unknown*. The function IsVisible conducts ray-shooting to check if the two samples are visible to each other. A sample is selected as an admissible frontier viewpoint only if it is in a free space and visible to an unknown space sample in its neighbourhood, i.e., the valid sample is returned by the function (TheValidOf).

To ensure efficient sampling and a reasonable set of viewpoints, we employ a neighbourhood checking step to avoid severe overlap of viewpoints coverage. For the coarse exploration layer, this is achieved by a k-d trees-based nearest neighbour check with a specified range [42] (see lines 6-9 in the Algorithm 2). The function NearestNeighbour checks if the viewpoint is in the neighbourhood of its nearest neighbour of the existing viewpoints such that all viewpoints “repel” each other by a range R . For the fine mapping layer, we use ray-casting to mark individual viewpoint coverage in the registered OctoMap (see lines 11-14 in the Algorithm 2). The function IsCovered serves to avoid viewpoints falling in the regions covered by any existing viewpoints (registered by function RegisterCoverage) regarding the sensor model. Doing so allows a relatively “rounder” sample filtering policy for the coarse exploration phase to achieve faster coverage of the environment contour while keeping relatively “finer” for the fine mapping phase to pick up details with smoother movements.

C. Viewpoints Pruning

As the exploration goes on, more viewpoints from the newly explored space may be added. Due to the updating map, necessary re-evaluation and filtering of the information gains are performed to prune the set of viewpoints such that a reasonable scanning coverage is maintained for efficient exploration. The information gains of the remaining viewpoints should be continuously evaluated as they may be affected by the recent scans. The remaining scans and newly inserted samples form a new series of viewpoints candidate. Any viewpoints whose information gain below certain threshold would be dropped from the set to maintain a reasonable size, as shown in Algorithm 3.

D. FSOTSP Solver

After the coverage planning stage, numerous viewpoints are available for scanning. The viewing sequence and the associated paths will affect the exploration and mapping efficiency. The straightforward methods are greedy schemes but are subject

Algorithm 4: Cost Matrix ($X_{viewpoints}$).

```

1 for  $i = 0 : 1 : \text{SizeOf}(X_{viewpoints}) - 1$  do
2   for  $j = 0 : 1 : \text{SizeOf}(X_{viewpoints}) - 1$  do
3      $Cost_{ij} = \bar{x}_i \bar{x}_j / (\text{gain}_{x_i} + \text{gain}_{x_j} * \exp^{-\eta \bar{x}_i \bar{x}_j})$ 

```

Algorithm 5: FSOTSP Solver ($X_{viewpoints}$).

```

1 for  $k = 0 : 1 : N - 1$  do
2    $\text{Population} \leftarrow \text{Randomize}(X_{viewpoints})$ 
3    $S_{best} \leftarrow \text{BestSequenceOf}(\text{Population})$ 
4   while  $\text{TerminationConditionMet}$  do
5      $\text{Subgroups} \leftarrow \text{RandomPartition}(\text{Population})$ 
6      $\text{Clear}(\text{Population})$ 
7     for  $n = 0 : 1 : \text{NumberOfSubgroups} - 1$  do
8        $\text{Parent} \leftarrow \text{BestSequenceOf}(\text{Subgroups}[n])$ 
9       for  $m = 0 : 1 : \text{SizeOf}(\text{Subgroups}[n]) - 1$  do
10       $\text{Population} \leftarrow \text{Mutate}(\text{Parent})$ 
11     $S_{best} \leftarrow \text{BestOf}(S_{best}, \text{BestSequenceOf}(\text{Population}))$ 
12 return  $S_{best}$ 

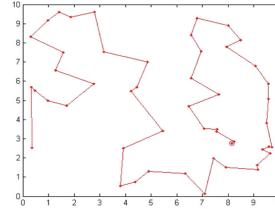
```

to local minima problem. The sense of global optimality is therefore important to keep the overall exploration efficiency.

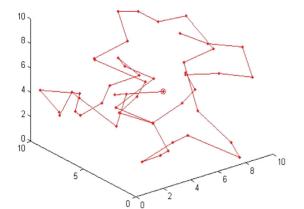
In this work, we consider a variant of the traveling salesman problem (TSP) which connects the viewpoints in an open-loop optimal sequence. The fixed start open traveling salesman problem (FSOTSP) takes the current position of the vehicle as a starting point and tries to visit each of the viewpoints once and ends at one of them with a minimal cost.

To achieve the global optimality, we formulate the cost in FSOTSP as a function of the information gain and traveling distance. The utility of the edge connecting two viewpoints is supposed to be information gain driven and penalized on the traveling distance. Therefore, the cost matrix of the viewpoints set $X_{viewpoints}$ should be the reverse of the utility as described in Algorithm 4, where \bar{x}_i, \bar{x}_j represents the Euclidean norm of the viewpoints x_i and x_j . The parameter η is tunable for emphasizing the depreciation of the utility along the direction of travelling, which makes a trade-off between the information gain and the travelling distance.

To solve the FSOTSP, we propose and implement an efficient genetic algorithm as presented in Algorithm 5 to obtain an optimal viewing sequence iteratively. The algorithm employs a simplified breeding mechanism in which the population is randomly partitioned into subgroups to undergo local selection and mutation without crossover operation (see lines 4-10 in Algorithm 5). In such a way, the evolution process iteratively updates the global optimal sequence, providing a reasonable speed without sacrificing the quality of the solution. The objective function is the total cost of a route, which is to be optimized in the algorithm. The best route is sought for every population. Then, the population is divided into sub-groups. In each sub-group,



(a) A 2D planning instance



(b) A 3D planning instance

Fig. 6. FSOTSP solution instances for 50 random viewpoints in 2D and 3D spaces (assuming all viewpoints have the same information gain).

the best route is chosen to generate offspring within the group. Portions of the best picked routes are mutated by the breeding process through sliding and swapping position to generate new routes. Such steps are repeated until each sub-group is filled by new routes. In the end, the best routes in the old and new population are compared to ensure the population is updated using reduced best-cost-sofar. The whole process is repeated until a desired goal is achieved. The optimality is evaluated under the condition of the instantaneous knowledge of the environment model which is unknown *a priori*.

Two exemplary optimized solutions of 2D and 3D cases with 50 random viewpoints using our FSOTSP solver are shown in Fig. 6.

E. Volumetric Motion Planning Interface

The navigation of the UAV and the UGV is supported by a volumetric motion planning interface which integrates the instantaneous environment map with an external motion planning library.

In this work, we employ an improved version of our recently developed sampling-based path planning library, i.e., BIT*-H [43] for navigation in cluttered environments. BIT*-H is specially customized for fast and optimal planning, independent of the collision checking schemes. The algorithm builds on the Batch Informed Trees (BIT*) [44] which employs ordered search and informed sampling of a heuristic (hyper) ellipsoidal subspace limiting the planning space to accelerate the convergence to optimal solutions. We extend the original algorithm with obstacle-guided sampling and biasing strategy to adapt the advantage of the BIT* to cluttered environments. The obstacle-guided sampling strategy embeds additional obstacle-based Gaussian sampling processes in the underlying uniform sampling. Such hybrid sampling strategy leads to a rise in density of the samples near the obstacles, enhancing the probability of sampling difficult-to-find regions like narrow passages. In addition, the embedded Gaussian sampling process uses a simple heuristic to “spot navigators” which are of crucial geometric positions linking to the difficult-to-find regions. This is used to bias the samples near those regions and thus speed up the convergence to globally optimal solutions.

The motion planning interface is facilitated with ray-shooting based collision checker, which is in principle the same as the ray-casting approach engaged for the information gain calculation at the view planning stage, except that the checking range is now

subject to the bounding boxes of the vehicle model. The collision check is performed using the instantaneous OctoMap at the time the planner is engaged. The planner samples the continuous space and traces back the Octree to check the corresponding volumetric cell occupancy.

The preliminary global path plan provides discrete exploration way-points with plain connecting edges which are ready for execution. The waypoints contains both position and orientation configurations, which makes up primary edges for the vehicle to set its poses along the way and conduct continuous mapping while in motion. We further optimize the edges using local trajectory planners such as CHOMP solver [45] for the UAV and the DWA planner [46] for the UGV (with standard cost metrics as per planner). In such a way we combine the global and local optimal planning strategies in a unified infrastructure to serve the navigation.

Some discussion on the computational complexity of Algorithms 1–5 can be found in the full-length version of this paper in https://drive.google.com/file/d/1XcHwOUT_StcVA3mWRCNoXtsha6s-eBVN/view?usp=drivesdk.

VI. COLLABORATIVE PERCEPTION SCHEME

Firstly, we would like to mention both the UAV and the UGV are capable of independent autonomous exploration and reconstruction work in unknown environments using our optimized view planning paradigm. Practically, the performance of such independent exploration and reconstruction is subject to physical constraints of the platforms. For instance, UAVs usually possess higher mobility and agility but are weak in payload and flight endurance. They are suitable for complex structures instead of large-scale environments. In contrast, UGVs usually have better endurance and greater on-board resources, including a variety of sensors and data processing units, but suffer from constrained planar motions. Usually, the scale and complexity of the environments are unknown *a priori*. Instead of relying on single type of vehicle alone, we propose a collaborative scheme using heterogeneous UAVs and UGVs team to conduct complementary perception tasks in unknown environments.

1) *Two-Layered Exploration and Mapping:* To wisely exploit the advantages of heterogeneous vehicles while considering their respective limitations, we decompose the exploration and mapping tasks into two layers. The first layer is designed to be a coarse mapping process, in which the UGV mounted with a 3D LIDAR is deployed to perform a preliminary exploration and produce a coarse mapping. Upon completion, a primary map built in the first layer is retrieved by the UAV as a fundamental environment model and a navigational reference. In the second layer, the UAV exploits its advantage in mobility to perform a complementary fine mapping using a tilting 2D laser module and vision sensors, filling up the cavities left over in the first layer.

2) *Assistive Perception for Local Scene:* Once the overall environment model construction is finished, the UGV can be engaged for navigation tasks. However, the previously built volumetric environment model may not be detailed enough to carry out complex tasks like contacting and grasping (e.g., subject to

occlusions and missing information) since the perception unit on the UGV is often subjected to blind zones due to its restricted motions. In this case, an augmented perception with finer resolution would be required frequently. The UAV is therefore a better choice to provide assistive perception due to its higher mobility. Specifically, targeting certain regions whose scene information is insufficient, the UAV can be deployed to perform augmented perception using the same optimized view planning framework. The reconstruction contribution from the UAV is again integrated with the existing map using the SLAM strategy as explained in Section IV.

3) *Map Transfer:* Our UAV-UGV cooperation strategy is essentially in a decentralized manner: the UAV and the UGV update their navigation strategy and map on their own to explore the environments extensively. Comparing with the existing approaches, we highly utilize the vehicle-specific mobility of heterogeneous system to construct and explore the unknown environments in different detailed levels. Here we emphasize the importance of map sharing issue for maximizing the performance of the UAV and the UGV. The basic target is towards a global coherent exploration result with different level of awareness of the environments. During the map transferring, we first treat the map from the UGV as point-cloud based format. Then, we down-sample the point-cloud size to reduce the size of the map into a delicate level, after which the map is further converted to a volumetric map. The UAV takes the volumetric map upon instantiation and updates the occupancy information of previous uncovered regions during the exploration.

By combining the efforts of the UAV and the UGV, we can realize an autonomous 3D fine mapping which is usually a challenging task.

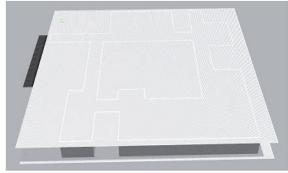
VII. SIMULATION AND EXPERIMENTAL RESULTS

This section presents the simulation and experimental results of effective perception for autonomous exploration, mapping and manipulation in unknown environments using our proposed heterogeneous UAV and UGV system.

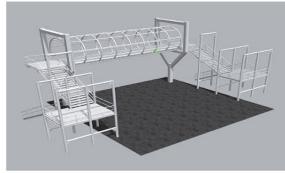
A. Simulation Results

1) *Independent Exploration and Mapping:* The simulation is conducted using Gazebo simulator RotorS [47]. We employ a LIDAR model with 16 channels and range up to 30 m for the UGV 2.5D mapping and a stereo-vision sensor model with $100^\circ(H) \times 60^\circ(V)$ FoV and range up to 10 m for the UAV 3D mapping.

Firstly, we demonstrate the ability of independent exploration and mapping on both the UAV and the UGV using our optimized view planning framework and the volumetric motion planning interface. Simulated worlds such as an office corridor and a footbridge are created for the UAV and the UGV respectively. As shown in Fig. 7, both vehicles are able to generate reasonable exploration routes and conduct complete overall mapping of the environment contours from scratch, which are sufficient for their respective navigation. Due to the vehicle and sensor constraints, the 2.5D mapping by the UGV is sparser and misses out certain details at corners, as expected.



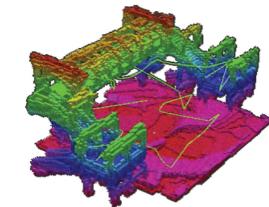
(a) Simulated office corridor



(b) Simulated footbridge

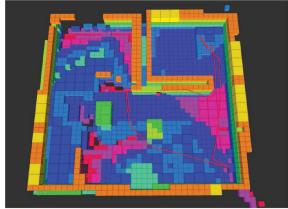


(c) Example of UGV 2.5D mapping

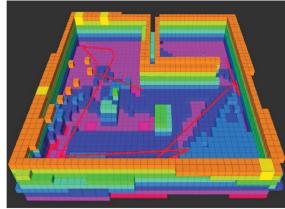


(d) Example of UAV 3D mapping

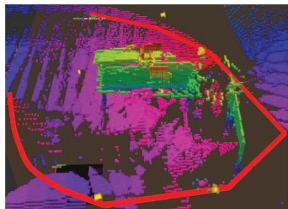
Fig. 7. Illustration of independent exploration and mapping using UGV and UAV alone. Exploration path highlighted with colored line segments. OctoMap colors correspond to height variation.



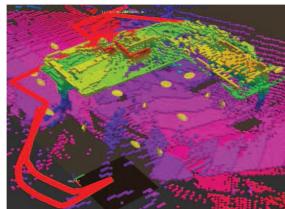
(a) 2.5D frontier based (UGV)



(b) Our approach (UGV)



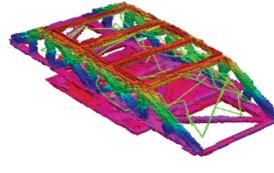
(c) 2.5D frontier based (UAV)



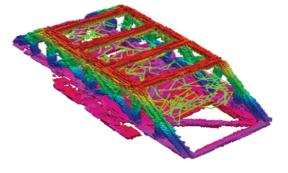
(d) Our approach (UAV)

Fig. 8. Mapping results for a simulated chamber and a tabletop environment using our approach and a 2.5D frontier based approach. The exploration paths are highlighted in red. The colors of OctoMap correspond to height variation.

In order to evaluate the efficiency of our proposed view planning framework, we compare our approach with a 2.5D frontier based scheme, where the exploration and mapping is conducted according to 2D environment information acquired, and one of the state-of-the-art NBV planner, the Receding Horizon NBV planner [6]. The 2.5D frontier based planner greedily plans for the exploration sequence of 3-DOF (x , y , yaw) frontier-based viewpoints sampled over the 2D environment map, while a 3D map is built using sensor scanning during the exploration, thus referred as 2.5D. The qualitative comparison results are shown in Fig. 8. From the figure we can see that both approaches are able to map a complete contour of the planar space. However, the 2.5D frontier based scheme apparently misses out some 3D features which are spotted in our approach. This elaborates the advantage of utilizing 3D information gain in our work, regardless of the vehicle configuration.

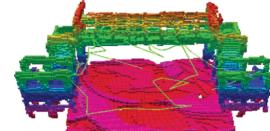


(a) Ours

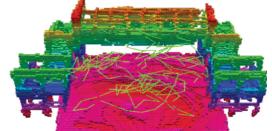


(b) Receding horizon

Fig. 9. Exploration results for simulated bridge using a stereo vision sensor. Exploration paths highlighted in 3D green line segments.



(a) Ours



(b) Receding horizon

Fig. 10. Exploration results for simulated footbridge using a stereo vision sensor. Exploration paths highlighted in 3D green line segments.

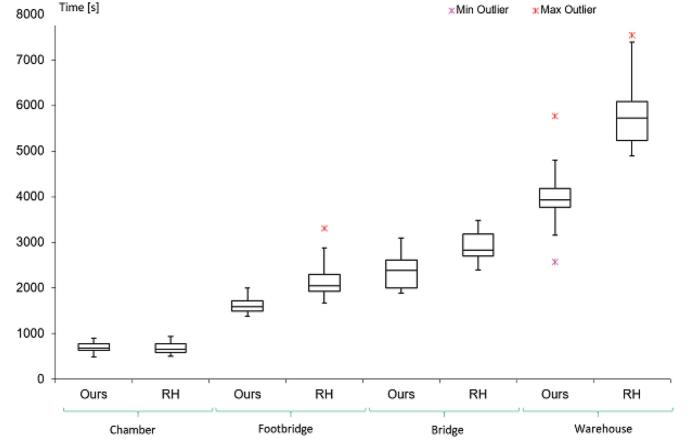


Fig. 11. Statistics of 10 successful runs for the four simulated environments using our approach and the receding horizon (RH) planner.

Repeated trials of 3D mapping using UAVs are also conducted to qualitatively compare our approach with the frontier based scheme. Moreover, our approaches are implemented in four different environments. The test results in footbridge and bridge environments are shown in Figs. 9 and 10, respectively. The statistical comparison results of for the four different environments are shown in Fig. 11. As the statistics reveal, our planner generally returns smoother and shorter exploration paths (and time) while the Receding Horizon (RH) planner tends to give more complex and longer paths, which becomes more and more apparent as the complexity and scale of the environment increase. In addition, the performances also reflect that our approach is less sensitive to environment size from the perspective of tuning parameters, as the RH planner require more frequent adjustment of the exploration tree parameters in order to adapt to new scenarios.

2) Collaborative Exploration and Mapping: Extensive simulations have been conducted to evaluate our proposed UAV-UGV collaborative exploration and mapping scheme. In Fig. 12,

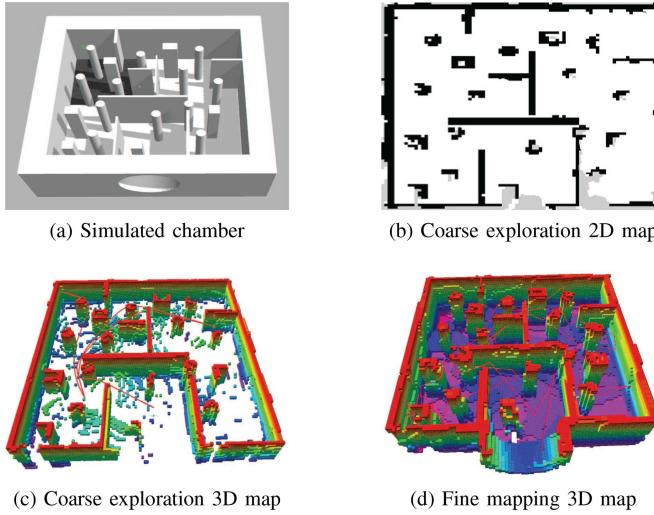


Fig. 12. Collaborative exploration and mapping for a simulated cluttered chamber. Exploration path highlighted with colored line segments. OctoMap colors correspond to height variation.

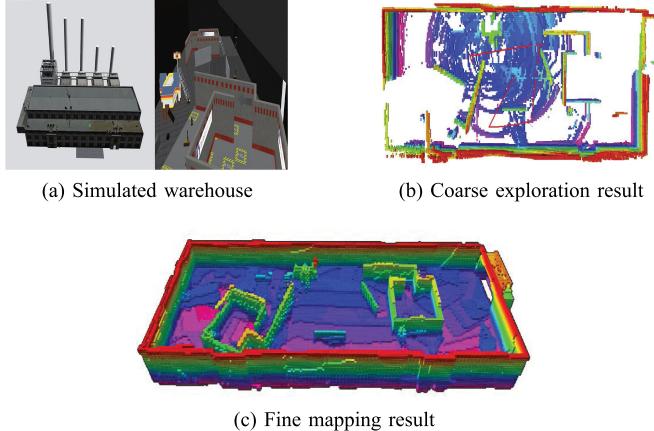


Fig. 13. Collaborative exploration and mapping for a simulated large-scale warehouse.

a simulated chamber with cluttered obstacles and barriers is explored by the UAV, which provides a coarse 2.5D volumetric model of the chamber (2D cost map+3D OctoMap). The 2D map is projected from the 3D map for UGV navigation purpose only. Starting from the coarse model, the UAV with a stereo vision sensor continues to perform the fine mapping and returns a complete 3D map of the chamber, covering up the uncollected information of regions not reached by the UGV. Similarly, the test results of a large-scale warehouse are presented in Fig. 13. The coarse mapping by UGV is able to provide a basic contour of the environment with a few steps of auto-exploration. However, the details of interior layout is missing due to the constrained motion of the UGV. As expected, the UAV successfully fills up the cavities with necessary geometric structures.

Quantitative evaluation is also performed to compare the collaborative exploration with independent exploration in terms of total exploration time. The simulated warehouse environment is chosen as the benchmark scenario since it is of large-scale and contains various complex structures. As we observe that the

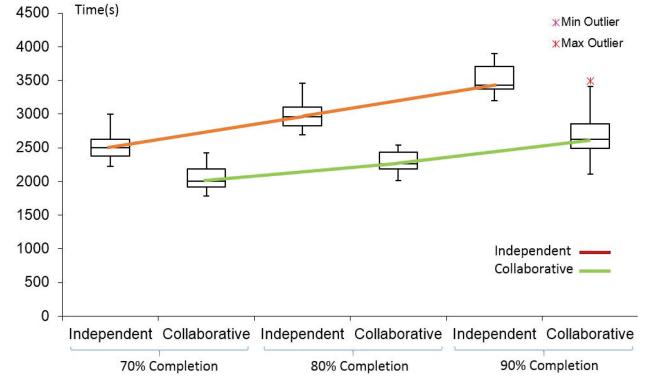


Fig. 14. Comparison of independent exploration using UAV and collaborative exploration in 10 successful runs for the simulated warehouse.

independent 2.5D mapping by the UGV is only able to cover up to approximately 60% of the specified bounding box of the 3D exploration space, we employ the UAV for independent exploration scheme, and record the exploration time at several check points-completion (reconstruction) of 70%, 80%, and 90% of the specified exploration space, respectively, to investigate the effectiveness of the proposed collaborative evaluation scheme. The statistics of 10 trials of the independent exploration and the collaborative exploration is shown in Fig. 14. The results demonstrate that the collaborative scheme is more advantageous in this large-scale environment in terms of total exploration time, which credits to the 2.5D coarse mapping saving majority of the view planning and motion planning efforts with reduced planning space (SE2 space compared to SE3 space). We also note that the increments in exploration time between the check points are less in the collaborative scheme than the independent scheme. This is probably because the coarse mapping layer provides global environment information to the fine mapping layer to be used as the navigation reference, such that the original global optimization problem is roughly solved and decomposed to local sub-optimization operations which take less exploration and mapping effort.

3) Assistive Scene Augmentation: A table-top scene as shown in Fig. 15 is simulated to illustrate the assistive augmentation using the UAV. In the setup, the UGV carrying the manipulator scans the scene with a RGB-D sensor and constructs the corresponding volumetric model. The UGV is able to offer an almost full view of the table with omnidirectional planar motion but fails to provide complete reconstruction of the objects on the top as well as lose some details underneath, as shown in Fig. 15(b)–(c), due to limited sensor view. Fig. 15(d) present the results of the augmented table-top scene by the UAV. Such augmentation enriches the local object information in details which could potentially be helpful for manipulation applications requiring pushing or grasping tasks, etc. To clarify, the manipulation tasks considered in this work are not regarding visual/semantic based manipulation capabilities, but making use of off-the-shelf manipulation packages such as MoveIt [48] and LineMOD [49] which can already be integrated to our framework, to perform position-based manipulation with geometric point cloud information.

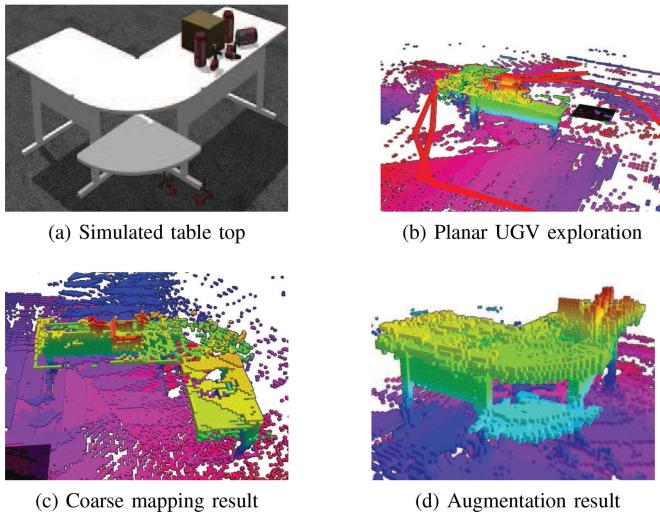


Fig. 15. UAV scene augmentation for a simulated table top scenario. Exploration path highlighted with colored line segments. OctoMap colors correspond to height variation.

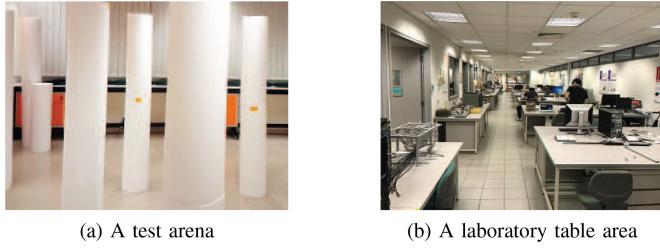


Fig. 16. Real flight test environments.

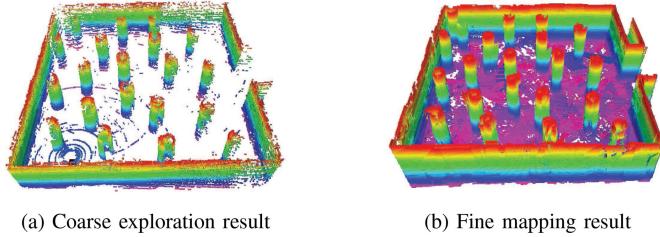


Fig. 17. Collaborative exploration and mapping for the test arena. Exploration path highlighted with colored line segments. OctoMap colors correspond to height variation.

B. Experiment Results

Real flight tests have also been conducted to verify the feasibility of our system in real-world applications. A test arena with cluttered pillars and a laboratory area as shown in Fig. 16 are involved for the two-layered collaborative exploration and mapping tests and the scene augmentation tests, respectively. The mapping results of the test arena are shown in Fig. 17. The coarse map is constructed using the UGV with a 3D LIDAR, producing a sparse 3D map and incomplete information of the obstacles. The UAV continues the mapping with a tilting 2D laser module and manages to build a denser map with finer local information, making up the missing parts on the ground, walls, and the pillars. In Fig. 18, the local reconstruction of a table

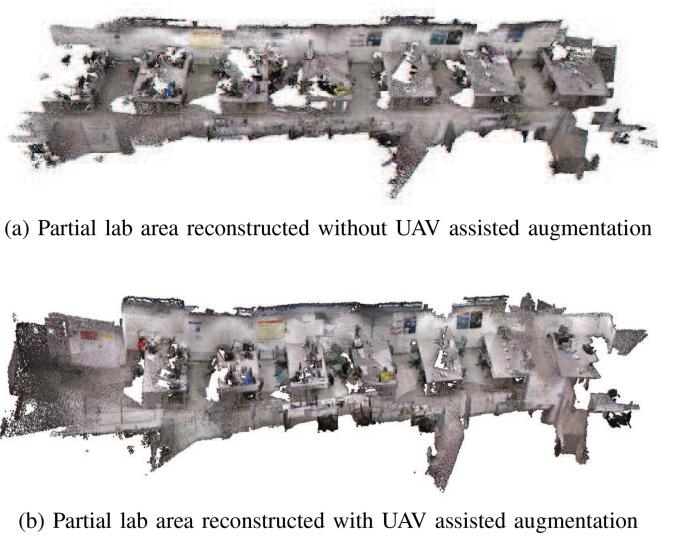


Fig. 18. 3D laboratory scene reconstructed.

area in a laboratory using a UGV (with a RGB-D sensor) and a UAV (with a vision sensor), is presented. The UGV performs local scans near the table area and reconstructs the scene. Due to the large footprint of the UGV limiting the scanning positions, loss of information is present as shown in Fig. 18(a) (the cavities). The UAV with a vision sensor continues the reconstruction with a focused augmentation around the middle table. Although slight distortion due to the localization errors is present, the missing table-top and obstacle information is compensated as shown in Fig. 18(b).

VIII. CONCLUSION

In this paper, we present a collaborative vehicular system using heterogeneous UAV and UGV agents with optimized perception scheme for autonomous exploration, mapping, navigation and manipulation tasks in GPS-denied 3D unknown environments. The system implements an optimized perception scheme to decompose the task as a coarse exploration layer and a fine mapping layer using our optimized next view planning framework. The overall system provides an efficient pipeline for fast environment perception. We have evaluated our proposed framework with simulation and experimental tests, which well demonstrate the effectiveness and robustness of our proposed framework in terms of travelling distance and map quality. The system is proven capable of implementing heterogeneous collaborative exploration and active SLAM, providing optimized perception for navigation and potential manipulation tasks.

REFERENCES

- [1] Y. Okada and J. Miura, "Exploration and observation planning for 3D indoor mapping," in *Proc. IEEE/SICE Int. Symp. Syst. Integration*, 2015, pp. 599–604.
- [2] A. Bircher *et al.*, "Structural inspection path planning via iterative viewpoint resampling with application to aerial robotics," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2015, pp. 6423–6430.
- [3] C. Luo, S. McClean, G. Parr, L. Teacy, and R. D. Nardi, "UAV position estimation and collision avoidance using the extended Kalman filter," *IEEE Trans. Veh. Technol.*, vol. 62, no. 6, pp. 2749–2762, Jul. 2013.

- [4] K. Lenac *et al.*, “Fast active SLAM for accurate and complete coverage mapping of unknown environments,” in *Proc. 13th Int. Conf. Intell. Autonomous Systems*, Padova, Italy, 2014, pp. 415–428.
- [5] C. Potthast and G. Sukhatme, “A probabilistic framework for next best view estimation in a cluttered environment,” *J. Vis. Commun. Image Representation*, vol. 25, no. 1, pp. 148–164, 2014.
- [6] A. Bircher, M. Kamel, K. Alexis, H. Oleynikova, and R. Siegwart, “Receding horizon next-best-view planner for 3D exploration,” in *Proc. IEEE Int. Conf. Robot. Automat.*, 2016, pp. 1462–1468.
- [7] H. M. Huang, K. Pavek, B. Novak, J. Albus, and E. Messin, “A framework for autonomy levels for unmanned systems (ALFUS),” in *Proc. AUVSIs Unmanned Syst. North Amer.*, 2005, pp. 849–863.
- [8] B. Grocholsky, J. Keller, V. Kumar, and G. Pappas, “Cooperative air and ground surveillance,” *IEEE Robot. Automat. Mag.*, vol. 13, no. 3, pp. 16–25, Sep. 2006.
- [9] W. Meng, Z. He, R. Su, P. Yadav, R. Teo, and L. Xie, “Decentralized multi-UAV flight autonomy for moving convoys search and track,” *IEEE Trans. Control Syst. Technol.*, vol. 25, no. 4, pp. 1480–1487, Jul. 2017.
- [10] Y. Hu and W. Meng, “ROSUnitySim: Development and experimentation of a real-time simulator for multi-unmanned aerial vehicle local planning,” *Simulation*, vol. 92, no. 10, pp. 931–944, 2016.
- [11] A. Brandão, M. Sarcinelli-Filho, and R. Carelli, “Leader-following control of a UAV-UGV formation,” in *Proc. 16th Int. Conf. Adv. Robot.*, 2013, pp. 1–6.
- [12] M. Kamel, K. Ghamry, and Y. Zhang, “Fault tolerant cooperative control of multiple UAVs-UGVs under actuator faults,” in *Proc. Int. Conf. Unmanned Aircraft Syst.*, 2015, pp. 644–649.
- [13] L. Paull *et al.*, “Towards an ontology for autonomous robots,” in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2012, pp. 1359–1364.
- [14] K. Guo, Z. Qiu, W. Meng, L. Xie, and R. Teo, “Ultra-wideband based cooperative relative localization algorithm and experiments for multiple unmanned aerial vehicles in GPS denied environments,” *Int. J. Micro Air Veh.*, vol. 9, no. 3, pp. 169–186, 2017.
- [15] T. Petrovic and T. Haus, “Can UAV and UGV be best buddies? Towards heterogeneous aerial-ground cooperative robot system for complex aerial manipulation tasks,” in *Proc. 12th Int. Conf. Inform. Control, Automat. Robot.*, 2015, pp. 238–245.
- [16] M. Blosch, S. Weiss, D. Scaramuzza, and R. Siegwart, “Vision based MAV navigation in unknown and unstructured environments,” in *Proc. IEEE Int. Conf. Robot. Automat.*, 2010, pp. 21–28.
- [17] L. Paz, P. Piniés, J. Tardós, and J. Neira, “Large-scale 6-DOF SLAM with stereo-in-hand,” *IEEE Trans. Robot.*, vol. 24, no. 5, pp. 946–957, Oct. 2008.
- [18] C. Forster, M. Pizzoli, and D. Scaramuzza, “SVO: Fast semi-direct monocular visual odometry,” in *Proc. IEEE Int. Conf. Robot. Automat.*, 2014, pp. 15–22.
- [19] C. Forster, M. Pizzoli, and D. Scaramuzza, “Air-ground localization and map augmentation using monocular dense reconstruction,” in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2013, pp. 3971–3978.
- [20] G. Klein and D. Murray, “Parallel tracking and mapping for small AR workspaces,” in *Proc. 6th IEEE ACM Int. Symp. Mixed Augmented Reality*, 2007, pp. 225–234.
- [21] R. Kuemmerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard, “g2o: A general framework for graph optimization,” in *Proc. IEEE Int. Conf. Robot. Automat.*, Shanghai, China, May 2011, pp. 3607–3613.
- [22] J. Bouguet, “Pyramidal implementation of the affine lucas kanade feature tracker description of the algorithm,” *Intel Corporation*, vol. 5, no. 1–10, pp. 1–10, 2001.
- [23] A. Howard, “Real-time stereo visual odometry for autonomous ground vehicles,” in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2008, pp. 3946–3952.
- [24] Y. Cheng, M. Maimone, and L. Matthies, “Visual odometry on the mars exploration rovers—A tool to ensure accurate driving and science imaging,” *IEEE Robot. Automat. Mag.*, vol. 13, no. 2, pp. 54–62, Jun. 2006.
- [25] A. Milella and R. Siegwart, “Stereo-based ego-motion estimation using pixel tracking and iterative closest point,” in *Proc. IEEE Int. Conf. Comput. Vis. Syst.*, 2006, pp. 1–4.
- [26] S. Kohlbrecher, S. Von, J. Meyer, and U. Klingauf, “A flexible and scalable slam system with full 3D motion estimation,” in *Proc. IEEE Int. Symp. Safety, Secur. Rescue Robot.*, 2011, pp. 155–160.
- [27] A. Segal, D. Haehnel, and S. Thrun, “Generalized-ICP,” in *Proc. Robot., Sci. Syst.*, 2009, vol. 2, pp. 1–8.
- [28] J. Zhang and S. Singh, “Visual-lidar odometry and mapping: Low-drift, robust, and fast,” in *Proc. IEEE Int. Conf. Robot. Automat.*, 2015, pp. 2174–2181.
- [29] H. Qin *et al.*, “A stereo and rotating laser framework for UAV navigation in GPS denied environment,” in *Proc. Annu. Conf. IEEE Ind. Electron. Soc.*, 2016, pp. 6061–6066.
- [30] P. Blaer and P. Allen, “Data acquisition and view planning for 3-D modeling tasks,” in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2007, pp. 417–422.
- [31] J. Vasquez-Gomez, L. Sucar, R. Murrieta-Cid, and E. Lopez-Damian, “Volumetric next-best-view planning for 3d object reconstruction with positioning error,” *Int. J. Adv. Robot. Syst.*, vol. 11, pp. 1–13, 2014.
- [32] R. Martin, I. Rojas, K. Franke, and J. Hedengren, “Evolutionary view planning for optimized UAV terrain modeling in a simulated environment,” *Remote Sens.*, vol. 8, no. 1, pp. 26–50, 2015.
- [33] S. Manyam, D. Casbeer, and K. Sundar, “Path planning for cooperative routing of air-ground vehicles,” in *Proc. Amer. Control Conf.*, 2016, pp. 4630–4635.
- [34] J. Li, G. Deng, C. Luo, Q. Lin, Q. Yan, and Z. Ming, “A hybrid path planning method in unmanned air/ground vehicle (UAV/UGV) cooperative systems,” *IEEE Trans. Veh. Technol.*, vol. 65, no. 12, pp. 9585–9596, Dec. 2016.
- [35] H. Tanner, “Switched UAV-UGV cooperation scheme for target detection,” in *Proc. IEEE Int. Conf. Robot. Automat.*, 2007, pp. 3457–3462.
- [36] R. Vidal, O. Shakernia, H. Kim, D. Shim, and S. Sastry, “Probabilistic pursuit-evasion games: Theory, implementation, and experimental evaluation,” *IEEE Trans. Robot. Automat.*, vol. 18, no. 5, pp. 662–669, Oct. 2002.
- [37] H. Yu, R. Beard, M. Argyle, and C. Chamberlain, “Probabilistic path planning for cooperative target tracking using aerial and ground vehicles,” in *Proc. Amer. Control Conf.*, 2011, pp. 4673–4678.
- [38] D. J. Barrett and R. E. Silverman, *SSH, The Secure Shell: The Definitive Guide*, O'Reilly Media, Inc., Sebastopol, CA, USA, 2001.
- [39] A. Nurunnabi, G. West, and D. Belton, “Robust methods for feature extraction from mobile laser scanning 3D point clouds,” *Res. @ Locate*, vol. 15, pp. 109–120, 2015.
- [40] Y. Yuan, “A review of trust region algorithms for optimization,” in *Proc. 4th Int. Congr. Ind. Appl. Math.*, 2000, vol. 99, pp. 271–282.
- [41] Z. Meng *et al.*, “A two-stage optimized next-view planning framework for 3-D unknown environment exploration, and structural reconstruction,” *IEEE Robot. Automat. Lett.*, vol. 2, no. 3, pp. 1680–1687, Jul. 2017.
- [42] C. Silpa-Anan and R. Hartley, “Optimised KD-trees for fast image descriptor matching,” in *Proc. CVPR IEEE Conf. Comput. Vis. Pattern Recognit.*, 2008, pp. 1–8.
- [43] Z. Meng, H. Sun, X. Shen, Z. Chen, and M. H. Ang, “System integration: Application towards autonomous navigation in cluttered environments,” in *Proc. IEEE/SICE Int. Symp. Syst. Integration*, 2016, pp. 786–791.
- [44] J. Gammell and S. Srinivasa, and T. Barfoot, “Batch informed trees (BIT*): Sampling-based optimal planning via the heuristically guided search of implicit random geometric graphs,” in *Proc. IEEE Int. Conf. Robot. Automat.*, 2015, pp. 3067–3074.
- [45] N. Ratliff, M. Zucker, J. Bagnell, and S. Srinivasa, “CHOMP: Gradient optimization techniques for efficient motion planning,” in *Proc. IEEE Int. Conf. Robot. Automat.*, 2009, pp. 489–494.
- [46] P. Ogren and N. E. Leonard, “A convergent dynamic window approach to obstacle avoidance,” *IEEE Trans. Robot.*, vol. 21, no. 2, pp. 188–195, Apr. 2005.
- [47] F. Furrer, M. Burri, M. Achtelik, and R. Siegwart, *Robot Operating System (ROS): The Complete Reference (Volume 1)*. New York, NY, USA: Springer, pp. 595–625, 2016.
- [48] S. Chitta, I. Sucan, and S. Cousins, “Moveit![ROS topics],” *IEEE Robot. Automat. Mag.*, vol. 19, no. 1, pp. 18–19, Mar. 2012.
- [49] S. Hinterstoisser *et al.*, “Model based training, detection and pose estimation of texture-less 3d objects in heavily cluttered scenes,” in *Proc. Asian Conf. Comput. Vis.*, 2012, pp. 548–562.

Authors' photographs and biographies not available at the time of publication.