

# Semi-Automated Emergency Landing Site Selection Approach for UAVs

BULENT AYHAN , Member, IEEE  
CHIMAN KWAN   
YOOL-BIN UM  
BENCE BUDAVARI   
JUDE LARKIN   
Signal Processing Inc., Rockville, MD, USA

The use of unmanned aerial vehicles (UAV) in military and industry today is becoming more widespread. There are a wide range of UAV models that are functional today. The size of these UAVs can be as small as a hawk and can be as big as a passenger jetliner. It is critical for these UAVs to have contingency plans before flight in case of unexpected situations, such as engine-out events which cause total loss of thrust during flight. An important part of contingency planning is to identify emergency landing sites along the flight path of the UAV. This paper discusses the development of an offline semi-automated approach for finding emergency landing sites in the shape of a rectangular runway to be used in preflight contingency planning. The approach introduces a total of five emergency landing measures and a surface type estimation, which are applied to the identified emergency landing site candidates for their safety assessment. The output is a list of emergency landing site candidates together with their surface type estimates that are ranked from the safest to least safe through a generalized safety score for each surface type. The approach can label the ranked landing site candidates according to their reachability in the presence of wind, given the UAV's altitude and coordinates at the time the total loss of thrust happened and the wind forecast for the area.

Manuscript received January 3, 2018; revised June 7, 2018; released for publication October 3, 2018. Date of publication November 9, 2018; date of current version August 7, 2019.

DOI. No. 10.1109/TAES.2018.2879529

Refereeing of this contribution was handled by K. D. Pham.

This work was supported by the U.S. Navy under Contract N68335-16-C-0154.

Authors' address: B. Ayhan, C. Kwan, Y.-B. Um, B. Budavari, and J. Larkin are with the Signal Processing Inc., Rockville, MD 20850 USA, E-mail: (bulent.ayhan@signalpro.net; chiman.kwan@signalpro.net; yoolbin92@gmail.com; bencebudavari@gmail.com; judelarkin93@gmail.com). (*Corresponding author: Chiman Kwan.*)

This paper has supplementary downloadable material available at <http://ieeexplore.ieee.org>, provided by the authors.

---

0018-9251 © 2018 IEEE

## I. INTRODUCTION

Reliability of unmanned aerial vehicles (UAV) can be strengthened using durable engines, strong structural materials, advanced structural health monitoring procedures [1]–[3], robust fault diagnosis and prognosis algorithms [4]–[6], and robust [7], [8] and fault tolerant controllers [9], [10]. Despite the above measures, some contingencies, such as engine failures, may still occur, and emergency landing places need to be determined. Planning for contingencies starts before flight missions, and in some cases, needs to be revised while the mission is taking place. An important part of the contingency planning is identifying alternative emergency landing sites for each flight segment along the flight path of the UAV [11]. Identifying these alternative landing sites could be a tedious task for human operators. Moreover, even if emergency landing sites are identified by human operators for the areas of interest, they need to be checked and validated periodically before a new mission since the earth surface is continuously subject to changes either because of man-made changes or changes due to nature (landslide, flooding, earthquake, etc.). Considering that an engine-out situation for the UAV has happened that causes total loss of thrust and forces the UAV to make an emergency landing, there could be numerous landing site candidates for the UAV. It would be preferable that the UAV lands at a site which does not cause any human loss and minimizes the damage to the UAV.

A semi-automated emergency landing site selection approach is introduced in this paper for supporting preflight contingency planning efforts. The approach eases the job of the human operators by providing a list of emergency landing site candidates within the geographical area of interest that are ranked from the safest to least safe. For each emergency landing site candidate, the surface type is also estimated. This list of emergency landing site candidates, together with their surface type estimates, can be easily reviewed by a human operator before any of these candidates is finalized as an emergency landing site in preflight contingency planning efforts. The proposed approach labels the ranked landing site candidates for each surface type with respect to their reachability given the UAV's last location and altitude at the time the total loss of thrust happened. The reachability analysis is conducted in the presence of wind using the wind forecast for the area of interest and utilizing the polar curve specific to the aircraft. The reachability scores are categorized with five labels: 1) reachable with a high excessive altitude; 2) reachable with a small excessive altitude; 3) barely reachable; 4) uncertain; and 5) not reachable.

There are several related works in the literature for emergency landing site identification. Some of these works are for in-flight use where the processed images are captured by on-board cameras [12]–[18]. There are some other works which assume that there is already a list of airport runways and they provide techniques for which runway to choose in an emergency [19]–[20]. A brief survey of some of the related past works are as follows. Levora *et al.* [21] introduced

a method to segment a given image into homogeneous regions for emergency landing purposes. In their work, the segmentation is based on the expectation–maximization algorithm and the feature vector contains pixel color, position, and texture of pixel neighborhood. In [15], an emergency landing site detection method using visible information acquired by aircraft-mounted cameras is described. The authors used a hierarchical elastic horizon detection algorithm to identify the ground in the image first. Then, they clustered the terrain image according to a “roughness” measure. In [16]–[18], Fitzgerald *et al.* proposed using low quality on-board aerial imagery for locating “safe” landing sites for UAV forced landings using computer vision, image processing techniques, and neural networks. The authors first processed the whole image to generate two binary images that correspond to edge detection and pixel intensity. They fused these two binary images with an OR logic operator. They then searched the fused binary image with four rectangular masks in combination with four different rotation angles in a so-called geometric acceptance phase. The four rectangular masks are passed over the fused binary image area to generate a preliminary site selection map. For a particular region being tested with the four masks, if all pixels under the pixels in the mask contain 1, then the pixels in this region are labeled as SAFE. The authors then generated a candidate landing site output map by fusing the preliminary site selection map and two other maps (surface type classification and course slope map) using a heuristic fuzzy linguistic based decision method. In [22], Garcia-Pardo, *et al.* designed a two-step autonomous safe landing site detection strategy. In [19], the authors used a weighted sum fusion for selecting the best airport runway using several attributes in a total loss of thrust emergency. The fused attributes correspond to eight airport and weather-related parameters. These are runway length, runway width, instrument approach quality, distance from the footprint boundary, headwind velocity, crosswind velocity, surface quality, and facility availability measure. Some of these parameters are normalized by their extreme values over the set of feasible landing sites before applying weighted sum fusion. Their landing site candidates are limited to airport runways that are retrieved from an airport database. In [20], the authors proposed a landing site selection from a number of available candidates using decision making Bayesian networks. Reachability and civilian proximity were some of the attributes considered in their work. Similar to [19], their approach assumed that there is a list of landing sites already available. In [14], the authors proposed a method to select safe landing zones for small-sized UAVs in dense settlement areas. Their method detects objects in the presence of camera movement and using the detected obstacles. The method then selects a safe landing zone for the UAV. In [23], the authors introduced a landing site selection method for UAVs based on  $k$ -nearest neighbor classification for segmentation considering both color and texture features. Their method did not take into account elevation profiles of the areas and it located suitable areas of a given size and shape for a UAV forced landing.

The technical aspects of our paper are summarized as follows: 1) introduction of an offline semi-automated approach for emergency landing site detection and ranking of the identified landing site candidates to support preflight contingency planning efforts; 2) use of color satellite images, elevation maps, and global human settlement (GHS) built-up layer [28] that are publicly available; 3) adjustable length and width for the rectangular mask when finding the rectangular-shaped emergency landing sites in order to support various aircraft with different emergency landing site size requirements; 4) integration of five measures to assess the safety of the emergency landing site candidates; 5) use of low-resolution satellite images to coarsely identify landing site candidates and use of high resolution image counterparts of those coarsely identified landing site candidates to estimate surface types; 6) surface type estimation of the identified landing sites using Gabor texture features and conducting a thorough investigation of which image resolution to use when extracting the texture features for surface type estimation; 7) a generalized ranking for each surface type using a weighted sum fusion of the safety measures; and 8) labeling of ranked landing site candidates for each surface type with respect to reachability in the presence of wind.

Although the individual processing steps of the proposed automated approach may not be new, to the authors’ knowledge, the system as a whole is new, unique, and fills a void in preflight contingency planning for UAVs. Our approach differs from the past related works by being an offline tool for supporting preflight contingency planning efforts. Because it is an offline approach, computation time constraints are not strict. One novelty in our approach is the utilization of both low and high resolution color satellite images within the same framework. We use low-resolution satellite images to coarsely identify landing site candidates and then use high-resolution images of these identified landing sites to estimate their surface types. This reduces the image download times since one does not need to download high-resolution images for the whole geographical area of interest, but only for the coarsely identified landing site candidates. Different from [16]–[18], instead of generating maps for the whole image for each measure, we use the whole image to coarsely identify rectangular-shaped landing sites and then focus only on these landing sites to apply our safety measures and surface estimation. A second attribute which we also consider as novel in our work is the integration of a ranking mechanism into the landing site identification framework. Many of the past works provide numerous image segments as output to pick emergency landing site candidates from [16], [18], [21], [23], and [24], which are all reasonable, yet, for a human operator to make a quick decision among these image segments of different sizes could be still challenging. Some of the works propose techniques to narrow down the decision to a few possible landing sites from multiple candidates, but these candidates are limited to a database of airport runways [19], [20] and do not provide other candidates of landing sites [25]. Fitzgerald *et al.* [16]–[18] incorporate

emergency landing site identification and apply a heuristic fuzzy based ranking, but the output emergency landing sites are in the form of image segment blobs and the ranking is limited to five fuzzy sets rather than quantitative values. Our approach identifies distinct rectangular-shaped emergency landing sites instead of blob-like segments and ranks them within the same framework. Our currently adapted ranking algorithm is a simple one and there is room to improve this with advanced decision-making techniques in the future. A third novel attribute of our proposed framework is using wind forecast data to determine the reachability of a ranked landing site.

We envision our approach to be used as an auxiliary tool for preflight contingency planning. After the primary flight path and flight profile are finalized by the flight management with waypoints and altitudes at those waypoints, the contingency plans would be generated for each waypoint along the primary flight path. Contingency plans are alternative flight plans in case the UAV faces an emergency like total loss of thrust and needs to make an emergency landing. The contingency plan would consist of alternative waypoints and an emergency landing site. We consider that our approach would help the mission planners when identifying the emergency landing sites for these contingency plans. Even though the landing site decision for a contingency plan would be still made by the operator or planner, our approach would provide a safety ranking of these landing site candidates for each surface type and provide information about their reachability, making it easier for the operator or planner to pick from.

The paper is composed of the following sections. Section II provides the overview of the processing steps of the semi-automated approach. Sections III–VII describe each of the processing steps of the approach in detail by providing demonstrations. These processing steps are: Automated image retrieval (see Section III), finding preliminary emergency landing areas (see Section IV), finding rectangular runway-shaped landing site candidates (see Section V), measures to assess the landing site candidates (see Section VI), generalized safety score generation (see Section VII), and labeling of the ranked landing site candidates with respect to reachability (see Section VIII). The results and analysis after applying the approach to three different geographical areas of interest are summarized in Section IX. Finally, concluding comments and future work are discussed in Section X.

## II. PROCESSING STEPS OF THE APPROACH

Fig. 1 shows the block diagram of the proposed approach. Here, a high-level description of the approach is provided. More details will be given in Sections IV–VIII. The inputs to the approach are: 1) coordinates of the area of interest in latitude and longitude; 2) size of the emergency landing site in length and width; and 3) a previously trained mathematical model for surface type estimation of the determined emergency landing site candidates. The optional inputs for reachability analysis are: 1) altitude and

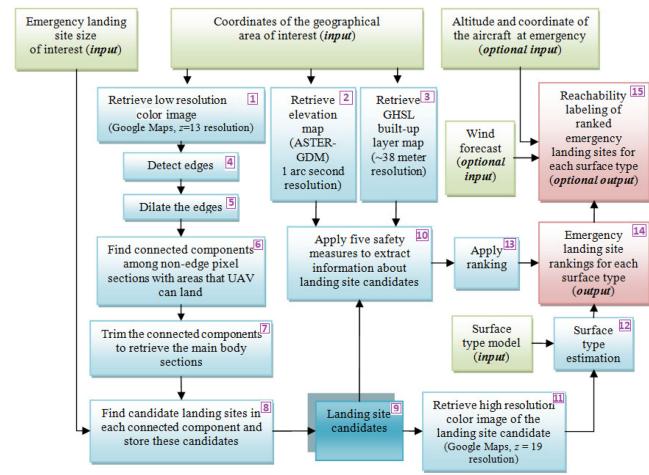


Fig. 1. Block diagram of the emergency landing site selection approach.

coordinates of the aircraft at the time of emergency; and 2) wind forecast data for the area of interest and flight time. The approach starts by retrieving the color satellite image of the area of interest from Google Maps at zoom level 13 ( $z = 13$ ), ASTER digital elevation maps [26] from USGS Earth Explorer [27], and the GHS built-up layer [28].

## III. AUTOMATED RETRIEVAL OF SATELLITE IMAGES, MAPS, AND LAYERS

### A. Color Image Retrieval

This subsection corresponds to Block 1 in Fig. 1. Google Static Maps application program interface (API) is used to retrieve the color images. This API allows downloading image tiles with a maximum width of 640 pixels and maximum height of 640 pixels. The resolution of the images can be as high as 0.25 m/pixels for inland areas. When sending a color image download request using the Google Static Maps API, the API requires several parameter values within the request. These parameters are: 1) center coordinates of the image of interest (latitude and longitude); 2) image height; 3) image width; and 4) the zoom level  $z$ . Google Maps uses Mercator projection [29]. In Mercator projection, the resolution of a map is dependent on the latitude [30]. The resolution of the map that uses a Mercator projection can be mathematically expressed as follows [30]:

$$\text{Resolution (m/pixel)} = \frac{156543.034 \times \cos(\text{lat} \times (\pi/180))}{2^z}. \quad (1)$$

In (1),  $z$  corresponds to the selected zoom level and  $\text{lat}$  corresponds to the latitude value in degree. Google Static Maps API has a limitation on the download of a single image size [29]. For large areas of interest that exceed this size limitation, there is a need to first partition the area of interest into multiple smaller image tiles for download. The original area needs to be then reconstructed using these smaller image tiles. Assuming that the area of interest is identified by the latitude and longitude coordinates of the upper right corner and lower left corner, first the necessary number of tiles along the latitude direction  $M$ , and the number of tiles

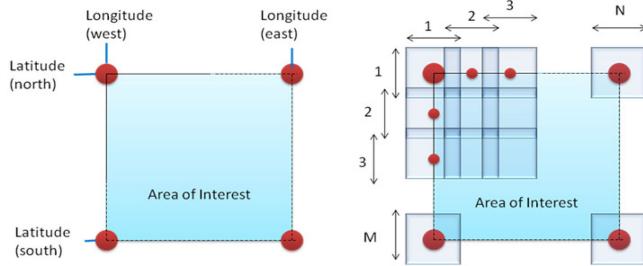


Fig. 2. Identifying the number of smaller tiles needed to reconstruct the large area of interest (which cannot be downloaded at once).

along the longitude direction  $N$ , are found out. Suppose  $A$  is the upper left corner point and  $B$  is the lower left corner point for the area of interest and the actual distance between points  $A$  and  $B$  in m corresponds to  $d_{AB}$ . Suppose the map resolution at zoom level  $z$  is denoted by  $r$ , and it is estimated using (1) where lat in (1) is set to the average of the latitude values of points  $A$  and  $B$ . One can then find the total number of pixels along the latitude direction ( $n_{p,\text{lat}}$ ) as

$$n_{p,\text{lat}} = \frac{d_{AB}}{r}. \quad (2)$$

Because Google Static Maps API does not allow more than 640 pixels for the image height, the number of tiles along latitude  $M$ , can be found as

$$M = \text{ceil}\left(\frac{n_{p,\text{lat}}}{640}\right) \quad (3)$$

where ceil is a mathematical operator which rounds the value to the nearest integers towards infinity. To guarantee downloading all pixel rows and not to miss any pixel rows for the area of interest, overlapping tiles are considered instead. The number of tiles to download for the same distance is increased by 25%,  $M = M * 1.25$ . The coordinates of the center of these  $M$  tiles are found. The same process is repeated along the longitude direction, and the  $N$  tiles and their center locations are computed. This process is demonstrated in Fig. 2.

Upon downloading the  $M \times N$  small-size image tiles, the next step is to reconstruct the large area of interest with these overlapping small-size image tiles. In the reconstruction, the overlapping sections for consecutive image tiles are found in both latitude and longitude directions. In finding the number of overlapping pixels among two consecutive image tiles along each direction, a validation check that involves both the cross correlation result and the mathematically computed result using the Mercator projection properties is used. Because there is some overlapping along consecutive image tiles in both latitude and longitude directions, the cross correlation results of the selected row pixels (or column pixels) in the consecutive image tiles provide information about the exact row index or column index (which corresponds to the start pixel for the nonoverlapping image area). If the results of the cross correlation check are comparable to the results from the mathematically computed values, the cross correlation results are used in reconstruction; otherwise the mathematically

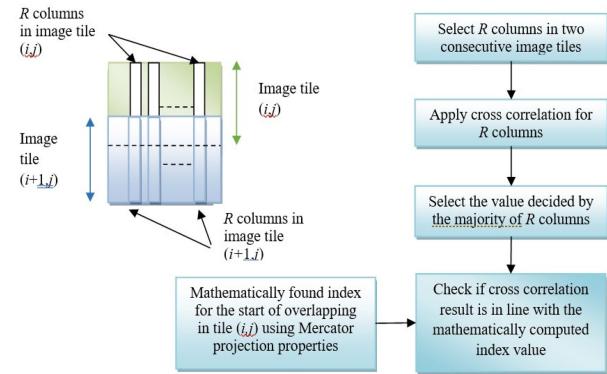


Fig. 3. Finding the overlapping area in two consecutive overlapping image tiles Image tile  $(i, j)$  and Image tile  $(i + 1, j)$  along latitude direction.

computed result using the Mercator projection properties is used. Fig. 3 demonstrates how the number of overlapping pixels is found along the latitude direction among two consecutive small-size image tiles. The same process is applied in the longitude direction as well.

For each downloaded image tile  $(i, j)$ , we find the pixel coordinate of the center point that is originally provided in latitude and longitude in degrees [31]. Using the pixel coordinate and the image tile's size information, we find the coordinates of all the pixels in the downloaded image tiles. Because the coordinate information of each image tile is known, it is straightforward to find the coordinate information of the reconstructed color image. The coordinate information of the reconstructed color image is then used to register the elevation maps and GHS built-up layer.

## B. Elevation Map Retrieval

The process discussed in this subsection corresponds to Block-2 in Fig. 1. The elevation map is made possible by the Advanced Spaceborne Thermal Emission and Reflection Radiometer (ASTER) instrument on NASA spacecraft Terra. Global digital elevation model (GDEM) is one of the products of this instrument which was released to users worldwide at no charge [26]. ASTER GDEM is a 1 arc-second elevation grid divided and distributed as  $1^\circ \times 1^\circ$  tiles. To register the ASTER elevation map area to the color satellite image, the corresponding ASTER GDEM tiles that contain the color image area of interest are downloaded from USGS Earth Explorer [27]. The coordinate system of the ASTER GDEM tiles is WGS 84. The latitude and longitude coordinates of the four corners and the pixel size are retrieved from the metadata information of ASTER GDEM tiles. This information is used to register the elevation map to the color satellite image counterpart. Fig. 4 shows the elevation map of the Holly Ridge, NC test area obtained by using ASTER GDEM tiles.

## C. GHS Built-Up Layer

The process introduced in this subsection corresponds to Block-3 in Fig. 1. GHS Layer (GHSL) project funded by the European Commission, Joint Research Centre aims

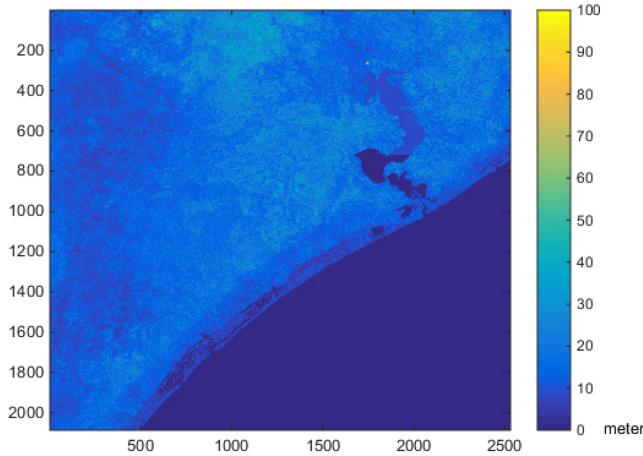


Fig. 4. Elevation map of the Holly Ridge, NC area (darker blue pixels are low elevation areas).

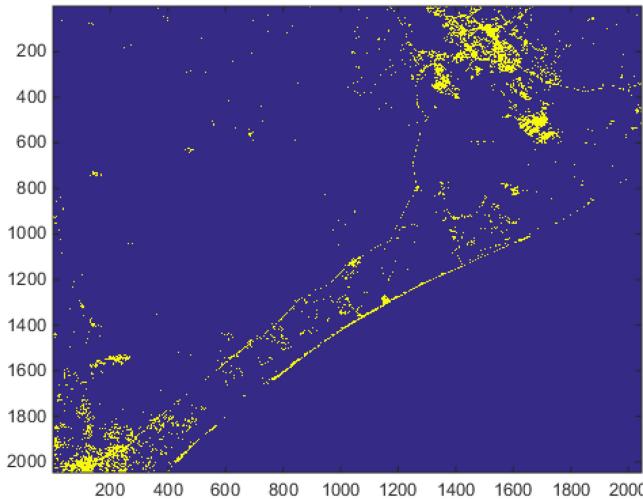


Fig. 5. GHS built-up layer of the Holly Ridge, NC area of about  $\sim 38$  m resolution (highlighted pixels show the built-up areas).

to provide scientific methods and systems for reliable and automatic built-up information gathering [28]. Forty years of Landsat imagery is utilized for mapping the global built-up areas over this period [28]. In GHSL, the built-up area class is defined as the union of all the spatial units containing a building or part of it [28]. Human settlements consist of population and physical infrastructures. Buildings are an essential part of the settlement infrastructure and are considered as the sign of human presence. Two of the safety measures used in our approach are man-made structure density inside the landing site and in the vicinity of the landing site. We use GHS built-up layer to determine these two measures. An example of the GHS built-up layer for one of the test areas we used is shown in Fig. 5. The most-up to date built-up layer belongs to year 2014. The GHS built-up layer is a like a binary image in which the pixels that correspond to buildings have a value of 101 and the pixels that do not contain any buildings have a value of 1. The coordinate system in the GHS built-up layer is WGS 1984 Web Mercator Auxiliary Sphere. The pixel coordinates are

converted to latitude and longitude such that the GHS built-up layer can be registered to the color satellite image.

#### IV. FINDING PRELIMINARY EMERGENCY LANDING AREAS

As shown in the block diagram in Fig. 1, an edge detection process followed by a connected component analysis is essential to identify the preliminary landing areas. Once these preliminary landing areas are identified in the form of connected components, the emergency landing site candidates are determined inside these connected components. This process is explained in detail as follows.

The color image retrieved from Google Maps is at a zoom level of  $z = 13$ . This color image is first transformed into a gray image and Matlab's Canny edge detection method is applied to the gray image using the default lower and upper thresholds [33]. Canny edge detection uses the gradient of the image to find the edges. The gradient of the image is calculated using the derivative of a Gaussian filter. At zoom level,  $z = 13$ , the image resolution for a geographical location at  $36^\circ$  latitude corresponds to approximately 15 m/pixel. The result of the edge detection is a binary image with edge pixels denoted by 1 and nonedge pixels denoted by 0. The edge detection process corresponds to Block-4 in Fig. 1.

Once the binary edge image is computed, the next step is to dilate this binary image with a square-shaped structure object that has a size of  $5 \times 5$ . The dilation process is shown by Block-5 in Fig. 1. The objective of the dilation process is to connect the edge pixels that are close to each other. This helps the connected component analysis, which will be applied to the nonedge pixels, to minimize the number of connected components, thus resulting in a lower computation time. The size of the morphological dilation object is selected such that it is in line with the image resolution of the color image. For a color image retrieved at  $z = 13$  zoom level, the impact of dilation with a  $5 \times 5$  square size morphological object could be thought of connecting the edge pixels that are  $\sim 40$  m apart from each other. After dilation, the binary image is inverted such that the nonedge pixels have value of 1 and the edge pixels have value of 0. This is because the applied connected component requires the pixels that will be connected to have a value of 1. Fig. 6 shows the results of the above-mentioned processes on one of the satellite color images (Holly Ridge, NC) that is used in this paper. The intermediate results for the edge detection and dilation process can be found in the Supplementary Materials Section (Part-1).

A connected component analysis with 8-connected neighborhood connectivity is applied to the inverted binary image using Matlab's image processing toolbox [34]. In the applied connected component analysis, each pixel with value 1 in the binary image is considered as a node. The neighboring node pairs are formed using the 8-neighborhood connectivity. Dulmage-Mendelsohn decomposition [35] of the adjacency matrix of the pairs provides the connected components [34]. This process is shown by

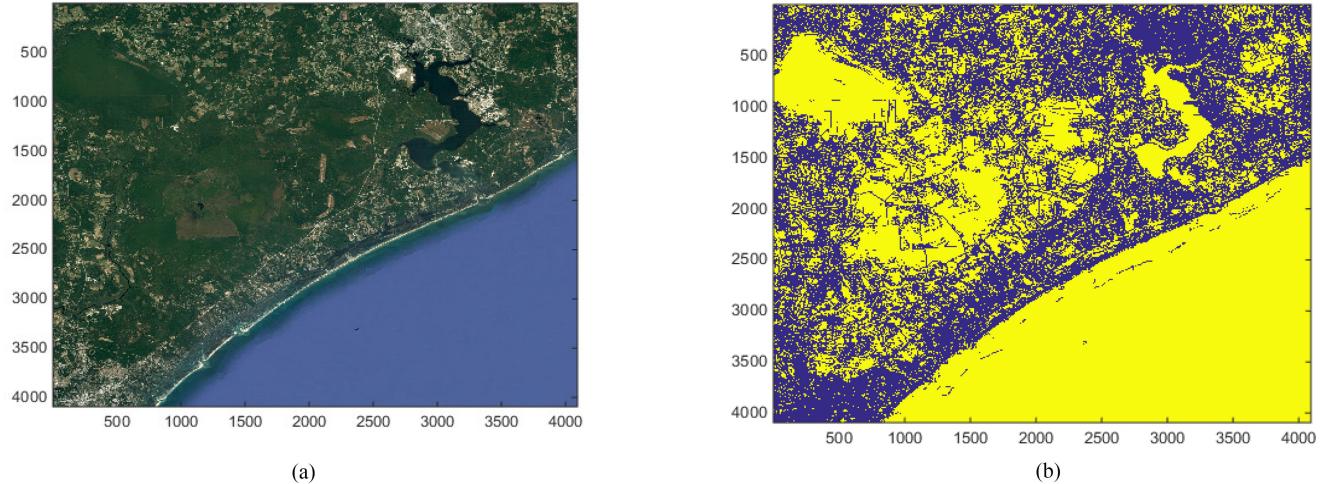


Fig. 6. Edge detection and dilation processes to identify connected components in Holly Ridge, NC area. (a) Original color image for Holly Ridge, NC area (zoom level  $z = 13$ ). (b) Inversed dilated binary edge image.

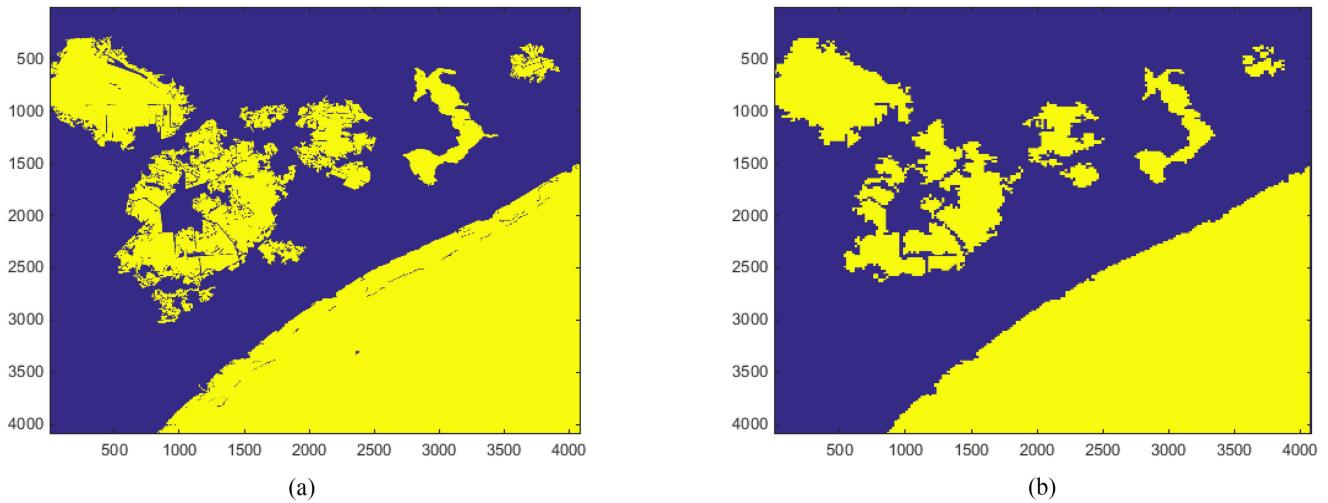


Fig. 7. Identified connected components before and after trimming process. (a) Connected components (before trimming). (b) Connected components (after trimming).

Block-6 in Fig. 1. A connected component analysis result on the Holly Ridge, NC area can be seen in Fig. 7(a). It can be noticed that there are several indented sections inside these connected components especially along the outer regions. Some connected components consist of multiple smaller bodies where the connection between these smaller bodies is via narrow path ways. A trimming process is applied to the connected components to exclude the indented and wavy sections. The connected components are transformed into bodies with majority of its pixels being non-edge pixels. The trimming process involves partitioning the connected component area into a grid of  $30 \times 30$  blocks and checking if 20% of the pixels within these blocks belong to edge pixels. If the percentage of the edge pixels in a block is more than 20%, the pixels in that block are excluded from the connected component. A result of the trimming process on the Holly Ridge, NC image can be seen in Fig. 7(b). It can be noticed that this process reduces the number of these indented sections and the emergency

landing site search can then be applied on nonedge pixel areas within the main body of the connected component. Among the trimmed components, the ones that are larger than a user defined number of pixels (that fulfill the user defined criterion for the emergency landing site size) are considered as the preliminary emergency landing sites and these are used in searching for landing sites in the shape of rectangular runways. The trimming process fills in the holes inside the trimmed connected components. The trimming process is shown by Block-7 in Fig. 1. This could potentially result in identifying some landing site candidates which may not be fully homogeneous. However, one of the safety measures applied to the identified landing sites is the homogeneity measure and this measure is highly sensitive to the detecting landing sites with nonhomogeneous formations. Due to this safety measure, the identified landing sites that have potentially nonhomogeneous formations inside would take lower rankings in the finalized landing site candidate list.

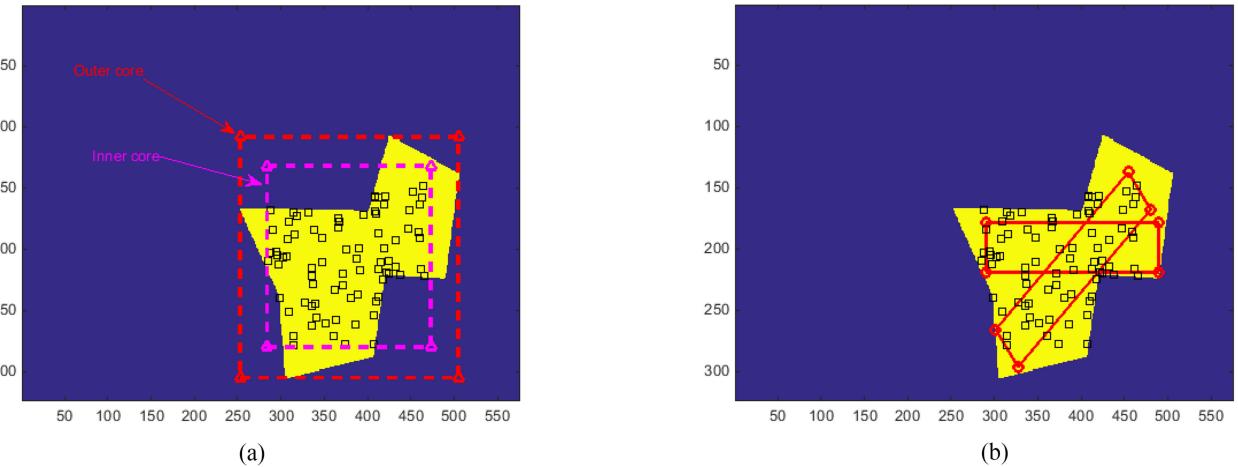


Fig. 8. Finding landing sites in a connected component ( $P = 80$  random points inside the connected component, the searched runway-shaped landing site has a width of 40 pixels and length of 200 pixels). (a) Inner core selection to pick random pixels for landing site centers. (2) Found emergency landing sites that fulfill the user-defined criterion for landing site's width and length (only two sites are found).

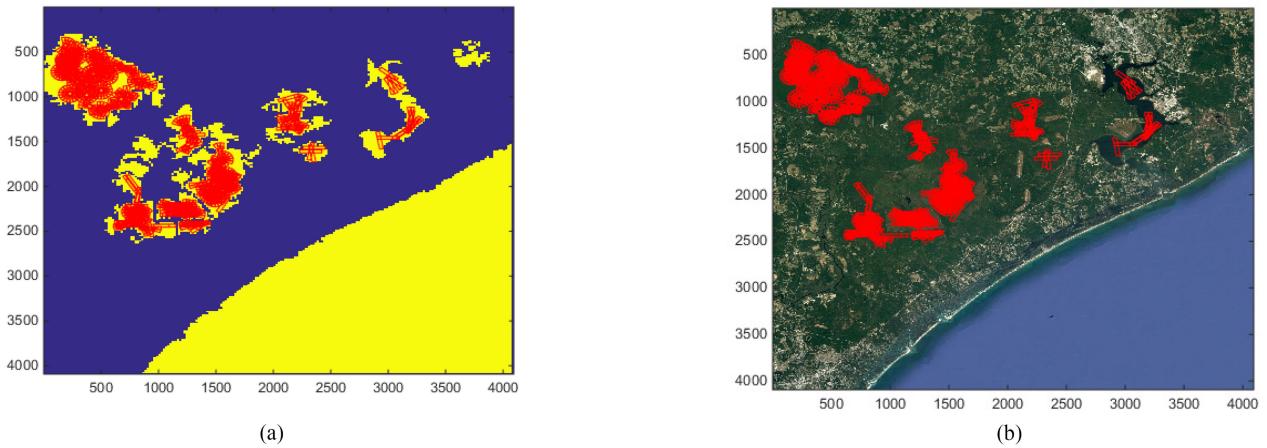


Fig. 9. Identified emergency landing sites in Holly Ridge, NC area. (a) All identified landing sites shown on the connected component (landing site). (b) All identified landing sites shown on the color image.

## V. FINDING EMERGENCY LANDING SITE CANDIDATES

This section corresponds to Block-8 and Block-9 in Fig. 1. In each connected component, emergency landing site candidates are determined as follows. It is assumed that the emergency landing site has a rectangular runway shaped size with width ( $W$ ) and length ( $L$ ). Different values can be set to these parameters depending on the aircraft landing specifications. First,  $P$  random points, inside the inner part of the connected component (preliminary landing site) are identified. It is assumed that these  $P$  points correspond to the rectangular runway-shaped landing site centers. The selection of the inner core of the connected component is demonstrated in Fig. 8(a). First, a rectangle is fit such that the connected component falls inside this rectangle. This rectangle represents the outer core. Then a smaller rectangle of 75% size in both width and length of the outer rectangle is determined. This second rectangle represents the inner core of the connected component. For each identified center, it is checked if a rectangular

runway-shaped mask of width  $W$  and length  $L$  (in pixels) can be placed inside the connected component. If a rectangular runway-shaped mask of width  $W$  and length  $L$  (in pixels) cannot be placed inside the connected component, a landing site cannot be found. Different orientations of the rectangular mask with angles scanned from  $0^\circ$  to  $180^\circ$  with a step size of  $20^\circ$  are considered. The ones that fulfill the searched criterions are kept. Fig. 8(b) shows how a landing site of user-defined size (width and length) is found inside a preliminary landing site (connected component). In this demonstration, the runway width of interest is set to 40 pixels; the runway length is set to 200 pixels. A total of 80 points are randomly picked inside the inner core of the connected component,  $P = 80$ . Fig. 9 shows the demonstration of landing site selection using the color image in the Holly Ridge, NC area. The image has a zoom level of 13,  $z = 13$ , ( $\sim 15$  m resolution) and the area corresponds to approximately 40 by 40 miles in size. The width and length of interest of the runway-shaped landing site corresponds to 500 and 3000 m, respectively.

Several measures are applied to these coarsely identified emergency landing site candidates to assess their safety and generate a ranking. These measures will be described in the following section.

## VI. MEASURES TO ASSESS THE SAFETY OF THE EMERGENCY LANDING SITE CANDIDATES

A total of five safety measures and surface type estimation are applied to the coarsely identified emergency landing site candidates to generate a ranking from the safest to least safe for each surface type. The process of applying the five safety measures to landing site candidates is shown by Block-10 in Fig. 1. In the examples used to introduce these measures, the emergency landing site candidates are set to have a length of 3000 m and width of 500 m. In the following, all five safety measures and surface type estimation are described. Demonstrations of these five measures and the surface type estimation can be found in the Supplementary Materials Section (Part-2–Part-8).

### A. “Homogeneity” Measure

This measure aims to find how homogeneous the emergency landing site candidate is by assessing the pixels that form the landing site. The color image with a zoom level of  $z = 13$  is used with this measure. If there is only one surface type of pixels (such as only grass, or only water) within the landing site, this measure provides lower values indicating that the landing site is composed of a homogeneous surface type. First,  $k$ -means clustering [36] is applied to the RGB values of the landing site candidate’s pixels. The clusters that have fewer pixels (a threshold for the minimum number of pixels is set by the user) are eliminated. This is followed by computing the norm values of the remaining cluster centers. The difference between the maximum and minimum norm values is then set as the homogeneity measure.

### B. Man-Made Structure Density Measure Within the Emergency Landing Site

We use the GHS built-up layer to compute the man-made structure density measure within the emergency landing site. The corresponding pixels for the landing site candidate are first determined in the GHS built-up layer image through a registration process. A binary mask image is then generated for the GHS built-up layer image in which the landing site candidate pixels have a binary value of 1 and 0 otherwise. This mask is then used to retrieve the GHS built-up layer values for all the landing site candidate pixels. Suppose  $N_{\text{total}}$  is the number of pixels of a landing site candidate. Suppose  $n_{\text{int}}$  of  $N_{\text{total}}$  is the number of pixels that correspond to buildings. The man-made structure density measure is then computed as

$$\text{Man-made structure density measure} = \frac{n_{\text{int}}}{N_{\text{total}}}. \quad (4)$$

### C. Man-Made Structure Density Measure in the Vicinity of the Emergency Landing Site

This measure, which also utilizes GHS built-up layer, provides a quantitative measure about the density of man-made structures in the vicinity of the landing site. The vicinity is defined as a circle-shaped region excluding the landing site candidate. The radius of the circular region is a user defined parameter and is set to 4.5 km in the demonstrations. The center of the circle is set to the center of the landing site candidate. Suppose  $N_{\text{vic}}$  is the number of pixels in the vicinity of a landing site candidate (excluding the landing site candidate). Suppose  $n_{\text{vic}}$  of  $N_{\text{vic}}$  is the number of pixels that correspond to buildings. The man-made structure density measure in the vicinity is then computed as

Man-made structure density measure in the vicinity

$$= \frac{n_{\text{vic}}}{N_{\text{vic}}}. \quad (5)$$

### D. Maximum Elevation Difference Along the Landing Site Candidate

This measure uses ASTER GDEM and extracts the elevation profile that passes through the center of the landing site along its length. Suppose the maximum elevation along the length of the landing site that passes through the center is  $E_{\max}$  and the minimum elevation is  $E_{\min}$ . The absolute difference between the maximum and minimum elevations is set as the measure, as is shown as follows:

$$\text{abs}(E_{\max} - E_{\min}). \quad (6)$$

### E. Elevation Difference Along the Two Approaching Sides of the Landing Site Candidate

The identified emergency landing site can be approached from two sides for landing. It is possible that even though the elevation difference within the landing site itself is small, the elevation difference on any of the two approaching sides could be too big which could pose some sort of danger to the UAV (due to a mountain, hill, tall structures, etc.) during landing. This measure extracts the elevation profiles on both approaching sides of the runway-shaped landing site. The same length as the landing site is considered on both approaching sides. Similarly, it finds the elevation difference on both approaching sides and sets the minimum value of the two as the final measure value. Suppose the maximum and minimum elevations along two approaching sides of the landing site are denoted by  $E_{\max_h}$  and  $E_{\min_h}$  where  $h = 1, 2, .$ , the measure is then mathematically expressed as follows:

$$\min (\text{abs}(E_{\max_1} - E_{\min_1}), \text{abs}(E_{\max_2} - E_{\min_2})) \quad (7)$$

This is considered based on the assumption that it would be preferable to use the approaching side with the minimum elevation difference for the UAV to land.

## F. Surface Type Estimation

The surface type of the landing site candidate is another criterion to take into consideration for emergency landing. As an example, it could be preferable for the UAV to land on grass instead of an area covered with trees since this will cause less damage to the UAV when compared to landing on a tree covered area. In conducting surface type estimation, texture features were extracted using Gabor filter [37] so that the surface type estimations could be more robust to varying illumination conditions, or seasonal color changes that affect the surface (such as trees and grass changing color in fall season, etc.) [38]. For comparison purposes, the performance of color related features was studied, but it is observed that the Gabor texture features had more discriminative power.

An investigation was conducted to identify which image resolution provided the highest separation and correct classification performance among the samples of three different surface types with Gabor texture features. For this, the collected image tiles for three different surface types at seven different zoom levels were used. The three considered surface types are: 1) grass/soil; 2) tree; and 3) inland water. Forty samples for each of the three surface types are collected at seven different image resolutions (zoom level) which make a total of 120 geographic locations, 840 image tiles total. The zoom levels are:  $z = 13, 14, 15, 16, 17, 18$ , and  $19$ . As an example, a geographical location at latitude of  $36^\circ$  N:  $z = 13$  corresponds to an image resolution of  $\sim 15.46$  m/pixel, while  $z = 19$  corresponds to an image resolution of  $\sim 0.24$  m/pixel. The image tiles at each different resolution are selected such that they correspond to approximately the same area in size “155 by 155 m.” The sizes of the retrieved image tiles at each different zoom levels are:  $10 \times 10$  (for  $z = 13$ ),  $20 \times 20$  (for  $z = 14$ ),  $40 \times 40$  (for  $z = 15$ ),  $40 \times 40$  (for  $z = 16$ ),  $80 \times 80$  (for  $z = 17$ ),  $160 \times 160$  (for  $z = 18$ ),  $320 \times 320$  (for  $z = 19$ ),  $620 \times 620$  (for  $z = 19$ ).

Gabor texture features are extracted at four orientations with three different sets of scales. These three different sets of features are listed as follows: 1) scale set1 (eight features): Four orientations ( $0^\circ, 45^\circ, 90^\circ, 135^\circ$ ) and two scales (wavelength = 3.2, 6.72 pixels/cycle); b) scale set2 (12 features): Four orientations ( $0^\circ, 45^\circ, 90^\circ, 135^\circ$ ) and three scales (wavelength = 3.2, 6.72, and 14.11 pixels/cycle); and c) scale set3 (16 features): Four orientations ( $0^\circ, 45^\circ, 90^\circ, 135^\circ$ ) and four scales (wavelength = 3.2, 6.72, 14.11, and 29.635 pixels/cycle). Gabor filters are applied to the image tiles of three surface types at seven different image resolutions and the mean amplitudes of the filtered images (at each scale and orientation combination) are set as the Gabor texture features. Gabor filters at wavelength = 3.2 pixels/cycle at four orientations can be seen in Fig. 10.

Two performance measures were used to identify which image resolution provides the highest separation and correct classification with the extracted Gabor texture features. These are: 1) Fisher’s criterion for multiple classes

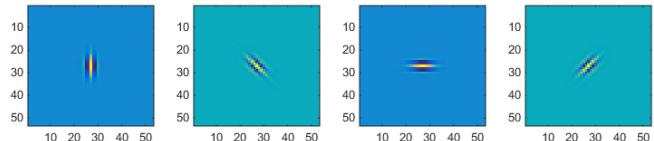


Fig. 10. Gabor filter responses (real component) at wavelength = 3.2 pixels/cycle at four orientations.

TABLE I  
Fishers Criterion values for Separability of the Gabor Features

NumScales	$z=13$	$z=14$	$z=15$	$z=16$	$z=17$	$z=18$	$z=19$
2 (8 features)	0.626	1.6868	5.9461	10.7759	12.3737	10.4477	4.5913
3 (12 features)	0.6828	1.8543	6.7908	12.2009	13.8598	16.0314	11.5554
4 (16 features)	0.7522	1.8678	7.5589	13.2511	13.9191	18.6737	17.4307

TABLE II  
Correct Classification Rates Using Leave-One-Out Classification SCHEME With an LDA Classifier

NumScales	$z=13$	$z=14$	$z=15$	$z=16$	$z=17$	$z=18$	$z=19$
2 (8 features)	0.575	0.7917	0.8833	0.875	0.8083	0.825	0.8333
3 (12 features)	0.5833	0.7333	0.875	0.9167	0.8417	0.825	0.8833
4 (16 features)	0.5833	0.7417	0.8667	0.8917	0.9	0.825	0.9

[39], and 2) correct classification rates in which a leave-one-out (LOO) classification scheme [40] is applied with linear discriminant analysis (LDA) [41] as the classifier. LDA was used for this analysis since it is a basic yet an effective classifier and does not require any parameter selection. In extracting Gabor texture features, four orientations with three different scale sets are considered.

Tables I and II show the Fisher’s criterion values for separability of the Gabor texture features and the correct classification rates using a leave-one-out classification scheme with an LDA classifier at seven different image resolutions. It can be noticed from Tables I and II that using the Gabor texture features extracted from higher resolution color images, with four scales (16 features), the correct classification rates and separability of the Gabor texture features improve significantly. The highlighted cells in Tables I and II correspond to maximum values.

From the applied analyses for surface type estimation, it is observed that applying surface type estimation using higher resolution color images has a significant positive impact on the estimation accuracy when compared to using low resolution color images with Gabor texture features. For surface type estimation, color images of high resolution ( $z = 19$ ) were used to extract 16 Gabor texture features (four orientations and four scales). The process of downloading high-resolution ( $z = 19$ ) color images is shown by Block-11 in Fig. 1. Even though an LDA classifier was used to determine which image resolution to use for surface type estimation with Gabor texture features, in the applied approach, a support vector machine (SVM) classifier was also used with the extracted Gabor texture features. The texture features to train the SVM model and to estimate the surface type are extracted from the center region of the landing site candidates corresponding to an area of approximately 155 by 155 m in size. The SVM classifier is a one-versus-one type that supports multiclass classification [42]–[45].

TABLE III  
Example of User-Set Weights for the Five Measures

Measure	Weight
Homogeneity	1.0
Man-made structure density within the landing site	1.0
Man-made structure density in the vicinity of the landing site	0.1
Elevation difference within the landing site	0.9
Elevation difference in the two approaching sides of the landing site	0.5

A Gaussian kernel is used in the SVM classifier. Because SVM is a nonlinear classifier, it provides better classification accuracy when compared to an LDA classifier. The surface type estimation process is shown by Block-12 in Fig. 1.

## VII. GENERALIZED SAFETY SCORE GENERATION

The identified emergency landing site candidates are ranked using a weighted sum fusion of the five measures. Each measure is normalized between 0 and 1 before fusion. The weighting factors assigned to the normalized measures can be changed by the operators to prioritize some of the measures over others. It should be noted that the surface type estimate is not used in the weighted sum fusion when computing the generalized safety score. The ranked landing site candidates are grouped for each surface type and the sites with lowest scores are considered as safer for that surface type. When one looks at these fused measures, the two man-made structure density measures relate to the safety of the people and property on land. The homogeneity and the two elevation measures relate to the safety of the UAV. Moreover, even though they are not included in the fusion, the surface type and reachability attributes also relate to the safety of people, property, and UAV. When calculating the generalized safety score of a landing site candidate after fusion, the score considers these three criterions (safety of people, property, and UAV).

An example set of weighting factors for the five safety measures is shown in Table III. In this set of weights, more importance is given to three of the five measures: 1) homogeneity; 2) man-made structure density within the landing site; and 3) elevation difference within the landing site. This will result in the landing sites that have lower values for these three measures to have better rankings. These weighting factors, in the end, need to be based on expert knowledge (mission planner, strategist, operator, etc.). Here, we selected these values for demonstration purposes only. The ranking generation process and the ranked landing site candidates for each surface type correspond to Block-13 and Block-14 in Fig. 1.

## VIII. REACHABILITY ANALYSIS

A reachability label generation capability in the presence of wind is incorporated into the proposed approach given the location and altitude information of the aircraft at the time of emergency and the wind forecast for the area. This is to help the operator with the decision process for identifying emergency landing site candidates along the primary flight path of the UAV in case a total loss of thrust

emergency happens. Since the waypoints of the primary flight path and the corresponding altitude at each waypoint are planned by the flight management system, the operator could use the reachability analysis to help select the emergency landing sites for each waypoint along the primary flight as a part of preflight contingency planning.

For reachability analysis, it is assumed that a wind forecast data for that spatial location and the flight time is available. We use the hourly updated wind forecast data from National Oceanic & Atmospheric Administration's National Centers for Environmental Prediction (NCEP) [46]. We retrieve the wind velocities and their angular directions for the area of interest at 37 different vertical pressure levels (100–1000 mb) where the pressure in millibar can be converted to pressure altitude. The wind forecast data at two of these pressure altitude levels that are closest to the aircraft altitude are interpolated to estimate the wind forecast data at the UAV altitude and position.

In the applied reachability analysis, the polar curve of the aircraft is utilized. A polar curve shows the relation between the airspeed and sink rate of the aircraft. Suppose  $W$  is the aircraft weight at the contingency point,  $S$  is the wing reference area,  $A$  is the wing aspect ratio,  $C_{D0}$  is the flags-up parasite drag coefficient,  $e$  is the airplane, and  $\rho_0$  is the air density at UAV's altitude. The equation that enables the computation of sink rate  $V_s$ , for any given airspeed of the aircraft  $V$ , is shown in (8) [47]. This equation is derived using the lift and drag information in which lift is approximated to be equal to weight assuming small glide angle and small angle of attack

$$V_s = \frac{0.5\rho_0 S C_{D0}}{W} V^3 + \frac{2V}{\rho_0 S \pi A e V}. \quad (8)$$

The reachability analysis conducted as a part of the proposed approach is not a true reachability analysis since the turn path, excessive altitude loss path, and final approach path trajectories are not designed. It is a coarse reachability analysis indicating whether the aircraft can reach the landing site or not. It assumes that the mishap aircraft can head to the emergency landing site right after the emergency happens and the time required to make the turn to the landing site is not considered. It is assumed that the aircraft can continuously adjust its airspeed  $V$ , using its surface controls such that its airspeed provides the best glide ratio value when gliding from the contingency point, CP, to the emergency landing site candidate LP, in the presence of wind. A straight path is considered between CP and LP. Suppose the course angle of the aircraft from CP to LP is  $\chi$ . The straight path between CP and LP is partitioned into  $N_s$  segments of one-mile length each, where each segment is denoted by  $d_i$  where  $i = 1, \dots, N_s$ . Within a segment, a steady wind condition is considered. Since the overall straight path  $P$ , consists of multiple segments, the wind forecast could be different in other segments. The wind forecast data consist of wind speed  $V_w$  and wind direction  $X_w$ . For each segment  $d_i$ , the airspeed that provides the best glide ratio  $V_{bg}^i$ , according to the wind forecast for that geographical coordinates

and altitude is found. When finding  $V_{bg}^i$ , all the airspeed ( $V_j$ ) and sink rate ( $V_s^j$ ) data points in the polar curve of the aircraft at that altitude are checked with the wind forecast values. This is to compute the corresponding groundspeed value  $V_{gs}^j$  that meets the required course angle  $\chi$  according to wind triangle computations [48], where  $j = 1, \dots, M_p$  and  $M_p$  is the total number of airspeed and sink rate data points in the polar curve. The index of the data point  $j_{bg}$  in the polar curve that provides the best glide ratio is mathematically expressed as

$$j_{bg} = \operatorname{argmax}_{j \in \{1, 2, \dots, M_p\}} \left( \frac{V_{gs}^j}{V_s^j} \right). \quad (9)$$

The best glide ratio airspeed  $V_{bg}^i$  is then set to the airspeed of the polar curve point with index  $j_{bg}$ ,  $V_{bg}^i = V_{j_{bg}}$ . Suppose the corresponding sink rate and the groundspeed of the aircraft for  $V_{bg}^i$  are  $V_{sg}^i$  and  $V_{gs_{bg}}^i$ , respectively. Suppose the length of  $d_i$  is  $l_i$ . One can then compute the travel time for the aircraft  $t_i$  as:  $t_i = l_i / V_{gs_{bg}}^i$ . The altitude drop  $\Delta h_i$  for  $d_i$  is computed using the sink rate,  $V_{sg}^i$  and the travel time  $t_i$  as:  $\Delta h_i = V_{sg}^i t_i$ . With the altitude drop information, the altitude of the aircraft at the start point of the new segment  $d_{i+1}$  can be computed as:  $H_{i+1} = H_i - \Delta h_i$  where  $H_i$  corresponds to aircraft altitude at the start point of segment  $d_i$ . The height of the aircraft corresponds to the difference between the altitude of the aircraft and the elevation of the emergency landing site of interest. The height of the aircraft at the end of segment  $d_i$  becomes,  $H_i - \Delta h_i - e_{LP}$ , where  $e_{LP}$  is the elevation of the emergency landing site. If the aircraft's height becomes negative before the aircraft reaches LP, the reachability score label is automatically set to "unreachable."

The process mentioned above is repeated until all the segments between CP and LP are completed. It is assumed that the heading angle adjustment to stay on the same course angle and airspeed adjustment to maintain the best glide ratio when moving from one segment to the next one occurs instantaneously. The segmentation of the path allows incorporation of varying wind forecast data into the applied reachability analysis. It is worth mentioning that the best glide ratio is equal to the maximum  $L/D$  where  $L$  is the lift and  $D$  is the drag in no wind conditions. However, in the presence of wind, the best glide ratio is not equal to the maximum  $L/D$  since glide ratio is dependent on the ground speed and sink rate of the aircraft. An illustration of polar curves and  $L/D$  plots of the used aircraft at two different altitudes can be found in the Supplementary Materials Section (Part-9).

For reachability score labels, we consider five different levels. These are: 1) reachable with high excessive altitude; 2) reachable with small excessive altitude; 3) barely reachable; 4) uncertain; and 5) not reachable. We assign these labels based on the final height of the aircraft when it is over the emergency landing site. An altitude margin of 1000 ft is considered for the unaccounted times for aircraft

control adjustments to maintain the best glide ratio speed and for heading angle adjustments to stay on the course according to varying wind at each new segment. The first label, "reachable with high excessive altitude," is assigned if the final height is over 5000 ft. The second label, "reachable with small excessive altitude" is assigned if the final height is between 5000 and 2000 ft. The third label, "barely reachable" is assigned if the final height is between 2000 and 1000 ft. The fourth label, "uncertain" is assigned if the final height is between 1000 and 0 ft. Finally, the label "unreachable" is assigned if the final height is negative. A detailed pseudocode for reachability score label generation in the presence of wind can be found in the Supplementary Materials Section (Part-10). We should mention that the assigned altitude thresholds are for demonstration purposes only and they could be set to more realistic values by experts (operators, mission planners, etc.) depending on the aircraft needs and specifications. The reachability labeling process is shown by Block-15 in Fig. 1.

## IX. RESULTS AND ANALYSIS

The proposed approach was applied to three different geographical areas of interest each approximately 40 by 40 miles in size to identify emergency landing sites which will be used for contingency planning of the flight missions that will take place in these areas. Among the identified landing site candidates, the human operator can review the candidate list that are ranked from the safest to the least safe for each surface estimate type and select the ones to be used in the contingency plans by considering their reachability score labels.

The UAV of interest here is large and close to a passenger jet in size. Because of that in the applied investigations, the emergency landing site size is kept large with a width of 500 m and length of 3000 m. The three geographical areas used in the investigations are: 1) Holly Ridge, NC; 2) Raleigh, NC; and 3) Columbia, SC. After the emergency landing sites are determined by the automated approach for these three areas, all landing site candidates are screened for labeling the ground truth surface type using their high-resolution images at zoom level,  $z = 19$ , since the ground truth surface types are needed for the classification analyses. Even though an SVM classifier is used in the proposed approach, for comparison purposes, an LDA classifier is applied as well. The LDA classifier results for surface type estimates can be found in the Supplementary Materials Section (Part-11). Confusion matrices are generated to compare the surface type classification performance of the two classifiers. Among the three areas, only in Holly Ridge, NC area, the top three emergency landing sites for each of the three surface types are provided with their generalized safety scores. For the remaining two geographical areas, all the identified landing site candidates by the automated approach are shown on the same map and the analysis for surface type estimations are provided. A demonstration of the reachability analysis and labeling of the identified landing site candidates with respect to reachability are provided

TABLE IV

Top Three Landing Sites for Three Surface Types in Holly Ridge, NC Area (Based on SVM Classifier Estimates)

Surface Type Estimate	Homo-geneity	Man-made structures (inside runway)	Man-made structures (vicinity)	Elevation difference inside runway	Min elevation difference (approach sides)	Generalized safety score
Inland water	6.730196	0	0.054808	0	0	0.015001
Inland water	10.33678	0	0.077891	0	0	0.031394
Inland water	6.499878	0	0.054773	0	6	0.048617
Grass/Soil	23.34406	0	7.95E-05	11	10	0.20654
Grass/Soil	20.84453	0	7.96E-05	10	15	0.218752
Grass/Soil	23.0973	0	0	13	10	0.224178
Tree	8.849349	0	0	6	5	0.090418
Tree	10.52932	0	0	5	7	0.097486
Tree	12.14301	0	0	6	5	0.099874

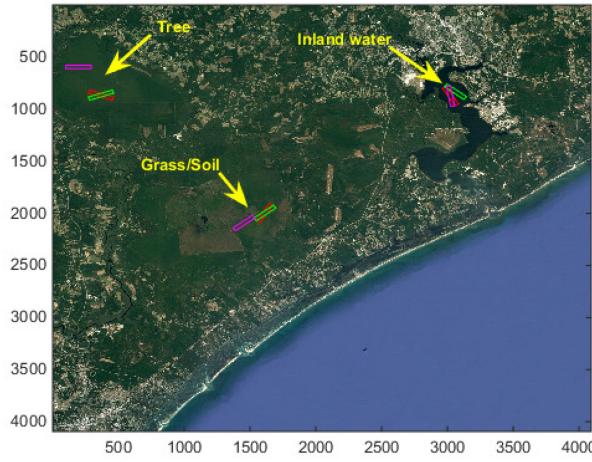


Fig. 11. Top three emergency landing sites for Holly Ridge, NC based on the generalized safety scores for inland water, grass/soil, and tree types (1st: Red, 2nd: Green, 3rd: Purple).

for one of the test areas (Holly Ridge, NC) assuming the location and the altitude of the aircraft at the time of emergency are available.

- A. Test Image: Holly Ridge, NC (latlim = [34.2331 34.8120], lonlim = [-77.9138-77.2112])

Fig. 6-a shows the color image and Fig. 9-b shows identified landing sites in Holly Ridge, NC area. The top three emergency landing sites based on the generalized safety scores for inland water, grass/soil, and tree types are shown in Table IV, which contains the values of the five measures and the generalized safety scores. The top three sites for the three surface types are shown on the same map in Fig. 11. The high-resolution images of the top three identified sites for each surface type can be also found in the Supplementary Materials Section (Part-12). The confusion matrix for the surface type estimations with SVM classifier can be seen in Table V.

- B. Test Image: Raleigh, NC (latlim = [35.4926 36.0715], lonlim = [-78.9973-78.2837])

Fig. 12 shows the color image and identified emer-

TABLE V

Confusion Matrix for Surface Type Estimations for Holly Ridge, NC area (SVM)

	Inland water	Tree	Grass/Soil
Inland water	18	0	0
Tree	0	442	6
Grass/Soil	0	1	105

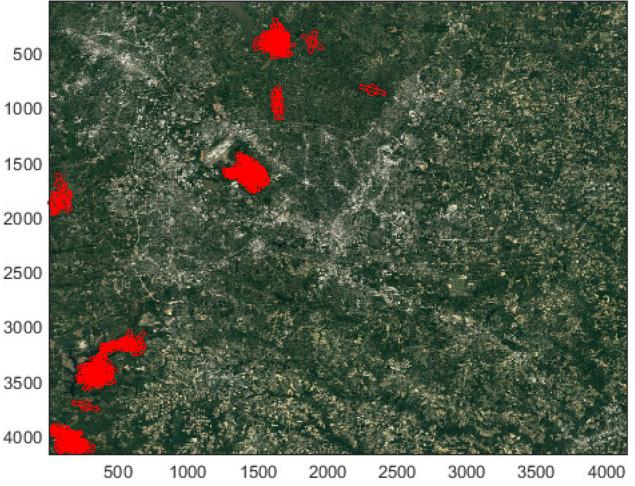


Fig. 12. Color image and identified landing sites in Raleigh, NC.

TABLE VI

Confusion Matrix for Surface Type Estimations for Raleigh, NC (SVM)

	Inland water	Tree	Grass/Soil
Inland water	50	0	5
Tree	0	166	1
Grass/Soil	0	0	5

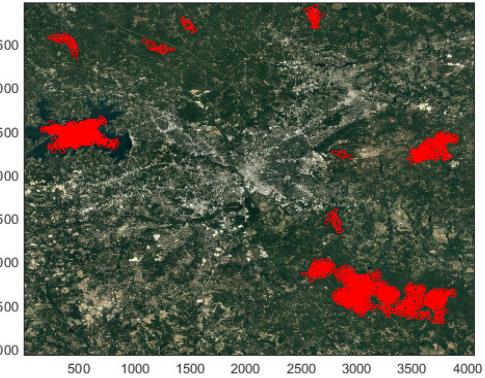


Fig. 13. Color image and identified landing sites in Columbia, SC.

gency landing sites in Raleigh, NC area. Table VI shows the confusion matrix for the surface type estimations with SVM classifier.

- C. Test Image: Columbia, SC (latlim = [33.6982 34.2772], lonlim = [-81.3850-80.6868])

Fig. 13 shows the color image and identified emergency landing sites in Columbia, SC area. There are landing sites that belong to only tree and inland water surface types in this example. Table VII shows the confusion matrix for the

TABLE VII  
Confusion Matrix for Surface Type Estimations  
for Columbia, SC (SVM)

	Inland water	Tree	Grass/Soil
Inland water	165	0	0
Tree	0	294	0
Grass/Soil	0	0	0

TABLE VIII  
Aerodynamic Parameters of the Airbus A320 Used  
in the Simulations [47]

Specs	Symbol	Value	Unit
Aircraft weight	$W$	150871	lbf
Wing reference area	$S$	1318.579	$ft^2$
Wing aspect ratio	$A$	9.5	unitless
Parasite drag coefficient (flags up)	$C_{D0}$	0.022	unitless
Airplane efficiency factor	$e$	0.7697	unitless

surface type estimations with SVM classifier. Both classifiers perform very well in this example.

#### D. Computation Times

Because the proposed approach is an offline approach for preflight contingency planning, the computation time constraints are more lenient than with on-board real-time landing site detection methods. For interested readers, in the Supplementary Materials Section (Part-13), we provided the computation times of our approach for the three sites.

#### E. Demonstration of Reachability Analysis and Label Generation

Even though the targeted air vehicles are large UAVs like Global Hawk [49], we used Airbus A320 in the reachability analysis demonstration since the aerodynamic parameters of this aircraft are available in public domain [47]. These parameters are shown in Table VIII.

In the demonstration, it is assumed that coordinates of the aircraft at the time the total loss of thrust emergency happened are: Latitude:  $34.490573^\circ$ , longitude:  $-77.337412^\circ$  which is considered to belong to one of the waypoints along its primary flight path. The aircraft altitude at the time of emergency is considered to be 10 000 ft. The emergency landing sites for reachability analysis and label generation belong to the 572 landing site candidates from the Holly Ridge, NC test area. The elevations of the emergency landing site candidates are found from elevation maps using ASTER DEM.

The wind forecast data used in this demonstration is an hourly updated wind forecast data from NOAA's NCEP and corresponds to the date of 6/19/2017 at 11:00 am. The wind amplitudes vary spatially, but they do not change with time for one hour. In Part-14 of the Supplementary Materials Section quiver plots of the used wind forecast data are also provided.

In Fig. 14, the landing site candidates are labeled with different colors according to the assigned reachability levels. The position of the aircraft at the time of emergency can be seen with a blue circle. It took about  $\sim 1.23$  s to compute the reachability score label of a single landing

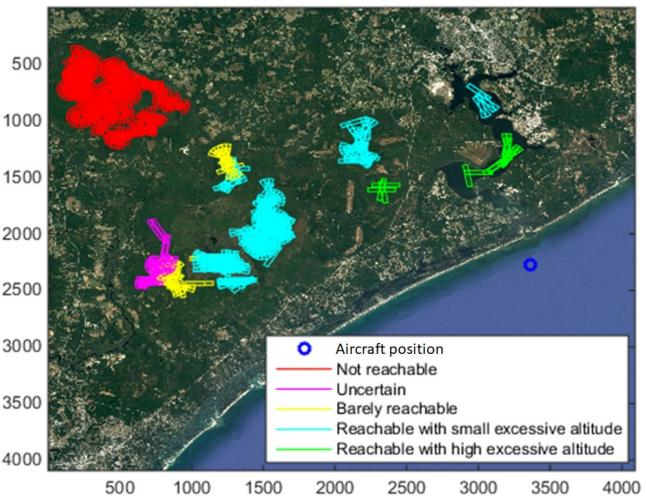


Fig. 14. Labeled landing site candidates with respect to reachability for Holly Ridge, NC test area.

site candidate in the Holly Ridge, NC test area site which had a total of 572 landing site candidates. The same computer system is used in the computations for reachability analysis. It is observed that the top three “tree” landing sites are not reachable when considering the wind forecast and the position of the aircraft at the time of emergency whereas the top three “inland water” and the top three “grass/soil” landing sites are reachable with small remaining excessive altitudes.

#### X. CONCLUSION AND FUTURE WORK

The proposed approach identifies emergency landing sites for preflight contingency planning. The approach is expected to ease the job of the human operators significantly. Instead of tediously looking for potential emergency landing sites, the human operators would only review a few candidates and assign them as emergency landing sites in the contingency plans along the flight path. Any potential human-based errors are thought to be reduced with the use of a semi-automated approach like this one. A few areas to further improve this approach have been identified throughout our work. One improvement area is the integration of publicly available high-resolution land type maps, from Landsat, [50] or even generating our own surface type maps for the area of interest. These maps could be directly used in our approach instead of performing the surface type estimation on the fly. Utilizing such a surface type map could significantly reduce the computation times since surface type estimation is the most time-consuming part. Another improvement is to conduct a true reachability analysis with trajectory design of the turn paths, excessive altitude loss paths, and final approach paths. Our approach in its current form can be considered to have Level-1 autonomy (semi-autonomous) regarding noncontextual autonomy potential level [51] since the human operator still needs to make the final decision. Related with that, another future improvement area would be to develop a machine intelligence that will imitate the decision process of the experts in this area (mission planner or strategist, pilot,

etc.) such that instead of providing separate safety rankings of landing site candidates for each surface type, the landing site can be directly selected without any human interaction.

This approach is new, unique, and fills a void in pre-flight contingency planning for UAVs. We have developed a software prototype for automated landing site detection, which would be an integral part of a preflight contingency planning system for UAVs that we are also currently developing. One of our plans is also to customize our approach for on-board use which would incorporate true reachability and weather condition for improved emergency landing site identification, fusion of cloud-based satellite imagery with on-board camera images, and real-time sensor feedback.

## ACKNOWLEDGMENT

The authors would like to thank the reviewers for their constructive comments. The views, opinions, and/or findings expressed are those of the author(s), and should not be interpreted as representing the official views or policies of the Department of Defense or the U.S. Government. Distribution Statement “A” (Approved for Public Release, Distribution Unlimited).

## REFERENCES

- [1] P. Ballal *et al.*  
Mechanical fault diagnosis using wireless sensor networks and a two-stage neural network classifier  
in *Proc. IEEE Aerosp. Conf.*, 2009, pp. 1–10.
- [2] X. Zhao *et al.*  
Active health monitoring of an aircraft wing with embedded piezoelectric sensor/actuator network: I. Defect detection, localization and growth monitoring  
*J. Smart Mater. Struct.*, vol. 16, no. 4, pp. 1208–1217, 2007.
- [3] G. Zhang, C. Kwan, R. Xu, N. Vichare, and M. Pecht  
An enhanced prognostic model for intermittent failures in digital electronics  
in *Proc. IEEE Aerosp. Conf.*, 2007, pp. 1–8.
- [4] C. Kwan and R. Xu  
A note on simultaneous isolation of sensor and actuator faults  
*IEEE Trans. Control Syst. Technol.*, vol. 12, no. 1, pp. 183–192, Jan. 2004.
- [5] R. Xu and C. Kwan  
Robust isolation of sensor failures  
*Asian J. Control*, vol. 5, no. 1, pp. 12–23, Mar. 2003.
- [6] X. D. Zhang, Y. Liu, R. Rysdyk, C. Kwan, and R. Xu  
An intelligent hierarchical approach to actuator fault diagnosis and accommodation  
In *Proc. IEEE Aerosp. Conf.*, 2006, pp. 1–15.
- [7] C. Kwan  
On variable structure output feedback controllers  
*IEEE Trans. Autom. Control*, vol. 41, no. 11, pp. 1691–1693, Nov. 1996.
- [8] C. Kwan, H. Xu, and H. Xu  
Robust spacecraft attitude control using adaptive fuzzy logic  
*Int. J. Syst. Sci.*, vol. 31, no. 10, pp. 1217–1225, 2000.
- [9] M. Ciuryla, Y. Liu, J. Farnsworth, C. Kwan, and M. Amitay  
Flight control using synthetic jets on a Cessna 182 model  
*J. Aircr.*, vol. 44, no. 2, pp. 642–653, 2007.
- [10] M. Polycarpou, X. Zhang, R. Xu, Y. Yang, and C. Kwan  
A neural network based approach to adaptive fault tolerant flight control  
In *Proc. IEEE Int. Symp. Intell. Control*, 2004, pp. 61–66.
- [11] E. Pastor, P. Royo, E. Santamaría, X. Prats, and C. Barrado  
In-flight contingency management for unmanned aerial vehicles  
*J. Aerosp. Comput., Inf. Commun.*, vol. 9, no. 4, pp. 144–160, 2012.
- [12] M. Warren, L. Mejias, X. Yang, B. Arain, F. Gonzalez, and B. Upcroft  
Enabling aircraft emergency landings using active visual site detection  
In *Field and Service Robotics*, New York City, NY: Springer International Publishing, pp. 167–181, 2015.
- [13] X. Sun, C. M. Christoudias, V. Lepetit, and P. Fua  
Real-time landing place assessment in man-made environments  
*Mach. Vision Appl.*, vol. 25, no. 1, pp. 211–227, 2014.
- [14] P. T. Eendebak, A. W. M. van Eekeren, and R. J. M. den Hollander  
Landing spot selection for UAV emergency landing  
In *Proc. SPIE-Int. Soc. Opt. Photon.*, 2013, pp. 1–9.
- [15] Y. F. Shen, Z. U. Rahman, D. Krusinski, and J. Li  
A vision-based automatic safe landing-site detection system  
*IEEE Trans. Aerosp. Electron. Syst.*, vol. 49, no. 1, pp. 294–311, 2013.
- [16] D. Fitzgerald  
Landing site selection for UAV forced landings using machine vision  
Ph.D. dissertation, School of Eng. Syst., Queensland Univ. Technol., Brisbane, Australia, 2007.
- [17] D. Fitzgerald, R. Walker, and D. Campbell  
A vision based forced landing site selection system for an autonomous UAV  
In *Proc. Int. Conf. Intell. Sensors, Sensor Netw. Inf. Process.*, Melbourne, Australia, 2005, pp. 397–402.
- [18] L. Mejias, D. Fitzgerald, P. Eng, and X. Liu  
Forced landing technologies for unmanned aerial vehicles: Towards safer operations  
in *Aerial Vehicles*, Thanh Mung Lam, Ed., Rijeka, IntechOpen, 2009, ch. 21.
- [19] E. M. Atkins, I. A. Portillo, and M. J. Strube  
Emergency flight planning applied to total loss of thrust  
*J. Aircr.*, vol. 43, no. 4, pp. 1205–1216, 2006.
- [20] M. Coombes, W. H. Chen, and P. Render  
Site selection during unmanned aerial system forced landings using decision-making bayesian networks  
*J. Aerosp. Inf. Syst.*, vol. 13, pp. 491–495, 2016.
- [21] T. Lerova, O. Bruna, and P. Paces  
Emergency landing site location using aerial image segmentation  
In *Proc. 29th Congr. Int. Council Aeronautical Sci.*, 2014, pp. 1–6.
- [22] P. J. Garcia-Pardo, G. S. Sukhatme, and J. F. Montgomery  
Towards vision-based safe landing for an autonomous helicopter  
*Robot. Auton. Syst.*, vol. 38, no. 1, pp. 19–29, Jan. 2002.
- [23] M. M. Dehsheibi, M. S. Fahimi, and M. Mashhadi  
Vision-based site selection for emergency landing of UAVs  
In *Recent Advances in Information and Communication Technology*, New York City, NY: Springer International Publishing, pp. 133–142, 2015.
- [24] P. Ahmadi, S. Sadri, R. Amirkattahi, and N. Gheissari  
Automatic aerial image segmentation based on a modified Chan-Vese algorithm  
in *Proc. 5th Int. Congr. Image Signal Process.*, Chongqing, China, 2012.
- [25] A. Howard and S. Homayoun  
A fuzzy rule-based safety index for landing site risk assessment  
in *Proc. 5th Biannual World Autom. Congr.*, 2002, pp. 1–6.
- [26] Advanced Spaceborne Thermal Emission and Reflection Radiometer. [Online]. Available: <https://asterweb.jpl.nasa.gov/gdem.asp>
- [27] USGS Earth Explorer. [Online]. Available: <https://earthexplorer.usgs.gov/>

- [28] *GHSL- Global Human Settlement Layer*. [Online]. Available: <http://ghsl.Joint Research Centre.ec.europa.eu/>
- [29] *Google Maps APIs Guides*. [Online]. Available: <https://developers.google.com/maps/documentation/javascript/maptypes#TileCoordinates>
- [30] *Microsoft Developer Network Documentation, Understanding Scale and Resolution*. [Online]. Available: <https://msdn.microsoft.com/en-us/library/aa940990.aspx>
- [31] *Google Maps API*. [Online]. Available: <https://developers.google.com/maps/documentation/javascript/examples/map-coordinates>
- [32] *USGS Long Term Archive, National Elevation Dataset (NED)*. [Online]. Available: <https://lta.cr.usgs.gov/NED>
- [33] *MathWorks Documentation, Image Processing Toolbox*. [Online]. Available: [https://www.mathworks.com/help/images/ref/edge.html?searchHighlight=canny&s\\_tid=doc\\_srcTitle](https://www.mathworks.com/help/images/ref/edge.html?searchHighlight=canny&s_tid=doc_srcTitle)
- [34] *MathWorks Documentation, Image Processing Toolbox*. [Online]. Available: [www.mathworks.com/help/images/ref/bwconvncomp.html](https://www.mathworks.com/help/images/ref/bwconvncomp.html)
- [35] A. Pothen and C. Fan  
Computing the block triangular form of a sparse matrix  
*ACM Trans. Math. Software*, vol 16, no. 4, pp. 303–324, 1990.
- [36] A. Jain  
Data clustering: 50 years beyond K-means  
*Pattern Recognit. Lett.*, vol. 31, no. 8, pp. 651–666, 2010.
- [37] B. S. Manjunath and W. Y. Ma  
Texture features for browsing and retrieval of image data  
*IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 18, no. 8, pp. 837–842, Aug. 1996.
- [38] J. K. Kamarainen, V. Kyrki, and H. Kalviainen  
Invariance properties of Gabor filter-based features-overview and applications  
*IEEE Trans. Image Process.*, vol. 15, no. 5, pp. 1088–1099, May 2006.
- [39] T. Li, S. Zhu, and M. Ogihara  
Using discriminant analysis for multi-class classification: An experimental investigation  
*Knowl. Inf. Syst.*, vol. 10, no. 4, pp. 453–472, 2006.
- [40] M. A. Brovelli, M. Crespi, F. Fratarcangeli, F. Giannone, and E. Realini  
Accuracy assessment of high-resolution satellite imagery orientation by leave-one-out method  
*ISPRS J. Photogrammetry Remote Sens.*, vol. 63, no. 4, pp. 427–440, 2008.
- [41] R. O. Duda, P. E. Hart, and D. G. Stork  
*Pattern Classification*. Hoboken, NJ: Wiley, 2012.
- [42] M. Galar, A. Fernández, E. Barrenechea, H. Bustince, and F. Herrera  
An overview of ensemble methods for binary classifiers in multi-class problems: Experimental study on one-vs-one and one-vs-all schemes  
*Pattern Recognit.*, vol. 44, no. 8, pp. 1761–1776, 2011.
- [43] T. Qian, X. Li, B. Ayhan, R. Xu, C. Kwan, and T. Griffin  
Application of support vector machines to vapor detection and classification for environmental monitoring of spacecraft in *Proc. 3rd Int. Symp. Neural Netw.*, 2006, pp. 1216–1222.
- [44] T. Qian, R. Xu, C. Kwan, B. Linnell, and R. Young  
Toxic vapor classification and concentration estimation for space shuttle and international space station in *Proc. Int. Symp. Neural Netw.*, pp. 543–561, 2004.
- [45] B. Ayhan and C. Kwan  
Application of deep belief network to land classification using hyperspectral images in *Proc. 14th Int. Symp. Neural Netw.*, 2017, pp. 269–276.
- [46] U.S. Department of Commerce, National Oceanic & Atmospheric Administration, National Centers for Environmental Prediction. [Online]. Available: <http://www.ncep.noaa.gov/>
- [47] M. Coombes, W. H. Chen, and P. Render  
Reachability analysis of landing sites for forced landing of a UAS  
*J. Intell. Robot. Syst.*, vol. 73, no. 1–4, pp. 635–653, 2014.
- [48] Department of the Air Force Air Navigation,"Department of the Air Force, USA, Rep. AFM 51-40, 1989.
- [49] Northrop Grumman Global Hawk—20 years of flight. [Online]. Available: <http://www.northropgrumman.com/Capabilities/GlobalHawk/Pages/default.aspx>
- [50] USGS Landsat Missions, [Online]. Available: <https://landsat.usgs.gov/ard>
- [51] P. J. Durst and W. Gray  
Levels of autonomy and autonomous system performance assessment for intelligent unmanned systems  
US Army Corps of Engineers, Geotechnical and Structures Laboratory, Vicksburg, MA, USA, Rep. ERDC/GSL SR-14-1, 2014.

**Bulent Ayhan** (M'01) received the Ph.D. degree in electrical engineering from North Carolina State University, Raleigh, NC, USA, in 2005.

He is currently a Principal Research Engineer with Signal Processing, Inc. (SPI), Rockville, MD, USA.

**Chiman Kwan** received the Ph.D. degree in electrical engineering from the University of Texas at Arlington, Arlington, TX, USA, in 1993.

Currently, he is the Chief Technology Officer with Signal Processing, Inc., Rockville, MD, USA, and with Applied Research LLC, Rockville.

**Yool-bin Um** received the B.S. degree in computer science and minor in statistics from the University of Maryland, College Park, MD, USA, in 2016.

He was a Software Developer with SPI.

**Bence Budavari** received the B.S. degree in audio engineering from Belmont University, Nashville, TN, USA, in 2015.

He is a Software Developer with SPI.

**Jude Larkin** received the B.S. degree in computer science from Franciscan University of Steubenville, Steubenville, OH, USA, in 2015.

He is a Software Engineer with SPI.