

Article

Ground Moving Target Tracking Filter Considering Terrain and Kinematics

Do-Un Kim ¹, Woo-Cheol Lee ¹, Han-Lim Choi ^{1,*}, Joontae Park ², Jihoon An ² and Wonjun Lee ³

¹ Department of Aerospace Engineering, Korea Advanced Institute of Science and Technology, Daejeon 34141, Korea; dukim@lics.kaist.ac.kr (D.-U.K.); wlee@lics.kaist.ac.kr (W.-C.L.)

² LIG Nex1, Yongin-si 16911, Gyeonggi-do, Korea; joontae.park@lignex1.com (J.P.); jihoon.an@lignex1.com (J.A.)

³ Agency for Defense Development, Daejeon 34186, Korea; wonjun_lee@add.re.kr

* Correspondence: hanlimc@kaist.ac.kr

Abstract: This paper addresses ground target tracking (GTT) for airborne radar. Digital terrain elevation data (DTED) are widely used for GTT as prior information under the premise that ground targets are constrained on terrain. Existing works fuse DTED to a tracking filter in a way that adopts only the assumption that the position of the target is constrained on the terrain. However, by kinematics, it is natural that the velocity of the moving ground target is constrained as well. Furthermore, DTED provides neither continuous nor accurate measurement of terrain elevation. To overcome such limitations, we propose a novel soft terrain constraint and a constraint-aided particle filter. To resolve the difficulties in applying the DTED to the GTT, first, we reconstruct the ground-truth terrain elevation using a Gaussian process and treat DTED as a noisy observation of it. Then, terrain constraint is formulated as joint soft constraints of position and velocity. Finally, we derive a Soft Terrain Constrained Particle Filter (STC-PF) that propagates particles while approximately satisfying the terrain constraint in the prediction step. In the numerical simulations, STC-PF outperforms the Smoothly Constrained Kalman Filter (SCKF) in terms of tracking performance because SCKF can only incorporate hard constraints.

Keywords: tracking filter; particle filter; soft constraint; DTED (digital terrain elevation model); gaussian process



Citation: Kim, D.-U.; Lee, W.-C.; Choi, H.-L.; Park, J.; An, J.; Lee, W. Ground Moving Target Tracking Filter Considering Terrain and Kinematic. *Sensors* **2021**, *21*, 6902. <https://doi.org/10.3390/s21206902>

Academic Editor: Joohwan Chun

Received: 19 July 2021

Accepted: 11 October 2021

Published: 18 October 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Ground tracking radars mounted on airborne platforms play a key role in many applications, especially those for military purposes; surveillance, airstrike, and escort missions done by aircraft commonly require precise tracking of ground targets. In several modern military campaigns, ground moving target indicator (GMTI) radar on-board the Joint Surveillance Target Attack Radar System (STARS) has been proven strategically and tactically significant [1]. Accordingly, algorithms that track ground targets running on radars are becoming more important. Although there have been great advances in target tracking, tracking ground targets is still a challenging problem. The reason is that the characteristics of ground target tracking are different from those of tracking other types of targets. (e.g., high clutter, terrain obscuration, etc.) [2].

Because exploiting appropriate assumptions other than the state-space model can help to improve the statistical inferences of the system [3], many studies have tried to introduce useful assumptions to ground target tracking. They can be classified based on two criteria of what or how assumptions were applied.

Based on the first criteria, existing studies can be classified into two further categories. The first category considers the behavior characteristics of the ground target that are distinguished from those of airborne targets. Fosbury [4] and Kastella [5] each created the terms 'trafficability' and 'hospitality of maneuver', which represent how easily a

vehicle can go through a particular area. These notions adaptively modify the target dynamics so that the dynamics can reflect the tendency of the target to prefer directions with small gradients. The second category involves empirical constraints. For instance, in the work of Streller [6], ground targets are assumed to move along the infrastructure such as roads, bridges, etc. More specifically, the assumption encourages the prior probability density which is propagated by the system model to align with the road network. The same assumption is utilized in the works of Pannetier [7,8]. The works of Mallick [9] and Kim [10] are also classified into the same category and share the same motivation as ours. To compensate for inaccurate GMTI measurements, both utilized the assumption that the position of a ground target is restricted to the terrain surface. This idea can be extended even further by adding another assumption that the velocity of the target is tangent to the terrain surface, which allowing the system to estimate the velocity more precisely [11,12].

From an other perspective, existing studies can be classified based on the second criteria, namely, how assumptions are applied. The first category involves modifying the target dynamics so that it can reflect the tendency of the target. Similar to the aforementioned works [4–6,13], the system dynamics of the filter are adaptively modified. In other words, external knowledge is embedded in the state-space model. Thus, we have the freedom to control only the tendency of a target. The second category involves transforming the assumption into a state constraint. This type of approach explicitly limits the state of a target to a specific subspace ([7,8] for example).

Extensive studies have attempted to deal with such constrained state estimation problems [11], including the methods that do not rely on the state-space model [14,15]. In the case of linear system dynamics and linear constraints, the following methods are applicable: model reduction [16], perfect measurement [17–19], estimate [20]/system [21]/gain [22] projection, pdf truncation [22], etc. If either system dynamics or constraint is nonlinear, the combination of linearization and linear methods is an available option. Other possible choices are variants of the Unscented Kalman Filter (PUKF [12,23], ECUKF [12], 2UKF [24], etc.), variants of the Particle Filter [25–28] (CLIP, COMP [29]), and the Smoothly Constrained Kalman Filter (SCKF) [30]. Moreover, many works in the literature have paid attention to state estimation problems with soft constraints [18,19,31–34]. Soft constraints, conditions that the state approximately satisfies, are utilized in most practical engineering applications [11,33] because uncertainty may appear during the transformation of external knowledge into the constraint. For example, in the case of ground target tracking constrained to a road, the roadmap may be inaccurate. Among promising methods dealing with soft constraints, some regard the degree of constraint satisfaction as measurement and extend the likelihood function [18,19,31,32,35]. Especially, this approach can be intuitively extended to a nonlinear soft constraint; scPF (soft-constrained Particle Filter) [35] is a good example. scPF has the advantage of preserving the nonlinearity of the constraint because it is based on an SIR (Sequential Importance Resampling) particle filter. However, scPF is not sample-efficient because the constraint is reflected by the generalized likelihood. More specifically, while particles are propagated through the system dynamics, they can be scattered in a direction that does not satisfy the constraint. Therefore, the propagated particles that do not satisfy the constraint would be given a low likelihood and eventually vanish, which makes the whole algorithm inefficient.

Thus, in this paper, we propose a particle filter that considers the stochastic terrain constraint. The term ‘terrain constraint’ not only represents the assumption that the position of a ground target should be located on the terrain surface but also that the velocity vector of the target should be tangent to the terrain surface. Contributions are the following:

- **We propose a sample-efficient particle filter to which the terrain constraint can be applied.** The proposed algorithm is named Soft Terrain Constrained Particle Filter (STC-PF). Given the assumption of target motion, STC-PF performs sampling in a direction for which the state satisfies the constraint during the propagation step. As a result, STC-PF is more sample-efficient than scPF. Furthermore, in the numerical

simulations, STC-PF using soft terrain constraint outperforms Smoothly Constrained Kalman Filter (SCKF)[30] using hard constraint in terms of tracking performance.

- **Using a Gaussian process, terrain constraint is formulated as a soft position constraint along with a soft velocity constraint.** Because kinematics states that position and velocity is not independent, a constraint on the position of a target implies that the velocity of the target will be constrained as well. Therefore, terrain constraint includes both position constraint and velocity constraint. Furthermore, terrain constraint requires exact terrain elevation and its gradient at an arbitrary position, but DTED (Digital Terrain Elevation Data) [36] cannot provide them. To overcome this issue, we model the ground-truth terrain elevation with a Gaussian process (GP) and treat DTED as a noisy observation [37] of it.

Technically, we used SRTM (Shuttle Radar Topography Mission). However, we will use the term DTED and SRTM interchangeably as they both are data that map terrain elevation of the entire globe.

The structure of this paper is as follows: In Section 2, tracking of a ground target with a terrain constraint is formulated. Section 3 presents the proposed algorithm, STC-PF. Section 4 provides detailed explanations, the results, and a discussion of the numerical simulation. Finally, in Section 5, we conclude.

2. Problem Formulation

In this section, tracking of a ground target with terrain constraint is formulated as a constrained state estimation problem.

Consider a system described by the following state-space model:

$$\mathbf{x}_{k+1} = \mathbf{f}(\mathbf{x}_k) + \mathbf{w}_k \quad (1)$$

$$y_k = \mathbf{g}(\mathbf{x}_k) + \mathbf{n}_k \quad (2)$$

where \mathbf{x}_k is the system state vector at time k , y_k the measurement vector, \mathbf{f} the system function, \mathbf{g} the observation function, \mathbf{w}_k the process noise vector, and \mathbf{n}_k the measurement noise vector. The system state vector $\mathbf{x}_k \in \mathbf{R}^6$ consists of the position (x_k, y_k, z_k) and the velocity $(v_{x,k}, v_{y,k}, v_{z,k})$ in local Cartesian coordinates at time k . The system function is a possibly nonlinear function but is assumed to be a constant velocity model in this paper. $y_k \in \mathbf{R}^3$ is the measurement, which consists of range, azimuth angle, and elevation angle measured from the radar. $\mathbf{w}_k \sim N(0, Q)$ is white Gaussian process noise, and $\mathbf{n}_k \sim N(0, R)$ is white Gaussian measurement noise. Subsequently, Equations (1) and (2) are realized as follows:

$$\mathbf{x}_{k+1} = \begin{bmatrix} \mathbf{I}_{3 \times 3} & \Delta t \cdot \mathbf{I}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & \mathbf{I}_{3 \times 3} \end{bmatrix} \mathbf{x}_k + \mathbf{w}_k \quad (3)$$

$$y_k = \begin{bmatrix} \sqrt{x_k^2 + y_k^2 + z_k^2} \\ \arctan \frac{y_k}{x_k} \\ \arcsin \frac{z_k}{\sqrt{x_k^2 + y_k^2 + z_k^2}} \end{bmatrix} + \mathbf{n}_k. \quad (4)$$

The final goal of the state estimation problem is to infer the state sequence of the dynamical system $\mathbf{x}_{0:k}$ from the series of observations $y_{1:k}$.

Now, the terrain constraint can come into play to incorporate the additional information that the state-space model cannot reflect. The terrain constraint not only represents the assumption that the position of a ground target should be located on the terrain surface but also that the velocity vector of the target should be tangent to the terrain surface. Both assumptions can be transformed into state constraints as follows:

$$\begin{aligned} h_k &= \bar{h}(\lambda_k, \varphi_k) \\ v_{h,k} &= \nabla \bar{h}(\lambda_k, \varphi_k) \cdot [v_{\lambda,k} \quad v_{\varphi,k}]^T \end{aligned} \quad (5)$$

where λ_k , φ_k , and h_k are the latitude, longitude, and altitude (LLA) of the target at time k . $\bar{h}(\lambda, \varphi)$ is ground-truth terrain elevation at latitude λ and longitude φ . Note that we do not have direct access to \bar{h} , but only noisy observations,

$$D = \{\text{DTED}(\lambda^i, \varphi^i) | i = 1 \cdots N_D\} \quad (6)$$

such that

$$\text{DTED}(\lambda, \varphi) = \bar{h}(\lambda, \varphi) + \epsilon(\lambda, \varphi). \quad (7)$$

3. Soft Terrain Constrained Particle Filter

In this section, the newly proposed algorithm, Soft Terrain Constrained Particle Filter (STC-PF) is derived. In Section 3.1, mathematical modeling of ground-truth terrain elevation is presented. Then, we propose a technique for the transformation of velocity between the LLA coordinates and the local Cartesian coordinates in Section 3.2. Necessary assumptions required for algorithm derivation are described in Sections 3.3. After the algorithm derivation in Section 3.4, we show the similarity between STC-PF and scPF [35] in Section 3.5.

3.1. Modeling of Ground-Truth Terrain Elevation

Although the terrain constraint (Equation (5)) requires the ground-truth elevation, it is almost impossible in practice to retrieve it at an arbitrary position. The reason is that DTED provides neither accurate ground-truth terrain elevation (Equation (7)) nor terrain elevation at arbitrary positions. (Equation (6)) This challenge can be met by reconstructing the ground-truth terrain elevation with a Gaussian process (GP) and treating the DTED as independent observations:

$$\begin{aligned} \bar{h} &\sim GP(m(\lambda, \varphi), k((\lambda, \varphi), (\lambda', \varphi'))) \\ \text{DTED}(\lambda, \varphi) &= \bar{h}(\lambda, \varphi) + \epsilon \\ \epsilon &\sim N(0, \sigma_{\text{DTED}}) \end{aligned} \quad (8)$$

where the observation noise σ_{DTED} can be estimated from the work of Rodriguez [37]. (see Appendix B) Because GP assigns a probability for each possible terrain, the terrain constraint becomes stochastic. An advantage of this approach is that it enables us to compute the gradient of \bar{h} analytically, which is required to apply the velocity constraint. (Equation (5)) More strictly, joint predictive distribution for ground-truth terrain elevation and its gradient can be expressed in a closed-form, (detailed description is in Appendix A)

$$\bar{h}, \nabla \bar{h} | \text{DTED} \sim N(\bar{\mu}, \bar{\Sigma}) \quad (9)$$

provided that the kernel function is differentiable. Figure 1 shows an example of prediction results when zero mean function and squared exponential kernel are utilized.

$$\begin{aligned} m(\lambda, \varphi) &= \mathbb{E}[\bar{h}(\lambda, \varphi)] \\ &= 0 \\ k((\lambda, \varphi), (\lambda', \varphi')) &= \mathbb{E}[(\bar{h}(\lambda, \varphi) - m(\lambda, \varphi))(\bar{h}(\lambda', \varphi') - m(\lambda', \varphi'))] \\ &= \alpha \exp\left(-\frac{1}{2} \begin{bmatrix} \lambda - \lambda' & \varphi - \varphi' \end{bmatrix} \Gamma \begin{bmatrix} \lambda - \lambda' \\ \varphi - \varphi' \end{bmatrix}\right) \end{aligned} \quad (10)$$

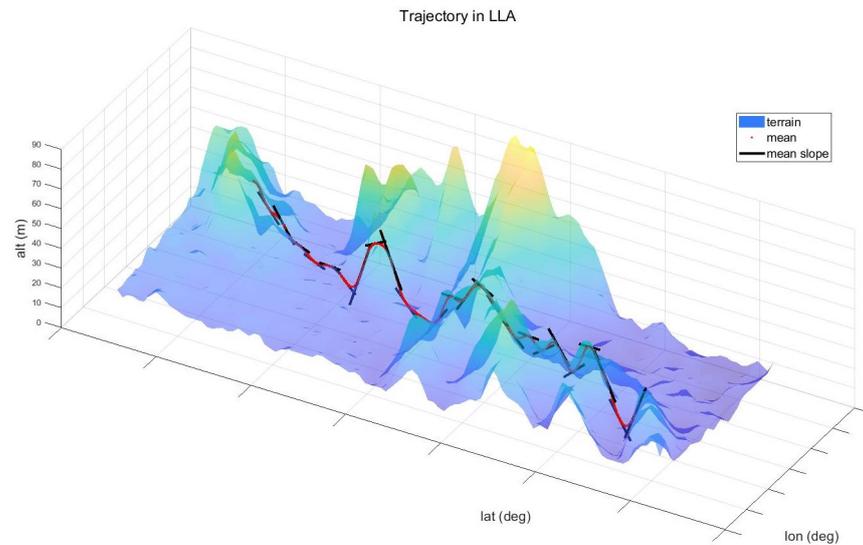


Figure 1. GP Prediction Example.

3.2. Velocity Transformation

Another major challenge when applying the terrain constraint to the filter is that the conversion of velocity between the local Cartesian coordinates and the LLA coordinates is not straightforward. More specifically, the terrain constraint (Equation (5)) requires the velocity in LLA coordinates.

This challenge can be met by multiplying the Jacobian, which is obtained by numerical differentiation. Additionally, because velocity in local Cartesian coordinates is relative while that in LLA coordinates is absolute, the velocity of the radar $V_{lla,ownership}$ should be added (or subtracted) after (or before) multiplying by the Jacobian.

$$\begin{aligned} \begin{bmatrix} \frac{\partial \lambda}{\partial t} \\ \frac{\partial \varphi}{\partial t} \\ \frac{\partial h}{\partial t} \end{bmatrix}_{rel} &= \begin{bmatrix} \frac{\partial \lambda}{\partial x} & \frac{\partial \lambda}{\partial y} & \frac{\partial \lambda}{\partial z} \\ \frac{\partial \varphi}{\partial x} & \frac{\partial \varphi}{\partial y} & \frac{\partial \varphi}{\partial z} \\ \frac{\partial h}{\partial x} & \frac{\partial h}{\partial y} & \frac{\partial h}{\partial z} \end{bmatrix} \begin{bmatrix} \frac{\partial x}{\partial t} \\ \frac{\partial y}{\partial t} \\ \frac{\partial z}{\partial t} \end{bmatrix} \\ &= J_{xyz2lla} V_{xyz} \\ V_{lla} &= J_{xyz2lla} V_{xyz} + V_{lla,ownership} \end{aligned} \quad (11)$$

Conversion from LLA to local Cartesian can be done in a converse way.

$$V_{xyz} = J_{lla2xyz} (V_{lla} - V_{lla,ownership}) \quad (12)$$

where $J_{lla2xyz} = J_{xyz2lla}^{-1}$.

3.3. Assumptions

Regarding the motion of the target, we assume the followings:

1. The vertical position (h) can be determined provided that the horizontal position (λ, φ) is fixed.
2. Then, the vertical velocity (v_h) can be also determined when the horizontal velocity (v_λ, v_φ) is fixed.

In Figure 2, assumptions 1 and 2 correspond to the red arrows that inbound to h and v_h , respectively. They comprise the 'elevation model'.

Then, by the principle of importance sampling, the corresponding weight is calculated as

$$w_k^i \propto \frac{P(x_{0:k}^i | y_{1:k})}{q(x_{0:k}^i | y_{1:k})}. \quad (17)$$

Because we have freedom to choose the proposal distribution, we consider a proposal distribution that has a form of

$$q(x_{0:k} | y_{1:k}) = q(x_k | x_{0:k-1}, y_{1:k}) q(x_{0:k-1} | y_{1:k-1}). \quad (18)$$

In other words, one can draw new support points $x_{0:k}^i \sim q(x_{0:k} | y_{1:k})$ by augmenting each of the previous support points $x_{0:k-1}^i \sim q(x_{0:k-1} | y_{1:k-1})$ with the new state $x_k^i \sim q(x_k | x_{0:k-1}, y_{1:k})$.

Starting from Equation (17),

$$w_k^i \propto w_{k-1}^i \frac{P(x_{0:k}^i | y_{1:k})}{P(x_{0:k-1}^i | y_{1:k-1})} \frac{q(x_{0:k-1}^i | y_{1:k-1})}{q(x_{0:k}^i | y_{1:k})}. \quad (19)$$

Together with Equation (18) and the recursive relation (Equation (13)), the weight update equation can be simplified.

$$w_k^i \propto w_{k-1}^i \frac{P(y_k | \lambda_k^i, \varphi_k^i, h_k^i) \cdot P(\lambda_k^i, \varphi_k^i, v_{\lambda,k}^i, v_{\varphi,k}^i | x_{k-1}^i) \cdot P(h_k^i | \lambda_k^i, \varphi_k^i) \cdot P(v_{h,k}^i | \lambda_k^i, \varphi_k^i, v_{\lambda,k}^i, v_{\varphi,k}^i)}{q(x_k^i | x_{0:k-1}^i, y_{1:k})} \quad (20)$$

A further assumption regarding the proposal distribution,

$$q(x_k^i | x_{0:k-1}^i, y_{1:k}) = P(\lambda_k^i, \varphi_k^i, v_{\lambda,k}^i, v_{\varphi,k}^i | x_{k-1}^i) \cdot P(h_k^i | \lambda_k^i, \varphi_k^i) \cdot P(v_{h,k}^i | \lambda_k^i, \varphi_k^i, v_{\lambda,k}^i, v_{\varphi,k}^i) \quad (21)$$

yields

$$w_k^i \propto w_{k-1}^i \cdot P(y_k | \lambda_k^i, \varphi_k^i, h_k^i) \quad (22)$$

This means that the weight of each particle is updated proportionally to its corresponding likelihood. Note that the above weight update equation implicitly includes the normalization given by Equation (15).

The proposed algorithm, STC-PF, is summarized in Algorithm 1. In a vanilla SIR PF, the next state is propagated through the dynamic model only. In contrast, in STC-PF, the next state is propagated through the dynamic model first and then propagated through the elevation model (line numbers 5 and 6 in Algorithm 1).

In Figure 3, a detailed implementation of the elevation model propagation is shown. It is worth mentioning that elevation model propagation can be accelerated by two techniques: parallelization and use of local data during the GP inference. Because the propagation process for each particle does not require information on other particles, it can be parallelized. Furthermore, during the GP inference, only the neighborhood data of DTED are utilized. The range of the neighborhood is defined by the spatial window size L .

Algorithm 1: Soft Terrain Constrained Particle Filter (STC-PF).

Result: Trajectory of Support Points

- 1 initialization of support points $\{x_0^i, w_0^i\}_{i=1}^{N_p}$;
- 2 **for** $k = 1 \dots T$ **do**
- 3 get new measurement y_k ;
- 4 **for** $i = 1 \dots N_p$ **do**
- 5 propagate through the dynamic model
 $\lambda_k^i, \varphi_k^i, v_{\lambda,k}^i, v_{\varphi,k}^i \sim P(\lambda_k, \varphi_k, v_{\lambda,k}, v_{\varphi,k} | x_{k-1}^i)$;
- 6 propagate through the elevation model

$$\left(\begin{array}{l} \bar{h}, \nabla \bar{h} | \text{DTED} \sim N(\bar{\mu}, \bar{\Sigma}) \\ h_k^i = \bar{h}(\lambda_k^i, \varphi_k^i) \\ v_{h,k}^i = \nabla \bar{h}(\lambda_k^i, \varphi_k^i) \cdot \left[v_{\lambda,k}^i \quad v_{\varphi,k}^i \right]^T \end{array} \right)$$
;
- 7 update the weight $\hat{w}_k^i \propto w_{k-1}^i P(y_k | \lambda_k^i, \varphi_k^i, h_k^i)$;
- 8 normalize $w_k^i = \hat{w}_k^i / (\sum_{j=1}^{N_p} \hat{w}_k^j)$;
- 9 **end**
- 10 (Optional) Resampling (e.x. multinomial resampling);
- 11 **end**

3.5. Remark on an Existing Work

As mentioned in Section 1, from a mathematical perspective, the proposed algorithm (STC-PF) is similar to scPF (soft-constrained Particle Filter) [35]. Similar to STC-PF, scPF is based on the SIR particle filter; however, the two differ in the sense that scPF utilizes generalized likelihood.

$$\hat{w}_k^i \propto w_{k-1}^i P(y_k | x_k^i) P(C_k | x_k^i) \quad (23)$$

where $P(C_k | x_k^i)$ is a pseudo-measurement that represents how much the given state x_k^i satisfies the constraint. If Equation (21) is replaced by

$$q(x_k^i | x_{0:k-1}^i, y_{1:k}^i) = P(\lambda_k^i, \varphi_k^i, v_{\lambda,k}^i, v_{\varphi,k}^i | x_{k-1}^i), \quad (24)$$

then the weight update rule is also changed.

$$w_k^i \propto w_{k-1}^i \cdot P(y_k | \lambda_k^i, \varphi_k^i, h_k^i) \cdot P(h_k^i | \lambda_k^i, \varphi_k^i) \cdot P(v_{h,k}^i | \lambda_k^i, \varphi_k^i, v_{\lambda,k}^i, v_{\varphi,k}^i) \quad (25)$$

Thus, the generalized likelihood function can be identified by equating the elevation model with the pseudo-measurement. As a result, scPF can be reduced to STC-PF as long as the assumption for target motion holds.

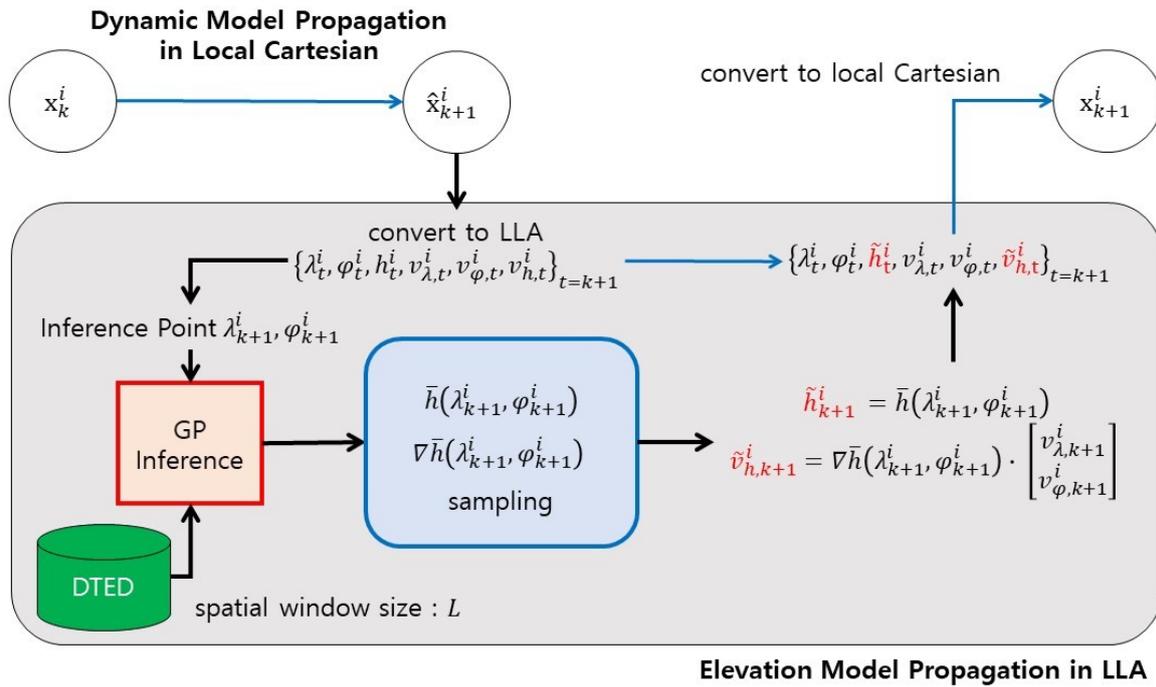


Figure 3. Implementation of Elevation Model Propagation.

4. Simulation

4.1. Scenario and Parameter Settings

To evaluate STC-PF, numerical experiments are performed with the following scenario: The radar is mounted on an aircraft that flies at a speed of 70 m/s at a height of 2500 m. The radar tracks a single target that moves along the surface at a speed of 25 m/s. (see Figure 4) The simulation runs for 100 s. Furthermore, to reflect the uncertainty in DTED, a noisy version of DTED is created. More specifically, iid zero-mean Gaussian noise with variance σ_{DTED} is sampled and added for each data entry in DTED. Because it is reasonable to bound the uncertainty of DTED, sampled noise is clipped to 50 m if its absolute value exceeds 50 m.

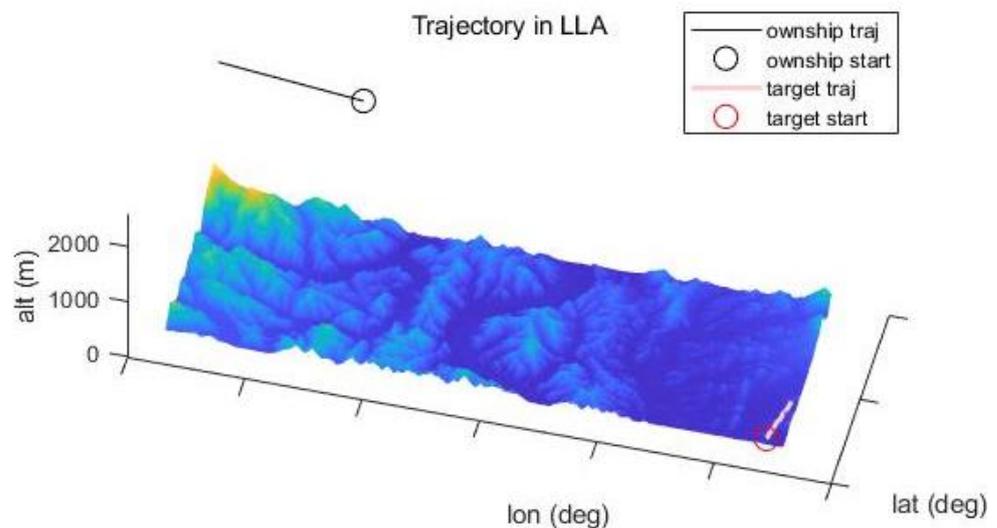


Figure 4. Trajectory in WGS84 LLA (0.05 degree interval).

Values of parameters used in the simulation are listed in Table 1. Detailed explanation about the choice of GP hyper-parameters is in the Appendix B. The simulations are performed with two settings that differ in the value of σ_{DTED} . The reasonable value for σ_{DTED} is 3.77 m, which is inferred from [37]. However, another setting whose σ_{DTED} is 1.89 m is also used to observe the sensitivity of the key parameter.

Table 1. Parameter Setting.

Name	Value
$\sigma_{\text{DTED}} (m)$	3.77, 1.89
α	100
$\Gamma (deg^{-2})$	$\text{diag} \left[\frac{1}{(2.78e-4)^2} \quad \frac{1}{(2.78e-4)^2} \right]$
$L (arcsec)$	13 ($\approx 390m$)
$\Delta t (s)$	1.0
Initial Cov.	$\begin{bmatrix} 1e2(m^2) \cdot \mathbf{I}_{3 \times 3} & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & 10(m^2/s^2) \cdot \mathbf{I}_{3 \times 3} \end{bmatrix}$
N_p	1e4
Q	$\begin{bmatrix} 20(m) \cdot \mathbf{I}_{3 \times 3} & \mathbf{0}_{3 \times 3} & 0 & 0 \\ \mathbf{0}_{3 \times 3} & 2(m/s) & 0 & 0 \\ 0 & 0 & 2(m/s) & 0 \\ 0 & 0 & 0 & 5(m/s) \end{bmatrix}^2$
R	$(\text{diag} [10(m) \quad 0.1(deg) \quad 0.1(deg)])^2$

4.2. Baseline Methods

To compare STC-PF with other filters that can incorporate nonlinear constraints, the Smoothly Constrained Kalman Filter (SCKF) is implemented as well [30]. Note that ‘Smoothly Constrained’ in the name of SCKF does not mean soft constraint. Because SCKF can incorporate only deterministic constraints, it requires approximations of ground-truth terrain elevation that require h and ∇h to be fixed to specific values. One approach used for the comparison is to ignore the noise inherent in DTED and use bilinear interpolation to retrieve the terrain elevation at arbitrary positions.

$$\begin{aligned}
 E(x_k) &= h_k - \bar{h}(\lambda_k, \varphi_k) = 0 \\
 E_v(x_k) &= v_{h,k} - \nabla \bar{h}(\lambda_k, \varphi_k) \cdot \begin{bmatrix} v_{\lambda,k} & v_{\varphi,k} \end{bmatrix}^T = 0 \\
 \bar{h}(\lambda, \varphi) &\approx BL(\lambda, \varphi | \text{DTED})
 \end{aligned} \tag{26}$$

where $BL(\lambda, \varphi | \text{DTED})$ is bilinear interpolation at (λ, φ) given DTED. For the computation of the gradient $\nabla \bar{h}$, central numerical differentiation is used instead of analytic differentiation to avoid non-differentiable cases.

$$\left. \frac{\partial \bar{h}}{\partial \lambda} \right|_{x_k} \approx \frac{\bar{h}(\lambda_k + \Delta, \varphi_k) - \bar{h}(\lambda_k - \Delta, \varphi_k)}{2\Delta} \tag{27}$$

where Δ is a small constant.

Another method is to use GP mean regression rather than bilinear interpolation. That is,

$$\begin{bmatrix} \bar{h} & \nabla \bar{h} \end{bmatrix}^T \approx \bar{\mu} \tag{28}$$

where $\bar{\mu}$ is the GP joint mean of \bar{h} and $\nabla \bar{h}$ in Equation (9). This enables us to reconstruct the most probable ground-truth terrain elevation considering the noise of DTED; however, this method still cannot consider the uncertainty of the inferred \bar{h} and $\nabla \bar{h}$ values, in contrast to STC-PF.

SCKF requires the Jacobian of the constraint functions:

$$G(x_k) = \left. \frac{\partial E}{\partial x} \right|_{x_k} = \left[\frac{\partial E}{\partial x} \quad \frac{\partial E}{\partial y} \quad \frac{\partial E}{\partial z} \quad \frac{\partial E}{\partial v_x} \quad \frac{\partial E}{\partial v_y} \quad \frac{\partial E}{\partial v_z} \right] \Big|_{x_k} \quad (29)$$

$$G_v(x_k) = \left. \frac{\partial E_v}{\partial x} \right|_{x_k} = \left[\frac{\partial E_v}{\partial x} \quad \frac{\partial E_v}{\partial y} \quad \frac{\partial E_v}{\partial z} \quad \frac{\partial E_v}{\partial v_x} \quad \frac{\partial E_v}{\partial v_y} \quad \frac{\partial E_v}{\partial v_z} \right] \Big|_{x_k}$$

However, it is impossible to differentiate $E(x_k)$ and $E_v(x_k)$ analytically because they involve coordinate transformation between local Cartesian and WGS84 LLA. Alternatively, the derivative can be obtained using the central numerical difference regardless of the regression method.

$$\left. \frac{\partial E}{\partial x} \right|_{x_k} \approx \frac{E(x_k + \Delta \cdot e_x) - E(x_k - \Delta \cdot e_x)}{2\Delta}, \quad (30)$$

where e_x is a canonical unit vector whose first component is nonzero. $\partial E / \partial y_k$, $\partial E / \partial z_k$, and $\partial E_v / \partial v$ can be obtained in a similar way. Because E is not a function of $v_{,k}$, corresponding derivatives automatically become zero.

4.3. Results

To evaluate STC-PF, SCKF using bilinear regression, and SCKF using GP mean regression, 100 Monte-Carlo simulations were carried out for each σ_{DTED} value. Tracking performance is assessed based on timewise RMS (Root Mean Squared) error. For example, timewise RMS for local Cartesian x position error at time k is

$$RMS_{x,k} = \sqrt{\frac{1}{N_{MC}} \sum_{n=1}^{N_{MC}} (x_k^n - \bar{x}_k)^2} \quad (31)$$

where N_{MC} is the number of repetitions (i.e., 100), x_k^n the filter mean value for x position at time k in the n^{th} trial, and \bar{x}_k the ground-truth x position at time k . The time average ($10 \leq k \leq 90$) for timewise RMS is also computed for evaluation.

Figure 5 shows the timewise RMS for local Cartesian position error and velocity error. In the figures, SCKF using bilinear regression shows the worst tracking performance. In terms of time average of RMS position error, as shown in Table 2, the superiority of STC-PF over SCKF using GP mean regression is clear, although it cannot be identified in Figure 5. In terms of RMS velocity error, STC-PF distinctly outperforms the other two methods. This trend also holds for the different parameter setting, namely $\sigma_{\text{DTED}} = 1.89$ m, as shown in Figure 6 and Table 3.

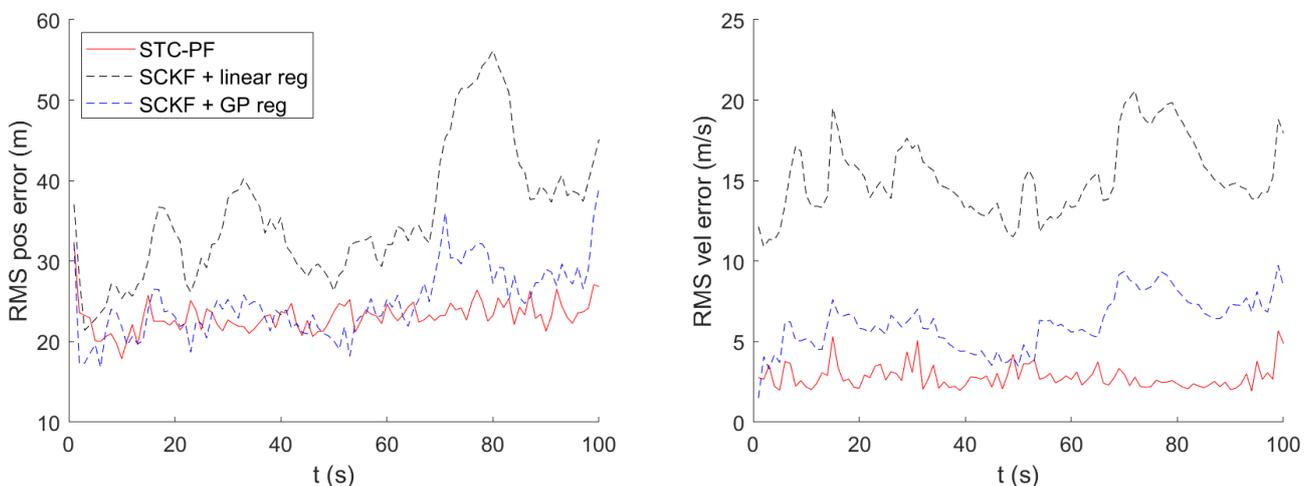
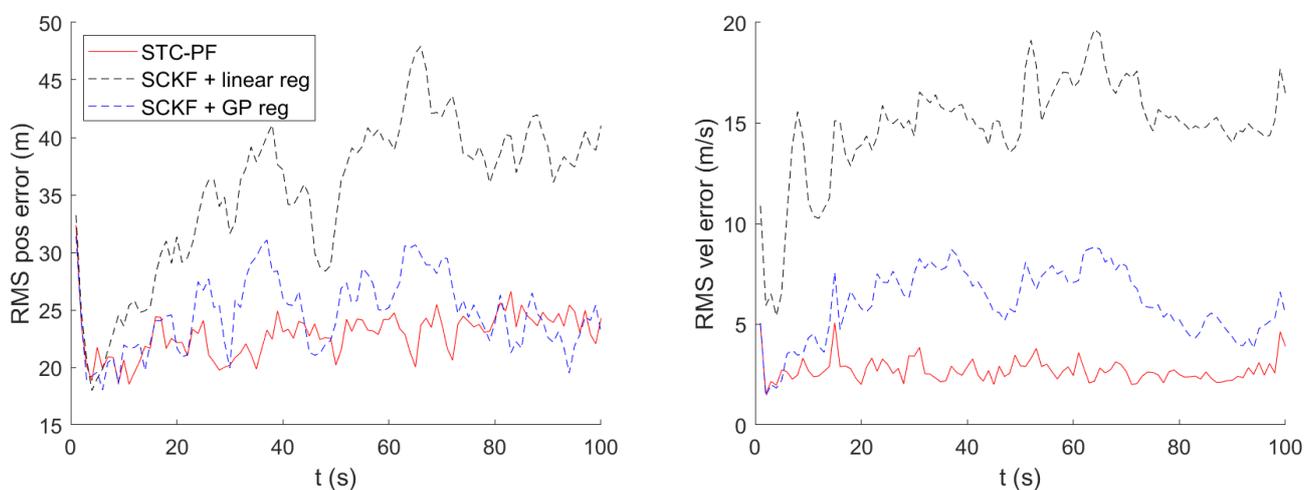


Figure 5. Timewise RMS for Local Cartesian Position and Velocity Error ($\sigma_{\text{DTED}} = 3.77$ m).

Table 2. Time Average of Timewise RMS ($\sigma_{\text{DTED}} = 3.77$ m).

	STC-PF	SCKF + Bilinear	SCKF + GP
x (m)	9.61	10.9	9.52
y	20.7	34.1	22.4
z	2.77	3.84	3.05
Position	23.0	36.1	24.6
v_x (m/s)	0.972	4.10	1.55
v_y	1.74	14.0	5.45
v_z	1.78	4.16	2.15
Velocity	2.75	15.4	6.15

**Figure 6.** Timewise RMS for Local Cartesian Position and Velocity Error ($\sigma_{\text{DTED}} = 1.89$ m).**Table 3.** Time Average of Timewise RMS ($\sigma_{\text{DTED}} = 1.89$ m).

	STC-PF	SCKF + Bilinear	SCKF + GP
x (m)	9.48	11.0	9.63
y	20.5	34.4	23.1
z	2.56	3.96	3.12
Position	22.8	36.4	25.3
v_x (m/s)	0.966	3.38	1.11
v_y	1.71	14.2	5.97
v_z	1.74	3.95	2.22
Velocity	2.69	15.4	6.56

On the other hand, the speed of the algorithms is assessed based on the average processing time for a single timestep. STC-PF and SCKF both were implemented in MATLAB and run on an Intel Core i7-8565U with 16.0GB RAM. Table 4 shows that the baseline method runs nearly in real time, while STC-PF does not. Nevertheless, with parallel computing, STC-PF gets much faster and shows the possibility of real time applications.

Table 4. Average Processing Time for a Single Timestep.

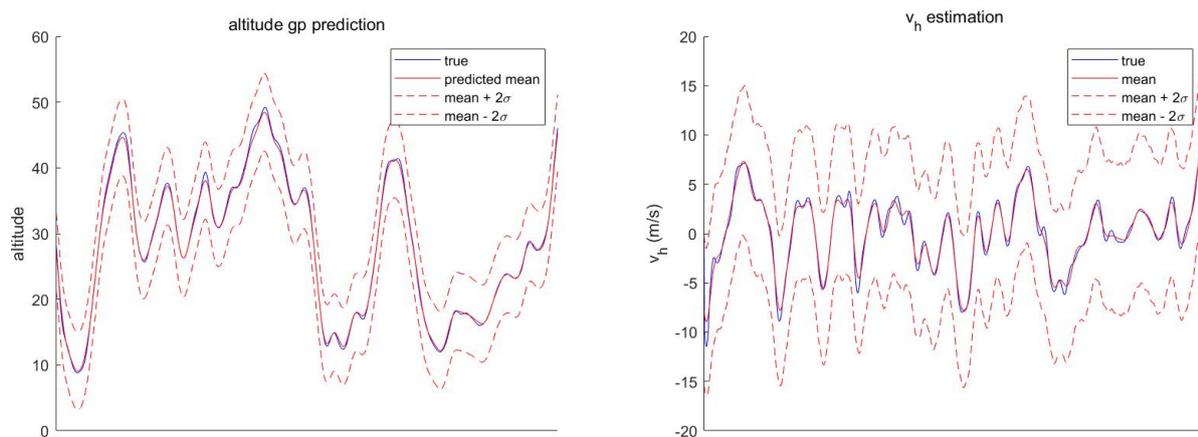
	STC-PF + Single	SCKF + Bilinear	SCKF + GP	STC-PF + Parallel
t (s)	43.5	1.18	1.33	14.2

4.4. Discussion

According to the simulation results, the tracking performance of SCKF depends on the regression method. More specifically, SCKF using GP mean regression has smaller RMS errors than SCKF using bilinear regression. This result could be due to the regression method used for target trajectory generation: GP regression, which is suitable for fitting of a smooth curve, might result in a trajectory closer to the ground-truth trajectory generated by bicubic spline interpolation.

Meanwhile, the superiority of STC-PF over SCKF in terms of tracking performance could stem from two factors. The first is that a particle filter is more expressive than a simple Gaussian filter. Because SCKF assumes that the posterior distribution is a simple Gaussian, SCKF adjusts its state to meet the constraint at one point. However, unlike SCKF, particle filters do not assume the form of a posterior distribution. Therefore, STC-PF can estimate the distribution of the velocity hypothesis for each position hypothesis, so that the combined hypothesis independently meets the terrain constraint. As a result, the filter mean value, a weighted sum of each hypothesis, is less biased.

The second reason is that the state estimation with soft constraint is less sensitive to the uncertainty of the constraint. Conversely, state estimation with hard constraint is very sensitive to the uncertainty of the constraint. Figure 7 shows that the terrain constraint (Equation (5)) holds almost perfectly assuming perfect knowledge of the position and DTED. However, the hard constraint, the GP mean regression, deviates from the ground-truth value if small amounts of position uncertainty (approximately 30 m in the longitudinal direction) and DTED noise ($\sigma_{\text{DTED}} = 3.77$ m) are introduced (see Figure 8). In other words, uncertainty of the horizontal position and DTED can result in a catastrophic state error if a hard constraint is applied. On the other hand, STC-PF can absorb the error to some degree as most of the ground-truth value resides inside the two-sigma bound.

**Figure 7.** GP Prediction Result Without Position Uncertainty and DTED Noise.

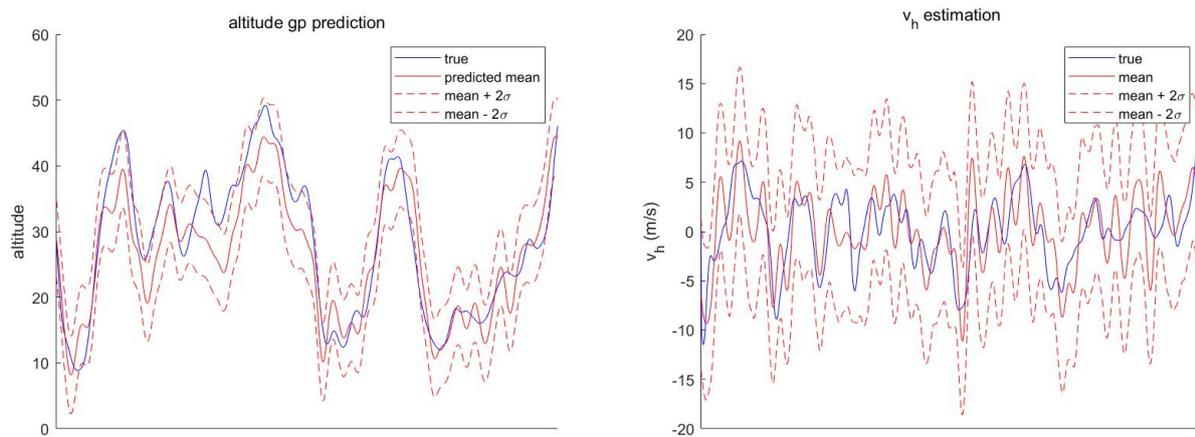


Figure 8. GP Prediction Result with both Position Uncertainty and DTED Noise.

Regarding the speed of the algorithm, it is natural that STC-PF consumes more computational resources than SCKF. This is because STC-PF requires expensive computation for every particle; GP inference whose time complexity is $\mathcal{O}(n^3)$, where n is the number of observation points involved in the GP inference. Fortunately, time consumption is drastically reduced by the virtue of parallel computing, and it can be reduced further by two measures. One is reducing the number of GP observation points (namely, the spatial window L in Figure 3) as far as the performance degradation is negligible, and the other is increasing the number of cores dedicated to the filtering.

5. Conclusions

To sum up, we have proposed a particle filter to improve the performance of ground target tracking. To estimate the velocity more accurately, not only a position constraint but also a velocity constraint has been introduced in the terrain constraint. Although DTED provides terrain elevation of the entire globe, it provides inaccurate values at discrete positions. Thus, the ground-truth terrain elevation included in the terrain constraint has been modeled with a Gaussian process, and DTED has been regarded as noisy observations of it. As a result, terrain constraint has become a soft constraint that can reflect the uncertainty of DTED. Finally, we have proposed a particle filter, STC-PF, given the assumption of the motion of the target. STC-PF is based on SIR PF, but the major difference is that STC-PF uses the elevation model. Due to the elevation model, knowledge of the horizontal position and velocity of a target enables us to infer the vertical position and velocity more precisely. In the numerical simulation, STC-PF has been compared with SCKF which can incorporate hard constraints only. Furthermore, to reflect the uncertainty in DTED, filters have made use of DTED contaminated by noise, whereas the ground-truth trajectory of the target is generated by the original DTED. The simulation results showed that STC-PF outperforms SCKF in terms of RMS error, for two possible reasons. The first is that particle filters are more expressive than simple Gaussian filters. The second is that the state estimation with soft constraint is less sensitive to uncertainty of the constraint than that with hard constraint.

Author Contributions: Conceptualization, D.-U.K., J.P. and H.-L.C. and W.-C.L.; Data curation, D.-U.K.; Formal analysis, D.-U.K. and W.-C.L.; Funding acquisition, H.-L.C. and W.-C.L.; Investigation, D.-U.K., W.-C.L., J.P. and J.A.; Methodology, D.-U.K.; Project administration, W.L. and J.A.; Resources, H.-L.C.; Software, D.-U.K. and W.-C.L.; Supervision, H.-L.C.; Validation, D.-U.K., W.-C.L., W.L., J.P. and J.A.; Visualization, D.-U.K.; Writing—original draft, D.-U.K., W.-C.L.; Writing—review & editing; D.-U.K., W.-C.L., H.-L.C. and J.P. All authors have read and agreed to the published version of the manuscript.

Funding: This work was conducted with the support of Agency for Defense Development and LIG Nex1. (contraction name: Fighter multi-mode Fire Control Radar).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: SRTM data was obtained from USGS and are available at <https://earthexplorer.usgs.gov/> with the permission of USGS. Detection data is not publicly available because this study was carried out as a defense-related project.

Conflicts of Interest: The authors have generic conflicts with individuals in the same affiliation. The fourth and the fifth authors have general conflicts with individuals in the defense contractors in Korea.

Abbreviations

The following abbreviations are used in this manuscript:

DTED	Digital Terrain Elevation Data
PF	Particle Filter
SIR-PF	Sequential Importance Resampling Particle Filter
GP	Gaussian Process
LLA	Latitude, Longitude, and Altitude

Appendix A. Derivative of Gaussian Process

A Gaussian process is a collection of random variables, any finite number of which have a joint Gaussian distribution. A Gaussian process is completely specified by its mean function and covariance function. The mean function and covariance function are defined as

$$\begin{aligned} m(x) &= \mathbb{E}[f(x)], \\ k(x, x') &= \mathbb{E}[(f(x) - m(x))(f(x') - m(x')))], \end{aligned} \quad (\text{A1})$$

respectively [40]. Following the most common choice, zero mean function and squared exponential kernel are used throughout this paper. Namely,

$$\begin{aligned} m(x) &= 0, \\ k(x^m, x^n) &= \alpha \exp\left(-\frac{1}{2}\|x^m - x^n\|_{\Gamma}^2\right), \end{aligned} \quad (\text{A2})$$

where $\|x\|_{\Gamma} = \sqrt{x^T \Gamma x}$. In most practical applications, we do not have direct access to function values themselves, only noisy versions thereof:

$$y = f(x) + \epsilon, \quad (\text{A3})$$

where $\epsilon \sim N(0, \sigma)$. Let X and Y be the concatenation of all observation points and corresponding measurements, respectively. Given the observation set (X, Y) , the predictive distribution of the function value f^* at arbitrary test points X^* can be derived. Starting from the joint distribution of the observation set (X, Y) and the test set (X^*, f^*) ,

$$\begin{bmatrix} Y \\ f^* \end{bmatrix} \sim N\left(0, \begin{bmatrix} k(X, X) + \sigma^2 I & k(X, X^*) \\ k(X^*, X) & k(X^*, X^*) \end{bmatrix}\right). \quad (\text{A4})$$

Using basic arithmetic operations, Equation (A4) can be transformed into the following predictive distribution [40]:

$$\begin{aligned} f^* | X^*, X, Y &\sim N(\mu, \Sigma), \\ \text{where} & \\ \mu &= k(X, X^*)^T \cdot (k(X, X) + \sigma^2 I)^{-1} \cdot Y, \\ \Sigma &= k(X^*, X^*) - k(X, X^*)^T \cdot (k(X, X) + \sigma^2 I)^{-1} \cdot k(X, X^*). \end{aligned} \quad (\text{A5})$$

Furthermore, consider the derivative of the given GP. As differentiation is a linear operator, the derivative of a Gaussian process remains a Gaussian process as long as the

kernel function is differentiable [41]. To find the joint probability of the observation set (X, Y) and the derivative observation set (X_d, Y_d) , covariance between the function value and the derivative value as well as covariance among the derivative values should be described. First, let the derivative of the underlying function be

$$f_d = \left[\frac{\partial f(x)}{\partial x_1} \quad \dots \quad \frac{\partial f(x)}{\partial x_D} \right]^T, \quad (\text{A6})$$

where D is the dimension of x . Then, the explicit expressions of the new covariance functions are

$$\begin{aligned} \text{cov}((f_d^m)_i, f^n) &= \frac{\partial}{\partial x_i} \text{cov}(f^m, f^n) \\ &= -\alpha \gamma_i (x_i^m - x_i^n) \exp\left(-\frac{1}{2} \|x^m - x^n\|_1^2\right), \\ \text{cov}((f_d^m)_i, (f_d^n)_j) &= \frac{\partial^2}{\partial x_i \partial x_j} \text{cov}(f^m, f^n) \\ &= \alpha \gamma_i \left(\delta_{i,j} - \gamma_j (x_i^m - x_i^n) (x_j^m - x_j^n) \right) \exp\left(-\frac{1}{2} \|x^m - x^n\|_1^2\right). \end{aligned} \quad (\text{A7})$$

Using basic arithmetic operations, the above expression is reduced to vector form:

$$\begin{aligned} \text{cov}(f_d^m, f^n) &= -k(x_d^m, x_d^n) \Gamma(x_d^m - x_d^n) \\ &= k_{dx}(x_d^m, x_d^n) \in \mathbf{R}^{D \times 1}, \\ \text{cov}(f_d^m, f_d^n) &= k(x_d^m, x_d^n) \left(\Gamma - \Gamma(x_d^m - x_d^n) (x_d^m - x_d^n)^T \Gamma \right) \\ &= k_{dd}(x_d^m, x_d^n) \in \mathbf{R}^{D \times D}. \end{aligned} \quad (\text{A8})$$

Suppose there are N observation points and M derivative observation points,

$$\begin{aligned} X &= [x^1 \quad x^2 \quad \dots \quad x^N]^T, \\ Y &= [y^1 \quad y^2 \quad \dots \quad y^N]^T, \\ X_d &= [x_d^1 \quad x_d^2 \quad \dots \quad x_d^M]^T, \\ Y_d &= [(y_d^1)^T \quad (y_d^2)^T \quad \dots \quad (y_d^M)^T]^T. \end{aligned} \quad (\text{A9})$$

Finally, the joint distribution of the observation set (X, Y) and the derivative observation set (X_d, Y_d) can be described, assuming noise-free observation for the simplicity of the notation,

$$\begin{bmatrix} Y_d \\ Y \end{bmatrix} \sim N(0, \tilde{K}) \quad (\text{A10})$$

where

$$\tilde{K} = \begin{bmatrix} \tilde{K}_{dd} & \tilde{K}_{dx} \\ \tilde{K}_{xd} & \tilde{K}_{xx} \end{bmatrix}, \quad (\text{A11})$$

$$\tilde{K}_{dd} = k_{dd}(X_d, X_d) = \begin{bmatrix} k_{dd}(x_d^1, x_d^1) & \dots & k_{dd}(x_d^1, x_d^M) \\ \vdots & \ddots & \vdots \\ k_{dd}(x_d^M, x_d^1) & \dots & k_{dd}(x_d^M, x_d^M) \end{bmatrix} \in \mathbf{R}^{MD \times MD}, \quad (\text{A12})$$

$$\tilde{K}_{dx} = k_{dx}(X_d, X) = \begin{bmatrix} k_{dx}(x_d^1, x^1) & \dots & k_{dx}(x_d^1, x^N) \\ \vdots & \ddots & \vdots \\ k_{dx}(x_d^M, x^1) & \dots & k_{dx}(x_d^M, x^N) \end{bmatrix} \in \mathbf{R}^{MD \times N}, \quad (\text{A13})$$

$$\tilde{K}_{xx} = K_{xx}(X, X) = \begin{bmatrix} k_{xx}(x^1, x^1) & \dots & k_{xx}(x^1, x^N) \\ \vdots & \ddots & \vdots \\ k_{xx}(x^N, x^1) & \dots & k_{xx}(x^N, x^N) \end{bmatrix} \in \mathbf{R}^{N \times N}, \quad (\text{A14})$$

and $\tilde{K}_{xd} = \tilde{K}_{dx}^T$. Similar to the argument in Equations (A4) and (A5), the joint predictive distribution of the function value f^* and derivative f_d^* at the test point X^* and derivative test point X_d^* , given the observation set (X, Y) and derivative observation set (X_d, Y_d) is

$$\begin{bmatrix} f^* \\ f_d^* \end{bmatrix} | X^*, X_d^*, X, Y, X_d, Y_d \sim N(\tilde{\mu}, \tilde{\Sigma}), \quad (\text{A15})$$

where

$$\tilde{\mu} = \hat{K}_0 \tilde{K}^{-1} \begin{bmatrix} Y_d \\ Y \end{bmatrix}, \quad (\text{A16})$$

$$\tilde{\Sigma} = \hat{K}_1 - \hat{K}_0 \tilde{K}^{-1} \hat{K}_0^T,$$

and

$$\begin{aligned} \hat{K}_0 &= \begin{bmatrix} k_{xd}(X^*, X_d) & k_{xx}(X^*, X) \\ k_{dd}(X_d^*, X_d) & k_{dx}(X_d^*, X) \end{bmatrix}, \\ \hat{K}_1 &= \begin{bmatrix} k_{xx}(X^*, X^*) & k_{xd}(X^*, X_d^*) \\ k_{dx}(X_d^*, X^*) & k_{dd}(X_d^*, X_d^*) \end{bmatrix}. \end{aligned} \quad (\text{A17})$$

Now, given the sequence of test points, function value and gradient value of each test point can be inferred together. Because Equations (A15)–(A17) give mean and covariance, the function value and the slope can be sampled easily as well. If only observations of the function values are available, then X_d and Y_d are empty vectors.

Appendix B. Choice of Hyper Parameters of Gaussian Process

According to the DTED specifications (Department of Defense(2000), Performance Specification, DTED, MIL-PRF-89020B), linear vertical absolute height error is less than 16 m and circular absolute geolocation error is less than 20 m for 90% of the data. Even though the global verification of the data is a challenging task, extensive ground truth data (e.g., Kinematic GPS data, GCPs) has been collected for the verification [37]. Consequently, it is found that linear vertical absolute height error is 6.2 m and circular absolute geolocation error is 8.8 m for 90% of the data.

Interestingly, the height error distribution is approximately a Gaussian, within an order of magnitude. Therefore, we can estimate the standard deviation of the height error σ_{DTED} to be 3.77 (m), because a 90% confidence interval is $\pm 1.65\sigma$.

Other hyper parameters are chosen empirically. Γ is inverse squared of 1 arc second ($\approx 2.78 \times 10^{-4}$ deg) which makes the exponent part of the kernel function non-dimensional. Finally, α is properly selected so that the predictive mean sufficiently converges to the ground truth terrain elevation.

Appendix C. Multinomial Resampling

Multinomial resampling was proposed with the first particle filter [42,43]. The multinomial resampling algorithm used in the implementation is illustrated in Algorithm A1.

Algorithm A1: Multinomial Resampling Algorithm.

```

Result: new support points  $\{xx^i, ww^i\}_{i=1}^{N_p}$ 
1 for  $i = 1 \dots N_p$  do
2    $u^i = \text{rand}()$  // sample from uniform  $[0, 1)$ ;
3 end
4  $u_o = \text{SORT}(\{u^i\}_{i=1}^{N_p})$  // ascending sort;
5  $sumW = 0$ ;
6  $i = 0$ ;
7  $j = 1$ ;
8 while  $j \leq N_p$  do
9    $i = i + 1$ ;
10   $sumW = sumW + w^i$ ;
11  while  $(j \leq N_p) \ \&\& \ (sumW > u_o^j)$  do
12     $xx^j = x^i$ ;
13     $ww^j = 1/N_p$ ;
14     $j = j + 1$ ;
15  end
16 end

```

References

- Council, N.; Sciences, D.; Board, N.; Groups, C. *C4ISR for Future Naval Strike Groups*; National Academies Press: Washington, DC, USA, 2006. [\[CrossRef\]](#)
- Chee-Yee Chong.; Garren, D.; Grayson, T.P. Ground target tracking—a historical perspective. In Proceedings of the 2000 IEEE Aerospace Conference, Proceedings (Cat. No.00TH8484), Big Sky, MT, USA, 25–25 March 2000; Volume 3, pp. 433–448. [\[CrossRef\]](#)
- Snidaro, L.; García, J.; Llinas, J. Context-based Information Fusion: A survey and discussion. *Inf. Fusion* **2015**, *25*, 16–31. [\[CrossRef\]](#)
- Fosbury, A.M.; Crassidis, J.L.; Singh, T.; Springen, C. Ground target tracking using terrain information. In Proceedings of the 2007 10th International Conference on Information Fusion, Quebec, QC, Canada, 9–12 July 2007; pp. 1–8. [\[CrossRef\]](#)
- Kastella, K.; Kreucher, C. Multiple model nonlinear filtering for low signal ground target applications. *IEEE Trans. Aerosp. Electron. Syst.* **2005**, *41*, 549–564. [\[CrossRef\]](#)
- Streller, D. Road map assisted ground target tracking. In Proceedings of the 2008 11th International Conference on Information Fusion, Cologne, Germany, 30 June–3 July 2008; pp. 1–7.
- Pannetier, B.; Benameur, K.; Nimier, V.; Rombaut, M. Ground moving target tracking with road constraint. In Proceedings of the Signal Processing, Sensor Fusion, and Target Recognition XIII, International Society for Optics and Photonics, Orlando, FL, USA, 13–15 April 2004; Volume 5429, pp. 138–149.
- Pannetier, B.; Benameur, K.; Nimier, V.; Rombaut, M. VS-IMM using road map information for a ground target tracking. In Proceedings of the 2005 7th International Conference on Information Fusion, Philadelphia, PA, USA, 25–28 July 2005; Volume 1, p. 8. [\[CrossRef\]](#)
- Mallick, M. Maximum likelihood geolocation using a ground moving target indicator (GMTI) report. In Proceedings of the IEEE Aerospace Conference, Big Sky, MT, USA, 9–16 March 2002; Volume 4, pp. 4. [\[CrossRef\]](#)
- Kim, D.U.; Lee, W.C.; Choi, H.L.; Seo, J.J.; Lee, W.J.; An, J.H.; Park, J.T. Softly Constrained Particle Filter for Digital Terrain Elevation Data Aided Ground Moving Target Tracking. In Proceedings of the 2021 AIAA Scitech Forum, Nashville, TN, USA, 11–15. 19–21 January 2021; p. 1396. [\[CrossRef\]](#)
- Simon, D. Kalman filtering with state constraints: A survey of linear and nonlinear algorithms. *IET Control. Theory Appl.* **2010**, *4*, 1303–1318. [\[CrossRef\]](#)
- Teixeira, B.O.; Chandrasekar, J.; Tôrres, L.A.; Aguirre, L.A.; Bernstein, D.S. State estimation for linear and non-linear equality-constrained systems. *Int. J. Control* **2009**, *82*, 918–936. [\[CrossRef\]](#)
- Xu, L.; Liang, Y.; Duan, Z.; Zhou, G. Route-based dynamics modeling and tracking with application to air traffic surveillance. *IEEE Trans. Intell. Transp. Syst.* **2019**, *21*, 209–221. [\[CrossRef\]](#)
- Li, T. Single-road-constrained positioning based on deterministic trajectory geometry. *IEEE Commun. Lett.* **2018**, *23*, 80–83. [\[CrossRef\]](#)
- Zhou, J.; Li, T.; Wang, X.; Zheng, L. Target Tracking with Equality/Inequality Constraints Based on Trajectory Function of Time. *IEEE Signal Process. Lett.* **2021**, 1330–1334. [\[CrossRef\]](#)
- Wen, W.; Durrant-Whyte, H. Model-based multi-sensor data fusion. In Proceedings 1992 IEEE International Conference on Robotics and Automation, Nice, France, 12–14 May 1992; Volume 2, pp. 1720–1726. [\[CrossRef\]](#)

17. Porrill, J. Optimal Combination and Constraints for Geometrical Sensor Data. *Int. J. Robot. Res.* **1988**, *7*, 66–77. [[CrossRef](#)]
18. Alouani, A.; Blair, W. Use of a kinematic constraint in tracking constant speed, maneuvering targets. In Proceedings of the [1991] 30th IEEE Conference on Decision and Control, Brighton, UK, 11–13 December 1991; Volume 2, pp. 2055–2058. [[CrossRef](#)]
19. Wang, L.S.; Chiang, Y.T.; Chang, F.R. Filtering method for nonlinear systems with constraints. *IEE Proc.-Control. Theory Appl.* **2002**, *149*, 525–531. [[CrossRef](#)]
20. Simon, D.; Chia, T.L. Kalman filtering with state equality constraints. *IEEE Trans. Aerosp. Electron. Syst.* **2002**, *38*, 128–136. [[CrossRef](#)]
21. Ko, S.; Bitmead, R.R. State estimation for linear systems with state equality constraints. *Automatica* **2007**, *43*, 1363–1368. [[CrossRef](#)]
22. Simon, D. *Optimal State Estimation: Kalman, H Infinity, and Nonlinear Approaches*; John Wiley & Sons: Hoboken, NJ, USA, 2006. [[CrossRef](#)]
23. Teixeira, B.O.S.; Torres, L.A.B.; Aguirre, L.A.; Bernstein, D.S. Unscented filtering for interval-constrained nonlinear systems. In Proceedings of the 2008 47th IEEE Conference on Decision and Control, Cancun, Mexico, 9–11 December 2008; pp. 5116–5121. [[CrossRef](#)]
24. Julier, S.J.; LaViola, J.J. On Kalman Filtering With Nonlinear Equality Constraints. *IEEE Trans. Signal Process.* **2007**, *55*, 2774–2784. [[CrossRef](#)]
25. Agate, C.S.; Sullivan, K.J. Road-constrained target tracking and identification using a particle filter. In Proceedings of the Signal and Data Processing of Small Targets 2003, International Society for Optics and Photonics, San Diego, CA, USA, 3–8 August 2003; Volume 5204, pp. 532–543.
26. Kyriakides, I.; Morrell, D.; Papandreou-Suppappola, A. A Particle Filtering Approach To Constrained Motion Estimation In Tracking Multiple Targets. In Proceedings of the Conference Record of the Thirty-Ninth Asilomar Conference on Signals, Systems and Computers, Pacific Grove, CA, USA, 30 October–2 November 2005; pp. 94–98. [[CrossRef](#)]
27. Boers, Y.; Driessen, H. Particle filter track-before-detect application using inequality constraints. *IEEE Trans. Aerosp. Electron. Syst.* **2005**, *41*, 1483–1489. [[CrossRef](#)]
28. Wang, X.; Li, T.; Sun, S.; Corchado, J.M. A survey of recent advances in particle filters and remaining challenges for multitarget tracking. *Sensors* **2017**, *17*, 2707. [[CrossRef](#)]
29. Kyriakides, I.; Morrell, D.; Papandreou-Suppappola, A. Multiple target tracking with constrained motion using particle filtering methods. In Proceedings of the 1st IEEE International Workshop on Computational Advances in Multi-Sensor Adaptive Processing, Puerto Vallarta, Mexico, 13–15 Decemebtr 2005; pp. 85–88. [[CrossRef](#)]
30. Geeter, J.D.; Brussel, H.V.; Schutter, J.D.; Decreton, M. A smoothly constrained Kalman filter. *IEEE Trans. Pattern Anal. Mach. Intell.* **1997**, *19*, 1171–1177. [[CrossRef](#)]
31. Mahata, K.; Söderström, T. Improved estimation performance using known linear constraints. *Automatica* **2004**, *40*, 1307–1318. [[CrossRef](#)]
32. Tahk, M.; Speyer, J. Target tracking problems subject to kinematic constraints. *IEEE Trans. Autom. Control* **1990**, *35*, 324–326. [[CrossRef](#)]
33. Simon, D.; Simon, D.L. Kalman filtering with inequality constraints for turbofan engine health estimation. *IEE Proc.-Control. Theory Appl.* **2006**, *153*, 371–378. [[CrossRef](#)]
34. Massicotte, D.; Morawski, R.; Barwicz, A. Incorporation of a positivity constraint into a Kalman-filter-based algorithm for correction of spectrometric data. *IEEE Trans. Instrum. Meas.* **1995**, *44*, 2–7. [[CrossRef](#)]
35. Liu, C.; Li, B.; Chen, W.H. Particle Filtering With Soft State Constraints for Target Tracking. *IEEE Trans. Aerosp. Electron. Syst.* **2019**, *55*, 3492–3504. [[CrossRef](#)]
36. Farr, T.G.; Rosen, P.A.; Caro, E.; Crippen, R.; Duren, R.; Hensley, S.; Kobrick, M.; Paller, M.; Rodriguez, E.; Roth, L.; et al. The Shuttle Radar Topography Mission. *Rev. Geophys.* **2007**, *45*. [[CrossRef](#)]
37. Rodríguez, E.; Morris, C.S.; Belz, J.E. A Global Assessment of the SRTM Performance. *Photogramm. Eng. Remote Sens.* **2006**, *72*, 249–260. [[CrossRef](#)]
38. Arulampalam, M.; Maskell, S.; Gordon, N.; Clapp, T. A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking. *IEEE Trans. Signal Process.* **2002**, *50*, 174–188. [[CrossRef](#)]
39. Doucet, A.; de Freitas, N.; Gordon, N. An Introduction to Sequential Monte Carlo Methods. In *Sequential Monte Carlo Methods in Practice*; Springer: New York, NY, USA, 2001; pp. 3–14. [[CrossRef](#)]
40. Rasmussen, C.E.; Williams, C.K.I. *Gaussian Processes for Machine Learning*; MIT Press: Cambridge, MA, USA, 2006.
41. Solak, E.; Murray-Smith, R.; Leithead, W.E.; Leith, D.J.; Rasmussen, C.E. Derivative observations in Gaussian process models of dynamic systems. In *Advances in Neural Information Processing Systems*; MIT Press: Cambridge, MA, USA, 2003; pp. 1057–1064.
42. Gordon, N.; Salmond, D.; Smith, A. Novel approach to nonlinear/non-Gaussian Bayesian state estimation. *IEE Proc. Radar Signal Process.* **1993**, *140*, 107–113. [[CrossRef](#)]
43. Hol, J.D.; Schon, T.B.; Gustafsson, F. On Resampling Algorithms for Particle Filters. In Proceedings of the 2006 IEEE Nonlinear Statistical Signal Processing Workshop, Cambridge, UK, 13–15 September 2006, pp. 79–82. [[CrossRef](#)]