J Intell Robot Syst (2013) 70:477–489
DOI 10.1007/s10846-012-9735-0

Development of a GPU Accelerated Terrain Referenced UAV Localization and Navigation Algorithm

Hikmet Yigit · Guray Yilmaz

Received: 30 June 2012 / Accepted: 13 July 2012 / Published online: 1 September 2012

Abstract This study focuses on localization and navigation of Unmanned Air Vehicles (UAVs) based on digital terrain map data. The solution to the Terrain Referenced Localization and Navigation (TERELONA) or Terrain Referenced Navigation (TRN) is described by using particle filter. In many UAV applications one of the most important points is to provide accurate location information continuously. TERELONA system can supply the air vehicle with the accurate position information with a bounded error. In this paper, the particle filtering method as an implementation of Bayesian approach to the terrain referenced localization and navigation is described. The radar altimeter measurements are used as an implicit representation of aircraft position. Whenever new measurements are taken from radar altimeter, they are compared to the Digital Terrain Map (DTM) data in order to fix a position. The solution is represented, in a Bayesian framework, by a set of particles with their corresponding weights. We have developed the terrain referenced localization and navigation algorithm based on the particle approximation. The proposed algorithm, which is developed in CUDA™, is also tested on the GPU environment using GPUmat software architecture. Thus, we can cope with the computational load of the very large initial horizontal position errors. The proposed algorithm has been implemented in MATLAB™ environment and evaluated on simulated data. Simulations are conducted over an ASTER GDEM product which belongs to a region in northwest of Turkey. The simulation results are provided.

Keywords Terrain referenced navigation · Particle filtering · Digital terrain map data · CUDA · GPU

H. Yigit (✉)
Aeronautics and Space Technologies Institute, Turkish Air Force Academy, Yesilyurt, Istanbul, 34149, Turkey
e-mail: h.yigit@hho.edu.tr, khyigit@gmail.com

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

1 Introduction

Each operation which is conducted with Unmanned Air Vehicles needs accurate position information in order to complete the mission successfully. Nowadays, location information is crucial in every application but it is vital for the navigation applications since the first flight of the aviation history. Basic terrain referenced navigation begins with the aviation history. Aviators have been navigated using topographical maps for matching different land forms visually in the map.

This visual adaptation of the surface and situational awareness is the basic way of navigation in well meteorological conditions. High altitude flights, night flights and meteorological conditions need more complicated methods of navigation. Because, it is unimaginable that it can be navigated at night conditions or in, no visual meteorological conditions without any assistance of navigational aids.

There are different kinds of navigational aids; however, a few of those can provide position information directly to the aircraft. One of the most important position information providers of the aircraft is the inertial navigation system (INS). INS can compute position, velocity, attitude and heading of an aircraft based on measured accelerations and angular rates in three dimensions. This system does not need any information outside to calculate these values but the accumulated error of the INS over time have to be corrected by means of other systems. Global Positioning System is the most common one of these which can provide highly accurate position information, such that the accuracy is 7.8 m with 95 % confidence [1]. However, GPS can be jammed or some failures may occur during operations. These circumstances can cause unrecoverable results which are not desired especially in military operations such as search and rescue, surveillance, reconnaissance and border patrol missions.

Terrain Referenced Navigation is another promising method in order to estimate location information of the flying aircraft by analogizing the measured terrain height values to stored digital elevation map. Therefore, association of these data with the inertial measurement unit data can be exploited to make a good position fix. Moreover, neither the stored data structure nor the measurements are prone to failures or can be easily jammed. Thus, Terrain Referenced localization and navigation can support the navigation system with the reliable position information without any help of outer sources.

The contribution of the present paper is the demonstration of the accurate location information and tracking of an aircraft with using a particle filter method along with the model uncertainties and bounded errors. Moreover, we applied parallel independent particle sets each has the same number of particles. Therefore, we could able to approximate the true position of the aircraft in acquisition phase with at least one particle set. On the other hand, it is demonstrated that the GPU computing can significantly increase the performance of the proposed algorithm especially when using independent particle sets. Thus, real time system requirements for computing could be met. The real time system requirement is dependent on the altitude update interval of the radar altimeter. It is assumed in the paper that the radar altimeter update interval is 33 Hz.

The paper is organized as follows; Section 2 details the Terrain Referenced Navigation and background of the method. Section 3 describes the particle filtering algorithm. Section 4 describes the CUDA™ software and GPU computing. Section 5 discusses the conceptual architecture which the algorithm has been implemented on and the simulation results. Conclusion and the future direction of the work are described in Section 6.

2 Terrain Referenced Localization and Navigation

The "navigation" stems from Latin "navigare" which means presently steering of an aircraft in aviation. Moreover, the navigation system must obtain the geographical position and velocity of the aircraft in order to fly from point A to point B for determining the desired course. These basic requirements of a main navigation system can be met by Terrain Referenced Navigation and other instruments of an aircraft. Especially, TRN provides the accurate position information. Therefore, initially the localization and then the tracking of the aircraft can be done by using this information.

Terrain referenced navigation method is applied to air and underwater vehicles and, a considerable amount of these studies were developed and tested in the eighties and nineties. On the other hand, the study in this domain has been going on especially since 1970's. In general, terrain referenced navigation system can be regarded as an insurance component of the main autonomous navigation system. This main structure is regarded as INS. The terrain referenced navigation system updates the INS in order to provide horizontal position accuracy. The mechanism of terrain referenced navigation utilizes the data of terrain elevation from sea level. The map containing this information should be placed on the vehicle in advance.

The barometric altimeter provides a measurement of altitude of an aircraft above mean-sea level which is denoted by h_b below. At the same time the ground clearance, i.e., the distance between the aircraft and the ground, is measured using a radar altimeter denoted by h_r. The radar altimeter measurements are used whenever available along with the stored elevation data for the purpose of making an inference about the aircraft position. The difference between these two measurements provides a measurement on the terrain height i.e., h_t given by Eq. 1. The mechanism of TRN is depicted in Fig. 1.

$$h_b = barometric\, height\ (altitude\, in\, MSL)$$

$$h_r = aircraft\, above\, ground\, level\ (AGL)$$

$$h_t = terrain\, height$$

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

Especially since 70's, too much research has been done for the purpose of exploiting terrain information in the aircraft navigation domain. Most of the study was made for the cruise missiles and for the fighter aircrafts. It can be seen that every decade has at least one crucial technique for the TRN domain.

In general, TRN algorithms can be simply split into two categories as in [2]; however, sometimes proposed algorithms can be regarded as a member of both categories. These main algorithm categories are batch algorithms and recursive algorithms [2]. Both methods have some disadvantages which they are basically described in their corresponding papers.

Terrain contour matching (TERCOM) is one of the most well-known terrain referenced navigation systems [3]. Basically, the TERCOM correlates the measured terrain elevation profile with the stored elevation map and tries to make a position fix choosing the best match as location information. In TRN methods, TERCOM can be regarded as representative of batch algorithms. This method is applied to terrain referenced navigation system for the navigation of cruise missiles as described in [4]. Moreover, the thesis project [5] gives a broad description on terrain navigation of cruise missiles.

Kalman filtering techniques are the members of another terrain referenced algorithm set. These methods can be regarded as recursive estimation algorithms in general. SITAN is one of the most widely known TAN algorithms which can be

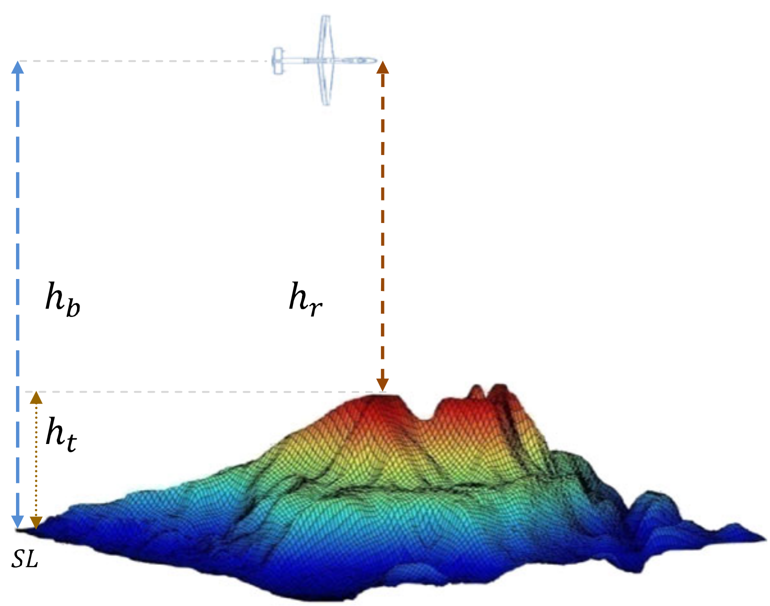


Fig. 1 Terrain referenced navigation

regarded as an example of recursive algorithms [6]. To date, this algorithm has been used by many researchers in considerable amount of academic and scientific researches. The SITAN algorithm is an example of terrain referenced navigation algorithm utilizing Extended Kalman Filter and a local linearization technique to put into practice an algorithm in recursive manner. The main difference between batch and recursive algorithms can be found in the processing of terrain height measurements. Recursive TRN algorithms process the terrain height measurements individually as they become available while the batch algorithms process these measurements after they keep a number of them in an array like data structure.

There are also terrain referenced navigation algorithms which takes the concept one step further and utilizes both batch-oriented and recursive methods such as SPARTAN [7], HELI/SITAN [8] and TERPROM® [9]. Some research has been made utilizing the combination of two techniques such as proposed in [10]. These techniques are correlation method and parallel Kalman filters.

Some other TRN techniques including terrain navigation using Bayesian statics [11], systems based on correlator method such as [12], particle filtering applications for air and underwater navigation [13, 14] were developed and tested since lately 90's.

One of the most challenging issues with the terrain referenced navigation is flying over the flat terrain. Basically, terrain referenced navigation algorithms need undulating terrain in order to give accurate position information for the aircraft. The navigation system with the terrain elevation data should be robust in the face of flat terrain flights. There are some studies on terrain referenced navigation which covers the flat terrain issue such as [7, 10, 15].

In [10], a new terrain navigation method for air vehicles has been described. It is stressed that traditional TRN methods cannot work well while flying over flat terrain. This paper proposes a Combined Terrain Aided Navigation (CTAN) system, which incorporates the correlation method with the parallel Kalman filters. CTAN method calculates the position by analyzing the difference between the DEM data and the set of height measurements collected in real time. The authors propose that using an array of altimeter sensors instead of one sensor under the aircraft. Therefore, at every sampling time, all the measured values from the radar array form a terrain height grid. Then the correlation of this terrain height grid with a given DEM data in larger size is calculated by moving this grid across the DEM data matrix. The correlation is calculated by summing the square sums. Then the expected position $x(t)$ is equal to argmin. of correlation sums. Other alternative to estimate position is Kalman filters as stated in [10], and provide measured values in a recursive manner. The authors explore that previous TAN methods like SITAN use a bank of three-state Kalman filters. It is explored in [10] that it is possible to use Parallel Kalman Filters to correct the error of correlation method. Meanwhile, the valid position fixes provided by the correlation can also be used to detect the Parallel Kalman filters' drift.

PDAF (Probabilistic Data Association Filter) technique is described in [15] in order to minimize association problems between actual position and estimated position of an aircraft. It is stressed that INS is utilized in modern, high accuracy navigation systems, however its estimation will draft away due to initial errors and measurement errors. This method is advantageous for missions flown over fairly flat or very rough terrain or when the aircraft is highly maneuverable. According to [15], main reasons that cause inaccurate navigation are terrain repetitiveness and flatness. Therefore it becomes hard to estimate the actual position of the aircraft. The authors have used PDAF approach in order to convert correlation function value to the probability of position estimate being actual position of the aircraft. Hence, the data association problem between positions estimated and actual position is solved. As the aircraft proceeds over a cell in the reference digital elevation map, the radar altimeter measurements of the aircraft altitude are averaged. This is done because digital terrain data values represent an average of a portion of the area. The paper uses the correlation method. The long flight path is broken into cell matrices which includes the position changes from the start. These position values and their corresponding average altimeter measurements compared with the reference digital terrain elevation to find the position for this data set.

On the other hand, TERELONA can be integrated potential field based navigation techniques as in [16] to enhance autonomous navigation capabilities. On the other hand, the stored digital map on the UAV is likely to be consisting errors which are inherited from the production phase. While the TAN system performance is highly dependent on the stored map quality, the horizontal and vertical resolution of the map should be considered for the overall system performance. The [17] details the application specific conditions which have influences on the terrain referenced navigation.

3 Application of Particle Filter to Terrain Referenced Localization and Navigation for UAVs

The particle filtering can be defined as recursive Bayesian estimation for the purpose of making an inference about dynamic system variables (states). Particle filters describe the posterior density by a number of particles. Bayesian implementation methods are widely used in localization and navigation domain in the recent years. This framework is adapted to estimation of a dynamic system states given independent observations in terrain navigation domain such as [11]. TRN is regarded as a non-linear problem; therefore the underlying Bayesian equations are hard to deal with. Numerical approximation of these equations can be done by Sequential Monte Carlo methods, or particle filters practically in an online dynamic system. There are some studies on particle filtering technique including [13, 14] in air and underwater navigation domain using terrain and depth elevation maps to support the INS. Divergence problems are common in these algorithms and some resampling methods can be used to solve these problems [18, 19]. The overview of the particle filtering technique is presented according to [14, 19–22].

A non-linear discrete-time system can be considered as in below. Equations 2 and 3 indicate generic motion and observation models respectively.

$$x_{t+1} = f\left(x_t, n_t\right) \tag{2}$$

$$y_t = h\left(x_t, e_t\right) \tag{3}$$

$x_t \in R^n$ denotes the state of the system and y_t is the observation at time t. The process noise n_t and measurement noise e_t are assumed independent with densities p_{n_t} and p_{e_t} respectively. Application specific models can be used while implementing particle filtering algorithm for various domains. These application domains are described in [23] with their suitable models. The particle filter method provides an approximate Bayesian solution to discrete-time recursive problem. The state is updated each iteration using measurements. These are radar altimeter measurements for proposed TERELONA algorithm at time t.

The state space which is denoted by x_t stores the variables of an UAV such as; geographical position, altitude, velocity and acceleration at time t.

$$x = \left[p_x\, p_y\, h\, v_x\, v_y\, v_z\, a_x\, a_y\, a_z\right]^T$$

These state space variables are updated in every iteration cycle whenever new radar altimeter measurements are available.

Let $Y_t = \{y_i\}_{i=1}^t$ be the set of observations until present time. The particle filter approximates the probability density $p(x_t|Y_t)$ by a large set of N particles $\left\{x_t^{(i)}\right\}_{i=1}^N$ where each particle has an assigned relative weight, $w_t^{(i)}$ such that all weights sum to unity and reflects the probability density in that region of the state space. From TERELONA algorithm point of view, these particle weights correspond to the probability that an UAV is at position (p_x, p_y) at time t.

Number of the particles may vary over a wide range for different applications, though it can be defined via trial and error method practically. This is a kind of trade-off between computational speed up and approximation accuracy. The particle filter updates the particle location and the corresponding weights recursively with each new observation. Therefore, as the iterations continue, the position uncertainty region of the UAV gets smaller and the real position is approximated.

The non-linear prediction density $p(x_t|Y_{t-1})$ and filtering density $p(x_t|Y_t)$ for the Bayesian inference are given by Eqs. 4 and 5 respectively;

$$p\left(x_{t+1}|Y_t\right) = \int_{R^n} p\left(x_{t+1}|x_t\right) p\left(x_t|Y_t\right) dx_t \tag{4}$$

$$p\left(x_t|Y_t\right) = \frac{p\left(y_t|x_t\right) p\left(x_t|Y_{t-1}\right)}{p\left(y_t|Y_{t-1}\right)} \tag{5}$$

The likelihood can be denoted as $p(y_t|x_t)$ and using the additive noise; $y_t = h(x_t) + e_t$. This is the observation model and y_t is the observation at time t. e_t is the measurement noise and the $h(.)$ is the terrain elevation as a function of the position on the map. For example, $h(p(1))$ is the height at point $p(1)$ according to the on the digital elevation data.

Using the weights the posterior can be written as in Eq. 6,

$$p\left(x_t|Y_t\right) \approx \sum_{i=1}^{N} \tilde{w}_t^{(i)} \delta\left(x_t - x_t^{(i)}\right) \tag{6}$$

where the normalized importance weights are defined as in Eq. 7,

$$\tilde{w}_t^{(i)} = \frac{w_t^{(i)}}{\sum_{j=1}^{N} w_t^{(j)}} i = 1, ..., N. \tag{7}$$

This is the estimation idea without resampling step. Resampling is a method of elimination of low weighted particles. This step is introduced as proposed in [19] and referred to as sampling importance resampling (SIR). The numerical approximation is given below;

1. Generate N samples $\left\{x_0^{(i)}\right\}_{i=1}^{N}$ from the initial distribution $p(x_0)$. Each sample of the state vector is referred to as *particle*.
2. Measurement update: Update the weights by the likelihood.
   Compute $w_t^{(i)} = p\left(y_t|x_t^{(i)}\right)$ and normalize the weight, i.e.,

   $$\tilde{w}_t^{(i)} = \frac{w_t^{(i)}}{\sum_{j=1}^{N} w_t^{(j)}} i = 1, ..., N. \tag{8}$$

   As an approximation to the minimum mean square (MMS) estimate $\hat{x}_t^{MMS}$,

   $$\hat{x}_t \approx \sum_{i=1}^{N} w_t^i x_t^i \tag{9}$$

3. Generate a new set $\left\{x_t^{(i)^*}\right\}_{i=1}^{N}$ by resampling with replacement N times from $\left\{x_t^{(i)}\right\}_{i=1}^{N}$, with probability

   $$\tilde{w}_t^{(j)} = Pr\left\{x_t^{(i)^*}, x_t^{(j)}\right\} \tag{10}$$

4. Predict new particles, i.e.,

   $$x_{t+1}^{(i)} = f\left(x_t^{(x)^*}, n_t^{(i)}\right), i = 1, ..., N \tag{11}$$

   using different noise realizations for the particles.
5. Let $t := t + 1$ and iterate to step 2.

The resampling step is crucial as it prevents a few of the particles to have the high amount of probability density. This is called the "degeneracy" problem. If this step is omitted the filter is not likely to work properly. Different types of resampling methods can be found in the literature as described in [24]. These methods are also evaluated in terms of computational complexity and resampling quality. According to [24], stratified and systematic resampling is favorable over multinomial resampling. Basically, the particle filtering as an implementation of Bayesian inference is one of the most appropriate tools in order to solve localization and navigation problems using stored digital terrain height map data and radar altimeter readings with a proper motion and observation model.

In recent years, particle filters have been utilized for solving many hard problems in robotics. It is a powerful method to cope with the nonlinear problems. While generating high number of particles, we have also noticed that utilizing GPUs would be effective. Due to these reasons we have applied particle filtering method for our work.

4 CUDA and GPU Computing

Air data computer systems have not been so capable of processing and storing huge amount of data when most of the algorithms for TRN were proposed. Nowadays, computer systems have very large disc capacities and even the computational overload of complicated algorithms can be handled. Thus, early proposed algorithms can be adapted and evaluated for up to date hardware systems. As a result, these improvements provide performance increase, accuracy and reliability both for the simulations and real systems. GPUs have been traditionally used only for computer graphics for many years. GPGPU (General-purpose computing on graphics processing units) method allows the GPUs to perform numerical computations usually handled by CPU. The advantage of using GPUs for general purpose computation is the performance speed up that can be achieved due to the parallel architecture of these devices. One of the most promising GPGPU technologies is called CUDA™, [25], introduced by NVIDIA in November 2006 [26]. CUDA™ is a general purpose parallel computing architecture that leverages the parallel compute engine in GPUs to solve many complex computational problems in a more efficient way than on a CPU. Moreover, GPUmat software enables MATLAB code to run on the Graphical Processing Unit (GPU) and provide access to CUDA™ libraries. Therefore, functions can be directly executed on the GPU and the execution is transparent to the user. More detailed information about CUDA™, GPUmat, GPGPU and related topics are provided in [27–29].

The proposed algorithm is developed according to the previous section and implemented in GPUmat. State transition matrix multiplications with the state space variable matrices are done via GPUs except resampling step of the proposed algorithm. Thus, the computational burden of the particle filter increasing with the particle number is handled by GPUs. One of the most important issues is the memory transfer between CPU and GPU while performing calculations on the GPU.

5 Simulations and Results

5.1 System Architecture and Environment

In this section, the simulations are performed according to the high level architectural diagram as depicted in Fig. 2, on a real digital terrain data which belongs to the north-western part of Turkey.

The landform of the region is ranging from flat to undulating terrain. The particle filtering method is applied to numerous TRN simulations on a simulated data. In Fig. 3 the generated height map is depicted which is used in the beginning of

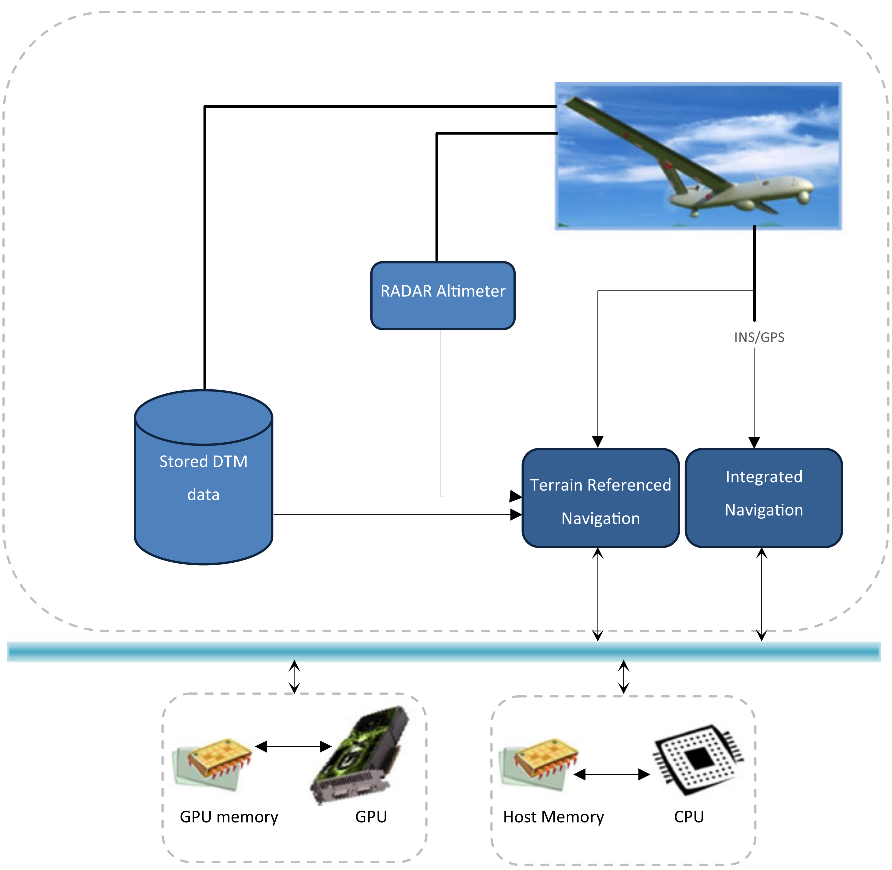


Fig. 2 High level architectural diagram

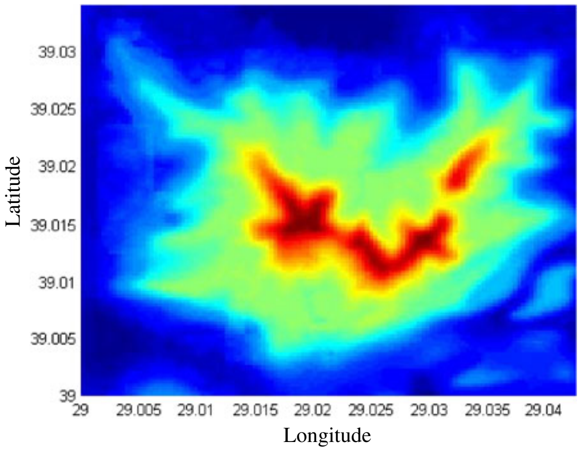


Fig. 3 Generic height map

the study. This is a 129 × 129 dimensional height map matrix.

The UAV is modeled after identifying system components. The position information and radar altimeter measurements are modeled in order to compare the values with the estimated ones. It is assumed that the acceleration signal of the aircraft is available. Thus, the state evolution model can be used for the simulations. The 129 × 129 height map is generated for small initial horizontal position errors and short time flight models as illustrated in Fig. 3. Then we take the study one step further and used a digital terrain map of a real terrain with the size of 3601 × 3601 which is ASTER GDEM product by METI and NASA, and a grid size of 30 m [30]. Therefore, we can evaluate the large initial horizontal position error performance of the proposed algorithm. This ASTER digital height map, which belongs to real terrain, is depicted in Fig. 4.

The process and measurement noise are assumed Gaussian. The simulations are performed with different number of particles for the purpose of evaluating the parallel structure of the particle filter TRN algorithm. In the measurement update phase, the particles are weighted using normal distribution with zero mean.

Performance analyses are performed after each run. The simulation hardware and data specifications are presented in Table 1. The specified hardware below can be utilized on a UAV. Furthermore, some mobile GPU processors can be applied for more small scale UAVs.

5.2 System Operation and Analysis

In the localization step, every radar altimeter reading is processed in order to find the best

Fig. 4 Digital terrain map

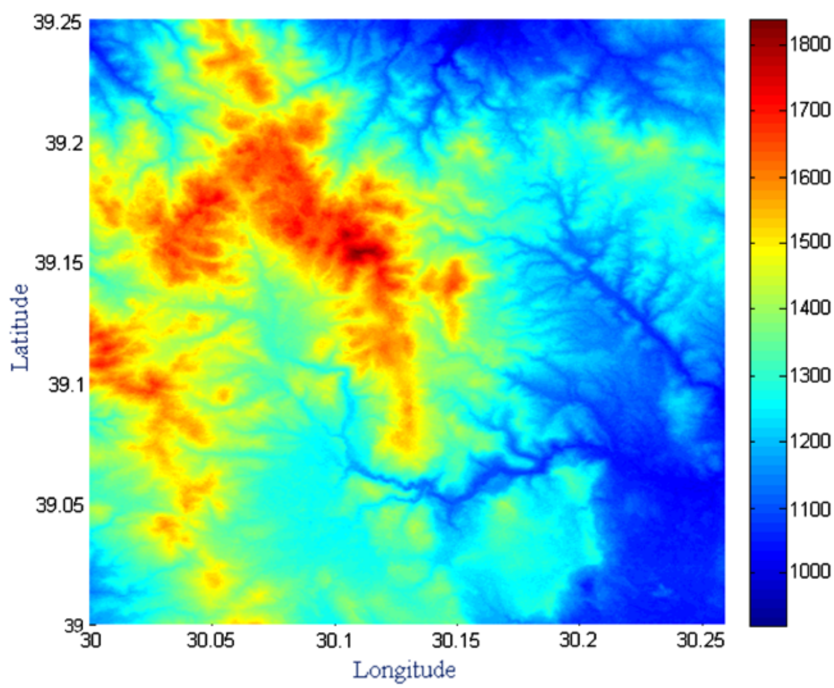


Table 1 Simulation environment

Central processing unit	Intel i7 2600 3.40 GHz	
Host memory	12 GB	
Graphics processing unit	Nvidia GeForce GTX 480	
	1536 MB	
	384 bit	
Height map	Grid posting	∼ 30 m
	Size	3601 × 3601

match in the elevation map by weighting each particle. In the beginning it should be known about the aircraft's position unclearly. The particles are generated and spread around according to this uncertain location information. These initially generated particles are depicted in the Fig. 5. These 3200 particles are generated for 1,44 km square geographical area. During iterations, it is observed that this particle amount is enough to find the exact location of the aircraft in this uncertainty area.

As the radar altimeter measurements are collected, altitude above ground level (AGL) information from each of possible aircraft location are processed and compared to the height map by particle weighting and resampling. Based on the *"Survival of the fittest"* rule, the particles, which have small weight, are omitted. Therefore, the uncertainty cloud of the particle filter gets smaller and the localization of the aircraft is made. This particle evolution is depicted in the Fig. 6.

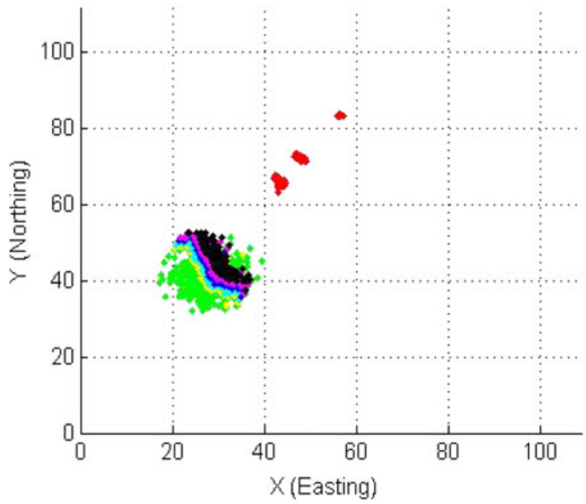


Fig. 6 Evolution of particles

The localization and tracking of the UAV would be done using only radar altimeter readings. Moreover, after the acquisition step, it is enough to use less number of particles in order to track the UAV. Acquisition and tracking steps are consequent processes. These steps are depicted in Fig. 7. In the figure, the results are shown with 4 independent particle sets each has 1000 particles and with approximately 1200 m. initial position error radius. Particle sets are evolving independently and the position fix is made. The dots in the figure are the mean of the geographical position estimates of each particle set during iterations. X and Y axes can be regarded as Longitude and Latitude respectively for geographical orientation.

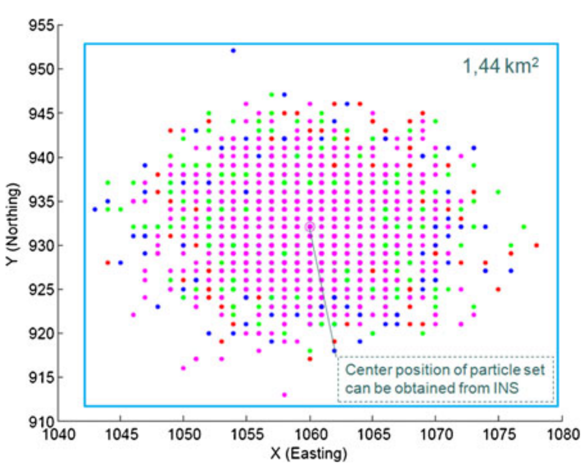


Fig. 5 Initial particle sets

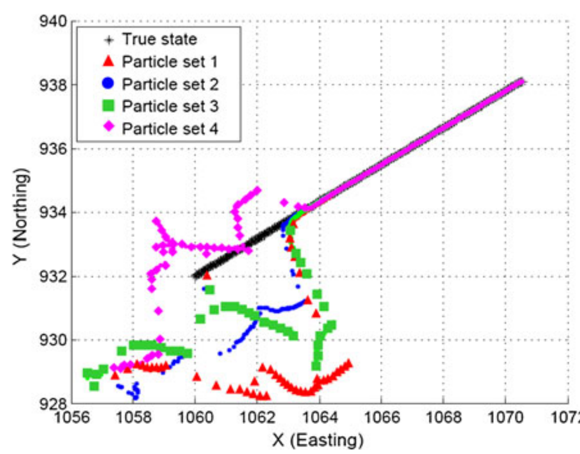


Fig. 7 Acquisition and tracking

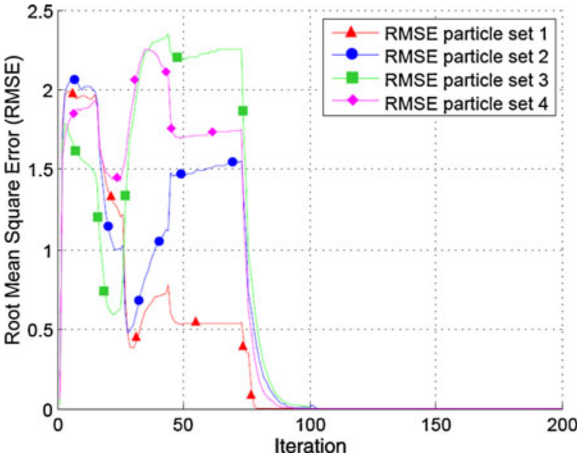


Fig. 8 Position fix of particle sets

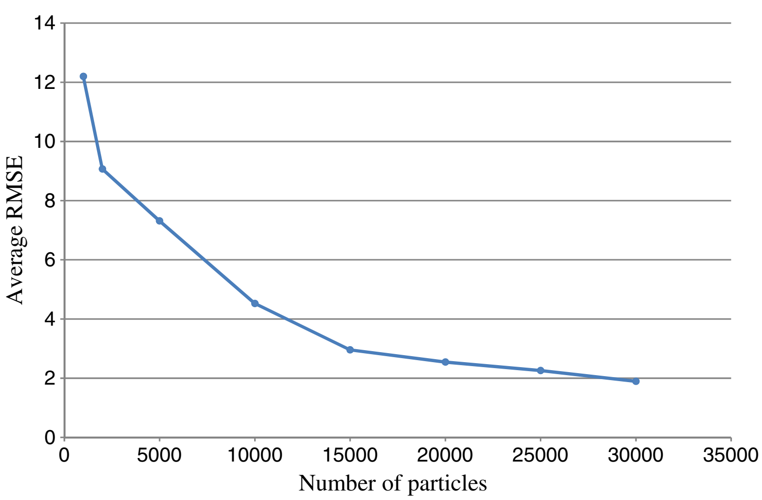
In localization step, the proposed particle filter algorithm for Terrain Referenced Localization and Navigation can estimate the position of the aircraft accurately after a small number of iterations with a few particles. For example, if we use 1000 particles for the ∼500 m bounded horizontal position error, we can succeed to approximate the real position with all of the particle sets as seen in Fig. 8. This figure depicts the root mean square error between the real position and estimated position information of the aircraft.

We applied independent set of particles in order to approximate the true state. However, while using these sets, we need more particles generated at the beginning of the simulations for the purpose of obtaining a position fix. It is clearly observed that, increasing the particle number reduces the average error for localization. Furthermore, increasing the number of particles or utilizing the particle sets requires considerable amount of computational burden which cannot be underestimated while reducing the average error. In the acquisition *(localization)* step, it is assumed that the average error of the position estimate of the aircraft would be ∼3.5 km. The values depicted in Fig. 9 are based on this bounded error in terms of geographical distance. In Fig. 9, it is shown that, as the proposed algorithm's particle number (N) increases, the Terrain Referenced Localization and Navigation system becomes immune to the initial "uncertainty cloud" growth.

Independent particle sets are generated and each particle cloud center is located at a different geographical position. Using this technique, it is assured to find the real position of the aircraft with at least 1 particle set. During the simulations we have evaluated the proposed algorithm for different initial horizontal position errors for aircraft. The position error has been increased incrementally. For example, if it is assumed that the horizontal position error is 4 km at maximum, it is possible to estimate the real state with each particle set with 10000 particles. The results for all particle sets are shown in Fig. 10.

Furthermore, when the initial horizontal position error is too much, for instance the aircraft

Fig. 9 Average RMSE vs. particle number



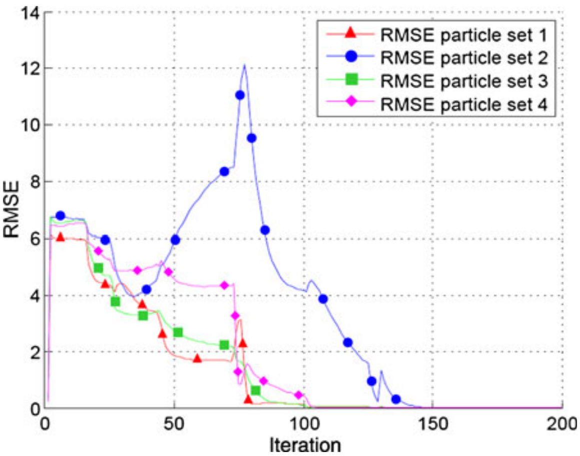


Fig. 10 Position fix in case of large position error

is somewhere in a region which has a radius of 50 km, the filter may not converge successfully to the real state. In such cases, some semi-batch schemes or more independent particle sets would be applied. Moreover, particle sets which their average weights are too low, should be omitted and discarded during iterations runtime for the purpose of avoiding unnecessary processing time.

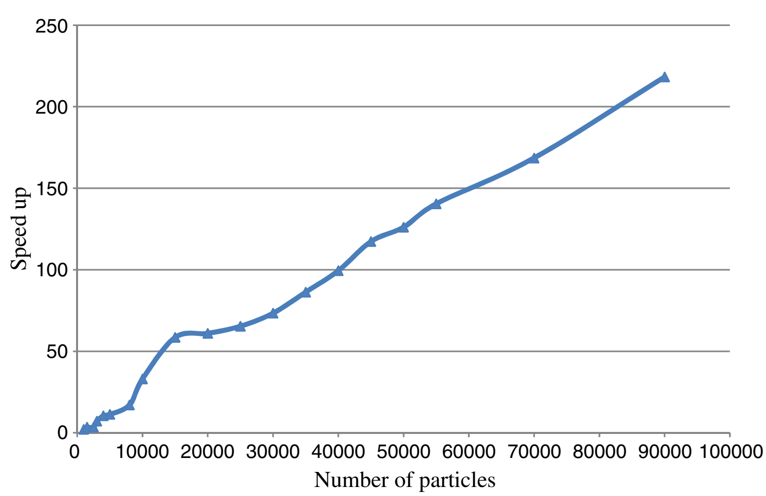
Computational load is dependent on the number of the particles in particle filter algorithm. Huge number of particles may lead to computation time increase which cannot be accepted in navigation domain. The processing time should be limited in order to complete the acquisition phase within reasonable time as the radar altimeter measurements are taken.

GPU provides high performance computing capability for general purpose computing as well as computer graphics. We observed that, positive performance effects of the GPU computing change with respect to the particle number in the proposed algorithm. In GPU computing, speedup refers to how much a parallel algorithm is faster than a corresponding sequential algorithm. Performance analysis is made after the vectorization techniques are utilized to perform multiple operations simultaneously.

Matrix multiplications are made on the GPU when using respectively high number of particles. Therefore, considerable amount of speed up could be observed as depicted in the Fig. 11. The CPU and GPU performance of the algorithm are compared and the results are shown. In the figure, *"speed up"* denotes that how many times faster is the GPU performance of algorithm than on a CPU.

Our parallel structure while performing calculations is based on SIMD (Single Instruction Multiple Data) architecture. As we generated 4 independent particle sets for present paper, we used 10000 particles for each set and 40000 particles in total. Maximum bandwidth could be obtained from Host to Device memory data transfer is 6 GBps with PCI 2.0 $\times$ 16 ports. For example, when we use 12 state variables and 40000 particles, we have a 12 $\times$ 40000 dimensioned state space matrix. If the transition matrix dimension is omitted, our amount of data which will be processed by GPU should be; 12 $\times$ 40000 $\times$ 32 bit = 15360000 bit. This amount of data can be transferred from host

Fig. 11 GPU vs. CPU computing performance



to device approximately in 3 ms, and in 3 ms from device to host. This transfer speed is high enough for performing calculations on time.

6 Conclusion

This paper has presented a particle filtering estimation framework for the problem of positioning and navigating an unmanned air vehicle over a various types of terrain. We could obtain the position fix and able to track the UAV in a small period of time by exploiting the GPU computing even if the measurements update intervals of the radar altimeter are short. The results indicate that the UAV localization is possible using DTM data and radar altimeter measurements, though the approximate position information is not supported or the initial explicit information on aircraft position is not accurate. The position fix and tracking of aircraft after acquisition can be made with a bounded error in terms of RMSEs of estimated states.

The particle filtering algorithm is used in order to take advantage of GPU computing during iterations. We have implemented the TRN algorithm and ported the code to GPUmat in order to use CUDA™ libraries. We plan to continue to investigate the performance of the GPU accelerated TRN algorithm. The source code implementation can be done in C language using directly CUDA™ libraries, so further work can be done based on this implementation in more realistic environment for the memory transfer durations. Furthermore, the authors intend to extend the work using more realistic flight dynamics model such as proposed in [31].

GPU computing has an effect almost on every domain which is required performance increase. This paper describes one of the significant areas where this opportunity can be used. GPUs are likely to be the one of the most crucial members of high performance computing systems. Simulation results showed that considerable amount of speed up can be provided for the proposed TRN algorithm. Moreover, particle filtering architecture in TRN system can be derived in different manner, so that the performance results can be changed respectively. The type of designed TRN architecture and implemented algorithm would be amended for the application purposes.

References

1. Global Positioning System Standard Positioning Service Performance Standard Document: 4th Edition September 2008. U.S Government Official Performance Standards & Specifications. http://www.gps.gov/technical/ps/ (2008). Accessed 15 January 2012
2. Johnson, N., Tang, W., Howell, G.: Terrain aided navigation using maximum a posteriori estimation. In: IEEE Position Location and Navigation Symposium (1990)
3. Baker, W.R., Clem, R.W.: Terrain contour matching (TERCOM) primer. Tech. Rep. ASP-TR-7-61. Aeronaut. Syst. Div., Wright-Patterson AFB, OH (1977)
4. Hicks, S.: Advanced cruise missile guidance system description. Aerospace and Electronics Conference (NAECON 1993) **1**, 355–361 (1993). doi:10.1109/NAECON.1993.290941
5. Ekutekin, V.: Navigation and control studies on cruise missiles. PhD thesis in Mechanical Engineering, Middle East Technical University, Ankara, Turkey (2007)
6. Hostetler, L., Andreas, R.: Nonlinear Kalman filtering techniques for terrain-aided navigation. IEEE Trans. Automat. Contr. **28**(3), 315–323 (1983)
7. Henley, A.J.: Terrain aided navigation: current status, techniques for flat terrain and reference data requirements. In: Position Location and Navigation Symposium, 1990, The 1990's A Decade of Excellence in the Navigation Sciences, IEEE PLANS '90, pp. 608–615 (1990)
8. Hollowell, J.: Heli/SITAN: a terrain referenced navigation algorithm for helicopters. In: Position Location and Navigation Symposium, 1990. The 1990's A Decade of Excellence in the Navigation Sciences. IEEE PLANS '90, pp. 616–625 (1990)
9. Cowie, M., Wilkinson, N., Powlesland, R.: Latest development of the TERPROM® Digital Terrain System (DTS). In: Position, Location and Navigation Symposium, 2008 IEEE/ION, pp. 1219–1229 (2008)
10. Jianchun, X., et al.: Combined terrain aided navigation based on correlation method and parallel Kalman filters. In: 8th International Conference on Electronic Measurement and Instruments, 2007, ICEMI '07, pp. 1-145–1-150 (2007)
11. Bergman, N., Ljung, L., Gustafsson, F.: Terrain navigation using Bayesian statistics. IEEE Contr. Syst. **19**(3), 33–40 (1999)
12. Nygren, I., Magnus, J.: Terrain navigation for underwater vehicles using the correlator method. IEEE J. Ocean. Eng. **29**(3), 906–915 (2004)
13. Flament, M., Lacave, J.N., Fleury, G.: Particle filtering for non-linear sensor fusion: Application to terrain-aided navigation. In: Proceedings of EUCASS'05 Moscow: EADS 1–7 (2005)

14. Karlsson, R., Gustafsson, F., Karlsson, T.: Particle filtering and Cramer-Rao lower bound for underwater navigation. In: IEEE International Conference on Acoustics, Speech, and Signal Processing Proceedings (ICASSP ’03), vol. 6, pp. VI-65-8, 6–10 April (2003). doi:10.1109/ICASSP.2003.1201619
15. Qingtang, F., Lincheng, S., Wenseng, C.: Terrain aided navigation using PDAF. In: 2003 IEEE International Conference on Robotics, Intelligent Systems and Signal Processing, Proceedings, vol. 2, pp. 1063–1068 (2003)
16. Cetin, O., Kurnaz, S., Kaynak, O., Temeltas, H.: Potential field based navigation task for autonomous flight control of UAVs. Int. J. Autom. Contr. **5**(1), 1–21 (2011)
17. Kedong, W., Yang, Y.: Influence of application conditions on terrain-aided navigation. In: 8th World Congress on Intelligent Control and Automation (WCICA), pp. 391–396 (2010)
18. Smith, A.F.M., Gelfand, A.E.: Bayesian statistics without tears: a sampling-resampling perspective. Am. Stat. **46**(2), 84–88 (1992)
19. Gordon, N.J., Salmond, D.J., Smith, A.F.M.: A novel approach to nonlinear/non-Gaussian Bayesian state estimation. IEEE Proceedings on Radar and Signal Processing **140**, 107–113 (1993)
20. Bergman, N.: Recursive Bayesian estimation: navigation and tracking applications. Ph.D. thesis, Linköping University, Dissertations No. 579 (1999)
21. Doucet, A., Freitas, N., Gordon, N.: Sequential Monte Carlo methods in practice. Springer, New York (2001)
22. Thrun, S., Burgard, W., Fox, D.: Probabilistic Robotics. MIT Press, Cambridge, MA (2005)
23. Gustafsson, F., Gunnarsson, F., et al.: Particle filters for positioning, navigation, and tracking. IEEE Trans. Signal Process. **50**(2), 425–437 (2002)
24. Hol, J.D., Schon, T.B., Gustafsson, F.: On Resampling algorithms for particle filters. In: Nonlinear Statistical Signal Processing Workshop, IEEE, pp. 79–82 (2006)
25. NVIDIA: NVIDIA CUDA C Getting Started Guide. NVIDIA Corporation. http://developer.nvidia.com/nvidia-gpu-computing-documentation (2011). Accessed 20 January 2012
26. NVIDIA: NVIDIA CUDA C Programming Guide. NVIDIA Corporation. Version 4.0. http://developer.com/nvidia-gpu-computing-documentation (2011). Accessed 26 January 2012
27. CUDA™: NVIDIA Corporation. http://www.nvidia.com/object/cuda_home.html# (2012). Accessed 10 January 2012
28. GPUmat: User Guide, Version 0.27. http://gp-you.org/ (2010). Accessed 20 January 2012
29. GPGPU: General purpose computing on graphics processing units. http://www.gpgpu.org (2012). Accessed 20 January 2012
30. ASTER: http://asterweb.jpl.nasa.gov (2011). Accessed 10 November 2011
31. Kurnaz, S., Cetin, O., Kaynak, O.: Fuzzy logic based approach to design flight control and navigation tasks for autonomous UAVs. J. Intell. Robot. Syst. **54**, 229–244 (2009)