International Journal of Control, Automation and Systems 16(X) (2018) 1-10
http://dx.doi.org/10.1007/s12555-016-0742-y
ISSN:1598-6446 eISSN:2005-4092
http://www.springer.com/12555

Generating Homogeneous Map with Targets and Paths for Coordinated Search

Hyeun Jeong Min

Abstract: This work presents a new solution for coordinated search with a team of heterogeneous robots executing a time-critical mission. It is challenging to specify and represent search locations (targets) in known but dynamic environments as well as to find robotic paths to visit the locations. We propose a technique to construct an information map that includes locations of uncertain targets, and generate optimal paths. We especially focus on combining a satellite map that has global coordinates with local images gathered from an aerial robot. Specific targets are represented on a homogeneous coordinate system, so that different types of robots, capable to gather necessary information, may cooperatively conduct a mission. Once a homogeneous map is constructed, a centralized path-finding algorithm can be applied. Our path-finding algorithm is to choose a set of paths, suggesting a proper number of robots along with their initial locations. In our work, robots can independently travel search locations, which may have dynamics or changes, but collaboratively cover all target locations. Through the experiments with real robotic platforms, we validate the generation of a map including targets and a choice of paths, and compare with existing algorithms.

Keywords: Coordinated search, homogeneous map, path planning.

1. INTRODUCTION

There exist growing interests in the use of robots for search and rescue applications. Contaminants in an area that needs to be detoxified, a bomb installed in a metropolitan area that needs to be defused, amongst others are representative examples in time-critical situations. In these situations, searching for specific objects or sources is challenging. Essential factors are the choice of some locations for inspection, the number of robots, their start locations, and the choice of their paths. Finding an optimal number of robots is an open problem in search and rescue robotics [1]. Selecting locations in which robots need to search with their respective sensors is also challenging. We consider the cooperation of heterogeneous robots between an Unmanned Aerial Vehicle (UAV)/UAVs and an Unmanned Ground Vehicle (UGV)/UGVs.

UAVs are fast in motion and can easily gather plenty of information. However, they may not be suitable for thorough inspection due to difficulties in extracting detailed information. UGVs, on the other hand, are useful for complete search, but target localization is required. Both have pros and cons: we, therefore, cooperate them together. For cooperation, we need a homogeneous system. A satellite map provides global coordinates for specific objects such as buildings, cars, and so on. However, it may not have enough high-resolution or is sometimes occluded. It may not give information about instant changes of movements, occluded objects, changes in before and after a disaster, and so on. Small drones (UAVs) can compensate lack of information acquired from a satellite [2]. This motivates us to combine a sequence of images acquired from UAV(s) with a satellite map. In our coordinated search, UAVs build a map that includes specific targets including instant changes, while UGVs (search robots) use the map to control and visit required targets.

For a target search problem, UAVs have been considered with their camera system, in order to maximize a covered area for cooperative search [3,4]. Multiple robots are utilized to optimize their controllability by combining local images [5]. A search strategy for UAVs is attempted to reveal priority of a target area in an unknown environment [6]. A problem of searching targets with UAVs requires localization of robots in an environment. There have been a lot of endeavors in localization and mapping in robotics. A 3D mapping method using UAVs provides necessary information for searching targets in an unknown environment [7]. Visual odometry along with the RANSAC (RANdom Sample Consensus) algorithm has been utilized [8–10].

Manuscript received November 29, 3016; revised May 23, 2017; accepted July 22, 2017. Recommended by Associate Editor Kyoungchul Kong under the direction of Editor Fuchun Sun. This material is based upon work supported by the BK21 plus program through the National Research Foundation (NRF) funded by the Ministry of Education of Korea.

Hyeun Jeong Min is with Computer and Telecommunication Engineering Division, Yonsei University, 1 Yonseidae-gil, Wonju, Gangwon-do, 26493, Korea (e-mail: solusea@yonsei.ac.kr).

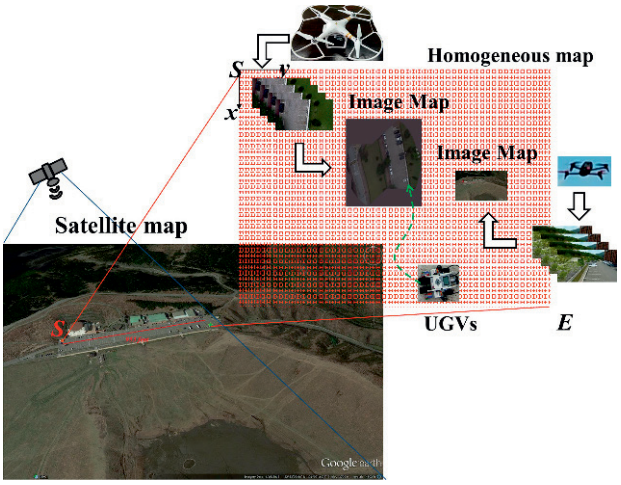


Fig. 1. Homogeneous map. It is combined with the images taken from drones and a satellite map.

For exploration by UAVs, a state-of-the-art vision-based approach is a generation of a 3D map [11]. It, however, tolerates a lot of computational load. Scaramuzza *et al.* used a global coordinate frame for a swarm of small flying robots [10, 12]. In terms of control laws of aerial robots, a 3-D image map is merged by local images acquired from the robots. The authors suggested an optimal control method for coverage; a target area is assigned to the nearest robot, a greedy method similar to the Nearest Neighbor (NN) algorithm [13]. Much research has exploited map construction, but robots that construct a map have utilized the map, so localization of the same series of robots is essential. On the other hand, we consider heterogeneous robots that need to share information; instead of providing a map to aerial robots (UAVs), we make search robots (UGVs) that have different perspective to use targets' information on the map.

To construct a homogeneous map, we combine consecutive images gathered by an UAV. Using SIFT features has increased reliability in image processing [14, 15]. A GPS-based localization method using SIFT features has been applied on Google Maps [16]. In our work, we adopt a SIFT-based method for our image processing. The use of GPS (Global Positioning System) has been increasing in localization [16–18]. In this work, we merge a sequence of images gathered by an UAV in order to transform targets' location from local coordinates to global coordinates for a coordinated mission.

Fig. 1 shows framework of our work describing a homogeneous map. It shows stacks of images taken from a camera system mounted on drones and a combined image map represented on transformed coordinates. It is represented on the grid map, which may contain dynamic targets. Targets can be either manually or autonomously selected, and are used to generate search paths for UGVs. In this way (*i.e.*, making a map from an UAV), a centralized method for search becomes realistic in a dynamic environment. The combined maps (Image Maps) represent targets on a local coordinate system, while the grid map reflects the global coordinates. We also present an efficient path-finding algorithm for multiple robots for coverage. Our path-finding algorithm is a centralized method in terms of the use of the constructed map with known targets.

Sensor-based coverage problem has been actively investigated in sensor network [19–21]. The objective of the problem is to maximally cover the entire environment through sensors. It has been conventionally solved to minimize the overlapped zones of sensing boundaries. Specific problems have been how to deploy sensors or which sensors need to be on/off in terms of optimization purposes [22, 23]. Multi-robot coverage problems, on the other hand, are interested in covering specific candidate regions by visiting them with robots. It has several advantages: i) it is robust to hardware limitation such as recharging exhausted batteries of sensors for long-term surveillance or to pick malfunctioning sensors up, ii) the mobility of robots assists to expedite search/rescue missions in hazardous areas, and iii) no pre-deployed sensors are required in advance. Regarding the mobility of robots, how to control robots in a coverage problem has been actively researched [19, 21]. Recent research has exploited in a robotic coverage problem constrained by batteries or fuels [24], but the number of charging robots was given.

Our multi-robot search is to minimize the maximum path cost among all chosen paths. The choice of a start location for multiple robots and the base number of robots affects to the optimization of the coverage cost [25]. In addition, the choice of the minimum search cost affects to the number of robots. In this work, we assume that robots may start at certain distinct locations. We present an algorithm that reduces the maximum path cost for the given conditions. Each travel cost for all chosen robots should be bounded to the maximum path cost while the union of targets in all paths is the entire targets. We represent a heuristic path-finding algorithm, which selects a set of paths, the number of robots, and their initial locations.

Our contributions are as follows: i) We propose a data association framework combining instant sensing information and satellite data for a coordinated search (UAVs and UGVs), so that changes of targets in an environment can be considered. ii) Our method creates a homogeneous map that is suitable for different types (in control laws or perspectives) of robots. iii) Our algorithm provides a solution of efficient paths for multiple robots.

2. PROBLEM DESCRIPTION

For coordination of a team of heterogeneous robots, especially when they have different resources and capabilities in sensing and actuation, localization of targets is necessary. UAVs gather information about changes of the en-

vironments in order to create a map with locations of targets. In this way, they act as active sensors, while multiple UGVs conduct their search mission as actuators. Note that targets can be created either autonomously or manually on the map. Our problems to combine images gathered from an UAV with a satellite map and to find a set of paths are discussed in Subsections 2.1 and 2.2, respectively.

2.1. Homogeneous map

Fig. 2 shows two reference frames, one from a satellite map and the other one from an UAV. Multiple targets, landmarks, or UGVs can be identified by using a dynamic UAV system. Using specified landmarks, we can transform a coordinate system in order to represent required targets on a homogeneous map. We estimate locations of targets (t_i) by transforming them from local image coordinates to a global homogeneous frame.

Our objective for the construction of a homogeneous map is to minimize the difference of the real target locations from the transformed locations acquired by a sequence of images from an UAV. It is represented as follows:

$$\min \| {}^{G}\xi - {}^{G}\hat{\xi} \|, \tag{1}$$

where ${}^{G}\xi = [{}^{g}x_1, \cdots, {}^{g}x_N]^T$ and ${}^{G}\hat{\xi} = [{}^{g}\hat{x_1}, \cdots, {}^{g}\hat{x_N}]^T$ are the vectors representing true locations of targets and the transformed locations of the targets acquired from the image measurements, respectively. Here, ${}^{g}x_i = [g_i(a)\ g_i(l)]^T$ represents the global coordinates of latitude and longitude of the i^{th} target, and N is the number of targets. Also, ${}^{g}\hat{x_i}$ is transformed from ${}^{c}x_i$, where ${}^{c}x_i = [u_i\ v_i]^T$ is the local coordinates on the combined image of the i^{th} target.

2.2. Multi-robot paths

We represent locations of targets in a homogeneous coordinate system, so that search robots control by themselves to visit the desired locations (targets) which are transformed from local images by an UAV. Let x_i in $\mathbb{R}^2$ be the i^{th} target, and $\xi = \{x_1, \cdots, x_N\}$ be a set of all targets, where N is the number of targets. For simplicity, we call the connection of two targets as an edge, and the connection of all elements in the j^{th} subset of ξ_j as a path.

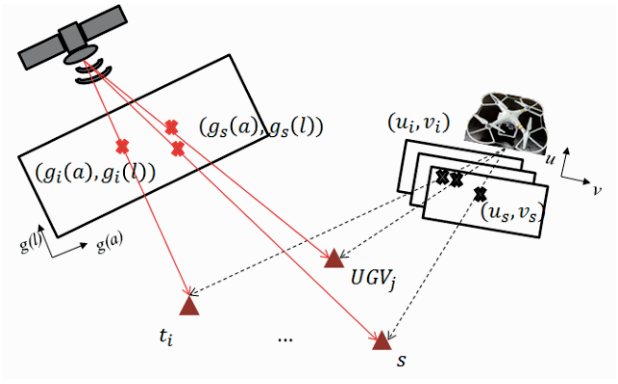


Fig. 2. Combining coordinate systems of the measurements representing the i^{th} target, the j^{th} UGV, and the landmark s. A satellite map has the latitude ($g(a)$) and longitude ($g(l)$), while a local image has the width (u) and height (v).

In order to find a set of paths for multiple robots, our objective is to reduce the maximum path cost and to choose an optimal set of start locations for unknown number of robots while maintaining all costs smaller than or equal to the chosen maximum cost as in (2a). Let us note that K be the chosen maximum cost. The conditions for the objective are given in (2a) $\sim$ (2d). For a given set of targets along with their edges, it is to find subsets of an ordered targets satisfying that the union of all subsets is ξ. Assuming that no zero cost exists for chosen robots, our objective also provides the number of robots while the path costs are bounded to the optimized cost (K).

$$\begin{aligned}
\text{minimize} \quad & K = \max cost(\pi_i), \forall i = 1, \cdots, R && \text{(2a)}\\
\text{subject to} \quad & \cup_{r=1}^{R} \pi_r = \xi, && \text{(2b)}\\
& \boldsymbol{x}_s^{(r_i)} \neq \boldsymbol{x}_d^{(r_j)}, \forall r_i \neq r_j, && \text{(2c)}\\
& 0 < cost(\pi_r) \leq K, \forall r = 1, \cdots, R. && \text{(2d)}
\end{aligned}$$

We want to find robotic paths π_r such that $cost(\pi_r)$ is bounded to the maximum cost in (2a) for $\forall r = 1, \cdots, R$, where R is the chosen number of robots which is initially undecided. Here, $cost(\pi_r)$ is the path cost corresponding to the robot path π_r, and $\boldsymbol{x}_s^{r_i}$ and $\boldsymbol{x}_d^{r_i}$ are the start and the destination targets in the r_i^{th} path, respectively.

3. COORDINATED MAP CONSTRUCTION

This section deals with the proposed algorithm, generating a homogeneous map, by combining local images with a satellite map. Fig. 3 shows the process for our robotic cooperation. Once a combined map is constructed from a sequence of local images taken from an UAV, we generate targets to be searched. Using this method, we

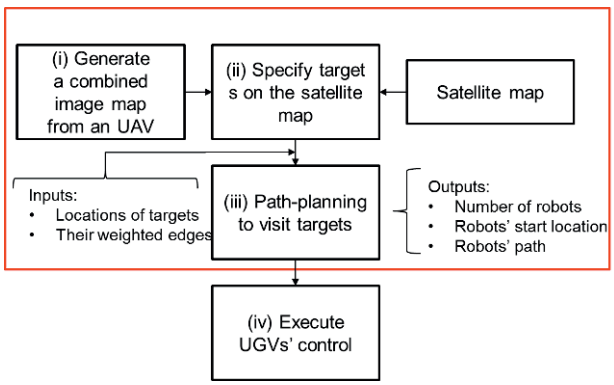


Fig. 3. Block diagram of our process.

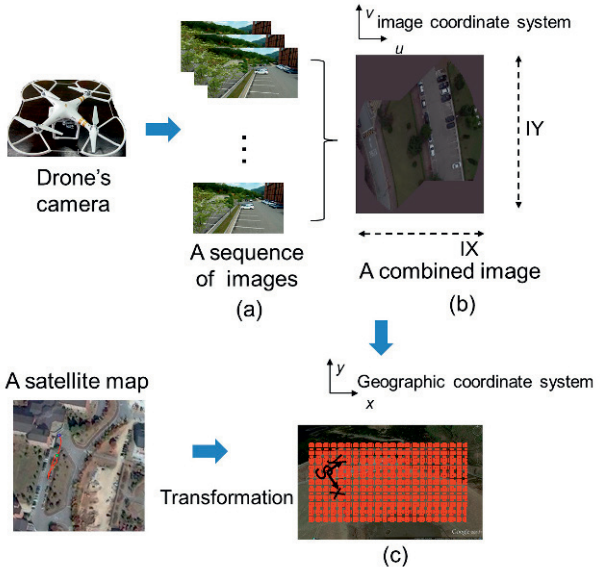


Fig. 4. Process of our map generation. Each shows (a) a sequence of images, which is incrementally merged by using SIFT features, (b) an image mosaic, and (c) a homogeneous coordinate system, combined with a satellite map. In (b), *IX* and *IY* are sizes of the image mosaic.

can identify UGVs from the image as well. UGVs can distributedly run according to our path-finding algorithm illustrated in Section 4 More details of a control method for UGVs are found in [26].

Examples of a satellite map are shown in Figs. 1 and 4. It has a global coordinate system: the latitude and the longitude of mark S on the map in Fig. 1 is $(32.885222, 131.05186)$. We only need landmarks to transform targets' locations. Fig. 4 shows the process of the generation of the combined map represented in a geographic coordinate system. Note that we try to match consecutive images from an UAV which has acceptable distortion. We call the merged image from local consecutive images as the image mosaic (or the combined image). The sequence of images taken from a drone is shown in Fig. 4(a). The image mosaic, matched and merged by the Scale-Invariant Feature Transform (SIFT) features and the transformed coordinate system, are shown in Figs. 4(b) and 4(c), respectively. In Fig. 4, a drone and a satellite image are shown on the left side.

3.1. Combined image (Mosaic)

To combine a sequence of images acquired from an UAV, we first decide a Region Of Interest (ROI) from a satellite map including geodesics as shown in Fig. 4. The specific region is represented in a rectangular shape according to the chosen top left-most and bottom right-most coordinates. Here are two landmarks representing the boundary of the region (ROI). Let us define that S and E be the top left-most and the bottom right-most corners, respectively. We take the latitude and the longitude of the start (S) and end (E) locations. We denote them as $(g_s(a), g_s(l))$ and $(g_e(a), g_e(l))$, respectively. Here, $g_\cdot(a)$ and $g_\cdot(l)$ are the latitude and the longitude of the target $\cdot$. Note that $i(t)$ and I be the local image acquired from an UAV at time t and the combined image mosaic, respectively. For merging a sequence of local images, we utilize SIFT descriptors and the RANSAC algorithm.

Two consecutive images are initially merged by using matched SIFT features. It then becomes a reference image (image mosaic) to combine with next images. It can be represented as $I = I \oplus i(t)$, where I and $i(t)$ are an image mosaic and the t^{th} image, respectively. The operation $\oplus$ of two images (of possibly different sizes) constructs a homography by using matched features. The t^{th} image $i(t)$ is accepted for our image mosaic according to the similarity of I and $i(t)$ defined as follows:

$$sim(d_I, d_{i(t)}) > \alpha, \tag{3}$$

where $d_{i(t)}$ and d_I are the SIFT features on the image $i(t)$ and the image mosaic I. We use a pre-defined threshold α to select good matches of the similarity. When they are matched well, the combined image I is used for the next consecutive image $i(t+1)$.

As mentioned above, the process of constructing an image mosaic starts with two consecutive images. In order to avoid heavy computation load, we limit the process of merging images within a certain ROI. For an effective merge of images, we especially consider to reduce the length of a video by monitoring only a circumscribed ROI and scheduling trajectory and height of an UAV for its flight. One can reduce a sequence of images by taking an UAV up high enough or making a pre-scheduled path for a flight. The process of finding SIFT features is applied on the grayscale intensity images. Two sets of SIFT features are matched and combined [15]. In order to reduce noise and distortion of images, we set a threshold α as in (3), which decides whether the image is accepted or not for an image mosaic. If the number of matched features are less than the threshold, we discard the image $i(t)$ and repeat the process starting from reading the following consecutive image $i(t+1)$.

3.2. Image transformation

For the choice of targets, we manually select them from the image mosaic. However, note that they can be autonomously selected by using vision-based algorithm. For instance, parked cars can be candidate targets. Next, we transform the targets represented in the combined image with respect to homogeneous coordinates. To transform the coordinates of the selected targets, we rotate the combined image in order to align with the geodesic from a

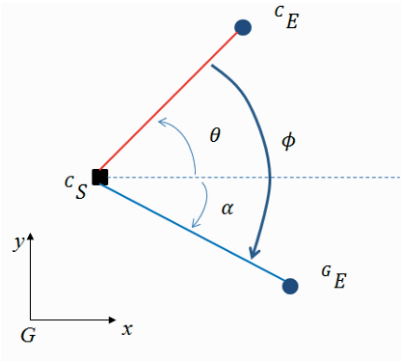


Fig. 5. Rotation angle to align two reference systems.

satellite map as shown in Fig. 5. The rotated coordinates on the image plane is computed as in (4). To do this, we set the landmark S as the center of rotation, and compute the rotation angle ϕ as shown in Fig. 5.

$$I_i = I_s + R(\phi)I_i, \tag{4}$$

where $I_i = (u_i, v_i)^T$, $I_s = (u_s, v_s)^T$, representing the coordinates on a combined image of the i^{th} target and the landmark s, respectively. The rotational angle ϕ is computed as $\phi = \alpha - \theta$, where two angles are defined as follows:

$$\alpha = atan2(g_e(a) - g_s(a), g_e(l) - g_s(l)), \tag{5}$$
$$\theta = atan2(v_s - v_e, u_e - u_s). \tag{6}$$

In (4), $R(\phi)$ is 2×2 rotational matrix. The estimated positions of the targets are then transformed by (7).

$$^{G}\hat{\xi} = T^{C}\xi \text{ and } T = (T_1|T_2), \tag{7}$$

$$T_1 = \begin{pmatrix} \ddots & & & \\ \cdots & \frac{g_s(l)-g_e(l)}{u_s-u_e} & 0 & \cdots \\ \cdots & 0 & \frac{g_s(a)-g_e(a)}{v_s-v_e} & \cdots \\ & & & \ddots \end{pmatrix}, \tag{8}$$

$$T_2 = \begin{pmatrix} & \vdots & & \\ \frac{g_e(l)}{u_s-u_e} & 0 & -\frac{g_s(l)}{u_s-u_e} & 0 \\ 0 & \frac{g_e(a)}{v_s-v_e} & 0 & -\frac{g_s(a)}{v_s-v_e} \\ & \vdots & & \end{pmatrix}, \tag{9}$$

where $^{G}\hat{\xi} = [\cdots, \hat{x}_i^T, \cdots]^T$, and $^{C}\xi = [\cdots, I_i^T, \cdots, I_s^T, I_e^T]^T$ represent the estimated targets in the global coordinates and the image coordinates, respectively. Here, $\hat{x}_i^T, I_i^T, I_s^T$, and I_e^T are the measured locations on the generated homogeneous map, the i^{th} target on the combined image from an UAV, the start position on the combined image, and the end position on a combined image. The estimated position of $\hat{x}_i^T$ includes the longitude and latitude for $i = 1, \cdots, N$, where N is the number of targets. Note that the combined map including the targets has the geographical system according to our method. We may represent them on a relative coordinate system transformed from the geodesic of latitude and longitude for computational simplicity.

4. PATH-FINDING ALGORITHM

This section takes into account our path-finding algorithm which selects a set of paths containing all targets on a map, addressed in Section 3. Considering to shorten completion time for covering all targets, we try to minimize the maximum path cost. In other words, it is to find the one that reduces the maximum path cost while other paths simultaneously cover all targets and their path costs are bounded to the maximum one. We propose a heuristic algorithm based on all-pairs shortest paths and an incremental path construction method. Our idea is to choose a path from a pool of all-pairs shortest paths as shown in Fig. 6. In the figure, all-pairs shortest paths starting at each target are expanded and represented in a tree structure. Blue squares and straight lines represent each target and available shortest paths to other targets (except the start one), respectively. For N targets $(\boldsymbol{x}_1, \cdots, \boldsymbol{x}_N)$, there are at most $N(N-1)$ shortest paths.

In our algorithm, we consider two serious of paths represented as $\Pi = \{\pi_r | r = 1, \cdots, R\}$ and $\Delta = \{\delta_i | i = 1, \cdots, N_S\}$, where $\pi_r = \{\boldsymbol{x}_s^{(r)}, \cdots, \boldsymbol{x}_d^{(r)}\}$ is the r^{th} robot path covering all elements (targets) in ξ, and δ_i is the i^{th} candidate shortest paths for given targets in ξ. The set Δ consists of candidate paths for robots, where $N_S \leq N(N-1)$. Considering directed edges for a complete graph, $N_S = N(N-1)$. We first generate all-pair shortest paths, which occupy most of computational time. The computational complexity for the worst case of a all-pair shortest path algorithm is $O(NElogN)$ since the Dijkstra algorithm requires $O(ElogN)$ where E and N are the number of edges and nodes (targets), respectively. For the kNN method, the time complexity of the worst case is $O(RN^{(1-1/R)})$, where R is the number of paths (robots).

Finding a set of multiple paths is challenging due to undetermined start locations and an undecided number of robots. A straightforward solution is to choose a path hav-

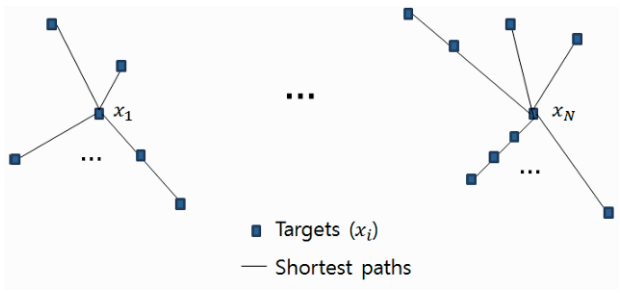


Fig. 6. A tree structure of available shortest paths.

ing the minimum path cost among all maximum path costs from Δ. It is represented as $\min_i \max_j cost(\delta_{ij})$, where δ_{ij} is the shortest path from x_i to x_j. It has proved when robots start at a single region [25]. However, it is not optimal for multiple robots starting at distinct locations. We define a weight function as follows:

$$w_i = f(\delta_i), \tag{10}$$

where $i = 1, \cdots, N_S$ indicates all-pairs candidate shortest paths. Note that we may include frequency of destined targets in each path in our weight function f; however, we only consider path costs of shortest paths for simplicity in this work. It is represented as $f(\delta_i) = 1 - \frac{c_i}{c_{max}}$, for $i = 1, \cdots, N_S$, where c_i and c_{max} are the i^{th} path cost and the longest path cost in Δ, and N_S and N ($N_S \gg N$) are the number of available shortest paths and the number of the targets, respectively.

For inputs of a set of targets Π and their edges E, our algorithm is as follows:

A:1. Find all-pair shortest paths ($\Delta = \cup_{i=1,\cdots,N_S} \delta_i$,)

A:2. sort Δ in ascending order w.r.t the weight as shown in (10).

A:3. Initialize $l = 0$ and $c_{max} = \max_i w_i$.

A:4. For each δ_i for $i = 1, \cdots, N_S$, choose δ_i if targets in the path are not covered yet, and do the followings:

- A:A. Set *dest* as the last target in δ_i.
- A:B. For each δ_j for $j = 1, \cdots, N_S$, choose δ_j if it satisfies that the last target in δ_j is *dest* and $cost(\delta_j) < c_{max}$. If it is chosen:
  - A:a. $\pi_l = \delta_j$ and $c_{max} = cost(\delta_j)$.
  - A:b. Update the path (π_l) with available targets
  - A:c. Increase l and set the status of targets to be covered.

A:5. Set $R = l$.

As a result, the algorithm generates a set of robot paths Π along with the number of robots R.

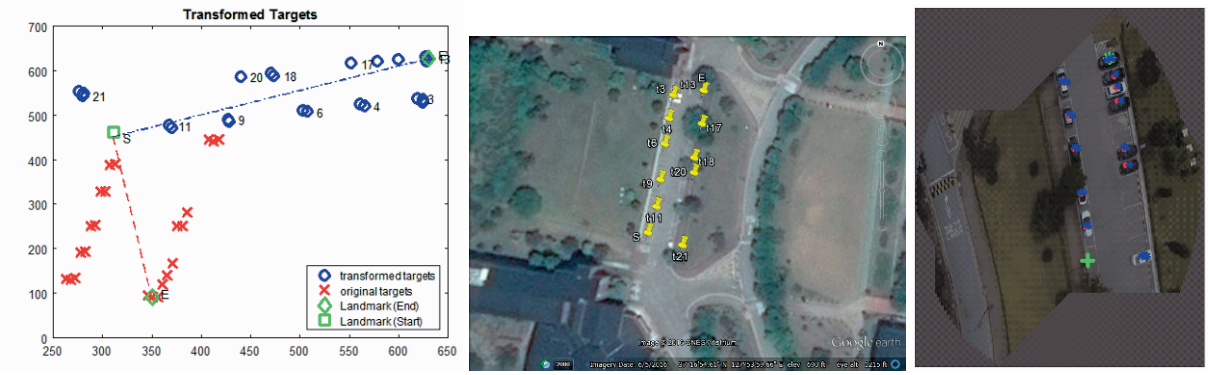
We denote that π_{max} is the maximum path cost, which is represented as follows: $\pi_{max} = \max_{j=1,\cdots,R} cost(\pi_j)$, where R is the number of robots which will be selected during the process. Each path, at the same time, is bounded to π_{max}, and $K = \pi_{max}$. Here, K is our objective that minimizes the maximum path cost as in (2a). Whenever the maximum path cost c_{max} is updated (Step A:a), we examine each path δ_j (Step A:B) whether there are available targets while each path maintains its cost smaller than the maximum path cost. The algorithm is repeated until there are no remaining targets.

Fig. 7. Trajectory of a drone while taking a video for the ROI 'A' shown in Fig. 8(f). Coordinate data represented as red line is overlapped on Google Earth. There are two ROIs represented by 'A' and 'B'.

5. EXPERIMENTAL RESULTS

This section validates our proposed solution: the generation of a coordinated target map and the selection of optimal paths in addition to the number of robots and their locations. In our experiments, we represent targets in a geographic coordinates system, which acts as a global coordinate system, and display them on Google Earth [27]. To demonstrate our path-finding algorithm, we compare with the well-known k-Nearest Neighbor (kNN) algorithm [13] in both real and simulated environments.

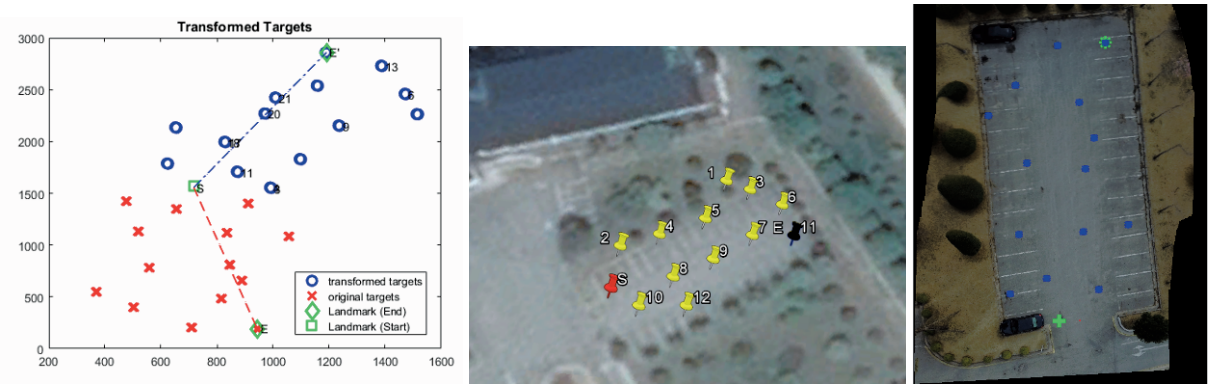
For the experimental environments, we used an aerial robot of the DJI Phantom 3 Professional. Our scenario for real robot experiments is that an aerial robot takes a video while it flies outside, and we manually choose search locations (targets). To generate targets' location, we examine two choices: the locations of parked cars (as shown in Fig. 8(c)) and randomly distributed locations (as shown in Fig. 8(f)). A trajectory of a flight is shown in Fig. 7. It is gathered from the GPS sensor mounted on Phantom 3 Professional, and is visualized on Google Earth. To get videos (a sequence of images) capturing required regions, we piloted the drone and recorded videos during its trips. Phantom-3 professional has the 1/2.3" CMOS camera and a gimbal that allows rotation in 3-axis (pitch, roll, and yaw). For each flight, we tried to set the camera point down (0° and 45°). The resolution of the image is 1280×720 with 24 fps or 640×480 with 30 fps. The lengths of the video clips recorded during its flight are about 4 and 9 minutes for each case shown in Figs. 8(c) and 8(f), respectively. We trim the video clips to select only the segment we need for an ROI.



(a) The rotated targets on the image mosaic by (4). Red crosses and blue circles are the targets representing before and after the rotation.

(b) The targets represented on Google Earth, showing the ground truth.

(c) Green signs represent the start and end locations, and blue stars are the targets.



(d) The rotated targets on the image mosaic by (4) applied on the map shown in Fig. 8(f).

(e) The targets represented on Google Earth, showing the ground truth.

(f) Green signs represent the start and end locations, and blue stars are the targets.

Fig. 8. Transformed targets from the image mosaic.

For the scenario shown in Figs. 8 and 9 , we set several outside parking lots as our ROIs. Fig. 9(a) shows the ROI denoted as 'B' in Fig. 7, and is represented on a satellite map including the start and end locations. The map displays the landmarks in geographic coordinates of the latitudes and the longitudes. Fig. 9(b) shows an image mosaic including the targets, corresponding to the same region 'B'. The targets are represented as red dots characterizing currently parked cars, which are manually chosen for this experiment. Note that our process of combining local images as shown in Fig. 9(b) is run off-line. For SIFT matches, we trimmed the captured video short according to a required ROI. The lengths of each video representing the regions of 'A' and 'B' were about 40 and 15 seconds, respectively, for the experiments shown in Figs. 8(c) and 8(f). We set the image resolution 640×480 with 30 fps for the ROI 'B'. We, on the other hand, attempted to set the resolution 1280×720 with 24 fps for the ROI 'A'. For both cases, we empirically chose about 7 fps in order to maximize the effect of image merging results.

In Fig. 8(a), the landmarks S and E (represented by green plus and star, respectively) correspond to the ones in Fig. 9(a). The geographic coordinates are $(37.281704, 127.899701)$ and $(37.2821, 127.899953)$ for S and E represented on a satellite map as shown in Fig. 9(a), while the local coordinates of S and E on the image mosaic are $(311, 452)$ and $(350, 90)$, as shown in Fig. 8(c). The size of the image mosaic shown in Fig. 8(c) is 643×521.

We applied a distance-based clustering algorithm with color detection for the targets (represented as red dots) generated on the image mosaic in Fig. 9(b). Using the method, target locations are automatically identified from the image mosaic. The identified targets are represented by blue stars as shown in Fig. 8(c). The rotational angles α, θ, and ϕ are about $29°, -84°$, and $113°$, respectively. The result of transformed locations for the targets is shown in Fig. 8(a): red crosses and blue circles represent the locations of targets on the image mosaic before and after the transformation, respectively. Fig. 8(b) shows the compared result of the transformed coordinates for the targets by representing them on Google Earth. The targets are marked with yellow pins, and we placed numbers around each target. For the representation, we imported our transformed longitude and latitude coordinates in Google Earth

(a) Google map showing the start and end locations including their latitudes and longitudes.

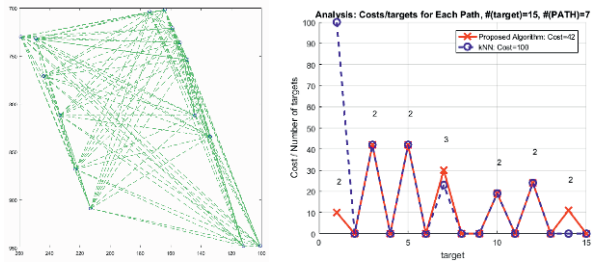
(b) Image mosaic including the targets manually selected and represented by the red dots.

Fig. 9. Results showing (a) a satellite map representing landmarks for a ROI, and (b) an image mosaic generated by local images from an UAV. Here, targets are parked cars.

(shown as placemarks).

Figs. 8(a) $\sim$ 8(c) and Figs. 8(d) $\sim$ 8(f) show the results for the ROIs corresponding to 'B' and 'A', respectively. Fig. 8(f) is the result of a combined image representing the ROI marked as 'A' on the map in Fig. 7. Here, the targets are randomly chosen. In the figure, the start and end locations representing the ROI are marked as green plus and star, and the size of the image mosaic is 1444×1865. Blue dots represent the targets which are randomly selected. As a result, Fig. 8(d) shows the image transformation from the local coordinates to the global coordinates. The transformed coordinates of the random targets (shown in Fig. 8(f)) are compared with yellow pins (for the targets) along with red and blue pins (for the landmarks) in Fig. 8(e). It validates that our transformation from the local image mosaic to the global coordinates works well.

Next, we examined our path-finding algorithm. We generated available edges among the targets while assuming that no obstacles exist among them. For the generation of edges in an environment with obstacles, one can refer our previous publication [25]. Fig. 10(a) shows the generated edges for the targets shown in Fig. 8(c). The numbers representing latitudes and longitudes express very small differences ($10^{-2} \sim 10^{-6}$ in the experiment); for example, the latitudes of S and E are 37.281704 and 37.2821, respectively. To avoid loss of significance, we applied $(g_a(\cdot) - 37.28) \times 10^5$ and $(g_l(\cdot) - 127.899) \times 10^6$ to each target when we utilize them in our path generation. The targets and the generated edges are shown in Fig. 10(a) for the experiment shown in Fig. 8(b). The corresponding paths with the path costs are shown in Fig. 10(b). The x-axis is each target and the y-axis is the path costs. If a target displays zero cost, it means there is no path initially assigned to the target.



(a) Targets represented on the image mosaic and generated edges

(b) Selected path costs and the start locations along with the number of paths, applied on the graph in (a)

Fig. 10. Comparison of each path cost. The x- and y-axes are each target and the path costs. The maximum path cost 42 by the proposed algorithm is compared to 100 by kNN.

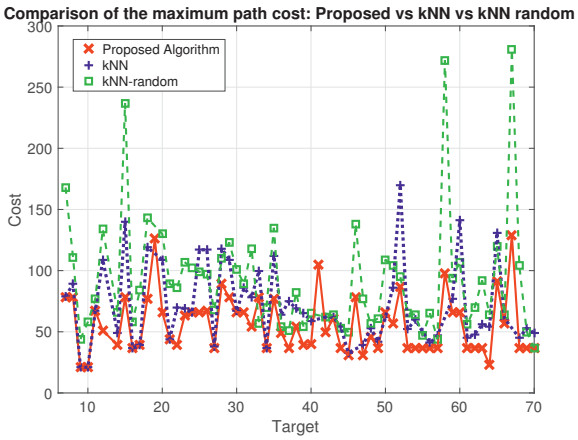


Fig. 11. Comparison of the maximum path cost for various numbers of random targets. Red crosses are the proposed method while green squares and blue pluses represent the results from kNN and kNN random, respectively.

As a result of generating the paths with the number of robots, there are 7 paths as shown in Fig. 10(b). Each path is compared with the kNN algorithm while we applied 15 targets. Since the kNN algorithm requires the number of robots along with their initial locations, we used the same start locations and the robots acquired by the proposed algorithm. For the experiment, we first applied our algorithm to get the set of paths (including the number of robots and their initial locations), and then utilized the information when applying the kNN method. The comparison exhibits that our proposed algorithm chose 42 as the maximum path cost while the kNN algorithm chose 100 as the maximum path cost. Note that the chosen initial locations for robots were $1, 3, 5, 7, 10, 12$ and 14 as shown in Fig. 10(b). The numbers denoted in the graph are the number of targets placed in each path.

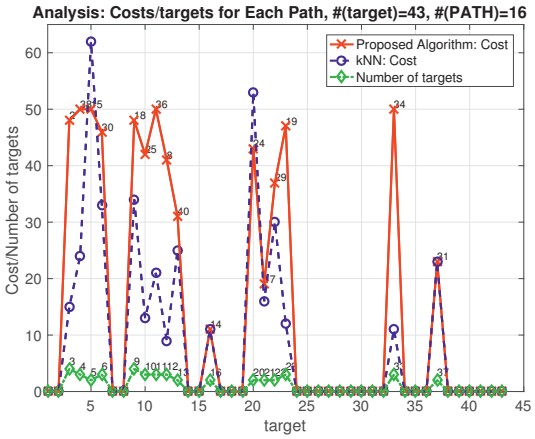


Fig. 12. Analysis of the chosen paths for a case of 43 targets. Blue circles and red crosses are the paths chosen by kNN and the proposed method, respectively. The selected robots are 16, and the maximum cost of 50 is compared with 62.

To demonstrate our path-finding algorithm, we compared the maximum path cost with the best results from kNN and kNN random. For kNN random, we used the same number of robots as ours, but the start locations were randomly chosen. We investigate the random case in order to compare the results while we choose different initial locations for robots. For the experiments, we used an obstacle-free environment with the size of 130×130, and randomly generated targets from 3 to 70 along with random edges. Fig. 11 shows the comparison of the maximum path cost with kNN and kNN random for each trial (various number of targets). It reveals that our path-finding algorithm chooses an efficient set of paths reducing the maximum path cost.

We analyzed each chosen path for the case of the randomly generated 43 targets. Fig. 12 shows the choice of the start targets (the ones have no zero cost), the destination targets (represented in text around red crosses in the figure), and their corresponding path costs (the y-axis). The selected robots are 16 and the maximum path cost is 50.

6. CONCLUSION AND FUTURE WORK

This work presented a solution for a team of heterogeneous robots, especially a cooperation of two groups of UAVs and UGVs, for a time-critical mission. Our framework has dealt with transforming local sensing images taken from an UAV to a satellite map that uses a geographic coordinate system. In our work, UAVs construct a map reflecting dynamic changes, which allow us a centralized path-finding method for targets in a dynamic environment. For future work, we will consider constructing a map for search robots evolving in long-term changes.

REFERENCES

[1] R. R. Murphy, K. L. Dreger, S. Newsome, J. Rodocker, B. Slaughter, R. Smith, E. Steimle, T. Kimura, K. Makabe, K. Kon, H. Mizumoto, M. H. amd F. Matsuno, S. Tadokoro, and O. Kawase, "Marine heterogeneous multirobot systems at the great eastern Japan tsunami recovery," *Journal of Field Robotics*, vol. 29, pp. 819-831, 2012. [click]

[2] D. Floreano and R. J. Wood, "Science, technology and the future of small autonomous drones," *Nature*, vol. 521, pp. 460-466, 2015. [click]

[3] J. Hu, L. Xie, K. Y. Lum, and J. Xu, "Multiagent information fusion and cooperative control in target search," *IEEE Transactions on Control Systems Technology*, vol. 21, no. 4, pp. 1223-1235, 2013. [click]

[4] S. K. Gan and S. Sukkarieh, "Multi-uav target search using explicit decentralized gradient-based negotiation," *Proc. of IEEE International Conference on Robotics and Automation*, 2011, pp. 751-756.

[5] M. Schwager, B. J. Julian, M. Angermann, and D. Rus, "Eyes in the sky: Decentralized control for the deployment of robotic camera networks," *Proceedings of the IEEE*, vol. 99, no. 9, pp. 1541-1561, 2011. [click]

[6] Q. Huang, J. Yao, Q. Li, and Y. Zhu, "Cooperative searching strategy for multiple unmanned aerial vehicles based on modified probability map," in *Theory, Methodology, Tools and Applications for Modeling and Simulation of Complex Systems: 16th Asia Simulation Conference and SCS Autumn Simulation Multi-Conference*, 2016, pp. 279-287.

[7] S. Weiss, M. Achtelik, L. Kneip, D. Scaramuzza, and R. Siegwart, "Intuitive 3d maps for mav terrain exploration and obstacle avoidance," *Journal of Intelligent & Robotic Systems*, vol. 61, no. 1, pp. 473-493, 2011.

[8] L. Heng, G. H. Lee, F. Fraundorfer, and M. Pollefeys, "Real-time photo-realistic 3D mapping for micro aerial vehicles," *Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2011, pp. 4012-4019.

[9] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381-395, 1981. [click]

[10] D. Scaramuzza, M. C. Achtelik, L. Doitsidis, F. Friedrich, E. Kosmatopoulos, A. Martinelli, M. W. Achtelik, M. Chli, S. Chatzichristofis, L. Kneip, D. Gurdan, L. Heng, G. H. Lee, S. Lynen, M. Pollefeys, A. Renzaglia, R. Siegwart, J. C. Stumpf, P. Tanskanen, C. Troiani, S. Weiss, and L. Meier, "Vision-controlled micro flying robots: From system design to autonomous navigation and mapping in gps-denied environments," *IEEE Robotics Automation Magazine*, vol. 21, pp. 26-40, 2014. [click]

[11] F. Fraundorfer, L. Heng, D. Honegger, G. H. Lee, L. Meier, P. Tanskanen, and M. Pollefeys, "Vision-based autonomous mapping and exploration using a quadrotor mav," *Proc. of Int. Conf. on Intelligent Robots and Systems*, 2012, pp. 4557-4564.

[12] M. Faessler, F. Fontana, C. Forster, E. Mueggler, M. Pizzoli, and D. Scaramuzza, "Autonomous, vision-based flight and live dense 3D mapping with a quadrotor micro aerial vehicle," *Journal of Field Robotics*, vol. 33, no. 4, pp. 431-450, 2016. [click]

[13] S. Arora, "Polynomial time approximation schemes for Euclidean traveling salesman and other geometric problems," *Journal on the ACM*, vol. 45, pp. 753-782, 1998. [click]

[14] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. Journal of Computer Vision*, vol. 2, no. 60, pp. 91-110, 2004. [click]

[15] A. Vedaldi and B. Fulkerson, "Vlfeat: an open and potable library of computer vision algorithms," *Proc. of Int. Conf. on Multimedia*, pp. 1469-1472, 2010.

[16] A. R. Zamir and M. Shah, "Accurate image localization based on Google maps street view," *Proc. of Computer Vision - ECCV 2010: 11th European Conference on Computer Vision*, Heraklion, Crete, Greece, Part IV, pp. 255-268, September 5-11, 2010.

[17] P. Agarwal, W. Burgard, and L. Spinello, "Metric localization using google street view," *Proc. of IEEE International Conference on Intelligent Robots and Systems*, pp. 3111-3118, 2015.

[18] J. Yuan and A. M. Cheriyadat, "Image feature based gps trace filtering for road network generation and road segmentation," *Machine Vision and Applications*, vol. 27, no. 1, pp. 1-12, 2016.

[19] M. Schwager, D. Rus, and J.-J. Slotine, "Unifying geometric, probabilistic, and potential field approaches to multi-robot deployment," *Int. Journal of Robotic Research*, vol. 30, pp. 371-383, 2011.

[20] G. Tuna, V. C. Gungor, and K. Gulez, "An autonomous wireless sensor network deployment system using mobile robots for human existence detection in case of disasters," *Ad Hoc Networks*, vol. 13, pp. 54-68, 2014. [click]

[21] M. A. Batalin and G. S. Sukhatme, "The design and analysis of an efficient local algorithm for coverage and exploration based on sensor network deployment," *IEEE Trans. on Robotics*, vol. 23, pp. 661-675, 2007. [click]

[22] A. Howard, M. J. Matarić, and G. S. Sukhatme, "An incremental self-deployment algorithm for mobile sensor networks," *Autonomous Robots*, vol. 13, pp. 113-126, 2002. [click]

[23] J. L. Ny and G. J. Pappas, "Adaptive deployment of mobile robotic networks," *IEEE Trans. on Automatic Control*, vol. 58, pp. 654-666, 2012. [click]

[24] N. Mathew, S. L. Smith, and S. L. Waslander, "A graph-based approach to multi-robot rendezvous for recharging in persistent tasks," *Proc. of the IEEE Int. Conf. on Robotics and Automation*, pp. 3497-3502, 2013. [click]

[25] H. J. Min and N. Papanikolopoulos, "The multi-robot coverage problem for optimal coordinated search with an unknown number of robots," *Proc. of IEEE Int. Conf. on Robotics and Automation*, 2011.

[26] —-, "Robot formations using a single camera and entropy-based segmentation," *Journal of Intelligent & Robotic Systems*, vol. 68, no. 1, pp. 21-41, September 2012. [click]

[27] "Google earth," https://www.google.com/earth/, accessed: 2017.

Hyeun Jeong Min received her Ph.D. degree in Computer Science from the University of Minnesota, USA in 2013. Her research interests include coordinated search, path planning, visual tracking, and robot formation.