2013 International Conference on Unmanned Aircraft Systems (ICUAS)
May 28-31, 2013, Grand Hyatt Atlanta, Atlanta, GA

A Novel Fast and Accurate Algorithm for Terrain Referenced UAV Localization

Orhan Eroglu[1], Guray Yilmaz[2]

[1]*Center of Research for Advanced Technologies of Informatics and Information Security, The Scientific and Technological Research Council of Turkey Gebze, 41470, Kocaeli, Turkey*

[2]*Computer Engineering Department, Turkish Air Force Academy Yesilyurt, 34149, Istanbul, Turkey*

[1]orhan.eroglu@tubitak.gov.tr

[2]g.yilmaz@hho.edu.tr

Abstract— **Recently, unmanned aerial vehicles (UAVs) have become one of the most popular and promising means for both military and civilian posts and academic research areas. Localization of the UAVs and persistent tracking of a UAV have vital importance to provide a UAV with navigation information and help to cope with getting lost permanently. Indeed, Inertial Navigation System (INS) and Global Positioning System (GPS) seem to be adequate for navigation of UAVs. However, an alternative augmented navigation system for UAVs should be taken into consideration since INS has accumulated errors and GPS always has the possibility of jamming and satellite signal loss. Terrain Referenced Navigation (TRN) could be a good alternative as a decision support system for these main systems. This study aims to detect the location of a lost or GPS-disabled UAV throughout a planned flight by using only the terrain data. In addition, assumptions and limitations are minimized for the sake of simplifying the process to apply this methodology on a real UAV in the future, e.g. flight through all directions with physically possible turn rates is allowed. In order to provide data of the terrain, Digital Elevation Model (DEM) of the flight region with 30m resolution is exploited. The proposed method is based on searching and matching the collected elevation values of the terrain below UAV within the DEM and makes use of simulation techniques to test the accuracy and performance. The whole algorithm utilizes a sequence of elevation values with a predefined length (i.e. profile). Mainly, all possible profiles are generated before the flight and stored in a huge search space. We identify, sort and classify these elevation profiles in order to perform search operations in a small subset of the huge set. During the flight, a sequence of terrain elevations, which is computed with the help of radar and barometric altimeter measurements, is searched within a small neighborhood of corresponding profile set.**

I. Introduction

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.

In aeronautics, the most favourite 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 [1].

INS is a well-known localization system that has been exploited in aeronautics since 1930s. Mainly, this system computes navigation parameters by measuring speed, acceleration and angular speed in all three dimensions. Since all inputs of the system are inertial and internal, INS is robust to external interferences and cannot be discomposed. However, localization in this system requires integrating measurements and this results in accumulated errors throughout flight. This behaviour makes INS need to be supported by some extra systems [2].

The other navigation system, GPS, which is based on the satellite technologies, is a worldwide navigation system, which has started to be developed in 1970s to tackle with constraints of other positioning systems. GPS overcomes the accumulated error of INS, because it enables positioning at each measurement [3]. On the other hand, GPS signals may be easily jammed [4] and may be inapplicable in some geographical conditions such as a deep valley or sides of a number of mountains.

Deficiencies of two systems under consideration create undeniable handicaps not only in military objectives, but also in some special civilian usages. This inadequacy case leads to need for a more accurate solution of navigation or at least a decision support system for the main positioning framework.

Actually, terrain based positioning has been used for all the history of aeronautical experiences. Even in the most primitive aircrafts, a navigation pilot used to try to localize the vehicle via using paper-printed territory maps on his hand. Besides, systematic studies on territory-based navigation started in early 1970s to minimize accumulated errors of INS. In these studies, TRN was examined by military aspects and was tried on Tomahawk missiles [5], and proved signal of success. Nevertheless, the performance of both navigation sensors on the aerial vehicles and computer systems to perform navigation, and insufficient resolution of terrain data restricted these systems from to be improved more in those years.

In the following decades, computers have undergone huge evolutions, altitude sensors have been fantastically improved also, and data of territories have reached to high resolutions. These developments have given new opportunities to researchers for studying on terrain based positioning more closely.

Basically, TRN systems work as follows:

1. Digital Elevation Model (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 equation (1),

$$\boldsymbol{h_t} = \boldsymbol{h_b} - \boldsymbol{h_r}, \quad (1)$$

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

From this sight of view, TRN may be easily and effectively applicable for navigation of aerial vehicles since it is independent of GPS and is firm to external attacks. On the other hand, accuracy of TRN is much dependent on the height values in DEM, i.e. it gets more accurate on more rough terrains.

By courtesy of being resistant to external factors, TRN is just what was needed especially in military practices. A military aerial vehicle can perform GPS-free localization successfully over a known territory by means of using TRN. TRN systems have already been used in several fighter aircrafts and missiles. Nonetheless, such military applications of TRN cannot be analysed in detail as an academic study due to privacy restrictions, unfortunately.

In this study, we aim to focus on all existing TRN approaches, to concentrate on applying these systems on UAVs and to describe a novel fast and accurate search algorithm. To achieve these goals, this study simulates flight of a UAV on a known DEM 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 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.

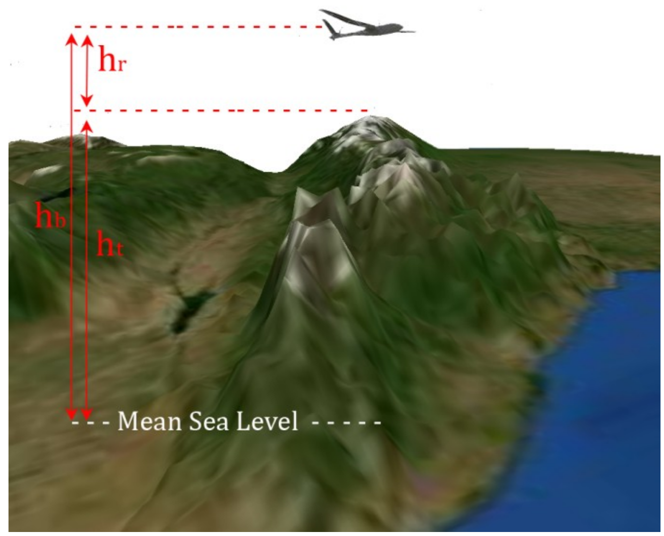

Fig. 1. Instantaneous computation of terrain elevation

II. Related Work

Terrain referenced navigation systems designed for UAVs are basically built up in order to find the position by searching real-time measurements from sensors in the territory. Besides, the kind(s) of sensors, the process of sensor measurements, and the types of searching approaches are all the factors that make TRN algorithms differ from each other.

First of all, embedded sensing media are the very initial actors playing crucial role on the operation of the algorithms. In other words, technology of sensors directly influences the method of localization. TRN systems can be classified in two categories with respect to their sensor components: *Passive imagers* and *active range sensors* [6].

The passive imaging sensors simply take spatial or spatio-temporal images of the territory. Conventional and fisheye cameras are the most popular and widely used examples of this category in UAVs [7]. This class of sensors offers a large-scale price and size range; on the other hand they have few disadvantages. A passive imager may easily be affected by changes in the illumination and the weather conditions, e.g. it cannot support beneficial localization data in the night, or in a foggy day. Data supported from this class of sensors may require image-processing knowhow.

The active range sensors are components, which can periodically support altitude measurements. The very examples of this category can be listed as: Radar, sonar, lidar and barometric altimeter. Although these sensors are consistent to light and weather conditions, they lead to some other difficulties, e.g. the higher the altitude, the bigger the measurement error [8]. Since these sensors may be very noisy at respectively high altitudes, they can be inoperable. On the other hand, the data gathered from active ranging units contain simply the altitude values, and can even be processed and evaluated utilizing the basic computational sciences and the engineering skills. Therefore, the active range sensors have acquired wider usage than the passive imagers on terrain aided positioning.

Secondly, TRN systems can be split into two categories considering their searching phenomena: *Correlation techniques* and *pattern matching techniques* [8].

Basically, in the 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 [8]. As the very first examples of this method, TERCOM (Terrain Contour Matching) is a well instance of altitude-to-map correlation [9] and DSMAC (Digital Scene Matching Area Correlator) can be given for image-to-map correlation [10].

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, On-board 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* [11].

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 [11].

The following subsections describe the most significant TRN algorithms:

2.1. 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) [12]. 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 [9].

2.2. 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 fighter aircrafts such as F-16, Eurofighter Typhoon etc. In fact, TERPROM utilizes TERCOM system for acquisition mode and SITAN algorithm for tracking mode [13]. TERPROM is started with acquisition mode at the beginning of a flight and just before the take-off, 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 Fig. 2 and Fig. 3, respectively. In tracking mode, radar altimeter periodically measures elevation below the vehicle, and INS failures are minimized via comparing observed values with the real elevation data from DEM [14].

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 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 [15], DSMAC [10] do not have been studied and designed for UAVs and there is not any specialized TRN framework known to be operating on a UAV.

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.

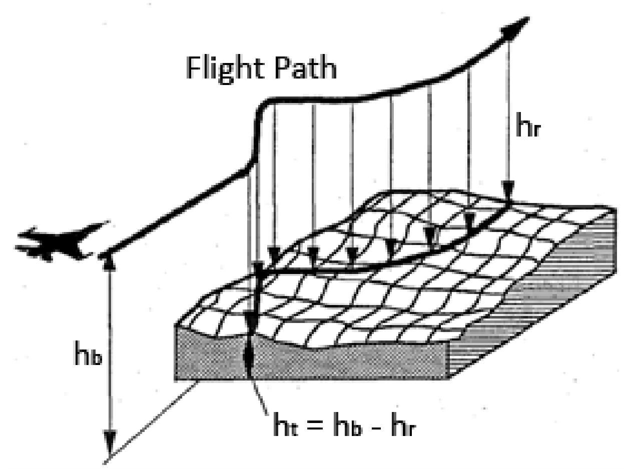

Fig. 2. Possible flight path in the TERPROM Acquisition Mode

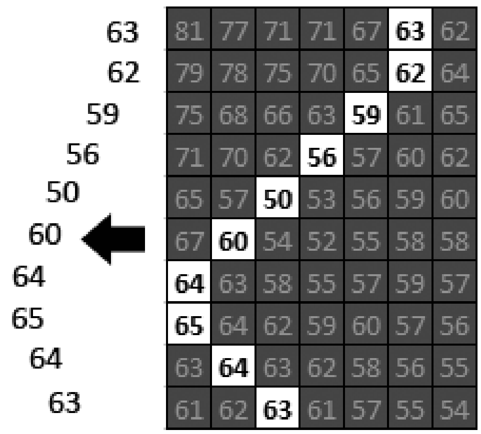

Fig. 3. Representation of a flight path in the terrain map in the TERPROM Acquisition Mode

III. Proposed Methodology

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, on the road operation, the method must be as fast as possible to cope with the case of localizing a UAV in an acceptable amount of time. 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.

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 solution space of terrain-referenced search is so huge that searching in raw data of the territory may result in an impossible hunting. Therefore, 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 three sub-phases:

3.1.1. ***Profile Generation:*** The very first decision of the algorithm is made on the required number of sensor observations of elevations to achieve a valuable search with agreeable performance. The system should provide correct position info until the lost UAV moves ahead a risky distance, i.e. <300 meters. Since the resolution of the DEM data is 30 meters, the number of measurements can be reckoned as 5 to 10 for the sake of positioning UAV in a safe range, and radar and barometric altimeters should be set up to take 1 measurement per second for a UAV moving with a velocity of ~30 meters per second. These measurements construct a sequence of elevation values and the term ‘profile’ is employed for this sequence in this study. In addition, less number of observations will be inadequate since lots of profiles may exist in terrain with similar features if profiles are not long enough.

Additionally, in order to simulate realistic flights, UAV is assumed to be able to change its heading within the range (-45, 45) degrees between two successive measurements as shown in Fig. 4. From this sight of view, the consecutive positions of the UAV can be seen as members of 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. 5. Even a small DEM may include millions of profiles (e.g. There exists almost a million 7 index-long profiles in a 64x64 sized DEM). These profiles are going to be compared with a measured profile of real-time flight in order to find the position of the UAV. Hence, each profile should be identified by means of significant characteristics for accurate comparison outputs.

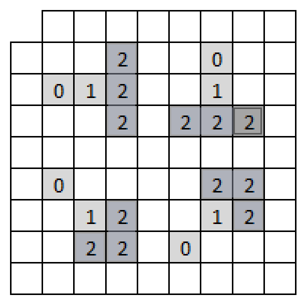

Fig. 4. Possible next positions (indices labeled with 2) when the first two positions of the UAV are indices numbered with 0 and 1, respectively. Opposite directions are available as symmetric of represented ones.

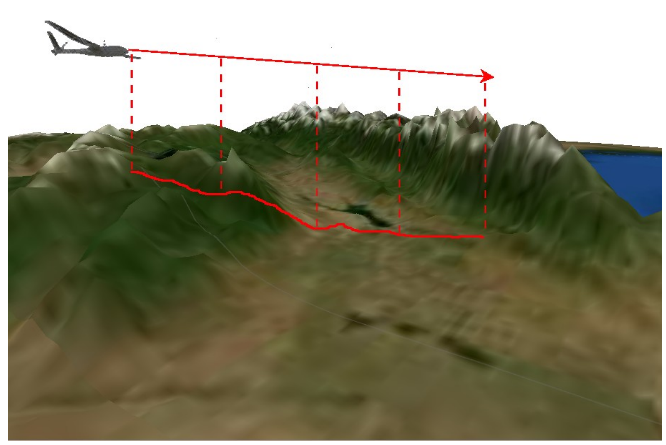

Fig. 5. A possible flight of a UAV and its profile

3.1.2. ***Profile Featuring:*** Since a profile is a sequence of altitudes of travelled coordinates by a UAV, it keeps track of elevation 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 [16] using each coordinate and depicted in the Fig. 6 below:

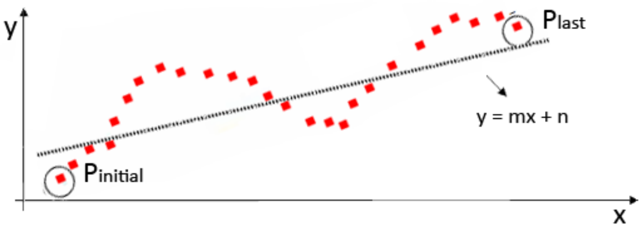

Fig. 6. Fitted line for several data points by Least Squares Line Fitting Method

To characterize a profile, we take advantage of some particular representative features of the line: *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:

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

Slope angle is calculated by the following equations:

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

$$\alpha = Arctan(m), \tag{3}$$

3.1.3. ***Profile Identification:*** These features are the main characteristics of a profile, however they are not still easily applicable for comparisons and a more practical identifier for each profile is needed. One of the strongest propositions and thus the novelty of our algorithm come into sight here. We proffer a unique scoring algorithm that generates a unique, real number identifier for each profile by using features extracted in the previous step.

Slope angle is the most significant feature of our profiles in the study. Since the slope of the line of a profile gives the most powerful idea about how rough the terrain section below the trajectory of a flight is, it must have the most influential contribution to the profile score. The average altitude comes after 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 the case 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 equation (4), 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 digit 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 diverse 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, 5000) 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. In order to obtain uniqueness in these multiplications, the amount of variation of a digit must be forecasted, thus we predict the slope angle to vary with at least 0.1 degrees and elevation features to vary with integer values. Additionally, providing more uniqueness can be obtained by selecting coefficients from prime numbers.

$$f(P_n) = k_\alpha\, \alpha + k_{h_{avg}} h_{avg} + k_h (h_{max} + h_{min}), \tag{4}$$

where P_n is the n^{th} profile, k_α is the coefficient of α, $k_{h_{avg}}$ is the coefficient of h_{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:

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

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 interval between integer slope angles 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.

3.2. *Search*

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

Files corresponding to the small neighbourhood includes at most a few hundred thousands of profiles, thus they can even be searched by means of a linear approach. These files are taken into memory and all the profiles are linearly compared with the profile supported from sensors. To describe similarities in the comparisons, we obey three levels of thresholds from tight to extensive and create three lists corresponding to these thresholds. The list of tight threshold keeps profiles of very close similarity, and that of extensive one keeps profiles from less similarities.

Obviously, the most of searches result in a case that tight list contains one or few profiles; intermediate list includes much more profiles and the wide list covers lots of profiles. However, occasional conditions may turn out with an empty tight list and it is the main reason behind the existence of additional list in the search phase.

Algorithm starts to final step with the first nonempty list starting from the tight one. If list only contains one profile, it is the most similar profile with the measurement in DEM. However, if there are multiple profiles, profile in the first index is selected as UAV's flight profile. The coordinate of the last point of profile recorded in the feature vector is announced as the position of UAV.

The flowchart of the complete algorithm can be seen in the Fig. 7.

IV. EXPERIMENTS AND RESULTS

In order to test our TRN system, we have developed a simulation system in C++ Programming Language and we ran our simulations on a single core of a machine, which has the technical specifications shown in the Table 1. In this simulation, a UAV is created on a random coordinate of DEM taken as input to the system. UAV is assumed to be lost initially, and starts to fly with our assumptions described in part III. Basically, when UAV altitude sensors 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

Necessity for having terrain data before flight makes the search space of the problem huge since 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 ~3.6x3.6 kilometres squared. This minimum size is considered to be useful during a close military mission; however we have prepared and tested up to almost ~15x15 kilometres squared data. Regarding that DEMs have 30m resolution; we have provided three different sizes of maps: 128x128, 256x256 and 512x512, and we can test our algorithm on ~3.6x3.6 km^2, 7.5x7.5 km^2 and 15x15 km^2 terrains, respectively. Furthermore, all three sizes of DEMs represent rough territories since this study does not aim to cope with drawback of TRN systems on flat terrains.

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	4096MB, Dual Channel DDR3-1333Mhz SDRAM
Cache Memory	4MB

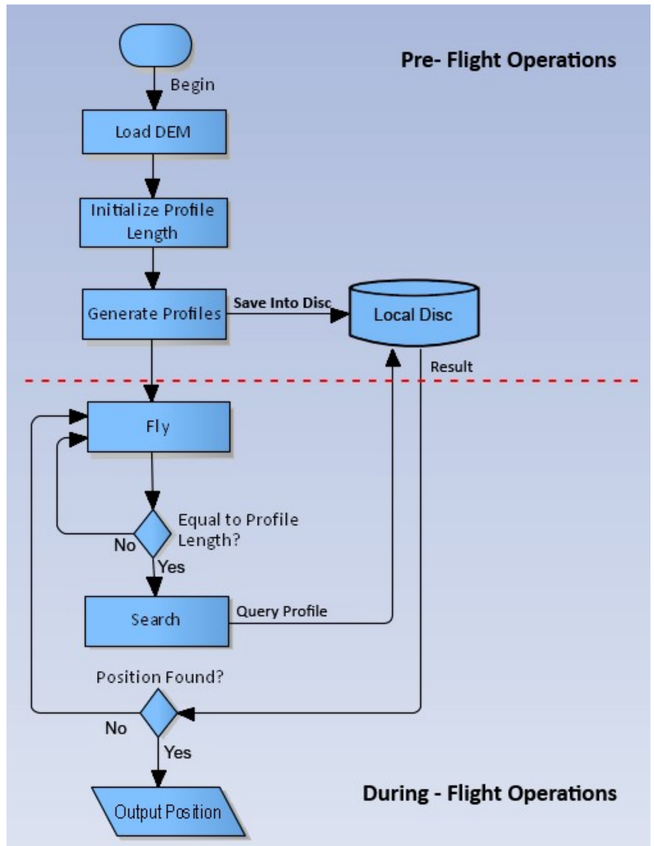

Fig. 7. The flowchart of the complete algorithm

After deciding the size of the DEM, the next decision should be made on the length of profiles. In our proposals, which are designed after several experiments on a large spectrum of lengths, profile length should be from 5 to 10 indices to provide both uniqueness of profiles in DEM and performance of the method. Thus, we generated 5, 6, 7 and 8 indices long profiles from each size level of DEMs for the purpose of examining different profile lengths. Table 2 shows the number of profiles generated in each DEM size with each profile length.

4.2. *Sensor Measurements*

Sensors are assumed to be radar and barometric altimeters in our study. These real system components exist as simulated entities in our TRN system. Due to the fact that a radar altimeter operates with ±3% accuracy at the worst case and the difference between barometric and radar altimeter measurements are used in the algorithm, we modelled these measurements by adding uniformly distributed noise with a 3 per cent rate to the real data in DEM as shown in equation

(5). By this way, we guarantee that our TRN system not only works with exact data, but also succeeds with approximate observations.

$$h'(x,y) = h(x,y) + h(x,y).X.P_{err}, \quad (5)$$

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

Table 2. Number of generated profiles with chosen length in each DEM size

Profile Length	DEM Size		
	128x128	256x256	512x512
5	461394	1748081	6992324
6	1355019	5300764	20608066
7	3945192	15098038	58392146
8	10733562	41434096	162236284

4.3. Results

For each DEM size, we ran our simulation 1000 times and recorded both termination time of the algorithm and the accuracy of results found with varying profile lengths.

As accuracy results of the algorithm with respect to profile lengths are shown in the Table 3 and the Fig. 8, 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, we cannot have an exact idea about the effect of DEM size on the accuracy since we can observe both increase and decrease on the accuracy while either minimizing or maximizing the DEM size.

Table 3. Accuracy of results (success percentage) for varying profile lengths in each DEM

Profile Length	DEM Size		
	128x128	256x256	512x512
5	76.8 %	79.2 %	78.3 %
6	79.8 %	80.1 %	80.7 %
7	82. 0 %	84.1 %	83.3 %
8	83.9 %	84.2 %	85.6 %

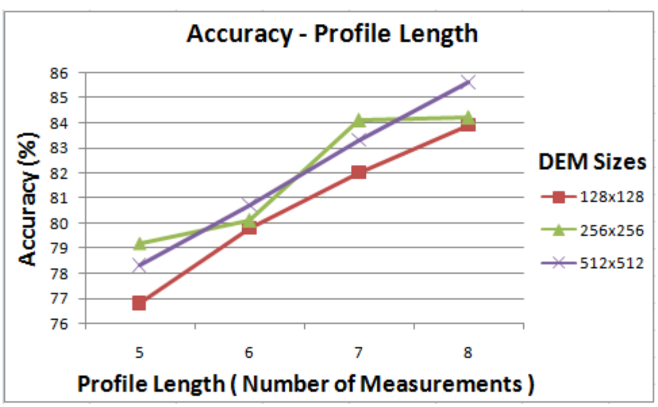

Fig. 8. The Accuracy – Profile Length Graph of each DEM size

Secondly, average running times of the algorithm with respect to profile length for each DEM size are given in the Table 4 and the Fig. 9. Observing execution times, we can have an idea about the applicability of the algorithm on a specific sized DEM with a tested profile length. For example, if we only consider running time results rather than accuracy concerns, we can sum up that 7-index profile length is the best choice for navigating the UAV on a 512x512 sized DEM, since it takes 10165 milliseconds in average to locate UAV, which means that UAV can be found in ~300 meters after starting to try to find the position. However, if we concern accuracy besides duration of algorithm output, we consider 8-index long profiles on the same terrain since the execution of the algorithm with this profile length provides outputs 2.2% more accurate than outputs with 7-index long profiles. Additionally, profiles with 8 indices require 25678 milliseconds in average in the largest DEM and this means the UAV to move ahead about 750 meters. Such a distance may be dangerous for a UAV, and then we must optimize the algorithm in order to work with longer profiles and to increase accuracy.

The navigation of a UAV is a real-time problem; hence it requires us to detect the position of the vehicle in milliseconds or few seconds in the worst case with a reasonable accuracy, which can be accepted only above 85% as a lower limit.

Table 4. Average running times (in milliseconds) of algorithm for varying profile lengths in each DEM

Profile Length	DEM Size		
	128x128	256x256	512x512
5	84	339	1254
6	213	762	2940
7	597	2421	10165
8	1693	6801	25678

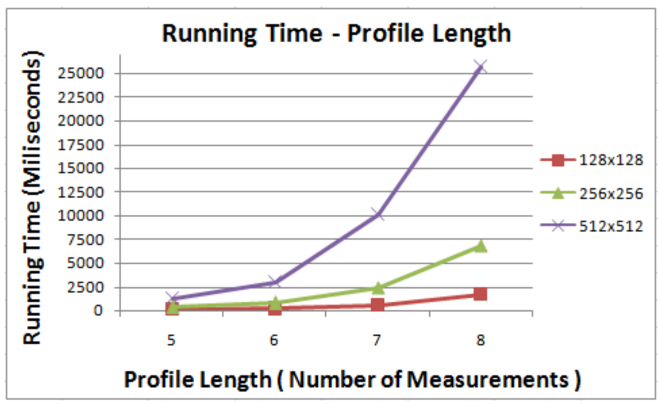

Fig. 9. The Algorithm Running Time - Profile Length Graph of each DEM size

V. CONCLUSION AND FUTURE WORK

In this proposed work, an accurate and comparatively fast terrain referenced navigation algorithm is implemented and tested enough. The main strength of the method comes from the well-designed preprocessing phase in which the terrain DEM is prepared for lookups. This study guarantees to diminish huge search spaces of traditional TRN approaches into linearly solvable small spaces.

Although the system is implemented as a simulation rather than applying on a UAV, assumptions are made considering to facilitate embedding on a real system. The direction of the flight is not limited, i.e. the UAV is thought to be able to fly through all direction with any turn radius in the range (-45, 45) degrees.

Experimental results of the study showed that the methodology is almost applicable for a real system on a 15x15 km^2 territory. Additionally as future studies, we can boost the performance of the existing system with the help of parallel programming techniques, then we can examine the parallelized system both on the same data with longer profiles to increase accuracy and on larger terrains to expand the domain of usage areas, and ultimately we may embed the powered algorithm on a UAV.

ACKNOWLEDGMENT

Thanks to Isa Tasdelen, Onur Ince, Kadir Kirtac, Zafer Sayar, M. Emin Paca and Alime Eroglu for valuable discussions.

REFERENCES

[1] S. Temel and N. Unaldi, “Opportunities and Challenges of Terrain Aided Navigation Systems for Aerial Surveillance by Unmanned Aerial Vehicles,” Augmented Vision and Reality, Berlin, Germany: Springer, pp. 1-15, 2013.

[2] S. Temel, “Insansiz Hava Araclarinda Bilgisayarla Grafik Algoritmalari Kullanilarak Arazi Verisine Dayali Seyrusefer Sistemi Tasarlanmasi,” MSc. Thesis, Aeronautics and Space Technologies Institute (ASTIN), Turkish Air Force Academy (TUAFA), 2008.

[3] M. S. Grewal, L. R. Weil and A. P. Andrews, “Global Positioning Systems, Inertial Navigation, Integration, 2nd Edition” John Wiley & Sons, Inc., 2007.

[4] J. Carroll, “Vulnerability Assessment of the Transportation Infrastructure Relying on the Global Positioning System,” Technical Report, Volpe National Transportation Systems Center, 2001.

[5] J. Copp, “Cruise Missiles,” Australian Aviation, http://www.ausairpower.net/notices.html.

[6] S. Temel, N. Unaldi and F. Ince, “Novel Terrain Relative Lunar Positioning System Using Lunar Digital Elevation Maps,” Proceedings of The 4th International Conference On Recent Advances In Space Technologies, pp. 597-602, 2009.

[7] A. J. Henley, “Terrain aided navigation: current status, techniques for flat terrain and reference data requirements,” Position Location and Navigation Symposium, The 1990's - A Decade of Excellence in the Navigation Sciences, IEEE PLANS '90, pp.608-615, 1990.

[8] A. E. Johnson and J. F. Montgomery, "Overview of Terrain Relative Navigation Approaches for Precise Lunar Landing", Aerospace Conference, 2008 IEEE Volume, Issue, 1-8 March 2008, pp. 1- 10, 2008.

[9] W. R. Baker and R. W. Clem, “Terrain contour matching (TERCOM) primer,” Tech. Rep. ASP-TR-77-61. Aeronaut. Syst. Div., Wright-Patterson AFB, OH, 1977.

[10] J. C. Carr and J. L. Sobek, "Digital Scene Matching Area Correlator (DSMAC)," Image Processing for Missile Guidance, Proceedings of the Society of Photo-Optical Instrumentation Engineers, 238, pp. 36-41, 1980.

[11] H. Yigit and G. Yilmaz, “Development of a GPU Accelerated Terrain Referenced UAV Localization and Navigation Algorithm,” Journal of Intelligent and Robotic System, vol. 7, no. 1-4, pp. 477-489, 2013.

[12] J. Golden, "Terrain contour matching (TERCOM): A cruise missile guidance aid," SPIE Vol. 238 Image Processing for Missile Guidance, 1980.

[13] O. K. Hagen, “Terrain Navigation Principles and Application,” Geodesi-og Hydrografidagene Lecture Notes, 2005.

[14] A. Robins, “Recent Developments in the ‘TERPROM’ Integrated Navigation System,” In Proceedings of the ION 44th Annual Meeting, 1998.

[15] J. Hollowell, “Heli/SITAN: A Terrain Referenced Navigation Algorithm for Helicopters,” 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.

[16] J. H. Williamson, “Least-squares fitting of a straight line,” Canadian Journal of Physics, vol. 46, no. 16, pp. 1845-1847, 1968.