Received: 18 April 2022 | Revised: 12 May 2023 | Accepted: 21 May 2023

DOI: 10.1002/rob.22213

RESEARCH ARTICLE

WILEY

Aerial online mapping on-board system by real-time object detection for UGV path generation in unstructured outdoor environments

Jehong Lee | Jeonggeun Lim | Sangjin Pyo | Jongho Lee

School of Mechanical Engineering, Gwangju Institute of Science and Technology, Gwangju, South Korea

Correspondence
Jongho Lee, School of Mechanical Engineering, Gwangju Institute of Science and Technology, Gwangju 61005, South Korea.
Email: jong@gist.ac.kr

Funding information
GIST Research Institute project; Korea Institute of Advancement for Police-technology

Abstract

An optimal path provides efficient operation of unmanned ground vehicles (UGVs) for many kinds of tasks such as transportation, exploration, surveillance, and search and rescue in unstructured areas that include various unexpected obstacles. Various onboard sensors such as LiDAR, radar, sonar, and cameras are used to detect obstacles around the UGVs. However, their range of view is often limited by movable obstacles or barriers, resulting in inefficient path generation. Here, we present the aerial online mapping system to generate an efficient path for a UGV on a two-dimensional map. The map is updated by projecting obstacles detected in the aerial images taken by an unmanned aerial vehicle through an object detector based on a conventional convolutional neural network. The proposed system is implemented in real-time by a skid steering ground vehicle and a quadcopter with relatively small, low-cost embedded systems. The frameworks and each module of the systems are given in detail to evaluate the performance. The system is also demonstrated in unstructured outdoor environments such as in a football field and a park with unreliable communication links. The results show that the aerial online mapping is effective in path generation for autonomous UGVs in real environments.

KEYWORDS

autonomous vehicles, cooperative systems, mobile robots, multirobot systems, object detection, path planning, sensor fusion, terrain mapping

1 | INTRODUCTION

Unmanned ground vehicles (UGVs), which provide a relatively high payload, stability, and reliable interactions with environments (Adamides et al., 2017; Li et al., 2021), are expanding their application areas, including delivery (Pani et al., 2020), transportation (Ljungqvist et al., 2019), exploration (Ebadi et al., 2020), search and rescue (Balta et al., 2017; Delmerico et al., 2019) and monitoring and inspection (La et al., 2017; Merriaux et al., 2018). In many cases, because UGVs rely on onboard sensors including cameras (Brooks & Iagnemma, 2012), LiDAR (Ren et al., 2021), radar (Milella et al., 2015), or ultrasonic sensors (Kolvenbach et al., 2020), their range of view is limited; thus, a generated path to follow may not be ideal, particularly, in environments with movable obstacles or barriers. For example, during search and rescue operations in large disaster areas, traversable paths may change continuously because of rescue workers, crowds, vehicles, and rubble may occupy potentially any place.

Many approaches have been reported for UGVs to generate optimal paths by perceiving obstacles or barriers that are not detectable by onboard sensors. In particular, heterogeneous cooperation from an unmanned aerial vehicle (UAV) can be useful as UAVs have a much longer range of view in open spaces (Bouman et al., 2020; Cantieri et al., 2020; Faessler et al., 2016; Krátký et al., 2021; Lakas et al., 2018).

Recent advancements in computer vision algorithms have enabled converting sequential aerial images to the three-dimensional (3D)-structured digital elevation model (DEM) to search for an optimal path that avoids relatively a high slope or roughness (Christie et al., 2017; Guastella et al., 2017). The approach is effective because DEMs have elevation information already once the DEMs are constructed for a large unstructured area; however, building DEMs in environments with not many features may require very intensive computations (Peterson et al., 2019).

Other approaches have focused on the online mapping of local terrain to update dynamic local obstacles through a centralized (Delmerico et al., 2017; Dubé et al., 2019; Ebadi et al., 2020; Potena et al., 2019; Schmuck & Chli, 2017, 2018) or decentralized cooperative navigation architecture (Fankhauser et al., 2016; Peterson et al., 2018; Song, 2019). The centralized architectures generate a large map incrementally by matching the visual information observed by UAVs onto the map server system. The vehicles transmit 3D point clouds (Dubé et al., 2019; Potena et al., 2019), aerial images (Schmuck & Chli, 2018), or local maps (Ebadi et al., 2020) to the map management system connected with the vehicles. These methods are useful when a high-bandwidth wireless network for a large area can be secured. For the feasibility of these approaches, a framework was studied in which UAVs and UGVs built and maintained large-area wireless networks themselves (Otsu et al., 2020). However, this framework also has the risk of intermediate disconnection and the limitations of portable wireless access points (APs). In decentralized architectures, a UAV and UGV generate the map and plan the optimal path without a network to a high-performance server. One approach shows that a UAV only transmits the aerial images and point clouds to a UGV generating a terrain map because the UGV can be equipped with relatively heavy and high-performance systems (Fankhauser et al., 2016). In other approaches, the UAV directly plans an efficient path of the UGV by using aerial images captured by the onboard camera (Peterson et al., 2018; Song, 2019). These approaches also require a local wireless network without communication drops between the vehicles or a long-range wireless network with a relatively large bandwidth. However, installing and maintaining a stable high-bandwidth wireless network in outside environments over large areas may be challenging. Especially, in disaster sites, wired networks may be damaged and wireless base stations processing all the communication requests may be lost or jammed. In many cases, therefore, it will be very helpful for a cooperative heterogenous system to have real-time online mapping to generate efficient paths for dynamic environments by transmitting small-sized data packets and predicting paths with data through low bandwidth communication.

In this study, we present an aerial online mapping system with robustness for communication drops that generates an efficient path for an UGV by updating obstacles on a 2D map by projecting the detected objects in the aerial online images from an UAV. The proposed system was designed to operate even when the UAV and the UGV are connected in a relatively weak communication link, so the UGV could keep proceeding by predicting locations of obstacles even with communication lag from the UAV. The proposed system requires relatively small or intermittent bandwidth to keep the UGV proceeding, when comparing to the centralized or decentralized mapping architectures that require to transmit relatively large data such as 3D point clouds, aerial images, or local maps to the map servers or UGVs. In other cases of decentralized mapping, a UAV is required to continuously transmit the generated paths from an onboard processor to a UGV through a reliable wireless network. In our work, (1) we made the minimized data packets (positions and widths) of the obstacles inferred from the UAV to transmit the minimized communication data to the UGV, and (2) we configured the Kalman filter so that the UGV could continuously estimate positions of the obstacles even with data transmission lag from the UAV. With the framework, even though the UGV cannot receive pose information (positions and widths) from the object detector on the UAV, it is possible to continuously predict positions of the obstacles by estimating the pose of the UGV. We designed, implemented and demonstrated these concepts in the fully working system and demonstrated in outdoor environments on campus with low bandwidth wireless network (2 Mbps or less). In Section 2, we describe the overall framework and hardware implementation. In Section 3, we detail each component of the system along with experimental results. In Section 4, we present demonstrations of the aerial online mapping system on a football field and in a park. Finally, we conclude the results in Section 5.

2 | SYSTEM DESIGN AND HARDWARE IMPLEMENTATION

In this section, we introduce the basic concept, overall framework, hardware implementation of the aerial online mapping system that provides information about obstacles detectable from the UAV as illustrated in Figure 1. With a prior map (green bushes only), the UGV tries to go through the shortest path but should come back and go around (red dashed line) after detecting the unexpected obstacles (cones). However, the aerial online mapping system can provide the minimal mapping information (blue dotted line) about the unexpected obstacles early on so that the UGV can generate and follow an efficient path (green line).

2.1 | Overall framework design of the aerial online mapping system

The aerial online mapping system is designed to detect obstacles globally from an UAV and to generate an updated obstacle map and path in a UGV as illustrated in Figure 2. The object detector detects obstacles (e.g., safety cone, bush, tree, etc.) and a UGV on the ground from the aerial image captured by the camera on the UAV. The object projector projects the detected obstacles and UGV on the ground plane in the navigation frame ($\mathbf{P}^{n}_{obs}$, $\mathbf{P}^{n}_{ugv}$) based on the UAV pose from the inertial navigation system (INS)/attitude and heading reference system (AHRS) of the UAV. The UAV transmits the information to the UGV through a relatively low-powered and low-bandwidth wireless network. The UGV pose generator calculates at a higher rate (20 Hz) the filtered UGV poses based on the projected positions ($\mathbf{P}^{n}_{ugv}$, ~4 Hz)

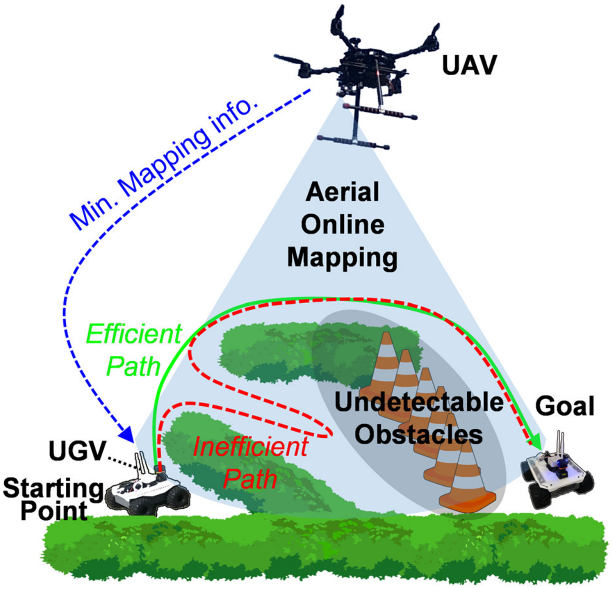

FIGURE 1 Schematic illustration of the aerial online mapping system for efficient path generation of the unmanned ground vehicle (UGV). When the obstacles in the path are undetectable by the UGV because of the blocked detection line or short detection range of the LiDAR, the UGV proceeds further and comes back to go around (red dashed line) after finding the obstacles. The aerial online mapping system provides the minimal mapping information (blue dotted line, and obstacle positions in the navigation frame through a low-powered wireless network) to the UGV to generate a more efficient path (green line).

from the UAV and the inertial navigation data (50 Hz) from the UGV. The obstacle map generator makes the 2D cost map on the prior map using the projected obstacles and local obstacles detected by 2D LiDAR on the UGV. The UGV pose is used to calculate the positions of the local obstacles from the UGV. The path planner generates a path with the lowest cost and velocity commands to the UGV based on the 2D cost map.

2.2 | Hardware implementation on the UAV and UGV

Figure 3 shows the UAV and UGV implemented with embedded computers and various sensors for the aerial online mapping system. The UAV based on a quadcopter is implemented with an embedded computer (Jetson TX2, Nvidia) with inertial sensors (IMUs), the RTK GPS module (HERE + V2 RTK, HEX), camera mounted on a two-axis gimbal, and 1D LiDAR (LW20, Lightware) that is used to estimate the altitude. It should be noted that 3D LiDARs are becoming more affordable and will be more frequently used because of their high accuracy. However, in this work, we used a camera on the UAV by considering its light weight (33.5 g, oCam-5CRO-U). Although 3D LiDARs are becoming smaller and lighter, they still weigh around 1–1.5 kg, which may be too heavy considering the payload (~700 g) of the regular-sized multirotor UAVs (size around 0.6–0.8 m). The UGV based on a four-wheel skid-steering mobile vehicle is also implemented with an embedded computer (Jetson TX2, Nvidia) with inertial sensors (IMUs), GPS module (HERE2 M8N, HEX), and 2D LiDAR (UTM-30LX, HOKUYO) that detects obstacles in front. Both of the embedded computers on the UGV and UAV are implemented with the robot operating system (ROS). The minimal information (UGV and obstacle positions) is transmitted to the UGV through a

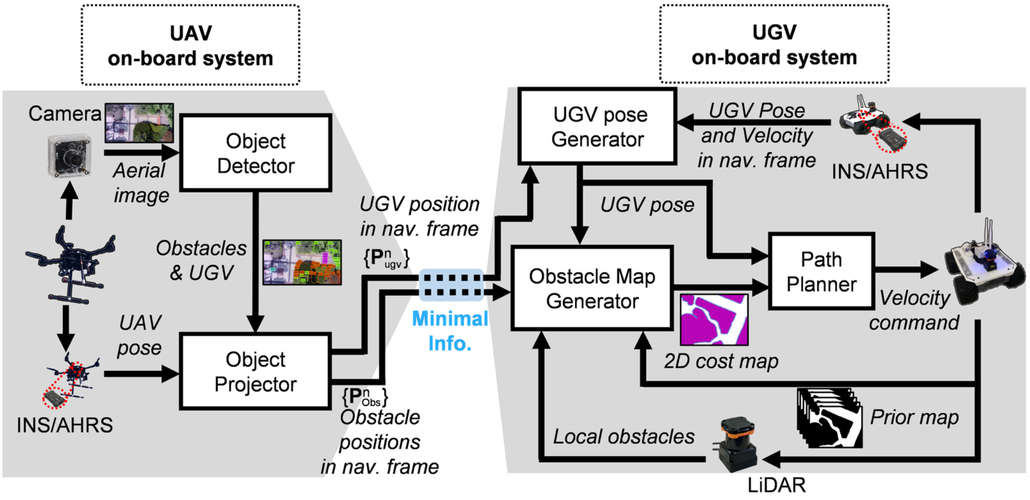

FIGURE 2 Overall framework of the aerial online mapping system. The framework includes the object detector and object projector fully implemented in an UAV on-board system. The UGV pose generator, obstacle map generator, and path planner are fully implemented in an UGV on-board system. The UAV on-board system provides minimal information (UGV and obstacles position in a navigation frame) to the UGV on-board system. UAV, unmanned aerial vehicle; UGV, unmanned ground vehicle.

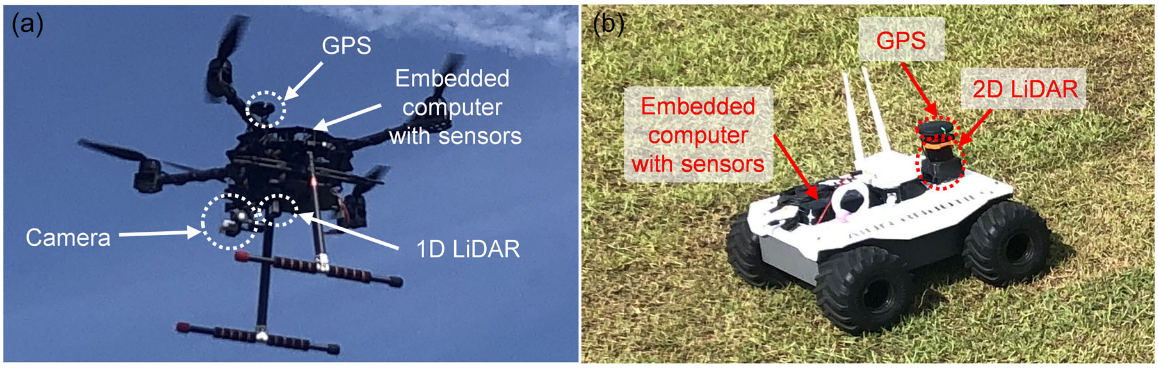
(a)
GPS
Embedded computer with sensors
Camera
1D LiDAR
(b)
GPS
2D LiDAR
Embedded computer with sensors

FIGURE 3 Photographs of the unmanned aerial vehicle and unmanned ground vehicle implemented with embedded computers and various sensors including a camera on a gimbal, LiDARs, GPSs, and inertial sensors. All the computations and processes are done on the on-board processors based on the on-board sensors.

relatively low-powered conventional wireless router (ipTIME N602E, Max. Power: 2.52 W).

2.3 | Experimental environments

Experimental sites were chosen within the campus to demonstrate feasibility of the proposed framework. For practical use of the system, further experimental validations are required in more general environments. Ground flatness at the campus is around within 1 m. At an altitude of 10 m or higher of the UAV, the expected size error of the ground obstacles is about 11%, for example, 11 cm size error of an 1 m obstacle. Although the experimental site includes objects such as trees, bushes, and cones, the framework is designed to build an obstacle map with the bushes and cones only as obstacles because the small UGV can go many cases through under the trees. The UGV can avoid tree trunks within a short range (30 m) with its onboard LiDAR. The UGV received the obstacle information from UAV through the low powered wireless router (ipTIME N602E, Max. Power: 2.52 W, Protocol: IEEE 802.11a), which was configured to have 20 MHz of bandwidth and 2 Mbps of data rate. Based on the updating rate (~4 Hz) of the obstacle map, the movable obstacles (cones) can be reconfigured at relatively low speed. For example, an obstacle moving at 0.4 m/s may have a position error of 10 cm considering the updating rate (~4 Hz).

3 | DESIGN AND PERFORMANCE OF EACH COMPONENT

3.1 | Object detector

The real-time object detection algorithm (Redmon & Farhadi, 2018) is implemented on the embedded computer (Jetson TX2, Nvidia) mounted on the UAV. We trained the detector with a customized aerial image data set captured by aerial vehicles at various sites over four seasons with two resolutions (720p, 4 K). We separately used 1240 images (720p, 4 K) for the training set and 260 images (720p) as the test set. The aerial images are labeled with four classes (cone, bush, tree, and UGV) for the system. Table 1 shows the performance of the implemented object detector as the measured average precisions (APs) for each class. AP50 means the ratio of the detected images that have an intersection over union (IoU, a ratio of overlapped area and union area of the ground-truth bounding box and predicted bounding box) higher than 50%. mAP stands for mean average precision of the classes. Its performance is very high on the UGV and the cones but relatively low on the bushes and trees because the bushes and trees are unstructured objects with repeated textures. The object detector can be further improved or replaced with more general object detectors including a higher number of classes.

TABLE 1 Performance of the real-time object detector (YOLOv3) implemented on the embedded computer (Jetson TX2, Nvidia) with the customized test data set.

Unit: % (Conf.: 50%)	AP_{50}	AP_{75}	$AP_{50:90}$
Cone	89.82	61.15	54.12
Bush	32.1	10.8	13.73
Tree	60.11	28.8	30.6
UGV	98.3	87.2	67.85
mAP	**70**	**46.98**	**41.558**

Note: The data set was collected at the experimental sites during four seasons.

3.2 | Object projector

The detected objects are projected on the ground in the navigation frame (cone: $\mathbf{P}^n_{\text{obj}}$, UGV: $\mathbf{P}^n_{\text{ugv}}$) based on the position ($\mathbf{P}^n_{\text{camera}}$) of the camera and the position ($\mathbf{P}^n_{\text{Image}} - \mathbf{P}^n_{\text{camera}}$) of the detected objects in the image as illustrated in Figure 4a,b and by the following equation:

$$\mathbf{P}^n_{\text{obj}} = \mathbf{P}^n_{\text{camera}} + k\left(\mathbf{P}^n_{\text{Image}} - \mathbf{P}^n_{\text{camera}}\right), \tag{1}$$

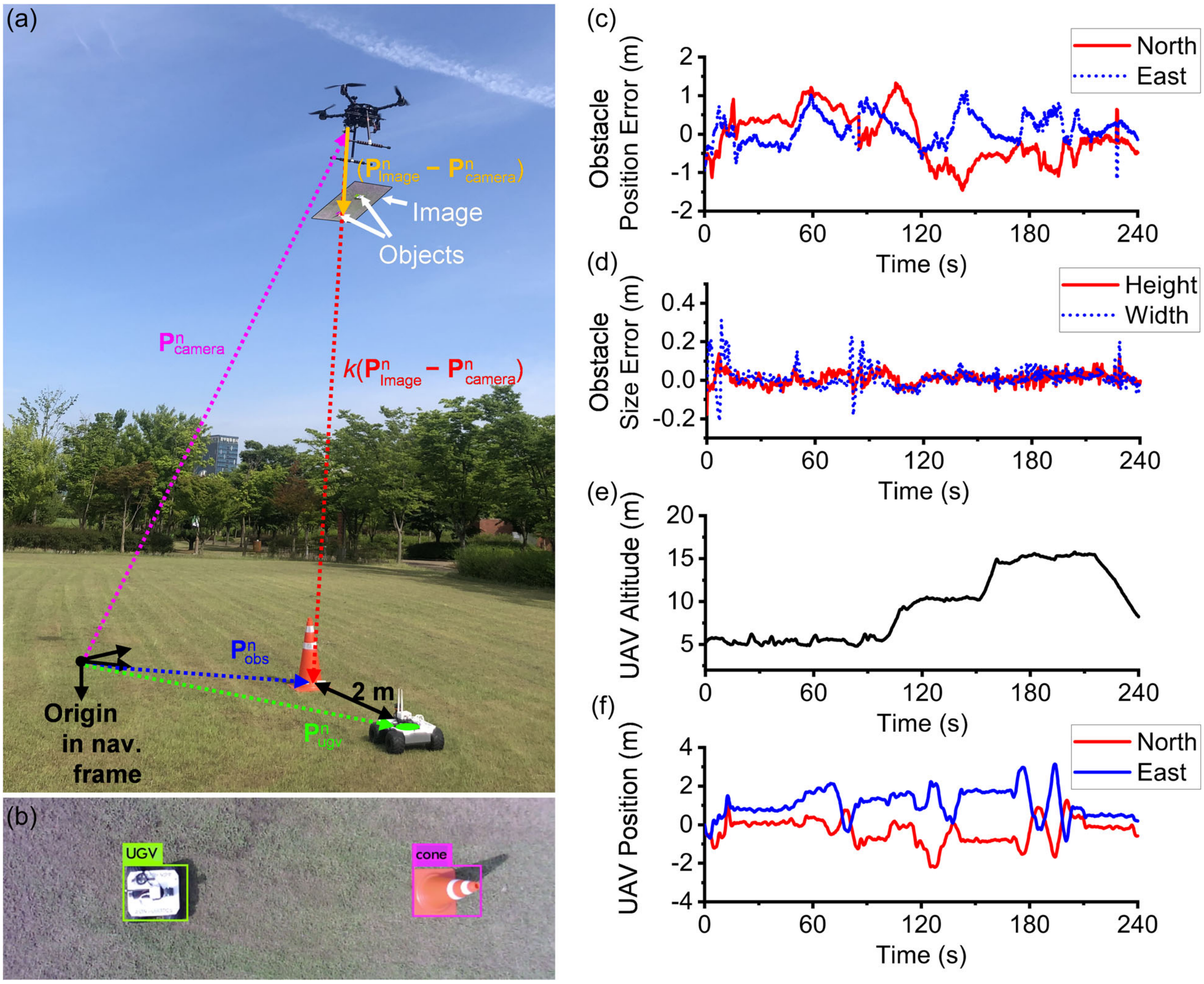

FIGURE 4 Experimental results of the object projector. (a) The positions of the obstacle (cone, P^n_{obs}) and UGV (P^n_{ugv}) on the ground in the navigation frame are determined based on the altitude and (b) the detected obstacles in the captured image. The true distance between the cone and unmanned ground vehicle is 2 m on the ground. (c) Position error (<±1.5 m), (d) size error (<±0.2 m) of the projected obstacle determined from the captured image, (e) while flying the unmanned aerial vehicle up from about 5 to 15 m, and (f) within the lateral bounds of 2.5 m.

where $\mathbf{P}^n_{\text{Image}}$ is the detected object position on the image plane (whose distance from the camera is assumed to be 1) in the navigation frame. $\mathbf{P}^n_{\text{camera}}$ is from the INS/AHRS by the extended Kalman filter. $(\mathbf{P}^n_{\text{Image}} - \mathbf{P}^n_{\text{camera}})$ is determined through the transformation of the pixel position $(\mathbf{P}^i_{\text{obj}})$ on the image by

$$\mathbf{P}^n_{\text{Image}} - \mathbf{P}^n_{\text{camera}} = \mathbf{R}^n_c\left(\mathbf{A}^{-1}\mathbf{P}^i_{\text{obj}}\right), \tag{2}$$

where $\mathbf{R}^n_c$ is the transformation matrix from the camera frame to the navigation frame; $\mathbf{A}^{-1}$ is the inverse matrix of the camera intrinsic matrix depending on the lens and image sensor. k is a scalar determining the length of the projection to the ground, calculated by the measured altitude (Z^n_{camera}) of the camera and the altitude of the image plane (Z^n_{Image}) using the following formula:

$$k = \frac{Z^n_{\text{camera}}}{Z^n_{\text{camera}} - Z^n_{\text{Image}}}. \tag{3}$$

Figure 4c,d show the experimental results of the object projector. The projected obstacle position and size errors are within ±1.5 and ±0.2 m, respectively, when the UAV is flying up from 5 to 15 m (Figure 4e) within the lateral bounds of 2.5 m (Figure 4f).

Figure 5 shows the experimental results of the object projector depending on the elevation ($h = 0$, 1 and 1.7 m as in Figure 5a) of the objects (cones, same size: 380 mm × 380 mm × 700 mm) from the ground. The projected image in the inset of Figure 5a clearly indicates that the most elevated cone looks larger than the other cones although all the cones are the same size. Figure 5b,c show the measured position errors ($|\mathrm{P}_{\text{error}}|$) and size errors ($|\mathrm{S}_{\text{error}}|$) of the projected obstacles when flying the UAV at an altitude from about 5 to 15 m (Figure 5d) within the lateral bound of 5 m (Figure 5e). At the low altitude (5 m) shown in the left image of Figure 5f, the position error and size error at the highest elevation (1.7 m) are larger than the other ones because

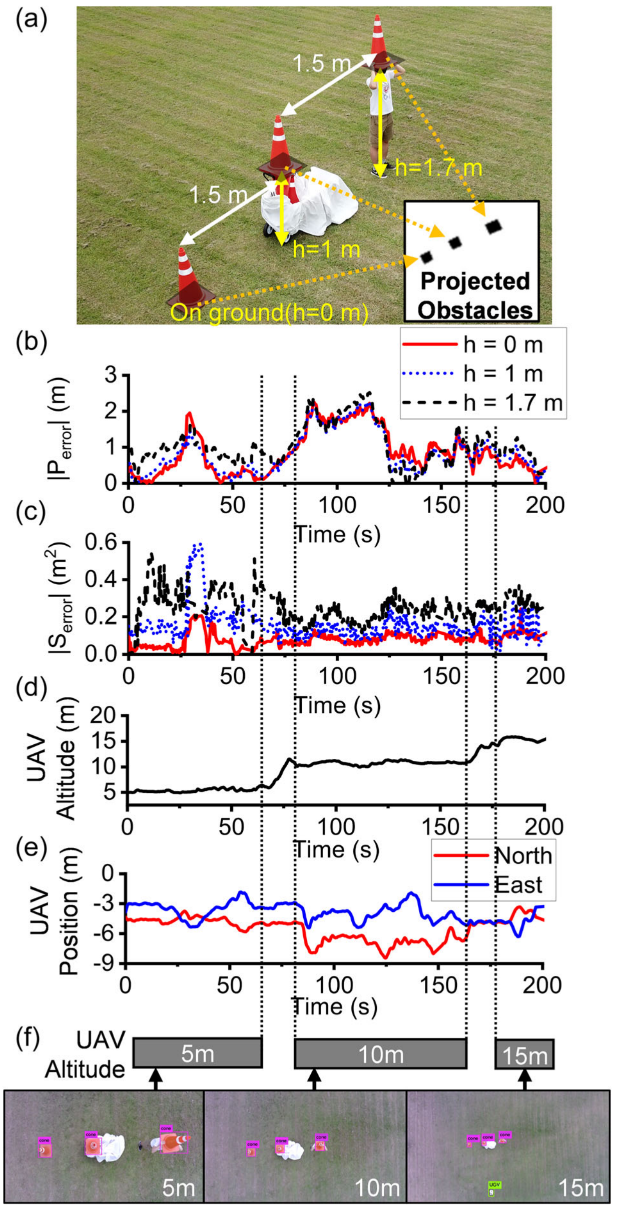

FIGURE 5 Experimental results of the object projector depending on the elevation of the objects. (a) The obstacles (cone: 380 mm × 380 mm × 700 mm) are placed on the ground ($h = 0$), at a height of 1 and 1.7 m. The inset image shows the projected obstacles. The elevated obstacles are projected larger depending on the elevated heights. (b, c) Magnitudes of the position errors ($|P_{\text{error}}|$) and size errors ($|S_{\text{error}}|$) of the projected obstacles at different heights ($h = 0$, 1, and 1.7 m) (d) while flying the UAV from about 5 to 15 m (e) within the lateral bounds of 5 m. The absolute position errors ($|P_{\text{error}}|$) and size errors ($|S_{\text{error}}|$) are within 3 m and 0.6 m^2, respectively. (f) Captured images from the UAV at altitudes of 5, 10 and 15 m. In the left image captured at the low altitude (5 m), the cone placed at the higher elevation (right, $h = 1.7$ m) has a higher size error because of the higher viewing angle.

of a higher perspective error by the viewing angles. However, when the altitude of the UAV is high (≥10 m), the perspective errors of the elevated cone become smaller (80–160 s in Figure 5b,c).

3.3 | Packet for projected objects

For collaboration between UAV and UGV, the packets using in relatively low bandwidth are beneficial. The packet that our UAV transmits consists of the positions ($\mathbf{P}^{n}_{obj}$) and sizes ($\mathbf{S}^{n}_{obj}$) of the projected objects, positions of the UGV ($\mathbf{P}^{n}_{ugv}$), and a number of detected UGVs ($\mathbf{C}_{ugv}$) through the TCP/IP wireless network of ROS as a simple data packet ($\mathbf{D}_{packet}$) as shown below in (4):

$$\mathbf{D}_{packet} = \left[\mathbf{P}^{n}_{obj,1}, \mathbf{S}^{n}_{obj,1}, \ldots, \mathbf{P}^{n}_{obj,\,n}, \mathbf{S}^{n}_{obj,\,n}, \mathbf{P}^{n}_{ugv,1}, \ldots, \mathbf{P}^{n}_{ugv,n}, \mathrm{C}_{ugv}\right], \tag{4}$$

where C_{ugv} is the number of UGVs detected by the object detector. Each data timestamp is recorded at the time when the input image is acquired.

In a low power and low bandwidth wireless network, the TCP based network guarantees more reliable transmission between the UGV and UAV than UDP-based network although the data transmission rate of TCP is lower than that of UDP. The simplified configuration of the packet (several kilobyte), compared to visual feature streaming (more than several megabyte), enables efficient communication with a low-bandwidth network. The configuration and protocol may be more dependable for the proposed aerial online mapping system with a low-powered wireless network in outdoor environments.

3.4 | UGV pose generator

The filtered UGV pose (dotted orange in Figure 6a) by fusing the projected position (P^{n}_{obj}, dotted red, ~4 Hz) from the UAV and inertial navigation data (dotted magenta, 50 Hz) from the UGV provides a higher precision (lower relative position error with the other obstacles) at a higher update rate (25 Hz). The UGV pose observed by the UAV is collected relatively slow (~4 Hz) in a low bandwidth wireless network than the pose data from the onboard sensors of the UGV. We implemented the Kalman filter (Grewal, 2014) as in the dashed box in Figure 6b to generate the UGV pose at the rate (~25 Hz) that is fast enough to control the UGV. The UGV pose by the INS on the UGV is fed to predict the states ($\mathbf{X}^{-} = [\mathbf{P}^{-}, \mathbf{V}^{-}, \Omega^{-}]$) consisting of the position ($\mathbf{P} = [\mathrm{P}_n, \mathrm{P}_e]$), velocity ($\mathbf{V} = [\mathrm{V}_n, \mathrm{V}_e]$) and attitude ($\Omega = [\varphi, \theta, \psi]$) in the navigation frame. From the system dynamic model, the predicted position ($\mathbf{P}^{-}_{k}$) at current state k is shown as the following Equation (5).

$$\mathbf{P}^{-}_{k} = \hat{\mathbf{P}}_{k-1} + \left(\hat{\mathbf{V}}_{k-1} + \hat{\mathbf{V}}^{\mathrm{INS}}_{k-1}\right)\frac{\Delta t}{2} + \mathbf{w}_{k}, \tag{5}$$

where $\hat{\mathbf{P}}_{k-1}$ and $\hat{\mathbf{V}}_{k-1}$ are the estimated position and velocity at the previous state, $\hat{\mathbf{V}}^{\mathrm{INS}}_{k-1}$ is the velocity estimated by INS on the UGV at the previous state, Δt is a time step between previous state and current state, $\mathbf{w}_{k}$ is the zero-mean uncorrelated noise of the predicted position. We assumed that the predicted velocity ($\mathbf{V}^{-}_{k}$) and attitude

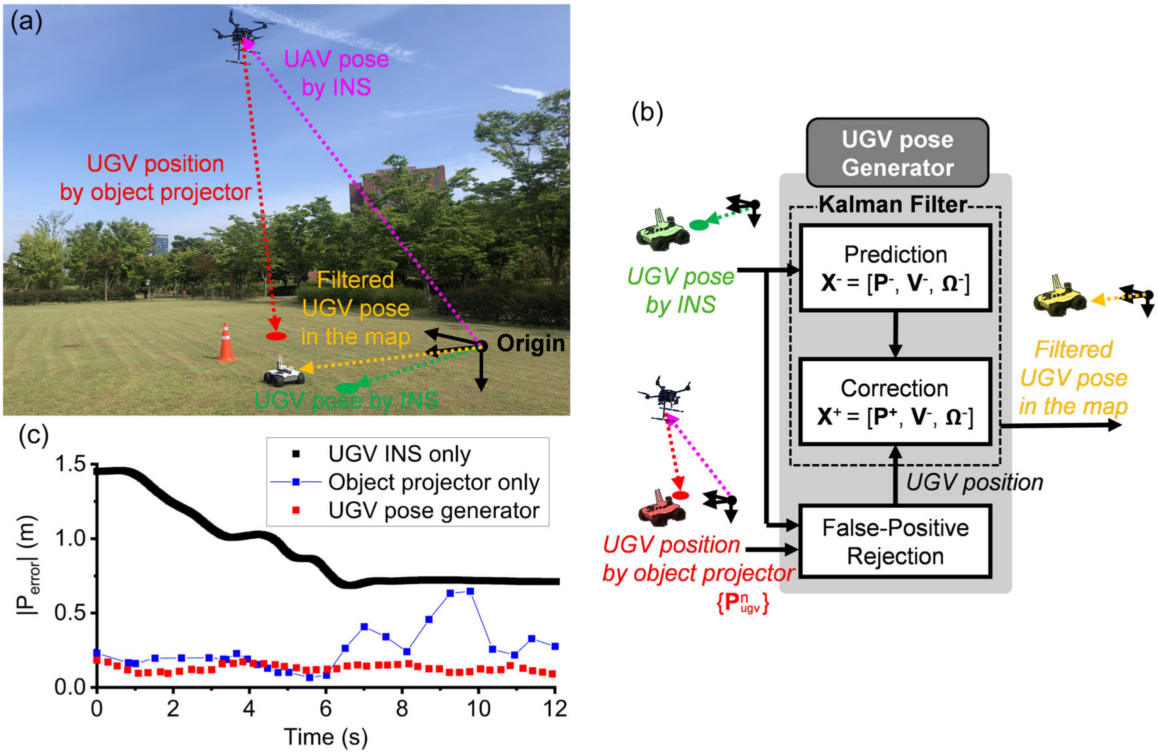

FIGURE 6 Process and experimental results of the UGV pose generator. (a) UGV pose generator generates the filtered UGV pose (position and attitude, orange) in the map by fusing the UGV position (red, estimated by the object projector and UAV INS) and the UGV pose (green, estimated by the UGV INS). (b) Block diagram of the UGV pose generator. The UGV INS pose and UGV position (false rejected) are feed to the Kalman filter to generate the UGV pose (**X** = [**P**, **V**, Ω]) in the map by prediction and correction processes with position (**P**), velocity (**V**), and attitude (Ω). (c) Position errors of the UGV that is stationary at a predetermined position.

($\boldsymbol{\Omega}_k^-$) at current state k are the same as the values estimated by INS with the gaussian noises given by (6)–(7).

$$\mathbf{V}_k^- = \mathbf{V}_k^{INS} + \boldsymbol{w}_k' \tag{6}$$

$$\boldsymbol{\Omega}_k^- = \boldsymbol{\Omega}_k^{INS} + \boldsymbol{w}_k'', \tag{7}$$

where $\mathbf{V}_k^{INS}$ and $\boldsymbol{\Omega}_k^{INS}$ are predicted velocity and attitude by INS on the UGV at current state k, respectively. $\boldsymbol{w}_k'$ and $\boldsymbol{w}_k''$ are the zero-mean uncorrelated noise of predicted velocity and attitude respectively.

The UGV position by the object projector was used to correct the states ($\mathbf{X}^+$). However, in the case of false detection of more than one UGV by the vision algorithm, we implemented the false-positive rejection algorithm to reject false detections of the UGV in two steps by comparing the moving distances of the UGV from the object projector with a physically allowable moving distance at the first step, followed by choosing the closest UGV position to the UGV poses from INS at the second step. The measurement model to correct the UGV position is shown as (8).

$$\mathrm{z}_k = \begin{bmatrix} P_k^N \\ P_k^E \end{bmatrix} + \boldsymbol{v}_k, \tag{8}$$

where P_k^N and P_k^E are the north and east position of the UGV projected by the object projector in the navigation frame, respectively. $\boldsymbol{v}_k$ is measurement noise assumed to be zero-mean Gaussian distribution.

Figure 6c shows the experimental result of the position errors of the stationary UGV by the UGV INS only (black), by the object projected only (blue), and by the UGV pose generator (red) while the UAV was hovering around an altitude of 14–15 m. The UGV position errors (red) by the UGV pose generator are relatively low while the UGV position errors (black) by the INS only are relatively large due to GPS errors (~1.5 m). The UGV position errors (blue) by the object projector also become sometimes high (6–10 s) because of the size and position errors of the bounding boxes and the position errors by the UAV INS. The UGV position data by the object projector (blue) are transmitted to the UGV pose generator. The periods are relatively slow and irregular (70–600 ms) with the low bandwidth wireless network in outdoor environments.

3.5 | Obstacle map generator and path planner

The 2D cost map is generated based on the projected obstacles and locally detected. Figure 7a shows the structure of the obstacle map

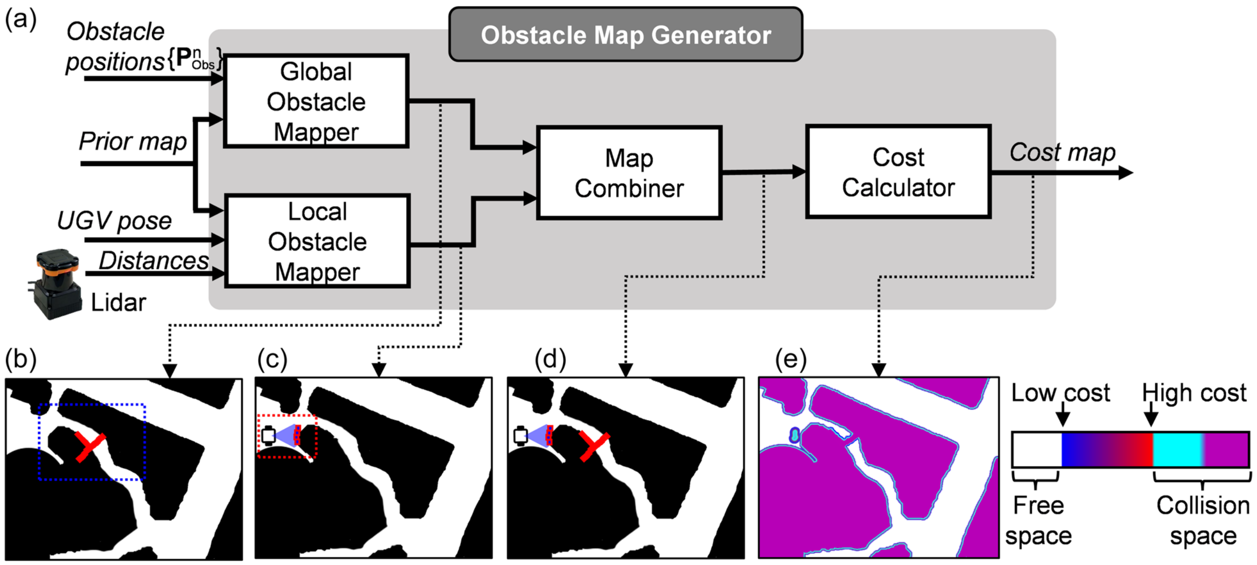

FIGURE 7 Framework of the obstacle map generator. (a) The global obstacle mapper draws global obstacles (red in (b)) acquired by the object projector from the UAV. The local obstacle mapper draws local obstacles (red in (c)) detected by the onboard LiDAR on the UGV. The map combiner combines (d) the global and local obstacles in the map. The cost calculator calculates (e) the 2D cost map from the map.

generator. The prior map is a map that shows obstructed regions (in black) marked based on a publicly available map. As the prior map does not have information about current obstacles, the aerial online mapping process marks global obstacles found from the UAV and local obstacles found from the UGV on the prior map. The global obstacle mapper draws the global obstacles (red) on the prior map (black) with the projected obstacles from the UAV as in Figure 7b, and the local obstacle mapper draws the local obstacles (red) detected from the 2D LiDAR mounted on the UGV as in Figure 7c. The UGV pose is used to calculate the local obstacle positions. The map combiner merges the global obstacle map and the local obstacle map (Figure 7d). Finally, the cost calculator allocates the costs to the cells of the grid-based 2D cost map (resolution: 0.05 m) depending on the distances to the obstacles shown in Figure 7e. The cost color bar is shown in Figure 7e. We generated the costs of the cells to increase exponentially when it comes closer to the obstacles with an assumption that the UGV is safe as long as the distance from the obstacle is higher than the radius of the UGV (Lu et al., 2014). Distances of the cells from the nearest obstacle are allocated based on the Euclidean distance field. The costs of the cells are generated based on the following equation.

$$C_{x,y} = \begin{cases} 254, & \text{when } D = 0, \\ 253, & \text{when } D \le R, \\ 252 \cdot e^{-\alpha(D-R)}, & \text{when } D > R, \end{cases} \tag{9}$$

where $C_{x,y}$ is the cost of a cell at (x, y), D is a Euclidean distance of the cell from an obstacle, R is a radius of the UGV, and the maximum cost of the 2D occupancy grid map is 254. α is a scaling factor determining an amount of cost reduction with distance. If α is low, the cost of the cell around the obstacle will decrease slowly with distance. In that case, the path planner generates a path that is farther away from the obstacle. We empirically set α as 3. The cost map clearly has a high cost in the passage around the middle due to the detected obstacles.

Based on the cost map, we customized and implemented the path planner (Move Base; Marder-Eppstein, 2018) consisting of a global and local path planner in the aerial online mapping system. The path planners are operated on UGV. The global path planner searches the global path on the cost map with the lowest cost from the start point to the goal through A* algorithm (Doran & Michie, 1966). The desired forward linear velocity and the steering angular velocity of the UGV are generated based on the generated cost within 3 m at a relatively high rate (10 Hz) by using the DWA algorithm (Fox et al., 1997).

4 | OUTDOOR DEMONSTRATIONS

We applied all the hardware (GPS, embedded computers, inertial sensors, camera, 1D and 2D LiDARs), software (C and C++ in ROS), and algorithm (object detector, projector, pose generator, map generator and path planner) components of the aerial online mapping system to a UAV to demonstrate the autonomous driving of an UGV through a path that takes into account undetectable obstacles ahead of the approaching UGV.

4.1 | Aerial online mapping with obstacles setups

Figure 8 shows the experimental setups and results for the autonomous driving of an UGV with the aerial online mapping

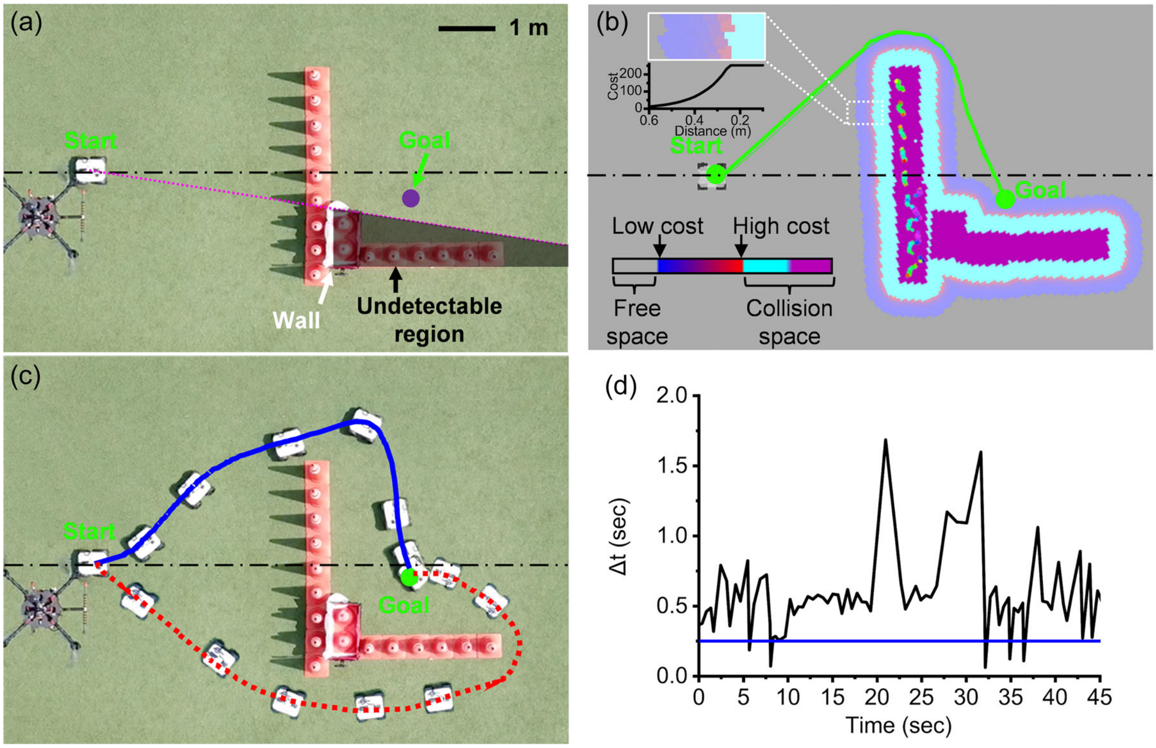

FIGURE 8 Experimental results of aerial online mapping and autonomous driving with obstacles setup on a football field. (a) Aerial view of the UAV, UGV, and obstacles (cones and a white wall) from an additional filming drone. The UGV starts to move from the center line (dashed-dot line), and the goal point is on the lower side of the center line. The obstacles (cones on the right) in the shaded region are undetectable from the LiDAR on the UGV as the area is blocked by the wall. (b) Cost map and generated path (green line). The generated path is on the upper side of the center line although the goal is on the lower side because the UGV uses the aerial online mapping system that detects the blocked obstacles by the wall. The inset plot shows the generated cost corresponding to the colors depending on distance to the obstacles. (c) Actual trajectories of the UGV from the time-lapse images. The blue- and red-dotted lines indicate the trajectories with and without the aerial online mapping. (d) Update periods of the UGV positions by the object projector. The period we set is 0.25 Section (4 Hz) as noted by the blue line but the measured update time (average: 0.5 s, max: 1.68 s) is longer than that due to communication lag. UAV, unmanned aerial vehicle; UGV, unmanned ground vehicle.

system with an UAV on a football field. We setup the obstacles in a kind of "L" shape with cones and a white cloth wall to make the horizontally aligned cones (shaded) undetectable by the LiDAR of the UGV at the starting point shown in Figure 8a. Without the aerial online mapping system, the UGV starting from the center line (dashed-dot line) is supposed to go around the lower cones because the goal (purple point) is located lower than that of the center line shown in Figure 8a. However, with the aerial online mapping system, the UAV detects all the obstacles from the air and provides the information to the UGV to generate the cost map and the shorter path (green line) shown in Figure 8b. Figure 8c shows the actual trajectories of the UGV with (blue line) and without (red dotted line) the aerial online mapping system. The time-lapse images of the UGV were taken every 5 s. With the aerial online mapping, the UGV takes the shorter path (blue line) because it can consider the obstacles behind the wall although the update time (Δt: average: 0.5 s, max: 1.68 s in Figure 8d) of the projected UGV position is longer than that we set (0.25 s for 4 Hz, blue line in Figure 8d) because of weak communication link with the low-powered wireless networks.

4.2 | Aerial online mapping in a park

We demonstrated the aerial online mapping system in a more real environment in a park on campus as shown with the aerial view in Figure 9a. The obstacles (cones) were placed around the middle between the starting point and the goal point. The prior map including the shaded nondrivable regions in Figure 9b can be acquired from free aerial or satellite images. Because the prior map does not include the obstacles placed or the bushes grown, the path is generated through the shortest path (green dotted line). The UGV proceeded through the shortest path but came back and detoured around the upper path (red dotted line) after detecting the obstacles with the onboard LiDAR.

The aerial online mapping system, however, detected the objects on the ground from the real-time aerial image taken by the UAV (Figure 9c) and then projected the obstacles (cones and bushes) on the prior map as noted in red in Figure 9d. In the experiments, the UAV was manually controlled at an altitude of 10 m or higher. Figure 9e shows the cost map and the generated path (green line) through the upper region as the cost map includes the detected obstacles that block the shortest path. Figure 9f shows the actual driven path (red line) of the

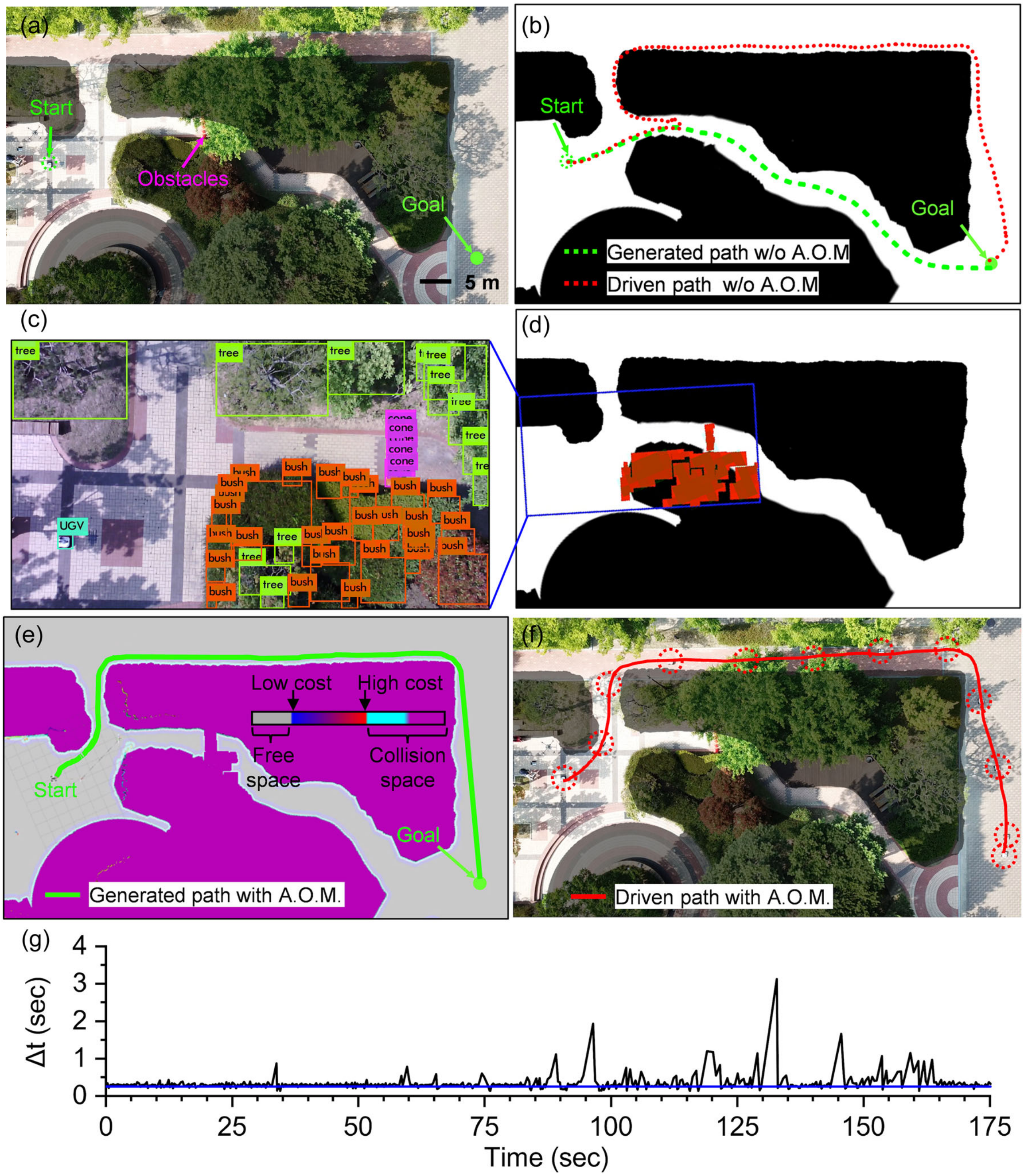

FIGURE 9 Autonomous driving of the UGV with the aerial online mapping system in the park. (a) Aerial view of the park. The obstacles (cones) are placed around the middle between the start and goal points. (b) Generated (green dotted line) and driven (red dotted line) paths without the aerial online mapping. (c) Objects detected with the aerial online mapping from the UAV. (d) Generated obstacle map on the UGV. (e) Generated path (green line) based on the cost map. (f) Time-lapse images (dotted circles) of the UGV driving through the generated path without going through the blocked path. (g) Update time of the projected UGV positions from the UGV pose generator. The period we set is 0.25 s (blue line). The UGV generates and proceeds through the desired path although the update time periods are irregular (average: 0.34 s, max: 3.1 s) due to low-powered wireless network. UAV, unmanned aerial vehicle; UGV, unmanned ground vehicle.

UGV. The time-lapse image includes the UGV (dotted circles) captured every 15 s. The UGV did not have to go and come back through the shortest path because the aerial online mapping system already considered the obstacles in the cost map even in the outdoor environment with the communication network updating the projected UGV positions irregularly. The UGV reliably moves through the efficient path (red line in Figure 9f) although the measured update time (Δt, Figure 9g) of the projected UGV positions is about 0.34 s on average and often spiking (max: 3.1 s) because of low-powered communication network.

5 | CONCLUSION

The results presented here indicate that the aerial online mapping system, which updates information about obstacles undetectable by the onboard sensors mounted on an UGV, is effective in the path generation of an autonomous UGV by detecting obstacles by the onboard camera in the UAV with conventional low-powered communication networks. All the components including the object detector, projector, UGV pose generator, obstacle map generator, and path planner can be implemented on the relatively small, low-cost onboard systems of the UAV and UGV. The system design enables the UGV to work continuously by predicting obstacle locations even with poor communication network from the UAV by minimizing the size of data packets containing processed size and location only and by implementing the Kalman filter-based UGV pose generator to use predicted obstacle positions continuously. Experiments of the implemented aerial online mapping system in the outdoor environments including the on-campus park demonstrated the feasibility of this approach. The realized system could update the local map to efficiently generate the paths of the UGV in a relatively low-powered network environment in which it is challenging to transmit large 3D point clouds, keyframe images, and generated local maps in real-time. It should be noted that the experiments demonstrate feasibility of the proposed framework in the relatively simple environment on campus. For practical use, the framework requires further studies to validate the performance within more general environments with dynamic obstacles such as pedestrians on nonflat grounds. Limited field of view of the UAV camera and limited communication distance between the UGV and UAV may affect the online mapping performance although these limitations can be relieved by active scanning the areas with the UAV. The proposed framework can be improved for more general environments after learning processes in the object detector with the data set containing diverse classes. The size error of the obstacles caused by using a regular camera can be reduced by using other types of sensors such as ultra wide band modules, depth cameras, or 3D LiDARs that can measure a distance more precisely. Recently released affordable and light-weight distance sensors may help improving accuracy of the obstacle mapper in more complex environments if they are used together with a camera. The results reported here should be useful for diverse applications requiring heterogeneous cooperation between the UGV and UAV for efficient tasks such as delivery, exploration, search and rescue, disaster monitoring, and maintaining maps.

ACKNOWLEDGMENTS

This work was supported by the GIST-MIT Research Collaboration grant funded by the GIST and the Policelab 2.0 grant (210121M04) funded by the Korea Institute of Advancement for Police-technology in 2021.

DATA AVAILABILITY STATEMENT

The data that support the findings of this study are available from the corresponding author upon reasonable request.

ORCID

Jehong Lee https://orcid.org/0000-0001-8797-9618

Jongho Lee http://orcid.org/0000-0003-0398-4220

REFERENCES

Adamides, G., Katsanos, C., Constantinou, I., Christou, G., Xenos, M., Hadzilacos, T. et al. (2017) Design and development of a semi-autonomous agricultural vineyard sprayer: human–robot interaction aspects. *Journal of Field Robotics*, 34(8), 1407–1426.

Balta, H., Bedkowski, J., Govindaraj, S., Majek, K., Musialik, P., Serrano, D. et al. (2017) Integrated data management for a fleet of search-and-rescue robots. *Journal of Field Robotics*, 34(3), 539–582.

Bouman, A., Ginting, M.F., Alatur, N., Palieri, M., Fan, D.D., Touma, T. et al. (2020) Autonomous spot: long-range autonomous exploration of extreme environments with legged locomotion. In: *2020 IEEE/RSJ international conference on intelligent robots and systems (IROS)*. IEEE, pp. 2518–2525.

Brooks, C.A. & Iagnemma, K. (2012) Self-supervised terrain classification for planetary surface exploration rovers. *Journal of Field Robotics*, 29(3), 445–468.

Cantieri, A., Ferraz, M., Szekir, G., Antônio Teixeira, M., Lima, J., Schneider Oliveira, A. et al. (2020) Cooperative UAV–UGV autonomous power pylon inspection: an investigation of cooperative outdoor vehicle positioning architecture. *Sensors*, 20(21), 6384.

Christie, G., Shoemaker, A., Kochersberger, K., Tokekar, P., McLean, L. & Leonessa, A. (2017) Radiation search operations using scene understanding with autonomous UAV and UGV. *Journal of Field Robotics*, 34(8), 1450–1468.

Delmerico, J., Mintchev, S., Giusti, A., Gromov, B., Melo, K., Horvat, T. et al. (2019) The current state and future outlook of rescue robotics. *Journal of Field Robotics*, 36(7), 1171–1191.

Delmerico, J., Mueggler, E., Nitsch, J. & Scaramuzza, D. (2017) Active autonomous aerial exploration for ground robot path planning. *IEEE Robotics and Automation Letters*, 2(2), 664–671.

Doran, J. E., & Michie, D. (1966) Experiments with the graph traverser program. *Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences*, 294(1437), 235–259.

Dubé, R., Cramariuc, A., Dugas, D., Sommer, H., Dymczyk, M., Nieto, J. et al. (2019) SegMap: segment-based mapping and localization using data-driven descriptors. *The International Journal of Robotics Research*, 39(2–3), 339–355.

Ebadi, K., Chang, Y., Palieri, M., Stephens, A., Hatteland, A., Heiden, E. et al. (2020) LAMP: large-scale autonomous mapping and positioning for exploration of perceptually-degraded subterranean environments. In: *2020 IEEE international conference on robotics and automation (ICRA)*. IEEE, pp. 80–86.

Faessler, M., Fontana, F., Forster, C., Mueggler, E., Pizzoli, M. & Scaramuzza, D. (2016) Autonomous, vision-based flight and live dense 3D mapping with a quadrotor micro aerial vehicle. *Journal of Field Robotics*, 33(4), 431–450.

Fankhauser, P., Bloesch, M., Krüsi, P., Diethelm, R., Wermelinger, M., Schneider, T. et al. (2016) Collaborative navigation for flying and walking robots. In: *2016 IEEE/RSJ International conference on intelligent robots and systems (IROS)*. IEEE, pp. 2859–2866.

Fox, D., Burgard, W. & Thrun, S. (1997) The dynamic window approach to collision avoidance. *IEEE Robotics & Automation Magazine*, 4(1), 23–33.

Grewal, M.S. (2014) *Kalman filtering: theory and practice with MATLAB*. Wiley-IEEE Press.

Guastella, D.C., Cantelli, L., Melita, C.D. & Muscato, G. (2017) A global path planning strategy for a UGV from aerial elevation maps for disaster response. In: *Proceedings of the 9th international conference on agents and artificial intelligence*.

Kolvenbach, H., Wisth, D., Buchanan, R., Valsecchi, G., Grandia, R., Fallon, M. et al. (2020) Towards autonomous inspection of concrete deterioration in sewers with legged robots. *Journal of Field Robotics*, 37(8), 1314–1327.

Krátký, V., Petráček, P., Báča, T. & Saska, M. (2021) An autonomous unmanned aerial vehicle system for fast exploration of large complex indoor environments. *Journal of Field Robotics*, 38(8), 1036–1058.

La, H.M., Gucunski, N., Dana, K. & Kee, S.H. (2017) Development of an autonomous bridge deck inspection robotic system. *Journal of Field Robotics*, 34(8), 1489–1504.

Lakas, A., Belkhouche, B., Benkraouda, O., Shuaib, A. & Alasmawi, H.J. (2018) A framework for a cooperative UAV-UGV system for path discovery and planning. In: *2018 international conference on innovations in information technology (IIT)*. IEEE, pp. 42–46.

Li, S., Zheng, P., Fan, J. & Wang, L. (2021) Towards proactive human robot collaborative assembly: a multimodal transfer learning-enabled action prediction approach. *IEEE Transactions on Industrial Electronics*, 69(8), 8579–8588.

Ljungqvist, O., Evestedt, N., Axehill, D., Cirillo, M. & Pettersson, H. (2019) A path planning and path-following control framework for a general 2-trailer with a car-like tractor. *Journal of Field Robotics*, 36(8), 1345–1377.

Lu, D.V., Hershberger, D. & Smart, W.D. (2014) Layered costmaps for context-sensitive navigation. In: *2014 IEEE/RSJ international conference on intelligent robots and systems*. IEEE, pp. 709–715.

Marder-Eppstein, E. (2018) Move base. Available from: https://github.com/ros-planning/navigation/tree/kinetic-devel/move_base

Merriaux, P., Dupuis, Y., Boutteau, R., Vasseur, P. & Savatier, X. (2018) Robust robot localization in a complex oil and gas industrial environment. *Journal of Field Robotics*, 35(2), 213–230.

Milella, A., Reina, G. & Underwood, J. (2015) A self-learning framework for statistical ground classification using radar and monocular vision. *Journal of Field Robotics*, 32(1), 20–41.

Otsu, K., Tepsuporn, S., Thakker, R., Vaquero, T.S., Edlund, J.A., Walsh, W. et al. (2020) Supervised autonomy for communication-degraded subterranean exploration by a robot team. In: *2020 IEEE aerospace conference*. IEEE. pp. 1–9.

Pani, A., Mishra, S., Golias, M. & Figliozzi, M. (2020) Evaluating public acceptance of autonomous delivery robots during COVID-19 pandemic. *Transportation Research Part D: Transport and Environment*, 89, 102600.

Peterson, J., Chaudhry, H., Abdelatty, K., Bird, J. & Kochersberger, K. (2018) Online aerial terrain mapping for ground robot navigation. *Sensors*, 18(2), 630.

Peterson, J., Li, W., Cesar-Tondreau, B., Bird, J., Kochersberger, K., Czaja, W. et al. (2019) Experiments in unmanned aerial vehicle/unmanned ground vehicle radiation search. *Journal of Field Robotics*, 36(4), 818–845.

Potena, C., Khanna, R., Nieto, J., Siegwart, R., Nardi, D. & Pretto, A. (2019) AgriColMap: aerial-ground collaborative 3D mapping for precision farming. *IEEE Robotics and Automation Letters*, 4(2), 1085–1092.

Redmon, J. & Farhadi, A. (2018) Yolov3: an incremental improvement. In: *Computer vision and pattern recognition*. Berlin/Heidelberg, Germany: Springer, 1804, pp. 1–6.

Ren, R., Fu, H., Xue, H., Li, X., Hu, X. & Wu, M. (2021) LiDAR-based robust localization for field autonomous vehicles in off-road environments. *Journal of Field Robotics*, 38(8), 1059–1077.

Schmuck, P. & Chli, M. (2017) Multi-uav collaborative monocular slam. In: *2017 IEEE international conference on robotics and automation (ICRA)*. IEEE. pp. 3863–3870.

Schmuck, P. & Chli, M. (2018) CCM-SLAM: robust and efficient centralized collaborative monocular simultaneous localization and mapping for robotic teams. *Journal of Field Robotics*, 36(4), 763–781.

Song, Y. (2019) *Inverse reinforcement learning for autonomous ground navigation using aerial and satellite observation data*. Doctoral dissertation, Master's thesis, Pittsburgh, PA.

How to cite this article: Lee, J., Lim, J., Pyo, S. & Lee, J. (2023) Aerial online mapping on-board system by real-time object detection for UGV path generation in unstructured outdoor environments. *Journal of Field Robotics*, 40, 1754–1765. https://doi.org/10.1002/rob.22213