Sensors and Materials, Vol. 31, No. 3 (2019) 889–903
MYU Tokyo

S & M 1823

Path Planning for Unmanned Surface Vessels with Multisensor Fusion Design

Yung Hsiang Chen,[1] Yung Yue Chen,[2*] and Joe Ming Yang[2]

[1]Department of Mechanical Engineering, National Pingtung University of Science and Technology,
No. 1, Xuefu Rd., Neipu Township, Pingtung County 91201, Taiwan R.O.C.
[2]Department of Systems and Naval Mechatronic Engineering, National Cheng Kung University,
No. 1, Daxue Rd., East Dist., Tainan City 70101, Taiwan R.O.C.

(Received October 15, 2018; accepted January 8, 2019)

Keywords: path planning, aerial photography, image resolution, quadrotor, real-time image

To achieve a successful sea voyage, an effective path planning algorithm plays an important role in avoiding vessel collisions in an autonomous surface vessel design . This algorithm depends on real-time global images of explored environments. Currently, satellite images used for the path planning algorithm always have rough resolutions, which increase the risk of vessel collision. To solve this problem, an alternative solution of integrating the aerial images captured by the image sensor of a flying quadrotor and the global positioning system (GPS), electrical compass, and height sensor data is proposed to provide real-time images in the global coordinate instead of satellite images. Naturally, aerial images possess higher resolution pixels than the satellite images and allow real-time modifications of geographic information. Images captured by the image sensor installed on the quadrotor are essentially represented in local coordinates; hence, in this investigation, a coordinate transformation for converting the local coordinate of the aerial images into the global coordinate is developed and named the distance–resolution transformation. With the fusion of these sensors and designs, the global coordinate of the high-resolution image can be calculated directly, and the real-time path planning using this aerial image can be applied to plan a collision-free path to navigate the unmanned surface vessel in any unknown ocean environment.

1. Introduction

Designs of unmanned vehicles have been an emerging research topic in recent years, since they can be used for military purposes, cargo transport, and environment monitoring.(1,2) Unmanned vehicles are highly useful for missions dangerous to humans. Recently, Amazon launched unmanned aerial vehicles to deliver parcels to clients. The US Navy launches unmanned surface vehicles (USVs) and underwater vehicles to scout and secure harbors. All these applications demonstrate that USVs are becoming popular solutions in different fields. Guiding a USV through complicated ocean environments is highly challenging, and effective path planning algorithms play an important role in overcoming this challenge. Research on

[*]Corresponding author: e-mail: yungyuchen@mail.ncku.edu.tw

https://doi.org/10.18494/SAM.2019.2161

ISSN 0914-4935 © MYU K.K.
https://myukk.org/

path planning design for vehicles has progressed over many years. Many robotic applications such as a robotic arm and an unmanned aerial vehicle(3) require path planning algorithms, which have been widely used in game design.(4–6) Dijkstra proposed a solution called Dijkstra algorithm for the shortest path problem.(7) Hart *et al.* first described the Astar (A*) algorithm, which is an extension of Dijkstra's algorithm(8) for the path planning design. The A* algorithm uses heuristic search that reduces computing time; therefore, the A* algorithm is more efficient than the Dijkstra algorithm when planning the shortest path. In the year 2007, the Any Time Weighted Astar Algorithm(9) based on an error bound concept that encompasses the differences between the upper and lower bounds to limit the weight of the weighted A* (WA*) algorithm was proposed by Hansen and Zhou. This method can yield the optimal path faster than the A* algorithm. These algorithms are based on gridded maps, and their solutions cause the planned paths to be constrained by the branching factor. The potential headings of the branching factor are restricted to 45 degrees or its multiple, so the paths found by A*-based methods are not the shortest. In order to ameliorate this shortcoming, Botea *et al.* developed a post-smooth method named A* Post-Smoothed (A*PS).(10) This method removes the redundant turns after the paths are found by the A* method. Utilizing A*PS, the shorter paths can be found, being more precise than the purely A* method, but the derived paths are still not the shortest. The Finite Angle A* (FAA*) proposed by Yang and co-workers(11,12) minimizes the problem of restricted turning angles by raising the branching factor unlimitedly to allow users' adjustments for the shortest path. In existing studies, path planning algorithms always use satellite images as the map. However, satellite images cannot provide real-time information for monitoring the ocean envronment, and the stored image resolution is worse than that of aerial images. Otherwise, in reported studies, the ships' dynamics were not considered in the path planning for USVs.(11) When the ship's dynamics are considered, the ship's initial heading and its turning ability should be taken into account as well. If the ship's initial heading is ignored, the path planning results will fail to navigate the ship to the right direction from the beginning. In this investigation, a constraint condition of the initial status of USVs is added into the path planning algorithm to obtain a feasible path. However, the real-time identification of this constraint condition needs clear images. For the above desired reason, previous path planning designs of USVs used satellite images as maps,(11,12) but the image from a satellite is not a real-time one and has a low quality resolution. Recently, the aerial images obtained with a quadrotor have replaced the satellite images and are used as the map for precisely controlling mobile robots.(13,14) Thus, to improve the quality of images, the aerial image obtained with a quadrotor will be adopted for path planning designs of USVs. Inherently, aerial images are represented in local coordinates; hence, a coordinate transformation should be developed to convert the local coordinate of aerial images into a global coordinate. This coordinate transformation can be achieved by global positioning system (GPS), electrical compass, and altimeter sensor fusion. Then, a new path planning algorithm that integrates aerial images, the FAA* method, and a distance–resolution transformation is proposed for the navigation of USVs. With the help of distance–resolution transformation, the global coordinates of the aerial image can be immediately calculated from the resolution of images, the height of the quadrotor, and the positions measured by the GPS and used in the path planning design of USVs. This paper is organized as follows: the real-time image mapping and path planning algorithm are introduced in Sect. 2, the practical implementation and tests of the proposed design are described in Sect. 3, and finally, the conclusions for this proposed design are summarized in Sect. 4.

2. Real-time Image Mapping and Path Planning Algorithm

To achieve a path planning design of a USV, two main functions, (1) real-time image mapping and (2) a path planning algorithm with a low-calculation-power structure, should be constructed. The first objective is to determine the correlation between the photographing distance and the image resolution of the camera; the method for this is named "distance–resolution transformation". The photographing distance is the distance between the camera and the photographed object. The image resolution is defined as the real-world length (m) of one pixel in the image. Distance–resolution transformation is important for the coordinating transformation between images taken with the quadrotor and the navigation coordinates. The image resolutions vary when the quadrotor takes pictures at different heights; therefore, it is essential to determine the distance–resolution correlation to quickly and accurately determine the image resolution for computing the navigation coordinate when the quadrotor flies at different heights.

2.1 Real-time image mapping

During the design stage, the camera of a smartphone is used to take a series of pictures for sample papers attached to a wall. Once the real size and pixel length of the sample papers are known, the image resolution can be computed by dividing the paper size by the pixel length (cm/pixel). The image processing techniques are utilized to determine the pixel length of the sample paper in the image. In addition, for each picture, the photographing distances are recorded; hence, the correlation between the photographing distance and the image resolution can be formulated using a regression model. The image resolution can be quickly computed by obtaining the flying height of the quadrotor with a barometric altimeter; then the distance–resolution transformation can be determined. Figure 1 shows the experimental setup, the red rectangle being the sample paper of known size. The following two sizes of sample paper are used: 30 and 100 cm^2. These sample papers are attached to the wall in this setup. The yellow

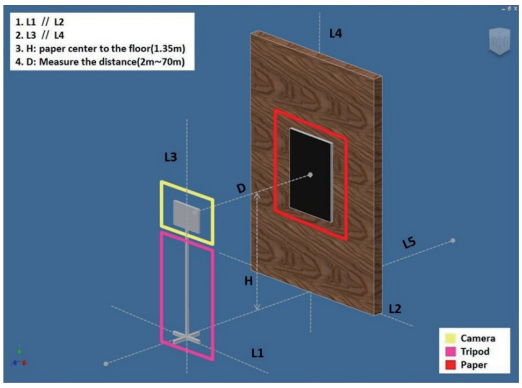

1. L1 // L2
2. L3 // L4
3. H: paper center to the floor(1.35m)
4. D: Measure the distance(2m~70m)
L4
L3
D
L5
H
L2
L1
Camera
Tripod
Paper

Fig. 1. (Color online) Schematic of the experiment for finding the distance–resolution transformation.

rectangle is the camera of a smart phone (Sony Xperia) used in this study. The pink rectangle is the tripod for the camera.

The procedure of the experiment is detailed below.

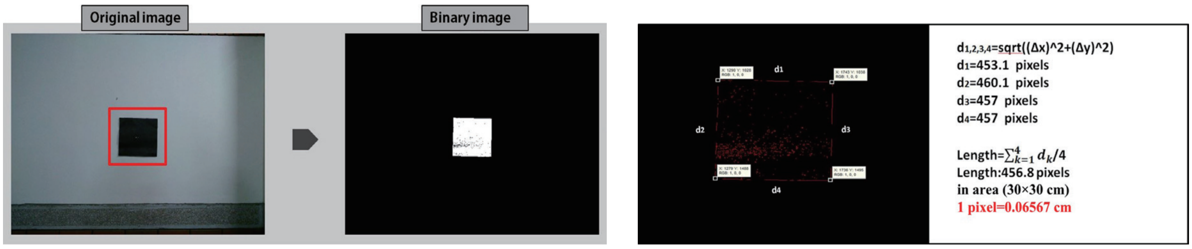
(1) To simulate real conditions of a quadrotor taking pictures from the air, the camera should be parallel to the wall to allow vertical alignment between the camera and its targets (e.g., L1 parallel to L2 and L3 parallel to L4).

(2) Since all edges of the sample papers are required to determine the image resolution, the square sample papers should be located in the middle of the image to guarantee that all edges are captured in the pictures. Moreover, to make the experiment consistent, the middle of the sample papers is aligned with the camera lens, and the height of the camera lens is 1.35 m in this experiment.

Repeat steps 1 and 2 while gradually increasing the photographing distance (*D*). In this experiment, *D* is increased 2 m for each repetition when the distance is less than 30 and 10 m for each step from 30 to 70 m.

The original size of the picture is 3264 × 2448 pixels, as shown on the left of Fig. 2(a), and it is composed of 3 color layers. To boost edge detection, the original color picture is binarized and shown in Fig. 2(a) on the right. The color threshold is used to transform the color picture to a binarized picture, and then the sample paper can be differentiated from the background. In this investigation, the canny edge detection method is then utilized to search for the edges of the detected sample paper in the binarized images. Once the edges are found, the pixel length of the sample paper can be calculated. To minimize the error of the calculation, the pixel lengths of the four edges are averaged. Since the real edge length of the paper is known, the image resolution can be calculated as the real length divided by the average pixel length (cm/pixel). Besides, the 100 cm^2 paper is too large to be imaged with the camera when the photographing distance is shorter than 8 m. Hence, the 30 cm^2 paper is used for a short distance from 2 to 8 m. Figure 2(a) shows the original picture photographed at 2 m and its binarized result. Figure 2(b) shows the coordinates of the corners of the white square, the lengths of each edge of the white square, and the resulting image resolution of the pictures.

By repeating the same process, practical tests can be done for 4, 6, …, 70 m. Data pairs of the true distances and the corresponding image resolutions are listed Table 1.


Original image
Binary image
d1
d2
d3
d4
X: 1290 Y: 1020 RGB: 1, 0, 0
X: 1743 Y: 1038 RGB: 1, 0, 0
X: 1279 Y: 1488 RGB: 1, 0, 0
X: 1736 Y: 1495 RGB: 1, 0, 0
d1,2,3,4=sqrt((Δx)^2+(Δy)^2)
d1=453.1 pixels
d2=460.1 pixels
d3=457 pixels
d4=457 pixels
Length=$\sum_{k=1}^{4} d_k/4$
Length:456.8 pixels
in area (30×30 cm)
1 pixel=0.06567 cm

(a) (b)

Fig. 2. (Color online) (a) Original and binary images (photographing distance: 2 m). (b) Result of distance resolution (photographing distance: 2 m).

Table 1
Data pairs of the true distances and the corresponding image resolutions.

Distance d (m)	Resolution Φ (cm/pixel)	Distance d (m)	Resolution Φ (cm/pixel)
2	0.06567	22	0.72463
4	0.13129	24	0.79365
6	0.19775	26	0.86206
8	0.26619	28	0.9372
10	0.32499	30	1.003
12	0.39108	40	1.3605
14	0.45516	50	1.731
16	0.52083	60	2.1052
18	0.58823	70	2.5
20	0.65573		

2.2 Regression model for distance–resolution relationship

In the previous stage, a series of photographing distances and image resolutions were recorded and are listed in Table 1. The data pairs shown in Table 1 can be treated as training data. On the basis of the training data, the distance–resolution relationship can be formulated as the regression model.

$$\Phi = \alpha d^2 + \beta d + \phi \ , \tag{1}$$

Here, Φ is the estimated image resolution (cm/pixel), d is the photographing distance (m), and α, β, and Φ are identifiable parameters that can be optimally obtained by the least-squares method (LS method). The LS method has been widely used in research to determine the parameters for the regression model(15–17) and can minimize the square sum of the errors between the measurement data and approximated results of Eq. (1). In this study, Matlab is used to implement the LS method, and the identified parameters are $\alpha = 6 \times 10^{-5}$, $\beta = 0.0315$, and $\Phi = 0.0044$. The resulting equation of the regression model is Eq. (2), and Fig. 3 shows the fitting curve.

$$\Phi = \left(6 \times 10^{-5}\right) d^2 + 0.0315d + 0.0044 \tag{2}$$

From Fig. 3, it is easy to see that the image resolution can be quickly estimated by using the photographing distance d in Eq. (2).

The accuracy of this identified regression model will be verified using out sample data pairs. A series of images are taken with different photographing distances. The distance ranges are 3–21 and 4.5–22.5 m. Note that these images are different from pictures of training data pairs. The out sample data pairs are used to examine the accuracy of the model against the actual image resolution. Tables 2 and 3 show the estimated and real image resolutions. The approximated error is calculated as

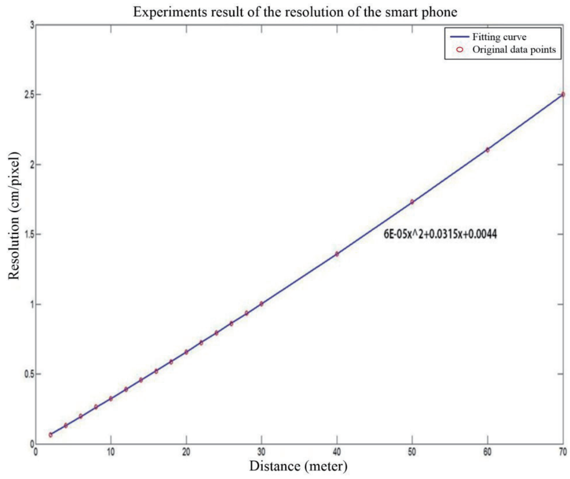


Fig. 3. (Color online) Fitting curve of distance-resolution relationship.

Table 2
Sample data pairs for different testing distances (3–21 m).

Distance (m)	Real (cm)	Estimated (cm)	Error
3	0.09868	0.09944	0.007702
5	0.16574	0.1634	−0.01412
7	0.23715	0.22784	−0.03926
9	0.28922	0.29276	0.01224
11	0.35714	0.35816	0.002856
13	0.4206	0.42404	0.008179
15	0.486	0.4904	0.009053
17	0.5757	0.55724	−0.03207
19	0.61255	0.62456	0.019607
21	0.67796	0.69236	0.02124

Table 3
Sample data pairs for different testing distances (4.5–22.5 m).

Distance (m)	Real (cm)	Estimated (cm)	Error
4.5	0.14888	0.147365	−0.01028
6.5	0.21505	0.211685	−0.0159
8.5	0.2777	0.276485	−0.00439
10.5	0.341	0.341765	0.002238
12.5	0.40733	0.407525	0.000478
14.5	0.46948	0.473765	0.009045
16.5	0.5305	0.540485	0.018474
18.5	0.5952	0.607685	0.020545
20.5	0.65789	0.675365	0.025875
22.5	0.72332	0.743525	0.027175

$$e = \frac{(\Phi - R)}{R}\ , \tag{3}$$

where e is the percentage error, Φ is the estimated image resolution, and R is the real image resolution.

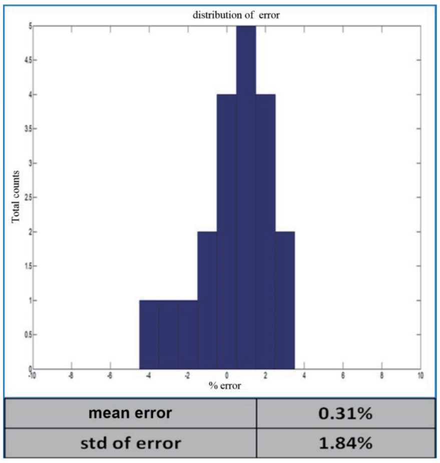
Figure 4 reveals the distribution of errors, the mean of errors, and the standard deviation of errors in Tables 2 and 3. In Fig. 4, the average error is only 0.31%, and the standard deviation of errors is only 1.84%. This means that 95% of all errors fall within −3.37–3.99% (two standard deviations). Furthermore, the histogram shows the distribution to be narrow. The error of the regression model in Eq. (2) causes no more than 1 cm error, which is much smaller than that of the navigation system (GPS). Thus, we can conclude that the regression model in Eq. (2) is accurate and acceptable for use.

2.3 FAA* path planning algorithm

The FAA* algorithm is adopted as the path planning algorithm in this investigation because it possesses the obstacle-free property. A branch factor of 16, whereby each signal potential heading only corresponds to one path length, is directly used. Because of the large branching factor, the FAA* method can effectively enhance the resulting paths on grids(11,12) and redundant expansions are eliminated. With this arrangement, the concept of line of sight is also used in FAA* to find a safer path. The detailed description of the proposed path planning algorithm is presented below.

2.4 Branching factors of FAA*

In Fig. 5, a branch factor of 16 is illustrated, and the corresponding lengths are listed in Table 4. In this list, 16 headings and the relative paths will be saved for generating an obstacle-free path.



mean error	0.31%
std of error	1.84%

Fig. 4. (Color online) Distribution and mean value of errors.

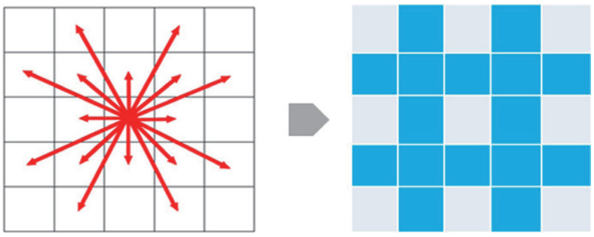


Fig. 5. (Color online) Illustration of the branching factor of 16.

Table 4
Corresponding lengths (squared magnitudes) for the branching factor of 16.

Angle	Length	Angle	Length
0	1	3π/8	5
π/8	5	π/2	1
π/4	2		

As seen in Fig. 5, all potential headings are clearly symmetrical about the central grid. Because of this symmetrical property, only the potential headings in the first quadrant are recorded to save memory. The corresponding lengths of these potential headings are shown in Table 4. For calculation convenience, the squared magnitude of the length is adopted.

The dimension of the branching factor of 16 is a matrix with a size of 5 × 5 as in Fig. 6. In this figure, the expanding lengths are set as 2 for the left, right, up, and down directions. The obstacle is indicated by the brown color.

2.5 Procedure of expansions in FAA*

The procedure of FAA* is very similar to that of A*. There are two additional conditions that are considered in the expansions of FAA*. One is that the visible neighbors of the expanding vertex are generated using branching factors, and the other is that the lines-of-sight (LOSs) between visible neighbors and the expanding vertex are examined. For example, in Fig. 6, the central grid C3 is selected to be expanded. There are 16 visible neighbors that were generated on the basis of the list of branching factors. However, five black paths were blocked by an obstacle, which means that the LOSs do not exist between vertex C3 and vertices B2, A2, B3, A4, and B4. Ultimately, the resulting number of paths successfully generated is 11. Algorithm 1 in Fig. 7 is the pseudocode of FAA*.

There is a scenario in which three expansions are introduced to detail the design procedure of FAA* with the branching factor of 16 in the following. Figures 8–10 illustrate the scenario for describing the manipulations of FAA* referring to the expansion scheme of FAA* with the branching factor of 16 in Fig. 6. In each vertex, two values *G* and *H* will be marked. The definitions of *G* and *H* values are in Remark 1. An effective expanded path is defined as "The safe distance along which the arrow points towards the current position must be larger than $\sqrt{2}/2$ unit lengths".(11)

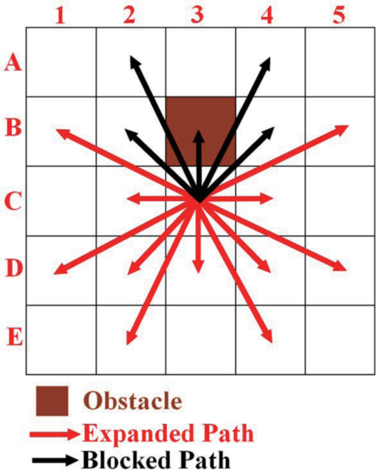
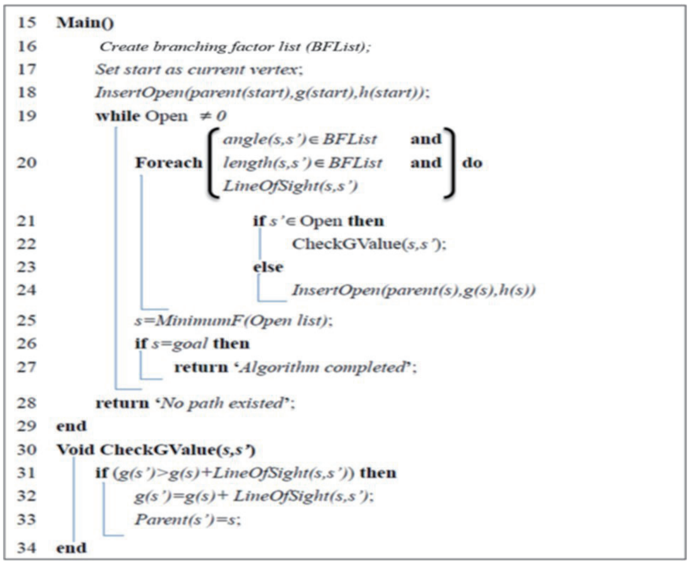


Fig. 6. (Color online) Scheme of the expansion of FAA* with the branching factor of 16.



```
Main()
    Create branching factor list (BFList);
    Set start as current vertex;
    InsertOpen(parent(start),g(start),h(start));
    while Open ≠ 0
            ⎧ angle(s,s')∈ BFList    and ⎫
        Foreach ⎨ length(s,s')∈ BFList   and ⎬ do
            ⎩ LineOfSight(s,s')         ⎭
                if s'∈ Open then
                    CheckGValue(s,s');
                else
                    InsertOpen(parent(s),g(s),h(s))
        s=MinimumF(Open list);
        if s=goal then
            return 'Algorithm completed';
    return 'No path existed';
end
Void CheckGValue(s,s')
    if (g(s')>g(s)+LineOfSight(s,s')) then
        g(s')=g(s)+ LineOfSight(s,s');
        Parent(s')=s;
end
```

Fig. 7. Algorithm 1: pseudocode of FAA*.

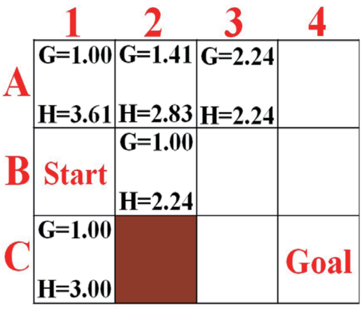


Fig. 8. (Color online) Vertex expansion 1 of FAA* for a start point in B1.

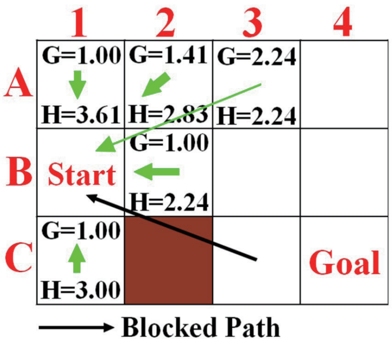


Fig. 9. (Color online) The LOS (B1,C3) does not exist.

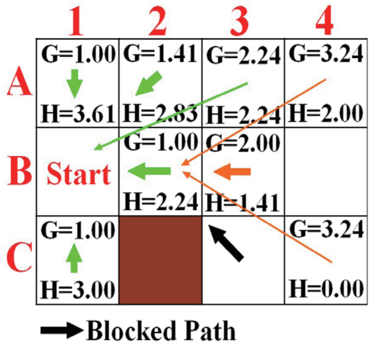


Fig. 10. (Color online) Results of expansion 1 (green arrows) and expansion 2 (orange arrows).

Remark 1: The *G*(s) value reveals the optimal length from the starting point to vertex "s". The *H*(s) value is the estimated distance from vertex s to the goal vertex and can be calculated using the heuristic function.

The manipulations of FAA* for a scenario with three expansions are given as follows.

Expansion 1: In Fig. 8, FAA* expands vertex B1 (Start) on the basis of the branching factor of 16 displayed in Fig. 6. This expansion generates the following seven vertices: A1, A2, A3, B2, C1, C2, and C3. Because there is an obstacle in C2, the vertex C2 should be eliminated in this process to avoid a collision.

In addition, the vertex C3 is also a meaningless point because, as seen in Fig. 9, the LOS between vertex B1 and vertex C3 cannot be obtained, i.e., the effective distance between the origin at grid C3 and the line length of B1C3 is less than $\sqrt{2}\,/\,2$ unit lengths because of blocking by the brown obstacle in C2. In expansion 1, five candidate vertices (A1, A2, A3, B2, and C1) are finally selected and stored in the open list. The initial vertex "Start" will move to the closed list from the open list after expansion 1 is complete. Furthermore, one of these five candidate vertices, which is with the minimal value, will be selected as the next one to be expanded, i.e., vertex B2.

Expansion 2: On the basis of the result of expansion 1, vertex B2 will be expanded continuously by FAA*. Vertex B2 as shown in Fig. 10 is the new initial point of this expansion, and vertices A4, B3, and C4 are new for this expansion; hence, according to FAA*, they can be directly stored in the open list. In this process, vertices A1, A2, and A3 are selected and well verified.

Because the new *G* values of the generated vertices A1, A2, and A3 are greater than the original *G* values, it is unnecessary to update the stored *G* values; for example, when the initial point B2 is expanded, then a new *G* value of 2.41 can be obtained for vertex A3 and the old *G* value of 2.24 is smaller than this new *G* value. Hence, the *G* value (2.24) of vertex A3 does not need to be updated. Because there is no LOS for vertices (B2, C3) and vertices (B2, C1), vertex C4 is selected as a candidate point to be expanded next and will be stored in the closed list after the expansion of vertex B2 is completed.

Expansion 3: FAA* will be terminated because the goal vertex C4 is reserved in the closed list. By extracting the initial point B1 from vertex C4, one optimal path without obstacles, C4→B2→B1, is then obtained.

3. Practical Implementation and Tests of the Proposed Design

Figure 11 shows a quadrotor 0.45 m in height and width, 1.5 kg in weight, and powered by a 14.8 V–4.8 Ah LiPo battery. The camera of an on-shelf smartphone is used in this investigation. The smartphone, Sony Xperia, works on the Android operating system. Android has become one of the most popular mobile operating systems in the world and is a free open-source system.

A smartphone-based camera is utilized in this study for its several advantages.

(1) The smartphone is multisensor-integrated, having a GPS module, gyroscope, accelerometer, electrical compass, global system for mobile (GSM) communication, wireless fidelity (WiFi) communication, and a high-resolution camera. It also has a computing ability and is compact in size.

(2) The smartphone can be easily maintained and replaced. If the camera is broken or a higher resolution camera is required, it can be easily exchanged or upgraded to a fancier smartphone.

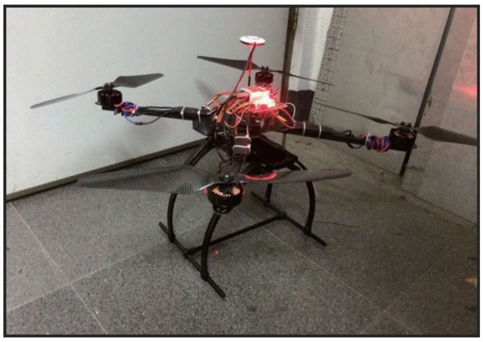


Fig. 11. (Color online) Quadrotor installed with a smartphone camera.

(3) The camera on the smartphone has antiblur, autofocus, and auto-white balance capabilities

The procedure of the android system developed in this research is described below. The USV continuously transmits its GPS coordinate to the quadrotor through the local network, as shown in Fig. 12. The quadrotor calculates the distance between the quadrotor itself and the USV, which serves to ensure that the camera on the quadrotor can capture the image of the USV. The communication protocol in this study is the 2.4 G WiFi signal, and the quadrotor and USV are connected to a local network by adopting the TCP/IP protocol.

The platform in the USV is a computer (Lenovo-i7-3610QM CPU@2.3 GHz, 8 GB of RAM), and the development environment is Labview. The developed user interface of the Android app is shown and described in Fig. 13. In this app-based user interface, the locations of the controlled USV and quadrotor are displayed in real time.

The aerial image taken by the controlled quadrotor will be used as the map for guiding USVs. The path planning algorithm, FAA*, is implemented on the captured aerial image. However, the captured aerial image represents the local coordinate, and the transformation of coordinates for resulting paths in the aerial image is essential, i.e., the coordinates in the aerial image should be transformed into GPS coordinates for the navigation of a USV. In this section, the procedures for coordinate transformation are detailed below, and the procedures are shown in Fig. 14.

(1) Picture taken by the quadrotor should be oriented to the Geographic North Pole. The electric compass built into the smartphone installed on the quadrotor is utilized to obtain the magnetic north, and then, the calibration for magnetic declination is implemented. NOAA provides the magnetic declination using the GPS location.(18) Once the magnetic declination is obtained, the picture can be oriented to the Geographic North Pole shown in Fig. 15.

(2) Determine the image resolution using distance–resolution transformation. A barometric altimeter was installed on the quadrotor, the altitude of the photograph can be transmitted to the USV through the barometric altimeter, and then the image resolution for that altitude can be determined.

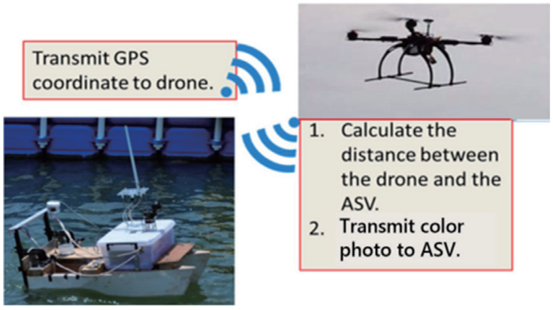
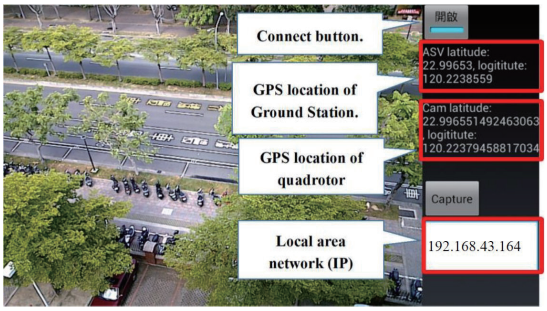

Transmit GPS coordinate to drone.
1. Calculate the distance between the drone and the ASV.
2. Transmit color photo to ASV.

Fig. 12. (Color online) Scheme of USV communication with the quadrotor.



ASV latitude: 22.99653, logititute: 120.2238559

Cam latitude: 22.996551492463063, logititute: 120.22379458817034

192.168.43.164

Fig. 13. (Color online) Screen shot of the app of the camera system.

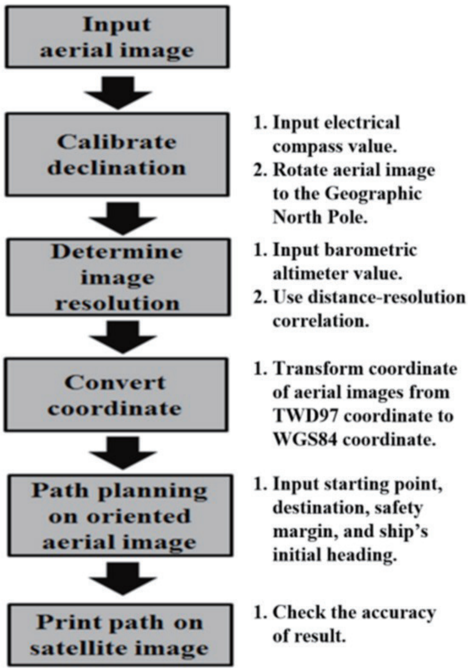

Input aerial image
Calibrate declination
1. Input electrical compass value.
2. Rotate aerial image to the Geographic North Pole.
Determine image resolution
1. Input barometric altimeter value.
2. Use distance-resolution correlation.
Convert coordinate
1. Transform coordinate of aerial images from TWD97 coordinate to WGS84 coordinate.
Path planning on oriented aerial image
1. Input starting point, destination, safety margin, and ship's initial heading.
Print path on satellite image
1. Check the accuracy of result.

Fig. 14. Flow chart of the coordinate transformation.

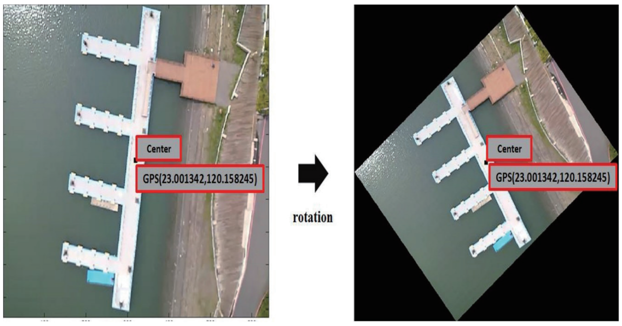


Fig. 15. (Color online) Rotate aerial image to Geographic North Pole.

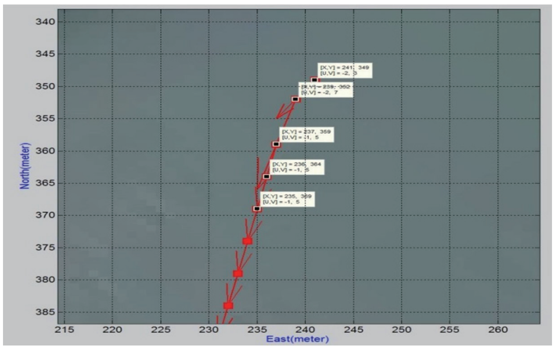
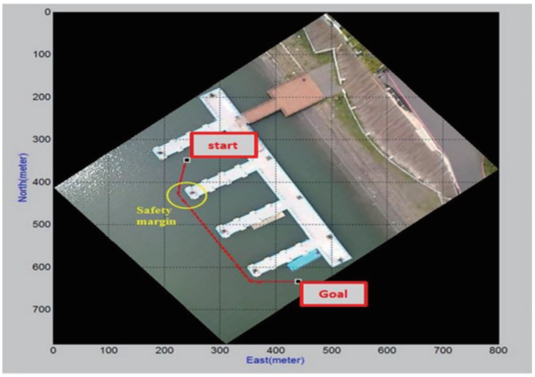


Fig. 16. (Color online) Portion of the path planning result.

(3) Use the smartphone to transmit the GPS location of the center of the image to the USV. Given the GPS location, image resolution, and oriented images, the coordinates of every pixel in the aerial image can be transformed into TWD97 coordinates. Detailed information of TWD97 can be obtained from Refs. 11, 19, and 20. Finally, the TWD97 coordinates can be transformed into WGS84 coordinates for navigation purposes.

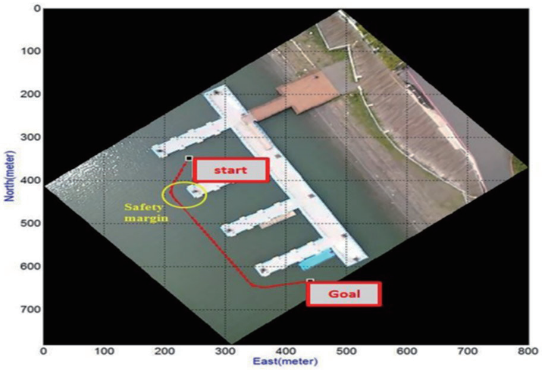
By the above designed procedure, the path planning algorithm on the aerial image can be demonstrated. Figure 16 shows a portion of path planning results. The resulting path is composed of a set of nodes and arrows. Every node points to its next node connected by arrows. Each arrow denotes the direction of motion of the USV shown in yellow circles in the figure. Two points are selected randomly as the start and end points for path planning on the aerial image shown in Figs. 17 and 18. These two path planning takes are performed using two different safety margins, namely, 1.12 and 2.77 m on the same image. Both results show that the planning paths are collision-free and maintain the safety margins near obstacles.

To validate the coordinate transformation of the aerial image to the navigation coordinate, the resulting paths are printed on the satellite image, which is a global and well-oriented image, shown in Fig. 19. It is found that the calculated paths are consistent with the aerial images, although the resolution of the satellite image is worse. However, it shows that the coordinate transformation of our proposed method is correct and useful. Note that the blue path (2.77 m safety margin) is farther from the dock than the red path, which is consistent with the results on the aerial image.



Safety margin=10 pixels (1 pixel=0.1156 m)
≈1.12 m

Fig. 17. (Color online) Path planning on aerial image (safety margin: 1.12 m).



Safety margin=24 pixels (1 pixel=0.1156 m)
≈2.77 m

Fig. 18. (Color online) Path planning on aerial image (safety margin: 2.77 m)

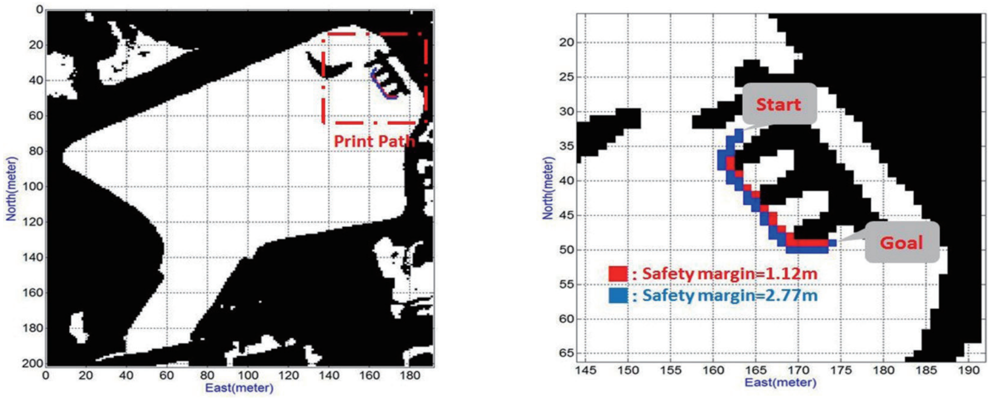


Fig. 19. (Color online) Printed resulting paths on the satellite image with respect to different safety margins: 1.12 (red line) and 2.77 m (blue line).

4. Conclusions

In this research, the high-accuracy path planning problem of USVs was successfully solved with the help of a multisensor fusion design. By integrating the FAA* method, distance–resolution transformation, and coordinate transformation of real-time aerial images, a new path planning algorithm with the collision-free property was proposed for the navigation of USVs. The FAA* method, which yields a smoother and more pragmatic path than previously used methods, was enhanced considering the USV's initial conditions. The distance resolution and coordinate transformation were developed and implemented in practice for the real-time transformation of the aerial images. With the distance–resolution transformation, a highly accurate transformation performance was obtained, i.e., the average error was only 0.31% and the standard deviation of errors was only 1.84%. This proposed method is verified by real USVs in the harbor. From comparisons of the path planning results with respect to aerial and satellite images, this proposed method was found to be promising for the path planning of USVs, and has the potential to be applied in the path planning designs of surface moving vehicles because of its high accuracy and real-time image updating property. In the near future, this proposed method will be practically applied to the guidance law design of unmanned surface vessels in messy ocean environments.

Acknowledgments

This work was financially supported by the National Science Council, Taiwan, R.O.C. (Grant No. MOST 104-2628-E-006-012-MY3).

References

1 M. Caccia, M. Bibuli, and R. Bono: Auton. Robots **25** (2008) 349.

2 M. Bibuli, G. Bruzzone, M. Caccia, and L. Lapierre: J. Field Rob. **26** (2009) 669. https://doi.org/10.1002/rob.20303

3 M. Achtelik, T. Zhang, K. Kuhnlenz, and M. Buss: Proc. Int. Conf. Mechatronics and Automation (ICMA09, 2009).

4 H. Choset, H. Lynch, K. Hutchinson, S. Kantor, G. Burgard, L. Kavraki, and S. Thrun: Principles of Robot Motion: Theory, Algorithms, and Implementations (MIT Press, Cambridge, MA, 2005).

5 R. Murphy: Introduction to AI Robotics (MIT Press, Cambridge, MA, 2000).

6 A. Patel: Amit's Game Programming Information. Available online: http://theory.stanford.edu/?amitp/GameProgramming/MapRepresentations.html (accessed in 2000).

7 E. W. Dijkstra: Num. Math. **1** (1959) 269.

8 P. Hart, N. Nilsson, and B. Raphael: IEEE Tran. Syst. Sci. Cyber. **4** (1968) 100.

9 E. Hansen and R. Zhou: J. Artif. Intell. Res. **28** (2007) 267.

10 A. Botea, M. Muller, and J. Schaeffer: J. Game Dev. **1** (2004) 1.

11 M. Yang, C. M. Tseng, and C. C. Fan: J. Taiwan Soc. Naval Arch. Marine Eng. **31** (2012) 173.

12 J. M. Yang, C. M. Tseng, and P. S. Tseng: Int. J. Nav. Archit. Ocean Eng. **7** (2015) 87.

13 L. Heng, L. Meier, P. Tanskanen, and M. Pollefeys: Proc. Int. Conf. Robotics and Automation (2011) 2472.

14 L. Meier, P. Tanskanen, L. Heng, G. Lee, F. Fraundorfer, and M. Pollefeys: Auton. Robots **33** (2012) 1.

15 A. Charnes, E. L. Frome, and P. Yu: J. Am. Stat. Assoc. **71** (1976) 169.

16 O. Bretscher: Linear Algebra with Applications (Prentice Hall, Upper Saddle River, NJ, 1995) 3rd ed.

17 S. M. Stigler: Ann. Stat. **9** (1981) 465.

18 National Oceanic and Atmospheric Administration. Earth's Magnetic Field Calculators. Available online: http://www.ngdc.noaa.gov/geomag/magfield.shtml (accessed in 2012).

19 S. P. Drake: DSTO Electronics and Surveillance Research Laboratory. Converting GPS Coordinates (ϕ, λ, h) to Navigation Coordinates (ENU) (DSTO No. DSTO-TN-0432). Retrieved from http://www.dsto.defence.gov.au/corporate/reports/DSTO-TN-0432.pdf (accessed in 2002).

20 D. Steve: Converting UTM to Latitude and Longitude (Or Vice Versa). Available online: http://www.uwgb.edu/dutchs/UsefulData/UTMFormulas.htm (accessed in 2012).

About the Authors

Yung Hsiang Chen received his M.S. degree from Da Yeh University, Taiwan, in 1998 and his Ph.D. degree from the Department of Electrical Engineering at National Cheng Kung University, Taiwan, in 2013. Currently, he is an assistant professor at National Pingtung University of Science and Technology, Taiwan. His research interests are in nonlinear control and sensors. (yhchen@mail.npust.edu.tw)

Yung Yue Chen was born in Pingtong, Taiwan, in 1972. He received his Ph.D. degree from the Department of Electrical Engineering, National Tsing Hua University, Hsinchu, Taiwan, in 2003. From 2004 to 2010, he served as a department manager and a product line leader in the Smart Microsystems Technology Center, Industrial Technology Research Institute. He is currently an associate professor at National Cheng Kung University, Taiwan, and the leader of the Information Team in the Engineering and Technology Promotion Center, NSC, Taiwan. He has published more than 50 scientific publications in refereed journals and conference proceeding and is the owner of 20 patents. His research interests include noise reduction design of cellular phones, micro- and nanocontrol design, MEMS-based inertial navigation system design, intelligent system design, system estimation, and control design. (yungyuchen@mail.ncku.edu.tw)

Joe Ming Yang received his B.S. degree from National Taiwan Ocean University, Taiwan, and his M.S. and Ph.D. degrees from the University of California-Berkeley, USA. Currently, he is an associate professor in the Department of Systems and Naval Mechatronic Engineering of National Cheng Kung University, Taiwan. His research interests are in novel ship design, path planning of vehicles, vibration system analysis, and sensors. (z7908036@mail.ncku.edu.tw)