J Intell Robot Syst (2014) 73:309–323
DOI 10.1007/s10846-013-9922-7

A Terrain Referenced UAV Localization Algorithm Using Binary Search Method

Orhan Eroglu · Guray Yilmaz

Received: 1 September 2013 / Accepted: 12 September 2013 / Published online: 2 October 2013

Abstract This study focuses on localization of Unmanned Aerial Vehicles (UAV) since permanent navigation has vital significance to support position information and to avoid getting lost. Actually, there exist effective aeronautical navigation systems in use. Inertial Navigation System (INS) and Global Positioning System (GPS) are two representatives of the most common systems utilized in traditional aerial vehicles. However, an alternative supporter system for UAVs should be mentioned since INS and GPS have serious deficiencies for UAVs such as accumulated errors and satellite signal loss, respectively. Such handicaps are coped with integrating these systems or exploiting other localization systems. Terrain Referenced Navigation (TRN) could be a good alternative as a supporter mechanism for these main systems. This study aims to localize a UAV accurately by using only the elevation data of the territory in order to simulate a TRN system. Application of the methodology on a real UAV is also considered for the future. Thus assumptions and limitations are designed regarding the constraints of real systems. In order to represent terrain data, Digital Elevation Model (DEM) with original 30 meter-resolution (Eroglu and Yilmaz 2013) and also synthetically generated 10 meter-resolution maps are utilized. The proposed method is based on searching the measured elevation values of the flight within the DEM and makes use of simulation techniques to test the accuracy and the performance. The whole system uses sequences of elevation values with a predefined length (i.e. profile). Mainly, all possible profiles are generated and stored before the flight. We identify, classify and sort profiles to perform search operations in a small subset of the terrain. During the flight, a measured flight profile is searched by the Binary search method (Eroglu 2013) within a small neighborhood of corresponding profile set.

Keywords Localization · Terrain referenced navigation · Unmanned aerial vehicles · Digital elevation model

O. Eroglu (✉)
The Scientific and Technological Research Council of Turkey, Gebze, Kocaeli, 41470, Turkey
e-mail: orhan.eroglu@tubitak.gov.tr

G. Yilmaz
Computer Engineering Department, Turkish Air Force Academy, Yesilyurt, Istanbul, 34149, Turkey
e-mail: g.yilmaz@hho.edu.tr

1 Introduction

Unmanned Aircraft Systems (UASs) are fixed or rotary wing aircrafts on which there is not any human pilot. UAVs may be controlled by human controllers on other vehicles or ground stations, or even autonomously. UAVs may be utilized in several missions, on which manned aircrafts are not applicable due to size constraints, requirements for high pilotage abilities, life-critical missions for a human pilot. Hence, the localization of UAVs has a major importance in order to tackle with such distinctive functions.

Basically, the aim of navigation systems is to determine the position of the vehicle in the 3D space and to track the spatio-temporal variables such as speed, heading, bank angle etc. to prevent the vehicle from getting lost [3]. Therefore, navigation is very crucial for accurate localization of aerial vehicles.

In aeronautics, the most common navigation methods can be listed as inertial, electronic (radio, radar and satellite) and terrain referenced navigation. INS and GPS are the most popular techniques of the inertia and electronics based navigation systems, respectively [4].

INS is a widely used navigation system, which has been used as a main localization system in aeronautics since 1930s. This system basically measures speed, acceleration and angular speed in all three dimensions [5] to compute navigation parameters. INS is robust to weather conditions and external interferences and cannot be discomposed [6] since all inputs of the system are inertial and internal [7]. However, this system integrates measurements to calculate placements and this results in accumulated errors throughout the flight. This behavior makes INS need to be supported by some extra systems [7].

The other navigation system, GPS, which is based on the satellite connections, is another common navigation system. This system has started to be developed in 1970s to overcome side effects of INS, especially [8]. GPS offers 95 % accuracy and a failure rate about 7.8 m [9]. GPS handles the accumulated error of INS, because it enables positioning at each measurement [10]. On the other hand, GPS signals may be easily jammed [11] or lost and may be inapplicable in some geographical conditions such as a deep valley or sides of a number of mountains.

Deficiencies of two systems are inacceptable not only for military objectives, but also for civilian usages. This inadequacy leads to need for a more accurate solution of navigation or at least a decision support system for the main positioning framework. In modern aircraft systems, integrated INS/GPS systems are used and able to overcome both systems individual problems. As an alternative way, terrain referenced localization may be regarded seriously to be able to supply the need.

Indeed, terrain referenced localization has been used for all the history of aeronautics. A navigation pilot used to try to estimate the position of the vehicle with the help of paper-printed maps on his hand, even in the most primitive aircrafts. In addition, systematic studies on terrain referenced navigation started in early 1970s to cope with the accumulated errors of INS. TRN was examined regarding military aspects in these studies and was tried on Tomahawk missiles [12]. Nevertheless, further improvements could not be achieved because of the inadequate performance of both localization sensors on the aerial vehicles and computer systems for navigation, and insufficient resolution of terrain data in those years.

The following decades have come with huge evolutions on the performance of computers and altitude sensors, and the resolution of territory data. These developments have given new opportunities to researchers for studying on terrain based positioning more closely.

In this study, we aim to discuss existing TRN approaches, to focus on application of these systems on UAVs and to propose a new TRN system that makes use of the Binary search algorithm. To achieve these goals, this study simulates flight of a UAV on the DEM of a known territory, and tries to determine position of the vehicle in a reasonable amount of time just after the vehicle is lost in a randomly chosen position of the flight.

Briefly, all possible flight trajectories of a predefined length is generated from DEM initially, terrain elevation measurements of the same length with the generated trajectories are performed then, and eventually the observed trajectory is searched within the huge search space of possible trajectories.

2 Related Works

This section describes the significant subjects of the TRN systems in general and mentions two well-known real TRN systems and one proven simulation study.

2.1 Terrain Elevation Data

In TRN systems, one of the most significant parts of the system is the terrain elevation data, which keep the height values of the terrain and must be obtained before flights. Despite there are a number of world-wide file formats for these maps, all these maps keep the elevation values of terrains in 2D matrices regarding a pre-defined resolution. Elevation values in the rows and columns of the matrix correspond to the longitudes and latitudes, respectively. The most popular file formats for these elevation maps are Digital Elevation Model (DEM), Digital Terrain Elevation Database (DTED) [13] and ASTER Global Digital Elevation Model (GDEM) [14].

2.2 Altimeters

Aerial vehicles have sensors placed on them that measure instantaneous distances between the vehicle and a reference point. Altimeters are a specialized kind of these sensors, which take vertical range measurements.

Generally, in TRN systems, both the altitude values from the sea level and the terrain below the UAV must be obtained in order to calculate the elevation of the terrain below, i.e. altitude value of the terrain from the sea level. Altitude of the air vehicle from the sea level can be measured by simply using pressure differences by means of barometric altimeters. Besides, there are a number of techniques and different devices designed regarding these techniques for the purpose of measuring the distance between the vehicle and the terrain below. Very first examples can be enumerated as sonar (sound), lidar (laser) and radar (radio wave) altimeters. Radar altimeters are most widely used altimeter thanks to their small sizes and very large price scale. These devices simply send radio waves to the earth periodically and measure the distance by exploiting the time to return and the speed of the radio waves. A modern radar altimeter can take approximately 50 measurements per second, i.e. it can supply one altitude measurement in each meter for a UAV flying with a speed of 50 m per second.

On the other hand, altimeters may have definitely errors, i.e. measurement noises. Moreover, noise becomes higher as the height of the flight increases. Therefore, radar altimeters for UAV navigation must have small errors and these errors must be mentioned designing the TRN system.

2.3 Basic TRN Flow

Mainly, TRN systems work as follows:

1. Digital Elevation Map (DEM), which is a well-known format and includes elevation data of a terrain with a defined resolution of the terrain to be flown on, is loaded on the aerial vehicle,
2. The vehicle observes instantaneous elevation of the terrain below by means of using the difference between barometric and radar altimeter measurements as shown in the Fig. 1 and given in the Eq. 1,

$$h_t = h_b - h_r, \tag{1}$$

3. Compare somehow these altitude measurements with DEM
4. Determine the position utilizing these comparisons.

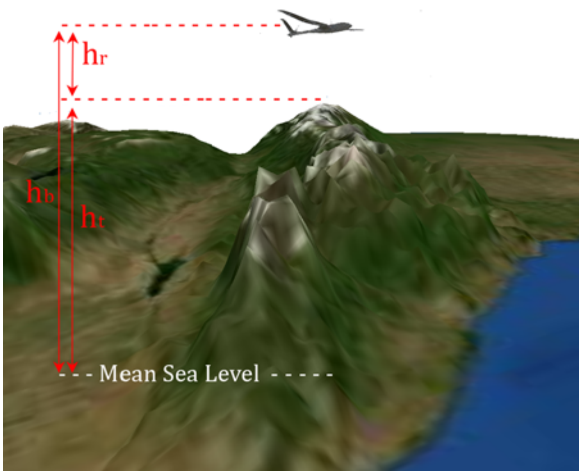

Fig. 1 Instantaneous computation of terrain elevation

TRN seems to be easily and effectively applicable for navigation of unmanned aircrafts since it is independent of GPS and is firm to external attacks. On the other hand, accuracy of TRN depends strictly on the variation of elevation values in DEM, i.e. it gets more accurate on more rough terrains.

A military aerial vehicle can perform GPS-free localization successfully over a known territory by taking advantage of TRN. From this sight of view, i.e. being resistant to external factors, TRN may especially satisfy the requirements of military localization practices. Therefore, TRN systems have been already used in several fighter aircrafts and missiles. Nonetheless, detailed analysis on such military applications of TRN cannot be handled because of privacy restrictions, unfortunately.

2.4 Classification of TRN Systems

TRN systems designed for aerial vehicles basically try to perform localization by searching real-time measurements from sensors within the terrain data. Additionally, differences between sensor technologies, data processing approaches various searching methodologies make TRN algorithms differ from each other.

Sensors placed on the vehicle are one of the most significant components of the UAV, which play crucial role on the operation of the algorithms. In other words, technology of sensors directly influences the method of localization. TRN systems can be clustered into two groups with respect to their sensor components: Passive imagers and active range sensors [15].

The passive imaging sensors simply take spatial or spatio-temporal images of the territory. The most popular and widely used members of this class are conventional and fisheye cameras for UAVs [16]. Members of this category offer a large price and size scale. On the other hand, a passive imager may be easily affected by illumination variations and the weather conditions, e.g. it cannot support beneficial localization data in the night, or in a foggy day. In addition, if the search operations are performed on the UAV, passive imagers require large memories on the vehicle since they take comparatively large data. Besides, these sensors need high bandwidth for the sake of transferring data to another station and handling the search processes. Moreover, since the data supported from this class of sensors consist of multiple still images or videos that may require image-processing knowhow to be evaluated.

The active range sensors are components that can periodically support altitude measurements. Radar, sonar, lidar and barometric altimeters can be listed as the very first examples of this category. Although these sensors are consistent to light and weather conditions, they lead to some other difficulties, e.g. operating at higher altitudes leads these sensors to have more measurement error [17]. Since these sensors may be very noisy at comparatively high altitudes, they may be inoperable for aerial navigation. On the other hand, the data gathered from active ranging units consist of simply the altitude values, and can even be processed and evaluated with the help of the basic computational sciences and the engineering skills. Therefore, the active range sensors are used more commonly than the passive imagers on terrain aided positioning.

Secondly, TRN systems can be mentioned in two categories considering their searching phenomena: Correlation techniques and pattern matching techniques [17].

Basically, in correlation methods, the sensors on the vehicle observe a small, contiguous subset of the territory. This subset is a sub-image of the territory when a passive imaging device is chosen, and it is a sub-matrix of the entire elevation data when the sensing device is a range measurer. After the observation of a sub-sample of the terrain data, this patch is searched within all the data via shifting it through each sub-part of the terrain. Search process is based on computing the similarity between the observed patch and terrain subsections. The sub-terrain region with the highest similarity is considered the position of UAV [17]. As the very first examples of this method, TERCOM is a well instance of altitude-to-map correlation [18] and DSMAC can be given for image-to-map correlation [19]. The crucial advantage of this technique is that, terrain sub-samples can be generated before the flight since the terrain data is pre-known.

Pattern matching algorithms stem from the fact that there exist a number of landmarks in any territory, which have specific characteristics that other regions in the terrain do not have. As a primary reference, hills, craters, lakes etc. can be cited as the examples. Certain samples of this category are SIFT (Scale Invariant Feature Transform), Shape-to-Signature Pattern Matching, Onboard Image Reconstruction for Optical Navigation [1]. This TRN approach requires high computer vision and image processing knowledge.

Thirdly, the processing type of data collected from sensors divides TRN systems into three classes: Batch data processing, sequential data processing and recursive data processing [20].

A number of TRN algorithms use sensor data, only after measurements can construct a patch, i.e. a small subset of the terrain. Such algorithms are said to have batch data processing. Sequential processing is indeed a special case of batch processing, in which data are kept and thought as a sequence of observations rather than a 2D array-like data structure. Recursive TRN data processing approaches access sensor data as each new value of them is generated [20]. Recursive processing may improve solution, i.e. narrow the search space at each iteration so that it may support to the navigation system more than batch and sequential data processing techniques. However, reducing the size of the search space in the right way with each new measurement is another difficult problem.

The following subsections describe the most significant TRN systems:

2.5 TERCOM (Terrain Contour Matching)

TERCOM is a well-known and primary altimeter-to-DEM correlation TRN method. This method has initially been used on cruise missiles with success. TERCOM essentially correlates active range sensor observations with a digitized elevation database of terrain (DEM) [21]. Chronologically, the initial position of the aerial vehicle (Cruise missile in practice) and DEM are loaded to the system before the flight, and the elevation data of the sub-territory below the vehicle and the real terrain data are given as input to the Extended Kalman Filter (EKF) throughout the flight. The result from the comparisons of EKF contributes intuition about the current position, and this inference is supported to INS in order to avoid increasing accumulated error. This procedure is repeated iteratively during the flight, thus cruise missile is kept from reaching a position, which cannot be located anymore [18].

2.6 TERPROM (Terrain Profile Matching)

TERPROM is a hybrid TRN system which enables both positioning and tracking of an aircraft and is used on a reasonable number of Possible flight path in the TERPROMhter aircrafts such as F-16, Eurofighter Typhoon etc. [22]. In fact, TERPROM utilizes TERCOM system for acquisition mode and SITAN algorithm for tracking mode [23]. TERPROM is started with acquisition mode at the beginning of a flight and just before the takeoff, and in any case that tracking mode is inapplicable and aircraft position is lost, besides the system is performed in tracking mode during an ordinary flight. In Acquisition mode, a possible flight path and its representation on the terrain map are shown in the Figs. 2 and 3, respectively. In tracking mode, radar altimeter periodically measures elevation below the vehicle, and INS failures are minimized by means of comparing observed values with the real elevation data from DEM [22].

The two most popular TRN systems, which are trained successfully on fighter aircrafts and missiles for the sake of accomplishing military goals, have been intuitively introduced in detail. The published studies are limited about these TRN methods because of military restrictions; hence further comprehensive observations of particular

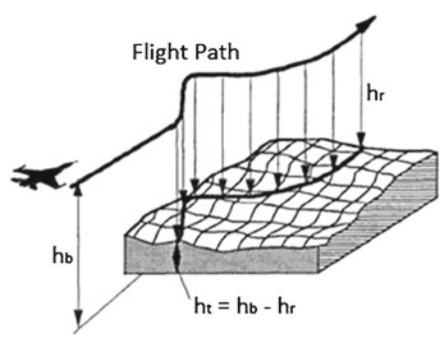

Fig. 2 Possible flight path in the TERPROM Acquisition Mode

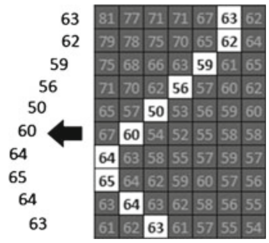

Fig. 3 Representation of a flight path in the terrain map in the TERPROM Acquisition Mode

algorithms in these systems could not be performed yet. In addition, neither these two TRN applications, nor the other well-known examples such as SITAN, HELI-SITAN [24], DSMAC [19] do not have been studied and designed for UAVs and there is not any specialized TRN framework known to be operating on a UAV.

2.7 TERELONA (Terrain Referenced Localization and Navigation)

In order to design and train TRN methodology on UAVs, several simulation studies have been practiced in the literature. In general, these studies utilize a number of assumptions and limitations to minimize the search space and to optimize the solution, but plenty of limitations remove systems away from being realistic. However, distinctive and potentially successful studies exist in the literature.

TERELONA (Terrain Referenced Localization and Navigation) system is a well-prepared and proven simulation study in a scientific journal. This study utilizes terrain data from eastern territories of Turkey as ASTER GDEM file format.

Searches are performed with the help of Particle Filter algorithm [25], which is an application of Bayesian theorem [26], in TERELONA system. Radar and barometric altimeters are assumed to be embedded on the UAV as altimeter set. One of the most typical characteristics of TERELONA is that, system exploits recursive data processing, i.e. solution is grown up at each altitude observation. Basically, a group of particles and their weights in the Particle filter represents possible positions of the UAV.

The most powerful proposal of TERELONA is running correlation methods in the CUDA library [27] and so processing in the graphical processing unit (GPU) of the computer system. TERELONA study aims to prevent initial position errors to overload Particle filter.

TERELONA system has reports on simulation studies and results on different terrain characteristics. These results show that utilization of GPU extremely contributes to the system on correct localization and tracking of the UAV even if altimeter measurement period is very short.

3 Application of Binary Search Algorithm to TRN

TRN systems considered so far are essentially based on usage of search techniques. However, we have developed a system, which takes its heart comparatively from pre-processing of the known data. Before describing the details of our newly proposed TRN approach, we are going to consider requirements and limitations of such a navigation issue.

Since TRN is a real time operation and must be solved in the flight-time, the method must be as fast as possible to cope with the case of localizing a UAV in an acceptable amount of time, i.e. this period must not exceed few seconds. At the same time, the system must support almost exact accurate position acquaintance in order to support any other navigation system like INS, and not to lose UAV eventually. In addition, although it is considered and designed as a simulation study, such a simulated system should be practically applicable for a real UAV after all. Therefore, assumptions and constraints must be chosen in the way that they can be handled in the future. In other words, motion dynamics of the UAV and assumptions and constraints on flight must satisfy the real system conditions and must not include impossible features.

Our methodology consists of two significant parts: The Pre-processing and the Localization parts. These two parts are closely explained through the end of this section.

3.1 Pre-Processing

The search space of terrain-referenced localization and navigation is so huge that searching in raw data of the territory may result in an impossible hunting. Furthermore, TRN systems have a remarkable convenience that reasonable pre-processing can be made since the elevation data of the flight territory are on the hand before the flight. As an inevitable consequence of these conditions, DEM, which is the previously known data of the study, must be pre-processed prior to the execution of the system during a flight. This preparation phase mainly comprises four sub-phases:

3.1.1 Profile Generation

In this section, generation of all possible flight profiles from the terrain data is described.

a. **Profile:** The term 'Profile' stands for the sequence of terrain elevation values observed within the DEM data or with the help of radar and barometric altimeters throughout the flight. A profile simply consists of successive height values of a predefined length. Figure 4 depicts that an example flight trajectory as shown creates a profile {17, 17, 13, 22, 36, 7, 36, 25, 14, 17, 35, 28, 17, 40, 36, 17, 8}.
b. **Generating profiles:** The very first decision of the algorithm is made on the required number of sensor observations of elevation to achieve a valuable search with agreeable performance. Any supporter navigation system should provide correct position info until UAV moves ahead a risky distance and this distance is assumed to be at most ~300 m. When the fact that original DEM data have 30 meter-resolution is regarded, the length of a flight can be reckoned as 9 to 12 indices in a DEM for the sake of positioning the UAV. In other words, radar and barometric altimeters should be set up to take 1 measurement per second for a UAV moving with a velocity of ~30 m per second to generate profiles from 9 to 12 index-length. Less number of observations will be inadequate since a huge amount of profiles may exist in the terrain with similar features if profiles are not long enough.

	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
0	1	14	24	22	21	10	5	0	0	9	13
1	11	**17**	26	20	19	12	7	2	4	19	23
2	18	**17**	**13**	10	11	4	4	4	13	44	55
3	22	25	**22**	17	19	9	12	14	24	14	7
4	28	23	**36**	**7**	0	8	9	17	19	0	36
5	18	25	48	**36**	**25**	33	38	20	8	8	8
6	9	21	5	25	**14**	**17**	14	14	5	14	11
7	6	22	7	48	25	**35**	**28**	5	7	5	17
8	33	39	36	8	7	35	**17**	36	5	19	21
9	29	41	25	39	39	33	**40**	**36**	25	36	5
10	32	28	17	49	31	17	24	**17**	**8**	36	7
11	16	29	19	15	23	12	5	19	13	36	20
12	10	34	48	36	29	15	17	28	16	9	18
13	3	8	12	15	9	33	38	20	36	9	23

Fig. 4 An example profile

Additionally, in order to simulate realistic flights, UAV is assumed to be able to change its heading within the range $(-45, 45)$ degrees as shown in Fig. 5. From this sight of view, it can be seen as a modified 8-neighborhood approach that is limited with the direction of the flight. In contrast with other studies (e.g. [1]), the direction of flights is not limited in this work, i.e. UAV is supposed to fly through all directions.

With respect to the phenomena described above, all possible profiles of predefined length in all directions are generated from DEM and written into files. An example flight of the UAV and its profile on the terrain is shown in Fig. 6. Even a small DEM may include millions of profiles (e.g. 64×64 sized DEM and 7 index-long profiles). These profiles are going to be compared with a measured profile of real-time flight in order to find the position of UAV. Hence, each profile

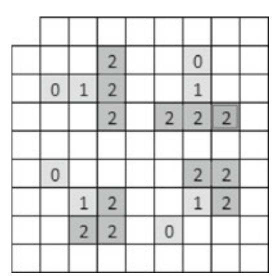

Fig. 5 Possible next positions (indices labeled with 2) when the first two positions of the UAV are indices numbered with 0 and 1, respectively. Opposite directions are available as symmetric of represented ones

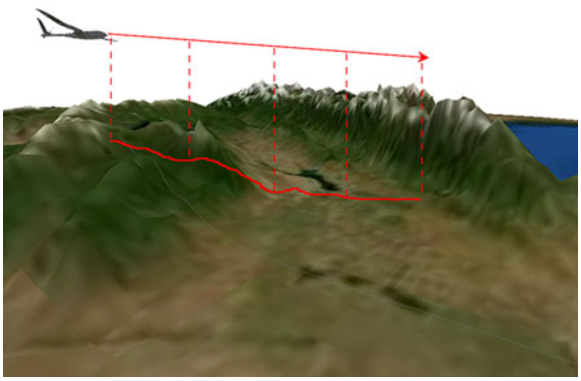

Fig. 6 A possible flight of a UAV and its profile

should be identified by means of significant characteristics for accurate comparison outputs.

3.1.2 Profile Featuring

Since a profile is a sequence of altitudes of travelled coordinates by a UAV, it keeps track of altitude variations of a flight and these variations can be utilized to distinguish profiles from each other. In our methodology, we represent each profile with a line, which is generated via Least Squares Line Fitting Method [28] using each coordinate and depicted in the Fig. 7 below:

a. **Least squares line fitting method:** This line fitting method is a specialized version of the well-known Least Squares regression method. The method basically fits a line through data points where the line is as possible as close to each point in the data.
Principally, Least Squares line fitting method tries to minimize the sum of the differences between each Y-value in the data and the corresponding Y-value generated from the line equation fed by X-value.
Line equation is given below:

$$y = mx + n \tag{2}$$

Making use of the line equation, the error between the real Y-values and the corresponding Y-values to the real X-values from the equation can be calculated as:

$$e_i = y_i - y\left(x_i\right) \tag{3}$$

$$e_i = y_i - m.x_i - n \tag{4}$$

where, i = 1,2...N, (xi, yi) pairs represent the horizontal distance between two adjacent profile index and the elevation of each index, respectively.
In the procedure that gives the name to the method, the sum of squares of the computed errors for each data point must be minimized. First of all, the sum of squares of the errors can be given as:

$$E = \sum\nolimits_{i=1}^{N} \left(e_i\right)^2 = \sum\nolimits_{i=1}^{N} \left(y_i - m.x_i - n\right)^2 \tag{5}$$

Looking for the best representing line for a set of data points means calculating the most appropriate m and n values. Hence, both the differentials of the sum of squares of the errors with respect to m and n must be equal to zero in order to minimize the sum. Solving m and n that satisfy this principal, the best representing line is achieved:

$$\frac{\partial E}{\partial m} = \sum\nolimits_{i=1}^{N} 2.\left(y_i - m.x_i - n\right)\left(-x_i\right) \tag{6}$$

$$\frac{\partial E}{\partial n} = \sum\nolimits_{i=1}^{N} 2.\left(y_i - m.x_i - n\right)\left(-1\right) \tag{7}$$

Despite real data points in a profile do not fit on a line, the profile can be represented by a line with the help of Least Squares line fitting method. Therefore, features of a profile can be generated by taking the advantage of line characteristics.

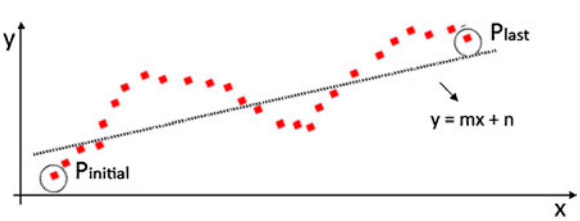

Fig. 7 Fitted line for several data points by least squares line fitting method

b. **Profile distinctive features:** To characterize a profile, we take advantage of some particular features of the line representing it: *Slope angle in degrees (α), the average (h_{avg}), the maximum*

($h_{\max}$) and the minimum altitude ($h_{\min}$) values of the profile, and we construct a feature vector from these features:

$$\left[\alpha, h_{\text{avg}}, h_{\max}, h_{\min}\right]$$

Slope is calculated by the following equation exploiting the line equation:

$$\alpha = Arc\tan(m) \tag{8}$$

3.1.3 Profile Identification

Distinctive features are the main characteristics of a profile, however they are not still easily applicable for comparisons and a more practical and unique identifier for each profile is needed. One of the strongest propositions of this study comes into sight here. The study proffer a unique scoring algorithm that generates a unique, real number identifier for each profile by using features extracted in the previous step. This individual number is named as 'score'.

Slope angle is the most significant feature of the profiles in the study. Slope of line of a profile gives the most powerful idea about how rough the terrain section below trajectory of a flight is, thus it must have the most influential contribution to the score. The average altitude comes after the slope angle, since it can tell the difference between two profiles of the same or similar inclinations. The maximum and the minimum heights may be helpful in some rare cases that both slope angle and average altitudes are close between a measured and reference profile, respectively.

We have developed an algorithm, i.e. evaluation function given in the Eq. 8, to generate different scores for each profile even if they have fairly inseparable feature values with inspiration from decimals. As is known, decimals have units, tens, hundreds etc. and each can be given a number in the range {0, 1, 2, ..., 9}. Mainly, the same number in different digits creates distinct contributions to the whole number thanks to different coefficients of each digit. Utilizing this knowledge on features set, we have accomplished to generate individual identifiers for profiles in the dataset.

Profile feature vectors include slope angle in degrees, thus it has a possible range $(-90, +90)$ of real numbers, and three altitude attributes that we assumed to be in the range (0, 5,000) in meters. Since inclination is the most significant attribute, a variation above a predefined threshold in the slope value must dominate any amount of variations in attributes having less significance and this holds for each attribute more significant than others, i.e. feature ranking. Feature ranking can be achieved multiplying less significant attributes with comparatively small coefficients and multiplying the next significant attributes with a coefficient dominating the previous multiplication. Providing more uniqueness can be obtained by selecting coefficients from prime numbers.

$$f(P_n) = k_\alpha \alpha + k_{h_{\text{avg}}} h_{\text{avg}} + k_h (h_{\max} + h_{\min}) \tag{9}$$

where P_n is the nth profile, k_α is the coefficient of α, $k_{h_{\text{avg}}}$ is the coefficient of $k_{h_{\text{avg}}}$ and k_h is the coefficient of $h_{\max}$ and $h_{\min}$.

After running the profile identification algorithm on every profile generated from DEM in the previous sub-phase, a unique identification number (i.e. score) is acquired for each profile. We place this score (sc) in the feature vector of its profile on account of using it in comparisons, and coordinates of the first and the last measurements in the profile to provide position info of the matched profile, so the new appearance of profile feature vector is as the following:

$$\left[\alpha, h_{\text{avg}}, h_{\max}, h_{\min}, sc,\ x_{\text{first}},\ y_{\text{first}},\ x_{\text{last}},\ y_{\text{last}}\right]$$

Generated profiles should be stored to use for comparisons during flights. Archiving operation must be handled in the way that comparison of a profile of measurements with stored profiles can be performed in the shortest path. To achieve this goal, we store profiles into different files regarding their slope angle and each integer slope angle interval has a corresponding file. This enables comparing measured profile with a small subset instead comparing with the whole DEM and reduces the search space from millions to ten thousands.

3.1.4 Storing Profiles

Generated profiles must be stored in order to be used in comparisons. However, for the sake of finding the flight profile in the whole set of terrain profiles in the shortest way, archiving procedure must be handled in a smart manner. To cope with such a requirement, profiles sorted regarding both their slope angle and score. Because of the fact that there exists unrestricted time for pre-processing of the terrain before the flight, sorting is assumed to be achievable in the real UAV study.

Owing to profiles being sorted with respect to slope angle and score, search space gets sorted during the flight. Thus, the profile of the flight trajectory can be tried to find by means of special search algorithms and the process may become dramatically faster.

3.2 Search

Whenever adequate number of measurements is performed by UAV, a profile is created from those measurements. This profile should be compared with DEM profiles and an accurate position must be achieved before UAV goes ahead for a dangerous distance. As stated earlier, profiles are sorted with their slope angle and score, hence we can search the observation profile within a small search space including the exact file containing the inclination interval with extra files from a small neighborhood ($\alpha \pm 2$) in order to cover failures due to possible measurement errors of sensors.

Files corresponding to the small neighborhood includes at most a few hundred thousands of profiles, however they must even be searched by means of a smart search algorithm. Since the profiles in each slope angle file are also sorted by their scores, files are taken into memory separately and all the profiles are searched with the Binary search algorithm [29], because this method guarantees to halve the search space at each iteration on sorted lists, i.e. the required number of comparisons is halved at each iteration.

In Binary search technique, query value is always compared with the value in the middle of the sorted list. If the values are the same, the result is found. If the searched value is smaller than the value in the middle, then the sub-list from beginning to the middle is kept while the remaining half is thrown away, and vice versa. Comparisons are performed recursively after each division step, whenever the result is found, search is finished.

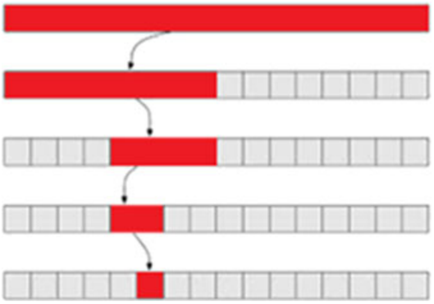

Fig. 8 Binary search

Even in the worst case, i.e. searched value is not in the list, algorithm has O($logn$) complexity rather than tracing through whole the list.

For example, Fig. 8. shows the binary search iterations on a sorted list with 16 entries.

The flowchart of the complete algorithm can be seen in the Fig. 9.

The contribution of the Binary search to the search cost (i.e. number of comparisons) is vital

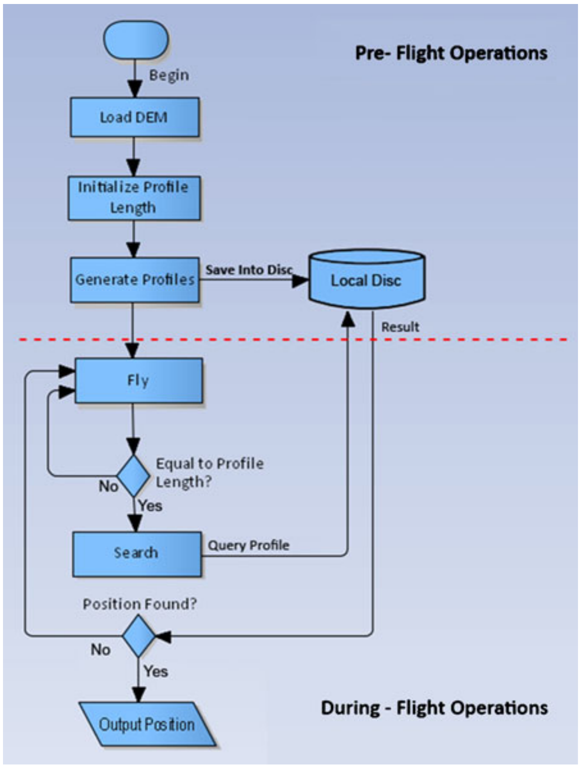

Fig. 9 The flowchart of the complete algorithm

that even millions of profiles can be searched in just a few iterations.

Finally, the original Binary search tries to find exact results, however in TRN systems, measurement errors due to the altimeters may result in flight profiles that does not exist in the terrain data. Indeed, such faulty profiles must be very similar to the existing ones; hence the search algorithm in this study must also have the ability to find these profiles. In order to accomplish this mission, the last profile at the end of comparisons is chosen as the result since it is the closest one.

4 Simulation Results

In order to test our TRN system, we have developed a simulation system in C++ Programming Language and we ran our simulations on each core of a machine in parallel, which has the technical specifications shown in the Table 1. In this simulation, a UAV is created on a random coordinate of the DEM, which is taken as input to the system. UAV is assumed to be lost initially, and starts to fly with the assumptions described before. Basically, when UAV altimeters measure a number of elevation values equal to the length of profiles, which is given as an input to the system; our TRN system starts to look for measured profile in the set of profiles, and returns the result if any.

4.1 Dataset

Firstly, the necessity for having terrain data before flight makes the search space of the problem huge since the possible flight route of a UAV usually covers comparatively large regions. Accordingly, examination of the study must be performed on a realistic map data and such a map is thought to be sized at least $\sim$3.6 $\times$ 3.6 km^2. This minimum size is considered to be useful during a close military mission; however up to almost $\sim$ 15 $\times$ 15 km^2 datafrom DEMs of real-world territories are gathered and used in the simulations of the study. Regarding that DEMs have 30 meter-resolution originally; three different sizes of maps are acquired: 128 $\times$ 128, 256 $\times$ 256 and 512 $\times$ 512, and the system can be examined on $\sim$3.6 $\times$ 3.6 km^2, 7.5 $\times$ 7.5 km^2 and 15 $\times$ 15 km^2 terrains, respectively.

Table 1 Technical specifications of the simulation machine

Property	Specification
CPU	Intel(R) Core(TM) i7-2620M CPU @ 2,70 GHz (4 CPUs), (2 Cores)
Memory	4096 MB, dual channel DDR3-1333 Mhz SDRAM
Cache memory	4 MB

In addition to real-world DEMs, synthetic 10 meter-resolution terrain maps are created by means of a simple interpolation method on account of proving the performance of the system with longer profiles on higher-resolution maps. To achieve 10 meter-resolution maps from a map of 30 meter-resolution, extra two indices are generated between adjacent indices in both rows and columns of the original one, and the elevation values of these extra indices are interpolated from the pre-existing indices. The number of indices of the new map is calculated as follows:

$$n = 3\,(m - 1) + 1 \qquad (10)$$

where the original map is a $m \times m$ and the generated map is a $n \times n$ matrix.

Therefore, a 64 $\times$ 64 index-sized small artificial terrain is generated from just a randomly chosen 22 $\times$ 22 index-sized portion of the largest map in order to try the system on 16-index-long profiles (i.e. again $\sim$90 M profiles of this size can be generated this small map). Both the size of this map and elevation values in it are not realistic, however this map can give hints about the potential performance of the suggested TRN system with longer profiles on terrains represented by high-resolution DEMs. Besides, 16-index-long profile just means that the UAV must fly only 160 m without localization until the position is found. This distance is almost exceeded by even 5-index-long profiles on a 30 meter-resolution DEM.

Secondly, because of the fact that this study does not challenge for flat areas, DEMs from all sizes are taken from the eastern region of Turkey, where roughness is very high. Therefore, analysis of the simulations does not contain any comment with respect to the roughness of the terrains.

Thirdly, after deciding the sizes of the DEM, the next decision should be made on the length of profiles. For the original, 30 meter-resolution DEMs, the study proposes that profile length should be at least 9 to achieve uniqueness of profiles in the terrain and the accuracy of the system. In addition, Binary search approach of the system allows increasing the length of the profile much more, but sequential data processing restricts since longer profiles (i.e. more measurements) mean longer distances without information about the position of the UAV. Thus, the present system with the sequential data processing generates from 9 to12 index-long profiles from each size of the original DEMs for the purpose of examining different profile lengths. However for the synthetic terrain, only 16-index-long profiles are generated to test the system with much longer profiles.

If the total number of profiles in different terrains is mentioned, the largest terrain in the original dataset seems to be containing approximately 90 M of 12-index-long profiles. This is the highest amount of profiles generated from the 30 meter-resolution DEMs in a simulation run for this TRN system. In addition, the interpolated 10 meter-resolution, 64 × 64 index-sized map contains also approximately 90 M 16-index-long profiles.

4.2 Modeling Altimeters

Altitude sensors on the UAV are assumed to be radar and barometric altimeters in this study. These real system components exist as simulated entities in this TRN system. Due to the fact that a radar altimeter operates with ±5 % accuracy at the worst case and the difference between barometric and radar altimeter measurements are used in the algorithm, these measurements are modeled by adding uniformly distributed noise with a 5 % rate to the real data in DEM as shown in Eq. 11. By this way, it is guaranteed that the proposed TRN system not only works with exact data, but also succeeds with approximate (i.e. faulty) observations.

$$h'\left(\mathrm{x},\mathrm{y}\right) = h\left(x,y\right) + h\left(x,y\right).X.P_{err}, \tag{11}$$

where $X \sim U(0,1)$, $h(x,y)$ is the elevation of the(x, y)coordinate in the DEM, $h\prime(\mathrm{x,y})$ is the sensor measurement of the same coordinate and P_{err} is the error percentage.

Speed of the UAV is simply assumed to be 30 m per second in this study. Regarding this speed, altimeters are assumed to be working with 1 Hz in the simulations of the original DEMs and 3 Hz (i.e. one measurement per 10 m of flight) in the simulations of the generated 10 meter-resolution terrain for the sake of taking one measurement per each map index.

4.3 Results

For each DEM size, 1,000 simulation runs are performed in order to obtain a more uniform distribution of flight trajectories in the terrain and the accuracy of results that are found with varying profile lengths are recorded. Additionally, execution time of the search algorithm can be ignored thanks to the Binary search algorithm (i.e. number of comparisons is about a few tens in the worst case), but time to reach the adequate profile length is mentioned in detail.

4.3.1 Accuracy Results

As shown in the Table 2 and the Fig. 10, accuracy results of the algorithm on the 30 meter-resolution DEMs with respect to profile lengths figure out that profile length directly influences the success rate of the algorithm, i.e. for each DEM size, when the profile length increases, the accuracy also increases. On the other hand, no exact idea can be gathered about the effect of DEM size on the accuracy since both increase and decrease can be observed on the accuracy while either minimizing or maximizing the DEM size.

The highest accuracy reported in the study for the original terrain DEMs is 90.4 % with 12 index-

Table 2 Accuracy of results (success percentage) for varying profile lengths in each DEM

Profile	DEM size		
length	128 × 128 (%)	256 × 256 (%)	512 × 512 (%)
9	85.3	84.8	85.6
10	87.1	87.2	87.2
11	88.4	88.9	88.7
12	89.7	90.1	90.4

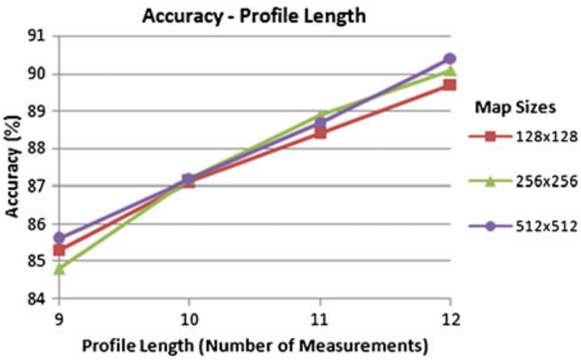

Fig. 10 The accuracy – profile length graph of each DEM size

long-profiles and it is about 5 % above the highest accuracy result of the previous study (i.e. 85.6 % with 8-index-long profiles) [1]. However, it is not enough to be applied on a real system. There are to possible ways to overcome this problem and to increase the success rate. The first option may be increasing the number of features of the profiles and rearranging the weights corresponding to each feature, so more distinctive scores may be acquired. This potentially contributes to the solution, but such an operation may be a subject to the real system study. The second choice is to make profiles longer, hence more distinct profiles can be generated from the same terrain.

In order to examine the accuracy of the system with longer profiles, there exists about 90 M 16 index-long-profiles generated from the synthetic 64 × 64 sized terrain map, which has 10 meter-resolution. The result with such a configuration is

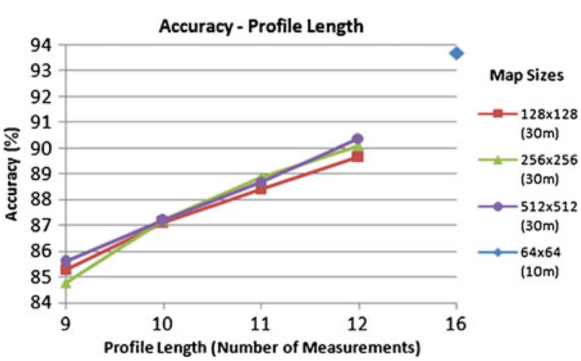

Fig. 11 The accuracy – profile length graph of each DEM size

satisfactory that the success rate is measured to be 93.7 % and is depicted in the Fig. 11. This result is about 9 % above the highest accuracy result of the previous study and proves that this study has the potential to be applied on a real system. The observed improvement on the accuracy with the help of longer profiles shows that the proposed TRN system provides results close to GPS and can be closer to fully correct localization by means of increasing the length of the profiles. On the other hand, even a small map contains almost a hundred million profiles, thus smarter solutions for data storage should be performed for larger terrains.

4.3.2 Time to Localize the UAV

Secondly, the average execution times (i.e. number of comparisons) of the system should be mentioned. First of all, Binary search method in the searching phase diminishes the execution time of the system to negligible levels on all sizes of terrains with all profile lengths. This is because the Binary search method allows the system to finish searching a profile in the largest data set (i.e. ∼90 M profiles) after only 29 comparisons in the worst case. Therefore, time to find a profile in any size of terrains does have no effect on the performance of the TRN system.

On the other hand, the lost UAV must take elevation measurements from the earth until the number of measurements is equal to the predefined length of a profile to search in the terrain data. For the original DEMs, the shortest profile means a flight through 270 m (9 indices*30 m/index) without localization, and the longest profile requires 360 m with the same condition. Together with the necessity to increase the accuracy, the distance to gather the longest profile must be regarded and that seems huge for a navigation study, i.e. flying 360 m with no exact position info is not applicable. In contrast with the original DEMs, the synthetic terrain map with 10meter-resolution allows the system both to fly only 160 m (16 indices*10 m/index) without TRN and to increase the accuracy to 93.7 %. In order to boost up the proposed TRN system at this point, either the resolution of the real-world terrain databases should be improved and the profiles should be made longer together, or the sequential data processing approach must be swapped with the recursive one. Although the latter option requires a new, individual study, the former solution seems to be handled comparatively easier. The reason behind this study does not include higher resolution DEMs is that such real data cannot be reached publicly.

The navigation of a UAV is a real-time problem; hence it requires the TRN system to detect the position of the vehicle in a short range with a reasonable accuracy, which can be accepted only above 90 % as a lower limit. Indeed this percentage is only acceptable in contemplation of having the potential to be improved dramatically through 100 %. To sum up how the accuracy and the performance of the system can be improved, utilization of higher resolution real terrain maps and longer profiles, a detailed review of the profile distinctive features and the weight distributions of those features, and usage of recursive data processing instead of sequential processing can be considered as the very initial ways of solution.

5 Conclusions

In this TRN system proposal, a comparatively accurate and fast terrain data based localization algorithm is implemented and examined with enough number of simulations. Basically, the study exploits the DEM file format as the terrain data, assumes to make use of active range sensors (i.e. radar and barometric altimeters), sequentially operates on sensor measurements, and tries to correlate between measured elevation profiles and the terrain data profiles.

The main strength of the method comes from the well-designed preprocessing phase in which the terrain database (i.e. profiles) is created from the DEM files for fast and accurate lookups. The study guarantees to diminish huge search spaces of traditional TRN approaches into linearly solvable small, sorted spaces. Additionally, the sorted search space is traced with a smart search algorithm (i.e. Binary search) rather than searching linearly. Thanks to the Binary search, hundred thousands of profiles can be looked up with just a few tens of comparisons.

Although the system is implemented as a simulation rather than applying on a UAV, assumptions are made considering to facilitate embedding the system on a real air vehicle. For example, the direction of the flight is not limited, i.e. the UAV is thought to be able to fly through all directions with any turn in the range $(-45, 45)$ degrees between adjacent indices of the terrain map.

Simulation results of the study show that the methodology is almost applicable for a real system on a 15×15 km^2 territory. Especially simulations on the synthetic terrain data proves that high resolutions (e.g. 10 m or higher) on terrain data allows to use longer profiles and eventually results in remarkable improvements on both success rate and the localization cost (i.e. time and distance to find the exact position).

Although the possible improvements to the existing TRN system were mentioned in detail, future studies can be finally concluded as: Larger real terrain data of higher resolutions should be acquired, much longer profiles must be generated from those higher resolution maps and simulated, elevation observations must be processed as each measurement is taken in order to lower the distance flew without localization, and ultimately the powered system may be embedded on a real UAV.

Acknowledgements Thanks to Kadir Kirtac, Zafer Sayar, M. Emin Paca and Alime Eroglu for valuable discussions.

References

1. Eroglu, O., Yilmaz, G.: A novel fast and accurate algorithm for terrain referenced UAV localization. In: Proceedings of International Conference on Unmanned Aircraft Systems ICUAS13, May 28–31, Atlanta, GA (2013)
2. Eroglu, O.: A simulation study on terrain referenced navigation of unmanned aerial vehicles without direction constraint. MSc. Thesis (in Turkish), Computer Engineering, Aeronautics and Space Technologies Institute (ASTIN), Turkish Air Force Academy (TUAFA) (2013)
3. Esmat, B.: Introduction to Modern Navigation Systems. World Scientific Publishing (2007)
4. Temel, S., Unaldi, N.: Opportunities and challenges of terrain aided navigation systems for aerial surveillance by unmanned aerial vehicles. In: Augmented

Vision and Reality, pp. 1–15. Springer, Berlin, Germany (2013)
5. Titterton, D.H., Weston, J.L.: Strapdown Inertial Navigation Technology. Peter Pregrinus Ltd. (1997)
6. IEEE Std.: IEEE Standart for Inertial Sensor Terminology, IEEE Std. 528–2001, IEEE (2001)
7. King, A.D.: Inertial navigation forty years of evolution. GEC Rev. **13**(3), 140–149 (1998)
8. Temel, S.: Developing terrain referenced navigation system for unmanned air vehicles using computer graphics algorithms. MSc. Thesis (in Turkish), Aeronautics and Space Technologies Institute (ASTIN), Turkish Air Force Academy (TUAFA) (2008)
9. Global Positioning System Standard Positioning Service Performance Standard Document, 4th edn. U.S Government Official Performance Standards & Specifications. http://www.gps.gov/technical/ps/ (2008). Accessed 7 July 2013
10. Grewal, M.S., Weil, L.R., Andrews, A.P.: Global Positioning Systems, Inertial Navigation, Integration, 2nd edn. John Wiley & Sons, Inc. (2007)
11. Carroll, J.: Vulnerability assessment of the transportation infrastructure relying on the global positioning system. Technical Report, Volpe National Transportation Systems Center (2001)
12. Kopp, C.: Cruise missiles. Australian aviation. http://www.ausairpower.net/notices.html (2005). Accessed 21 Apr 2013
13. DTED Performance Specification. MIL-PRF-89020B, National Imagery and Mapping Agency (2000)
14. ASTER GDEM Readme File. ASTER GDEM Version1 (2011)
15. Temel, S., Unaldi, N., Ince, F.: Novel terrain relative lunar positioning system using lunar digital elevation maps. In: Proceedings of the 4th International Conference on Recent Advances in Space Technologies, pp. 597–602 (2009)
16. Henley, A.J.: Terrain aided navigation: current status, techniques for flat terrain and reference data requirements. In: Position Location and Navigation Symposium, The 1990's - A Decade of Excellence in the Navigation Sciences, IEEE PLANS '90, pp. 608–615 (1990)
17. Hollowell, J.: Heli/SITAN: A terrain referenced navigation algorithm for helicopters. In Position Location and Navigation Symposium, 1990. Record. The 1990's - A Decade of Excellence in the Navigation Sciences. IEEE PLANS '90. IEEE, pp. 616–625 (1990)
18. Baker, W.R., Clem, R.W.: Terrain contour matching (TERCOM) primer. Tech. Rep. ASP-TR-77-61. Aeronaut. Syst. Div., Wright-Patterson AFB, OH (1977)
19. Carr, J.C., Sobek, J.L.: Digital scene matching area correlator (dsmac). In: Image Processing for Missile Guidance, Proceedings of the Society of Photo-Optical Instrumentation Engineers, vol. 238, pp. 36–41 (1980)
20. Yigit, H., Yilmaz, G.: Development of a GPU accelerated terrain referenced UAV localization and navigation algorithm. J. Int. Robotic Syst. **7**(1–4), 477–489 (2013)
21. Golden, J.: Terrain contour matching (TERCOM): a cruise missile guidance aid. In: SPIE Image Processing for Missile Guidance, vol. 238 (1980)
22. Robins, A.: Recent developments in the 'TERPROM' integrated navigation system. In: Proceedings of the ION 44th Annual Meeting (1998)
23. Hagen, O.K.: Terrain Navigation Principles and Application. Geodesiog Hydrografidagene Lecture Notes (2005)
24. Hollowell, J.: Heli/SITAN: A terrain referenced navigation algorithm for helicopters. In: Position Location and Navigation Symposium, 1990. Record. The 1990's – A Decade of Excellence in the Navigation Sciences. IEEE PLANS '90. IEEE, vol., no., pp. 616–625 (1990)
25. Gustafsson, F., Gunnarsson, F., et al.: Particle filters for positioning, navigation, and tracking. IEEE Trans. Signal Process. **50**(2), 425–437 (2002)
26. Bergman, N., Ljung, L., Gustafsson, F.: Terrain navigation using Bayesian statistics. IEEE Contr. Syst. **19**(3), 33–40 (1999)
27. NVIDIA: NVIDIA CUDA C Programming Guide. NVIDIA Corporation. Version 4.0. http://developer.nvidia.com/nvidia-gpu-computing-documentation (2011). Accessed 13 July 2013
28. Williamson, J.H.: Least-squares fitting of a straight line. Can. J. Phys. **46**(16), 1845–1847 (1968)
29. Nievergelt, J.: Binary search trees and file organization. ACM Comput. Serv. **6**(3), 195–207 (1974)