공학석사 학위논문

플래시 라이다를 사용한 템플릿 매칭 기반의 지형참조항법

Template matching based TRN using Flash LiDAR

2018 년 8 월

서울대학교 대학원

기계항공공학부

윤 인 도

플래시 라이다를 사용한 템플릿 매칭 기반의 지형참조항법

Template matching based TRN using Flash LiDAR

지도교수 박 찬 국

이 논문을 공학석사 학위논문으로 제출함

2018 년 6 월

서울대학교 대학원

기계항공공학부

윤 인 도

윤인도의 공학석사 학위논문을 인준함

2018 년 6 월

위 원 장 _____ (인)

부위원장 _____ (인)

위 원 _____ (인)

Abstract

플래시 라이다를 사용한 템플릿 매칭 기반의 지형참조항법

Indo Yoon
Department of Mechanical and Aerospace Engineering
The Graduate School
Seoul National University

This thesis compares and analyzes a performance of template matching based terrain referenced navigation (TMTRN) using correlation functions according to different error types and correlation functions. Conventional batch processing TRN generally utilizes the radar altimeter and adopts mean square difference (MSD), mean absolute difference (MAD), and normalized cross correlation (NCC) for matching a batch profile with terrain database. If a flash LiDAR is utilized instead of the radar, it is possible to build a profile in one-shot. A point cloud of the flash LiDAR can be transformed into 2D profile, unlike a vector profile obtained from batch processing. Therefore, by using the flash LiDAR we can apply new correlation functions such as image Euclidean distance (IMED) and image normalized cross correlation (IMNCC) which have been used in computer vision field. The simulation result shows that IMED is the most robust for different types of errors.

Keywords: Similarity measure, Terrain referenced navigation, LiDAR
Student Number: 2016-28891

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Motivation and background

Terrain referenced navigation (TRN) is an popular alternative of an integrated system of inertial navigation system (INS)/Global Navigation Satellite System (GNSS) because GNSS is vulnerable to jamming and spoofing [3]. TRN system has pre-manufactured database in on-board computer and remote sensor. The system acquire the estimate position of the aircraft by utilize database and the measurement. Therefore, the navigation solution which is independent of GNSS status.

TRN can be categorized as a batch processing TRN (BPTRN) and a sequential processing TRN according to their positioning method [4]. The BPTRN periodically updates the position by correlating stacked measurements with DEM. The sequential processing TRN recursively updates the estimates of INS error using estimators such as extended Kalman filter [5] and point mass filter [6].

The BPTRN system measures a clearance between the aircraft and the terrain using radar altimeter. Terrain contour matching (TERCOM) [7] introduced in 1958 is an example of BPTRN. The BPTRN is also called correlation based method because correlation functions (CF) are used to calculate the correlation. The BPTRN measures clearances of last N epochs to create a batch profile.

The BPTRN correlates a batch profile to candidate profiles from database, which is digital elevation map (DEM) [8]. Among candidates, the one with the highest correlation is chosen as the solution. The CFs used in the BPTRN are mean square difference (MSD), mean absolute difference (MAD) [9–11], and normalized cross correlation (NCC) [12].

Many TRN applications have utilized radar altimeter (RADALT) as a remote sensor, but recent studies are adopting LiDAR. The flash LiDAR is one type of the LiDAR and is being spotlighted in various applications such as automotive navigation [13, 14]. The flash LiDAR has advantages over scanning LiDAR. It is accurate, compact, and power efficient [15]. However, in TRN, the most of papers focus on application of scanning LiDAR.

The update speed of the BPTRN is slow due to stacking previous measurements. To assure the navigation solution is within the candidate, searching area should be enlarged. To get the accurate solution, it is required that searching interval should be reduced, generating more number of candidates, and stacking more measurement. However, computational load is increasing when searching area for candidates is big and searching interval is small and update speed is slowed if more measurements are stacked. Even though the BPTRN has some drawbacks, a correlation method still can provide accurate and bounded error. Furthermore, the correlation method extra information when the navigation solution of BPTRN is used as measurement for sequential processing TRN.

Image-based TRN (IBN) is an alternative navigation method to TRN, which exploits aerial terrain image as a reference. IBN generally extracts the feature points from the image from the camera and the pair from the database, and then the matching is performed. The proposed method of this thesis is different from the IBN, because the proposed method does not have a scale issue, and the number of pixels are very fewer than aerial camera. Furthermore, the LiDAR can

operate for day and night, but the camera needs ambient light such as sunlight. Even though infrared camera is utilized, the temperature of the terrain would be changed by the time of the day and the seasonal changes, so it is almost impossible to collect the data and create the database.

1.2 Objectives and contributions

In this thesis, we propose TMTRN using flash LiDAR. The performance of TRN according to CFs are compared and analyzed. The main contributions of this thesis are as follows.

- We implemented the flash LiDAR in TRN by extending the idea of batch processing to template matching. The flash LiDAR measures multiple ranges simultaneously, and this measurement can be transformed into measurement profile, which is a synthesized terrain elevation. TMTRN has two benefits against BPTRN thanks to this 2D measurement profile. This enables application of novel CFs of IMED and IMNCC. IMED and IMNCC requires an adopting a Gaussian spatial function, which is incalculable in a vector profile of BPTRN. Secondly, the TMTRN can fix the position every time the ranges are measured. The BPTRN needs to stack previous measurement for correlation, and BPTRN has a trade-off in accuracy and update speed. Therefore, TMTRN achieves higher accuracy with faster update speed.

- Each correlation functions has different characteristics, so there is no correlation functions working for all kind of templates [16]. There has been no studies that comparing or analyzing the suitableness of CFs according to the types of templates. Previous study [17] regarding the performance of TRN according to CFs only covers BPTRN, and another study only deals

with computer vision problems [18]. Therefore, analysis and comparison of performance of TMTRN according to CFs needs to be studied. First, CFs are categorized as distance-based, correlation-based, and Image-class, and their respective characteristics are organized. Secondly, the suitableness of CFs is examined for the matching terrain PC. Moreover, it needs to be verified that a certain CF works the best for matching PC and also for TMTRN. The performance is compared with matching test of various terrain templates and the Monte-Carlo simulation of TRN.

Chapter 2 provides literature survey of TRN methods with various remote sensors and template matching. Chapter 3 introduces the concepts of correlation-based TRN methods, which are conventional batch processing TRN and matching-based TRN method. In addition, we will address the characteristics of correlation functions. In chapter 4, to analyze the robustness against different types of errors, Monte Carlo flight simulation is conducted. The result clearly shows that specific correlation functions have advantage over other functions. Lastly, the thesis is summarized and concluded in chapter 5.

Chapter 2

Related Works

In this chapter, we provide the literature survey of related studies. In section 2.1, the concept of terrain referenced navigation (TRN) is introduced. Various methods of TRN according to their sensors and processing method are explained. Lastly, template matching and correlation functions are discussed in section 2.2.

2.1 Terrain Referenced Navigation

A characteristic of Inertial navigation system (INS) is diverging, and Global Navigation Satellite System (GNSS) has bounded error. When two systems are integrated, that system has bounded error with fast update rate. For this reason, INS/GNSS integrated system has been widely used in navigation. However, because GNSS is vulnerable to jamming and spoofing [3], the integrated system has a limitation for safe mission. To overcome the limitation, new algorithm which is independent with GNSS was needed to be invented.

Terrain referenced navigation (TRN) is one of the alternatives to GNSS. TRN is a system with remote. sensing sensors referencing database of terrain features to find current position, and usually digital elevation map (DEM) is used as reference (Fig. 2.1). TRN system has INS to continuously update position, velocity, and attitude and has sensors to measure terrain altitude.

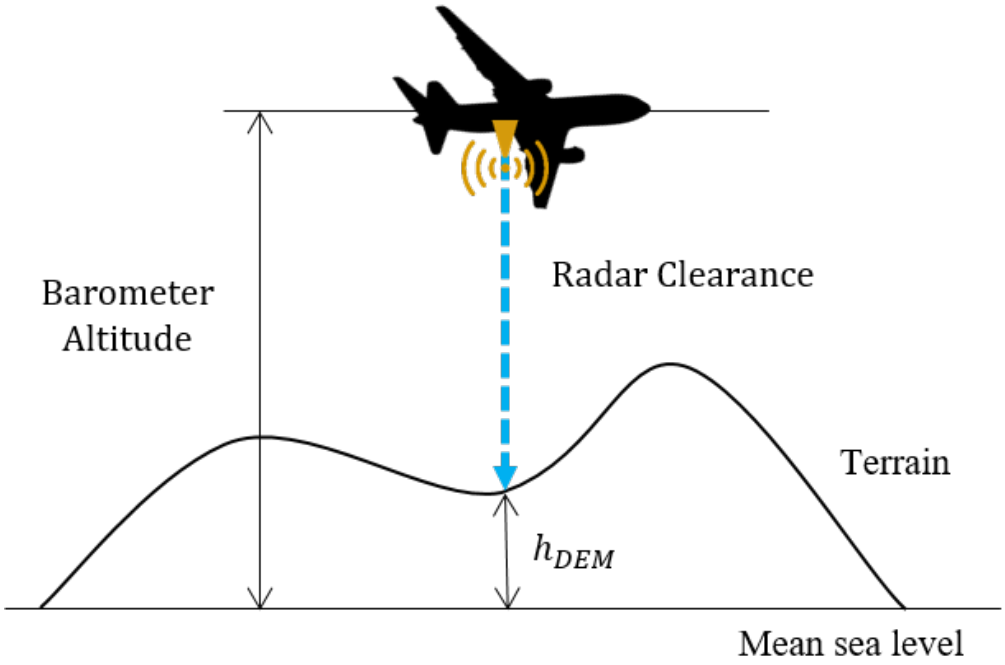


Figure 2.1: Scheme of TRN

A lot of research have been covered TRN using radar altimeter (RADALT) 2.1. The first automated TRN system is ATRAN (automatic terrain recognition and navigation), and the system was studied from 50s to 60s. After 70s, radar altimeter and barometric altimeter was applied to TRN which are TERCOM and SITAN (Sandia inertial terrain-aided navigation). TERCOM is the most representative batch processing and correlation-based TRN method using RADALT [7]. A block diagram of TERCOM is shown in Fig. 2.2. TERCOM synthesizes batch profile, which is a series of clearance measurements and correlates the profile with DEM to fix a position of the vehicle. In correlation process, MSD and MAD are frequently used as CFs.

SITAN integrated INS, barometric altimeter, and RADALT by implement-

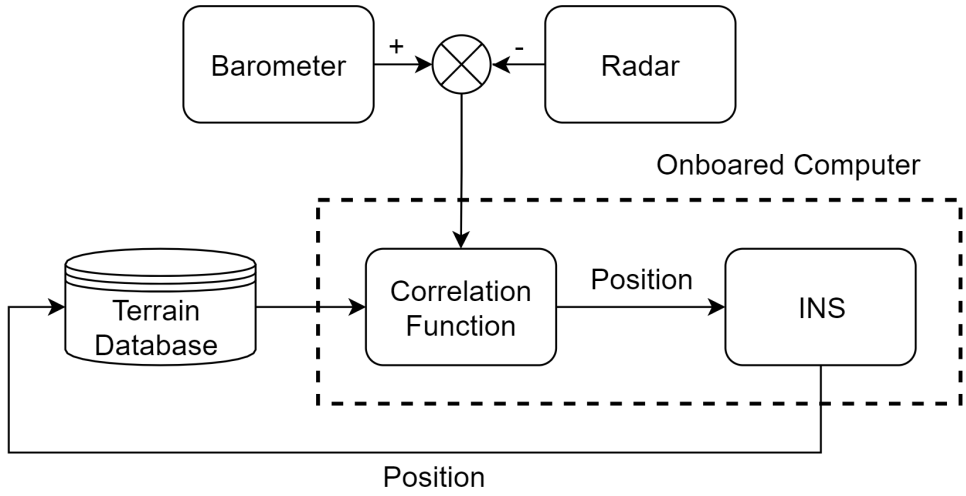


Figure 2.2: Block diagram of a BPTRN

ing extended Kalman filter (EKF). The innovation is difference between calculated terrain slope from INS and RADALT, and actual terrain slope from DEM. In 80s, Bayesian estimation is applied to SPARTAN (Stockpot algorithm robust terrain aided navigation) and TERPROM (Terrain profile matching), which is currently loaded on F-16 is developed.

TRN systems can estimate the position based on batch processing and sequential processing. In batch-TRN, the system stacks the measurements from previous N epochs, while sequential-TRN processes only the measurement from current epoch and estimates the state estimations by nonlinear filters. TRN systems also can be subdivided according to the remote sensors [19] and the processing method [20]. The sensors such as radar altimeter (RADALT), LiDAR or laser range finder, and camera or vision sensors have been applied. There are advantages and disadvantages of each sensors.

- RADALT using C-band, X-band or interferometric radar is all-weather

sensor. Depend on the composition, the accuracy differs, but generally the accuracy and precision is inferior to LiDAR.

- LiDAR can achieve several range measurement with very fast speed, with broad FOV. Because the laser is emitted from the emitter, regardless of the lighting condition the sensor can be used. LiDAR is very sensitive to water and vapor. It has a very higher accuracy, higher resolution and faster update rate than RADALT. The accuracy and resolution are possibly reduced in daylight.

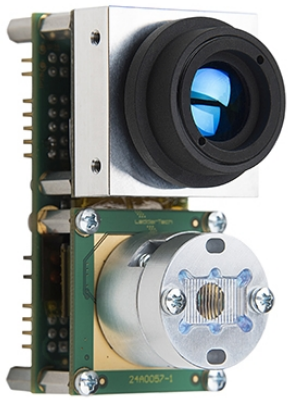
- Camera/Vision sensor lacks depth information if used only single camera. It is very sensitive to illumination condition, and therefore unable to operate in night time.

In the following subsections, TRN applications utilizing LiDAR and vision sensor are introduced.

Table 2.1: Development of TRN

System	RADALT	Period	Application	Horizontal CEP (m)	Database	Remarks
ATRAN	X-band	'50 ∼ '60	Mace missile	305	35mm film	Analogue
TERCOM	C-band	'70∼	Tomahawk	30.5	Res. 122m	Batch
SITAN	C-band	'70∼'80	Aircraft	75	Unknown	Sequential
SPARTAN	C-band	'80	Aircraft	Unknown	Unknown	Bayesian statistics
TERPROM	C-band	'80∼	Aircraft/missile	30	Unknown	
APALS	X-band	'90∼	Aircraft	3	Spotlight SAR	
PTAN	Interferometric C-band	'90∼	Tomahawk	3∼30	DTED level 4	Susceptible to vehicle velocity

(a) Hokuyo scanning LiDAR



(b) LeddarTech Flash LiDAR

Figure 2.3: An example of the two types of LiDAR

2.1.1 LiDAR-based TRN

LiDAR is an optical sensor that measures time of flight (ToF) of laser. ToF times the speed of the light gives the traveled distance of the laser. There are two types of conventional LiDAR, one is scanning LiDAR and the other is flash LiDAR (Fig. 2.3).

Scanning LiDAR consists of rotor part and sensor part. The rotor rapidly rotates the sensor part to collect point cloud (PC) as a shape of line with some field of view (FOV). Scanning LiDAR has good resolution but it is expensive and less robust. Flash LiDAR is often called as solid-state LiDAR due to the LiDAR is fixed to the platform without any moving parts. The geometry of the flash LiDAR is depicted in Fig. 2.4. Unlike the scanning LiDAR, the flash LiDAR has a lens that refracts the laser. The emitter shoots broad laser pulse to the target. The laser pulse is then reflected from the target and returned lasers are recognized by the receiver. The number of range measurement is defined by

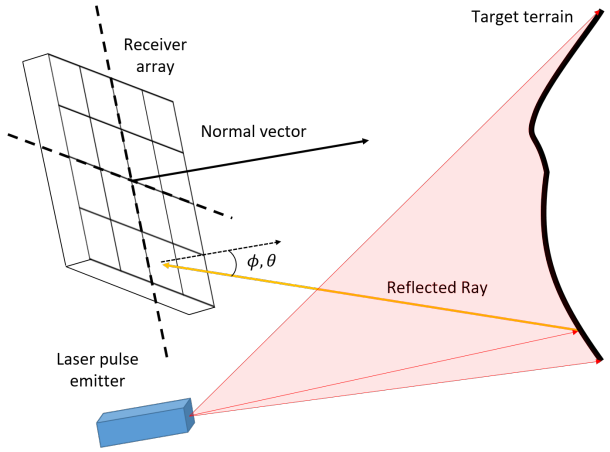
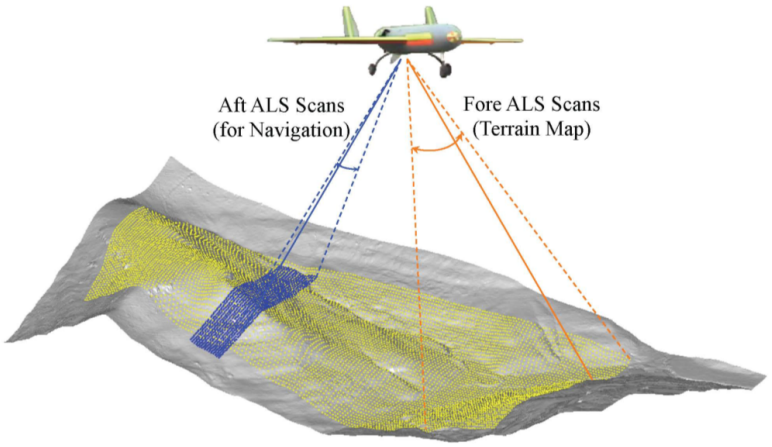


Figure 2.4: Geometry of flash LiDAR

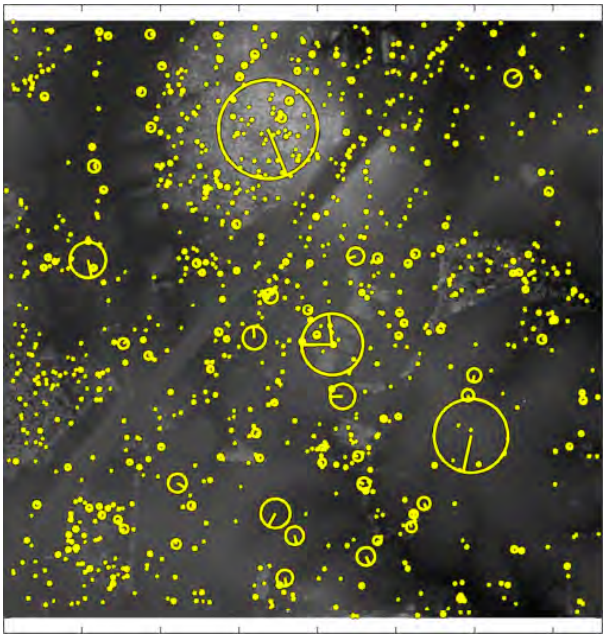
the number of the cells in the receiver.

Advantages and disadvantages of the flash LiDAR over scanning LiDAR are listed below:

- Advantages :
  - Wider FOV of cross- and along-track direction
  - Immediate mapping of terrain
  - No moving parts
  - Robust to noise due to longer exposure of laser pulse
  - Smaller form factor
- Disadvantages :
  - Less achievable numbers of PC (Low resolution)
  - FOV limitation in cross-track direction
  - Limited range detection due to reduced return signal



(a) Forward/Backward dual ALS [21]



(b) SIFT features [22]

Figure 2.5: Applications of scanning LiDAR

Next, scanning and flash LiDAR applications in TRN are introduced.

Airborne laser range scanner (ALS) is an application of scanning LiDAR to the aircraft. ALS measures slant ranges instantaneously. ALS has a filed-of-view in cross-track direction, and the movement of the aircraft enables ALS to collect PC along-track direction. In [19], they suggested a TRN mechanism using airborne laser scanner (ALS). PC is correlated by MSD with DEM to find the estimate position. Haag et al. [21] proposed feedforward of elevation map of PC and feedback of the position and velocity by correlation with database using dual ALS (Fig. 2.5a). Leines et al. [22] proposed feature points from PC by using SIFT descriptor (Fig. 2.5b).

For flash LiDAR applications, Johnson et al. [23] firstly suggested flash LiDAR based hazard avoidance of Mars lander. Hwang et al. [24] also utilized flash LiDAR for TRN and they compared the performances of batch processing and sequential processing techniques. Jeon et al. [25] proposed sequential TRN scheme based on flash LiDAR. Their algorithm shows improvement in robustness by applying Gaussian process to elevation and the covariance.

2.1.2 Image-based TRN

About image-based TRN, various research has been conducted. In [1], they proposed TRN method using terrain information from hazard detection and avoidance (HDA) (Fig. 2.6). The information from HDA is transferred to stereo vision measurement when Mars lander takes low-altitude flight for localization. Lee et al. [26] suggested replacement of INS with monocular camera. Homography of camera can estimate ground relative motion of the aircraft, the motion is integrated with range measurement of radar by point mass filter (PMF). Similar approach was proposed by [27], and they expanded this scheme to update not only position but also velocity and attitude.

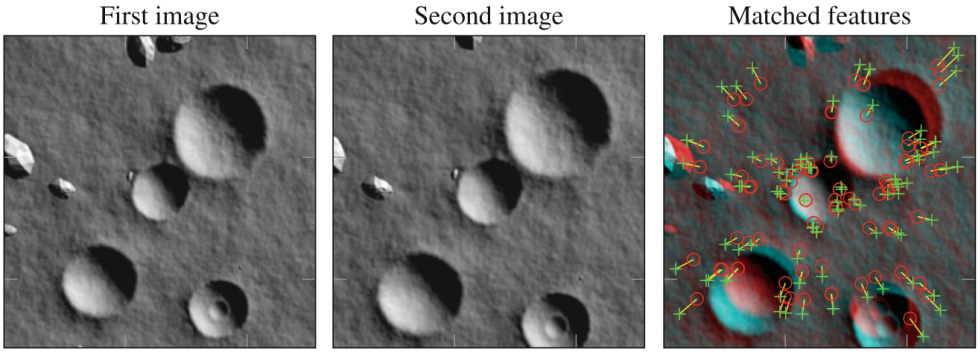


Figure 2.6: Matched features over two consecutive images. [1]

If template matching method is applied in the image-based TRN, the problem arises. In order to perform template matching without scale information, it is necessary to generate image pyramids for various scales and perform matching for each. If the scale information is inaccurate, it is necessary to generate several pyramid layers, the amount of computation also increases dramatically. Therefore, the most common method is to extract the features of the image and obtain the affine transform matrix. At this time, the landmarks are used to extract the feature points of the edges, but the corners are rare in the terrain, and the number of pixels of the flash LiDAR Since it is very small, it is generally difficult to directly apply the method used in IBN. On the contrary, in the case of Template matching, since the depth information of the LiDAR is given very precisely, accurate and robust matching is possible by using the average elimination technique regardless of the altitude of the aircraft.

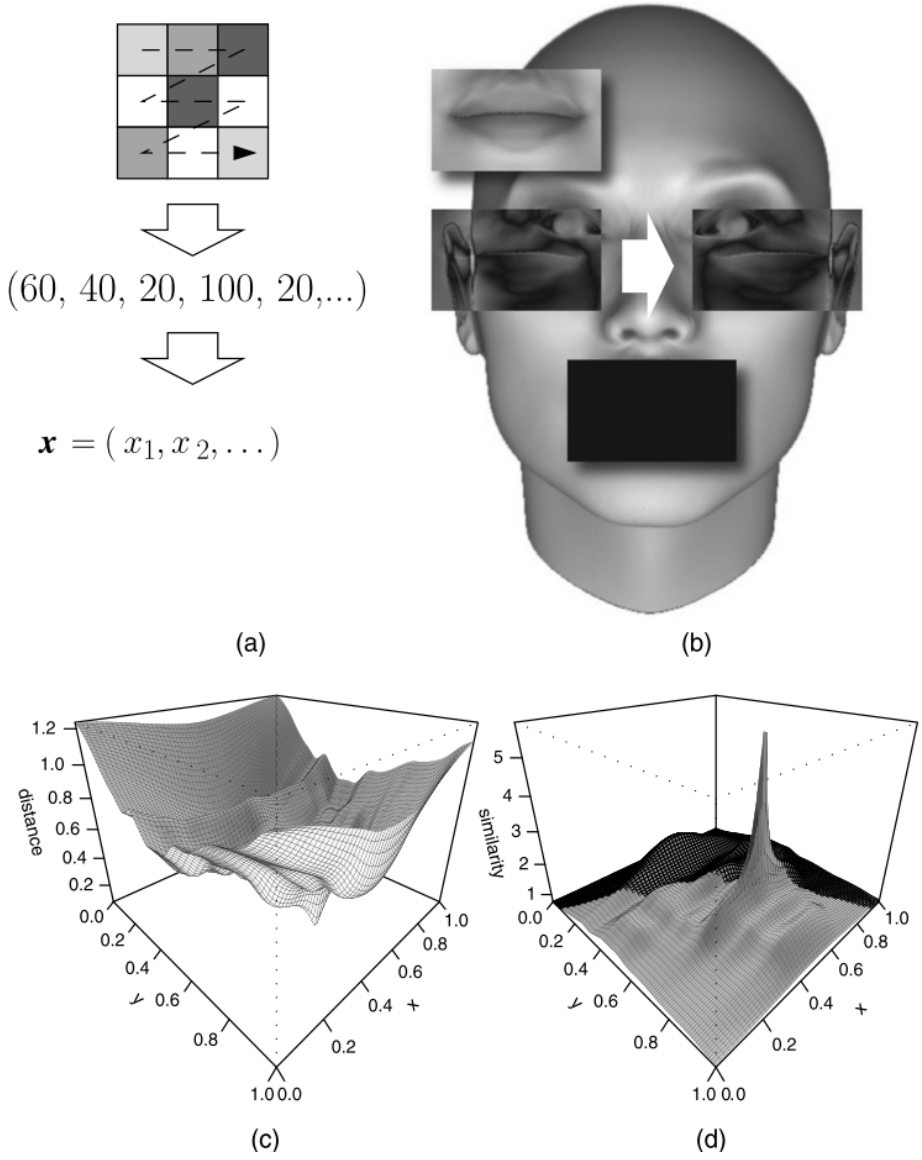


Figure 2.7: Simple template matching example [2]

2.2 Template Matching

2.2.1 General idea of template matching

Template matching is finding the best matching result by calculating the correlation between the template and the candidates [2] (Fig. 2.7). Conventional template matching algorithms exploit feature point of the image, such as corner, edge, and etc. Templates are often transformed and projected to other space. For example, to calculate a cross-correlation faster, templates are transformed using fast Fourier transform (FFT). FFT is also used to interpret the image in frequency domain. Another example of transformation is Radon/Hough transform, which are transforming a line to a point, and a point to a line. Using these techniques the similarity measures can find the correlation more accurately and robustly.

A major issues of template matching is robustness of similarity measures. The image is possibly corrupted by additive noise, constant illumination change, translation or perturbation, camera lens distortion, and other unknown distortions.

An aerial gray scale image and a measurement profile which is a terrain PC have different characteristics. The pixel value of the image has the maximum. For example, the maximum value of 8-bit image is 256. The PC has no limitation in the maximum value of a pixel, because basically the values of pixels are range measurements. The camera has higher resolution than the flash LiDAR, and this causes the vicinal pixels of the image have higher correlation than that of the PC. Furthermore, major error source of the measurement profile is atmospheric condition, while that of the aerial image is illumination condition. Because the aerial image and the terrain PC have different characteristics, different approach is needed for the matching task of terrain PC.

2.2.2 Correlation function

Correlation method is one of the template matching method. Measuring the distance or similarity between images is a fundamental and open problem in both psychology and computer vision. Correlation function (CF) is a function that gives correlation between two sets. The function is defined as below.

$$C_{X,Y} = Corr(X, Y) \tag{2.1}$$

The inputs X, Y of the function are sets which have the same length or size. Because each CFs has different characteristic, there is no CFs working for all kind of templates [16]. Templates often include additive noise, constant illumination change, and unknown distortion.

Many researches have been done on developing a CF which is robust to the noise and returning steady result. MSD is the most fundamental and simple CF, which returns the squared value of Euclidean distance. However, MSD has many shortcomings, and to overcome this shortcomings, various distances have been proposed. Histogram cosine distance [28], fractional distance [29], tangent distance [30], Hausdorff distance [31], fuzzy feature contrast [32], part-based methods [33], Isomap [34], and local linear embedding (LLE) [35]. Isomap and LLE calculates the distance in the manifold, and others are non-metric. The metric axioms are not satisfied by those CFs. In other word, self-similarity, symmetry, and the triangle inequality are not satisfied.

Low-level distance CFs, such as MSD and MAD, have little computational burden, but they are vulnerable to constant illumination change [36]. MSD and MAD are represented as follow, respectively.

$$\text{MSD} = \frac{1}{N} \sum_{i,j} \Big(h_{prof}(i,j) - h_{\text{DEM}}(i,j) \Big)^2 \tag{2.2}$$

Table 2.2: Characteristics of correlation functions

Functions	Correlation		Computational burden
	High	Low	
MSD	0	∞	Low
MAD	0	∞	Low
NCC	1	0	Mid
IMED	0	∞	High
IMNCC	1	0	Very High

$$\mathrm{MAD} = \frac{1}{N}\sum_{i,j}\left|h_{prof}(i,j) - h_{\mathrm{DEM}}(i,j)\right| \tag{2.3}$$

where $h_{prof}(i,j)$ and $h_{DEM}(i,j)$ are (i,j)th element of measurement and candidate profiles.

Cross correlation methods, such as NCC and IMNCC are known they have the best result for real image applications [12, 16]. Though they have high computational load and bit vulnerable to white noise, they are highly robust to illumination change [37]. NCC and IMNCC are represented as follow, respectively.

$$\mathrm{NCC} = \frac{1}{N}\frac{\sum_{i,j} h_{prof}(i,j)h_{\mathrm{DEM}}(i,j)}{\sum_{i,j} h_{prof}(i,j)^2 \sum_{i,j} h_{\mathrm{DEM}}(i,j)^2} \tag{2.4}$$

$$\mathrm{IMNCC} = \frac{\sum_{i,j}\sum_{i',j'} g_{i,j,i',j'} h_{prof}(i,j)h_{\mathrm{DEM}}(i',j')}{\sqrt{s_1 s_2}} \tag{2.5}$$

where

$$s_1 = \sum_{i,j} \sum_{i',j'} g_{i,j,i',j'} h_{prof}(i,j) h_{prof}(i',j') \tag{2.6}$$

$$s_2 = \sum_{i,j} \sum_{i',j'} g_{i,j,i',j'} h_{\mathrm{DEM}}(i,j) h_{\mathrm{DEM}}(i',j') \tag{2.7}$$

where $g_{i,j,i',j'}$ is a Gaussian function of spatial distance between pixels, originally proposed and adopted in IMED. The vicinal pixels may have similar intensities, and therefore this function reflects the relationships of vicinal pixels and gets larger weight for near pixels than distant pixels.

$$g_{i,j,i',j'} = \frac{1}{2\pi\sigma^2} exp\left(- \frac{dist\Big((i,j),(i',j')\Big)^2}{\sigma^2} \right) \tag{2.8}$$

Lastly, IMED is robust to small translation. The function is represented as follow.

$$\mathrm{IMED} = \sqrt{\sum_{i',j'} \sum_{i,j} g_{i,j,i',j'} ab} \tag{2.9}$$

$$a = h_{prof}(i,j) - h_{\mathrm{DEM}}(i,j) \tag{2.10}$$

$$b = h_{prof}(i',j') - h_{\mathrm{DEM}}(i',j') \tag{2.11}$$

IMED shares spatial function with IMNCC, but the functions was applied in IMED firstly. According to the authors [38], it explores spatial connection, and achieves improved robustness.

All the above CFs ignored the spatial relationship between the pixels. On the other hand, IMED proposed by Wang et. al [38] explores spatial connection, and achieves improved robustness. IMED is defined as

$$d(x,y) = \big[(x-y)^T G (x-y)\big]^{1/2} \tag{2.12}$$

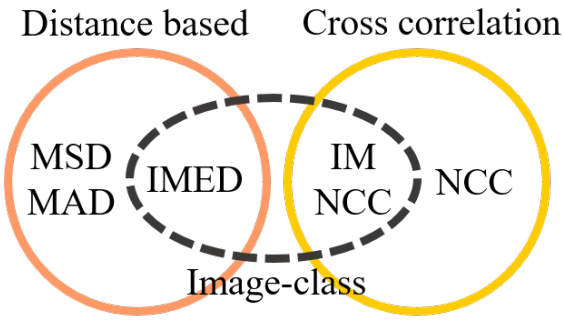


Figure 2.8: Relationship of CFs

However, still IMED and IMNCC has some limitations. First, the width parameter σ of the spatial Gaussian function has some ambiguity on its value. The authors and followed researchers [39, 40] did not mentioned the setting of the parameter. The ambiguity of the parameter causes naturally the second problem. Secondly, the IMED only considers the relationship of pixels of vicinity. The terrain PC surely has correlation between pixels, but it is hard to find the appropriate function that represents the relationship, even though we changed the spatial functions as the exponential.

In summary, the characteristics of CFs are in Table. 2.2 and the relationship of CFs can be displayed as Fig. 2.8.

Mean removal technique [41, 42] is used to eliminate the effects of bias in range measurements, the barometer, or DEM. Means of measurement profile and candidate profiles are subtracted from each.

$$\tilde{\mathbf{h}}^m = \mathbf{h}^m - \frac{1}{MN}\sum_{a,b}^{MN} h^m_{a,b} \tag{2.13}$$

$$\tilde{\mathbf{h}}^d = \mathbf{h}^d - \frac{1}{MN}\sum_{a,b}^{MN} h^d_{a,b} \tag{2.14}$$

$\mathbf{h}^m$ is measurement profile, and $\mathbf{h}^d$ is candidate profile. In this thesis, mean removal technique is applied for conventional CFs, and they are referred with a prefix Z:ZMSD, ZMAD, ZNCC, ZIMED, and IMZNCC.

Chapter 3

Template matching based TRN

3.1 Relationship with BPTRN

Terrain referenced navigation (TRN) is an absolute navigation method which is GNSS-independent. Once a database is manufactured and uploaded in the computer of a vehicle, the vehicle is able to navigate autonomously, and there is no need to connect the vehicle with the ground station.

Batch processing TRN (BPTRN) is the most fundamental TRN method. Often BPTRN is integrated with Kalman filter or with batch Kalman filter (BKF). In this section, measurement update process is omitted, and acquisition mode of BPTRN will be discussed. BPTRN synthesizes a profile by stacking previous N range measurements form radar, according positions from INS, and altitudes from barometric altimeter. The radar is assumed to be looking downward to measure clearance. After a clearance obtained from the radar and an altitude from barometer are subtracted, a synthesized terrain elevation, which is a batch profile, is obtained as follows.

$$\mathbf{h}_{batch}(i) = h_{baro}(i) + \delta h_{baro}(i) - (\rho(i) + \delta\rho(i)) \tag{3.1}$$

where $h_{batch}(i)$ is ith element of batch profile, $h_{baro}(i)$ is an altitude from barometer, and $\rho(i)$ is a clearance obtained from the radar. $\delta h_{baro}(i)$ and $\delta\rho(i)$ are error components of the barometer and the clearance. This concatenated profile

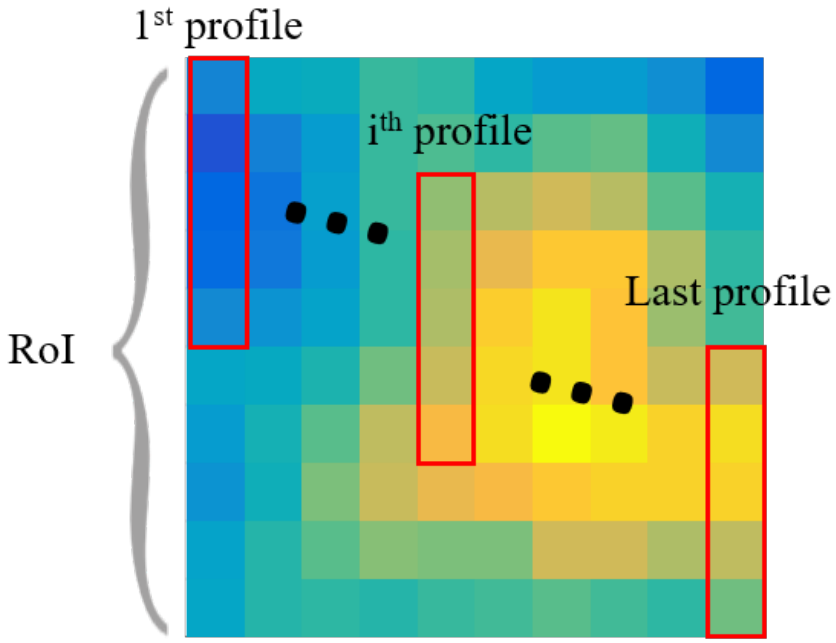


Figure 3.1: Generating candidate profiles

$\mathbf{h}_{batch}$is now defined as a batch or measurement profile. The candidate profiles are created via the following process (Fig. 3.1).

1. Estimate positions of each elements of the batch profile are concatenated in a vector form.

2. In region of interest (RoI), virtual flight paths are created using the concatenated positions, according to pre-set intervals (Latitude and longitude direction).

3. Terrain elevations for each virtual flight paths are computed by linear interpolation of DEM.

Using MSD, MAD, and NCC, correlations between the batch profile and candi-

date profiles are calculated. The candidate profile having the highest correlation is chosen and the according position is updated.

There are three drawbacks of batch TRN. The first is poor performance due to low accuracy of the radar compared to the LiDAR [8]. Secondly, building a batch profile takes a time because the system needs to collect measurement of last N epochs. Big size of the batch profile can improve the accuracy but the update rate will be slowed. Lastly, it is possible to achieve better performance by increasing the size of the batch profile and the size of the searching window/RoI. In this case, computational load is precipitously increased as well.

Classic BPTRN uses correlation method to fix the position of the vehicle. Correlation method is one of the template matching algorithm. However, rather matching vector-shaped profile, matching task of matrix-shaped profile which is generally an image is more commonly referred as template matching. This thesis adopts the basic idea of correlating measurement profile with DEM and expands that from 1D vector profile to 2D matrix profile. In other words, TMTRN is an extended version of correlation-based TRN.

3.2 TMTRN algorithm

TMTRN is an expanded version of batch processing TRN. The idea of correlating the measurement with the DEM is adopted, and the measurement dimension is extended from 1D to 2D.

Recent studies of TRN have covered sequential processing TRN, which is filtering based. More specifically, literatures focus on dealing with nonlinearity of the terrain, and their solution is utilizing nonlinear filters such as extended Kalman filter, unscented Kalman filter, particle filter, and point mass filter.

In conventional TRN applications, measurement from RADALT was vul-

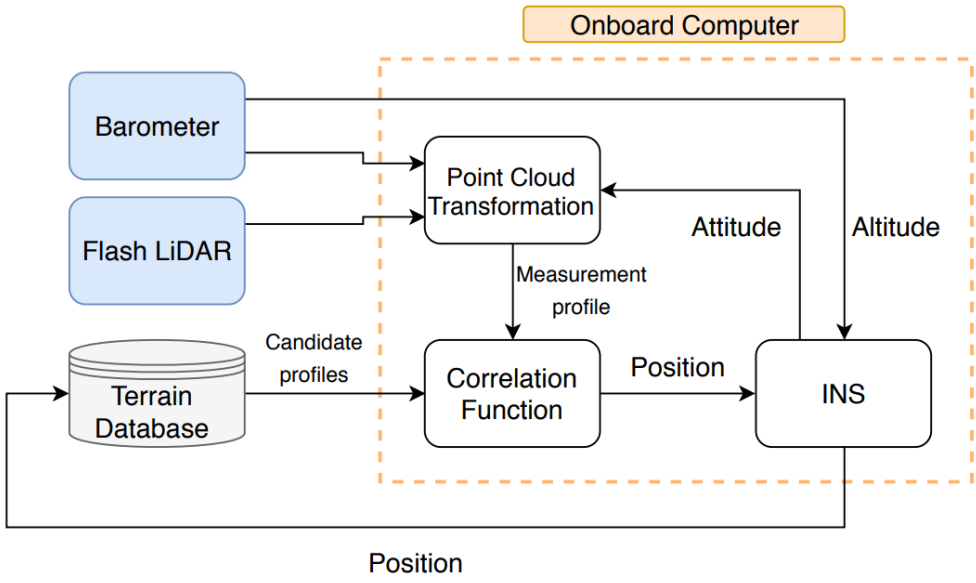


Figure 3.2: Block diagram of a template matching-based TRN

nerable to disturbance and the precision and accuracy of the radar was low. Moreover, reliability of DEM was low as well. Due to those reasons, adopting of new nonlinear filters could improve the performance of TRN. However, now very accurate range measurement is achievable from LiDAR and newly manufactured DEM is accurate. Therefore, the performance changes according to the filters have almost no differences.

The proposed method is able to be integrated with conventional TRN scheme easily. This method gives the position of the vehicle by correlation process, this position can be used as pseudo-measurement of filter.

A block diagram of the TMTRN is shown in Fig. 3.2. The TMTRN system consists of an IMU, a barometric altimeter, and a flash LiDAR. When the aircraft fly over the terrain, the flash LiDAR obtains PC composed of slant ranges and angles of incident. In this study, we assume a virtual flash LiDAR

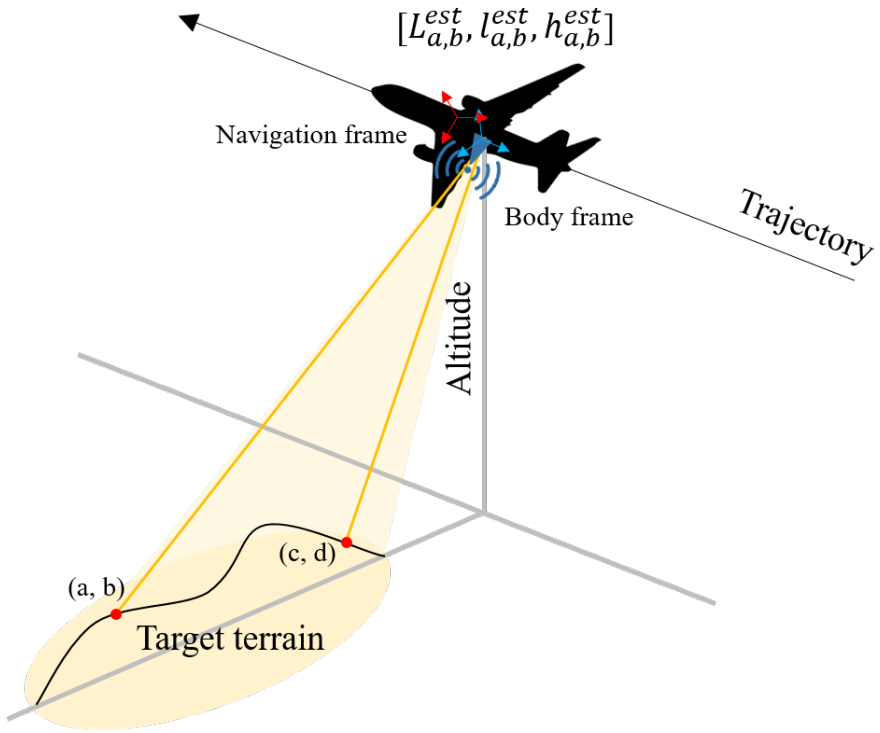


Figure 3.3: Generating measurement profile of TMTRN

that has angles of incident ϕ and θ increasing with the same offset for azimuth and elevation directions from the center.

The TMTRN for implementing the flash LiDAR is inspired by conventional batch-TRN, which is a correlation-based method. Compared with batch-TRN, TMTRN has two major differences. The first is that it can adopt new CFs thanks to 2D measurement profile of TMTRN. New CFs such as IMED and IMNCC have not been studied in conventional batch-TRN. Secondly, the flash LiDAR measures multiple clearance at once while batch-TRN method collects single radar range measurements along the time track to build a measured profile. Therefore, the update rate of TMTRN is much faster. In other words, available information per one update is much more in TMTRN. For example,

if the update rate of the radar is 1Hz and the batch length is 10, it takes 10 seconds for fixing one position. TMTRN can fix the position every 1 second when the flash LiDAR has the same update rate.

In batch-TRN, a position of an element of batch profile is equal to that of the aircraft. However, TMTRN measures slant ranges and therefore 3D positions of elements should be calculated. 3D positions of target terrain relative to the aircraft profile can be calculated as below (Fig. 3.3).

$$\begin{bmatrix} \mathbf{L}^m_{a,b} \\ \mathbf{l}^m_{a,b} \\ \mathbf{h}^m_{a,b} \end{bmatrix} = \begin{bmatrix} \mathbf{L}^{est}_{a,b} \\ \mathbf{l}^{est}_{a,b} \\ \mathbf{h}^{est}_{a,b} \end{bmatrix} + C^n_b(\rho_{a,b} + \delta\rho_{a,b})D_{a,b} \tag{3.2}$$

where $D_{a,b}$ is

$$D_{a,b} = \begin{bmatrix} \sin(\phi_{a,b} + \delta\phi_{a,b})\cos(\theta_{a,b} + \delta\theta_{a,b}) \\ \sin(\phi_{a,b} + \delta\phi_{a,b})\sin(\theta_{a,b} + \delta\theta_{a,b}) \\ -\cos(\theta_{a,b} + \delta\theta_{a,b}) \end{bmatrix} \tag{3.3}$$

Here, $\mathbf{L}^m_{a,b}$, $\mathbf{l}^m_{a,b}$, and $\mathbf{h}^m_{a,b}$ denote 3D positions of the measurement profile. $\mathbf{L}^{est}_{a,b}$, $\mathbf{l}^{est}_{a,b}$, and $\mathbf{h}^{est}_{a,b}$ denote estimate position from INS. h_{baro}, δh_{baro} are the barometer altitude and the error component. (a, b) is indices of the element. $\rho_{a,b}$, $\phi_{a,b}$, and $\theta_{a,b}$ represent a measured range, azimuth, and elevation angles of incident. Their respective error components are $\delta\rho_{a,b}$, $\delta\phi_{a,b}$, and $\delta\theta_{a,b}$. C^b_n is body-to-navigation direction cosine matrix calculated using attitude obtained from INS. This 3D position will be referred as a measurement profile in the rest of the thesis.

Candidate profiles are generated analogously with BPTRN as following steps (Fig. 3.3).

- Latitudes and longitudes from $\mathbf{p}_m$ are concatenated. Two matrices $\mathbf{L}^{est}$

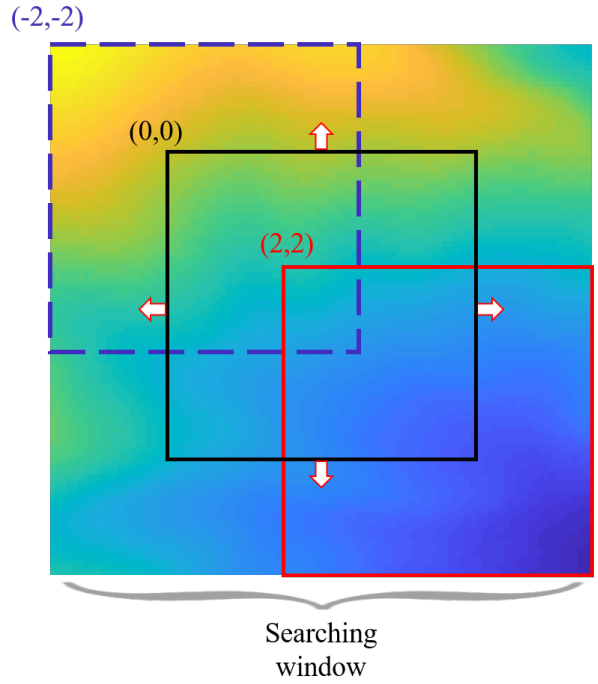


Figure 3.4: Generating candidate profiles of TMTRN

and $\mathbf{l}^{est}$ of latitude and longitude are then created.

- An gradual offset is added in $\mathbf{L}^{est}$ and $\mathbf{l}^{est}$. The resultant matrices are $\mathbf{L}^{cand,k}$ and $\mathbf{l}^{cand,k}$, where $k = \{1, 2, \ldots, N\}$. For example, if the center position is (3,3) and searching window size and searching interval are 2 and 1, the offset is $\{(-2,-2), (-1,-1), \ldots, (2,2)\}$. The resultant matrices are $\mathbf{L}^{cand,1}$, ..., $\mathbf{L}^{cand,25}$ and $\mathbf{l}^{cand,1}$, ..., $\mathbf{l}^{cand,25}$.

- kth candidate profile $\mathbf{h}^{d,k}$ is created by linearly interpolating the DEM at the position of $\mathbf{L}^{cand,k}$ and $\mathbf{l}^{cand,k}$. Repeat the step from $k = 1$ to N.

The TMTRN can adopt IMED and IMNCC in addition to MSD, MAD, and NCC, thanks to flash LiDAR. Similar to the BPTRN, the correlation between

the measurement profile and candidate profiles are calculated and the candidate with the highest correlation is selected.

The TMTRN has some drawbacks sharing with the BPTRN. The first is a curse of dimensionality, which means high computational load. To assure the navigation solution is among the candidate, searching area should be set wide enough. To get the accurate result, it is required that searching interval should be small and as a result generating more number of candidates. However, when searching area for candidates is big and searching interval is small, computational load is increasing. Second, finding measurement covariance matrix is challenging problem. The TMTRN can be integrated with INS by EKF or other nonlinear filters like loosely coupled INS/GNSS system, but without proper measurement noise model, there is always a possibility that the system diverges.

Chapter 4

Simulation Results

In this chapter, matching performances of CFs are compared by two simulations. In section 4.1, terrain PCs are cross-matched with each other, and their respective mean square error (MSE) is calculated and compared. In section 4.2, TMTRN is simulated with 5 types of CFs with various error conditions. Mean removal technique is applied to each CFs. The DEM for the simulations is SRTM level 1 [43], with a resolution of 3 arc-second. For generating a flash LiDAR measurements, the points were made by a map of 3/20 arc-second resolution. The map is generated by linear interpolation of SRTM level 1. The terrain is assumed as the Lambertian surface in the simulations, so emitted laser rays reflected at the target terrain, and come back directly to the receiver cell.

4.1 Template matching of terrain PC

In this section, matching performances of CFs are compared with each other. White noises with different levels are applied to the candidates and the mean squared error is calculated.

441 templates of each candidates with 30×30 PC are created by interpolating DEM in the region of interest (RoI) according to the procedure in 3.2. The process is repeated for 100 candidate profiles. The candidates are shown

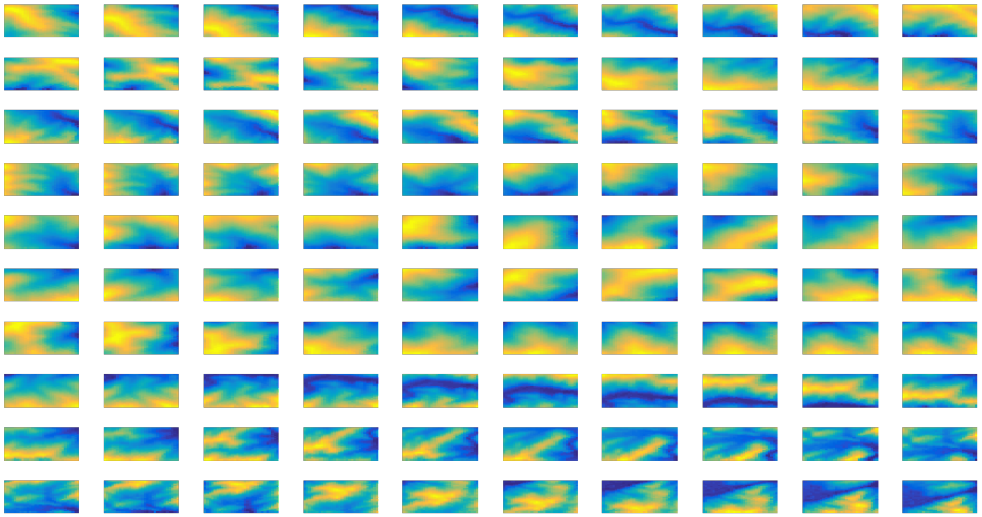


Figure 4.1: Terrain profile candidates

in Fig. 4.1. Each candidates has the same size and they are created with the same interval. In a terrain PC, each candidates has the same size and they are created with the same offset. The center candidate (0,0) is selected as a template, and this template is corrupted by additive Gaussian white noise with various levels. To compare the performance of CFs, mean squared error (MSE) is measured as follows. The template is correlated with other candidates from (-2,-2) to (2,2). If a CF returns that (2,1) has the highest correlation, an error becomes $2^2 + 1^2 = 5$. The process is then repeated for other 100 PCs, and and all errors are averaged. This error metric is chosen because TMTRN deals with intra-class comparison. In other words, the candidates in one terrain PC are actually slightly deformed from each other, so it is desirable to find a candidate the most similar with the template.

In this simulation, white noises of different levels of 5, and 10m are added to the candidates. For a reference, the candidates without noise ($\sigma = 0m$) are also used. ZMSD, ZMAD, ZNCC, ZIMED, and ZIMNCC are applied to

Table 4.1: Matching result of terrain PCs

σ(m)	ZMSD	ZMAD	ZNCC	ZIMED	IMZNCC
0	0	0	0	**0**	0
5	2.52	10.95	2.05	**0.98**	1.81
10	7.52	34.77	5.93	**4.3**	5.89

calculate the correlation. NCC and IMNCC are modified as they return 0 if two templates are identical, as the values are bounded as [0, 1]. The result is depicted in Table. 4.1. IMED shows the best performance among conventional CFs for $\sigma = \{0, 5, 10\}$, followed by IMZNCC.

This simulation is organized to identify the performance of intra-class matching of terrains. In a candidate, 441 templates were created with small interval. Therefore, they can be regarded as slightly deformed version of each other. In other words, this task is to find the closest brother to each other. The result shows ZIMED is suitable to apply when levels of additive Gaussian noise.

From the result, it is also shown that IMED is robust to severe additive noise. Expected noise in range measurement of flash LiDAR is from 1 to 10cm, and uncertainty of the DEM is also about 1 to 3m, when the DEM is assumed to created by concatenating PCs from airborne scanning LiDAR and post-processing. Therefore, it is concluded that IMED has possibilities of being applied in real applications of terrain PC matching.

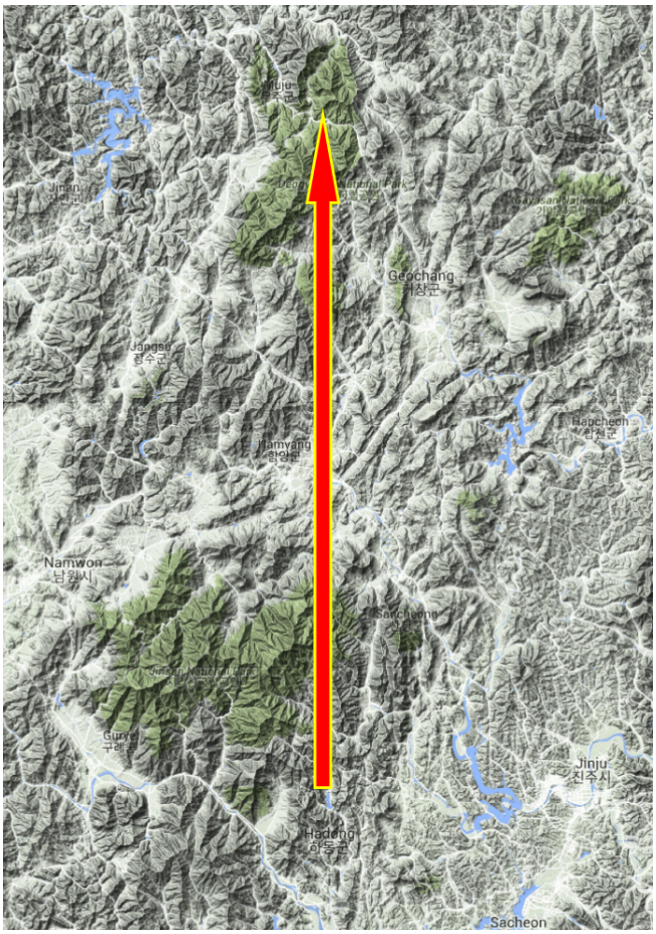


Figure 4.2: Trajectory for simulation

4.2 TMTRN simulation

In this section the performances of CFs in TMTRN are compared and analyzed. The conditions of the simulation are listed in Table 4.2. Flight trajectory (Fig. 4.2) starts from (35.15°, 127.70°), and ends at (35.18°, 127.70°). The aircraft flies with a constant velocity of 500km/h and a constant height of 2km. To measure the motion of the aircraft, the navigation grade IMU is used. Initial position error is 30 m. Monte Carlo simulations are performed for 30 times. The

flash LiDAR used in simulation has a receiver of which size is 9×9 with FOV of $30°$, same for azimuth and elevation directions.

The second simulation differs from the first simulation in compared pairs. In the second, the template is synthesized terrain profile, and the candidates are interpolated from the DEM, same as the simulation 1. If the estimate position of the vehicle is far away from the true position, the resultant measurement profile may a lot differ from the candidates, so the exact matching could be difficult.

Performance comparison of CFs is done by changing errors of range and angle of the flash LiDAR. Range white noise $\delta\rho$ is changed as 1, 3, 5, and 10m, and angle bias error $\delta\theta$ is changed as 0.05, 0.15, $0.3°$. To see the effect of barometer bias, not only MSD, MAD, NCC, IMED, and IMNCC but also their zero-mean versions are compared. Hence, overall 12 cases for each 10 CF are considered for the simulation. 30 times of Monte Carlo simulations are applied for all cases, and the RMSE results are shown in Table 4.3. The best results for each case are marked as bold.

Table 4.2: Simulation condition

Conditions		Values
DEM		SRTM level 1
Initial position error		30m
Monte Carlo		30
Gyroscope	Bias	1°/hr
	Random walk	$0.1^\circ/\sqrt{hr}$
Accelerometer	Bias	1mg
	Random walk	$0.1\text{mg}/\sqrt{Hz}$
Flash LiDAR	Resolution	9×9 px
	FOV (ϕ, θ)	30°, 30°
	Range white noise (1σ)	1, 3, 5, 10m
	Angle bias	0.05, 0.15, 0.3, 0.5°
Barometer bias		5m

Table 4.3: RMSE result

	RMSE									
$\delta\theta$	MSD	MAD	NCC	IMED	IMNCC	ZMSD	ZMAD	ZNCC	ZIMED	IMZNCC
0.05	34.23	34.96	13.96	36.29	14.24	9.09	9.87	10.93	**7.83**	8.60
0.15	45.77	48.01	17.55	48.42	18.41	9.68	10.79	12.19	**8.58**	9.34
0.3	90.52	91.03	39.85	93.30	44.81	26.58	27.11	38.79	**26.21**	26.91

(a) $\delta\rho = 1$m

	RMSE									
$\delta\theta$	MSD	MAD	NCC	IMED	IMNCC	ZMSD	ZMAD	ZNCC	ZIMED	IMZNCC
0.05	34.28	35.21	14.22	36.19	14.40	9.78	10.76	11.67	**8.51**	9.79
0.15	45.89	48.27	18.26	48.66	18.51	10.42	11.52	13.44	**9.15**	10.73
0.3	90.69	90.76	42.84	93.23	45.98	27.07	27.68	43.36	**26.42**	32.23

(b) $\delta\rho = 3$m

	RMSE									
$\delta\theta$	MSD	MAD	NCC	IMED	IMNCC	ZMSD	ZMAD	ZNCC	ZIMED	IMZNCC
0.05	34.18	35.25	15.33	36.29	14.75	10.63	12.07	12.68	**9.42**	10.75
0.15	45.83	47.95	18.82	48.83	18.99	11.54	12.93	14.95	**10.2**6	12.20
0.3	90.70	91.18	44.14	93.23	45.31	27.55	28.17	46.41	**26.77**	36.46

(c) $\delta\rho = 5$m

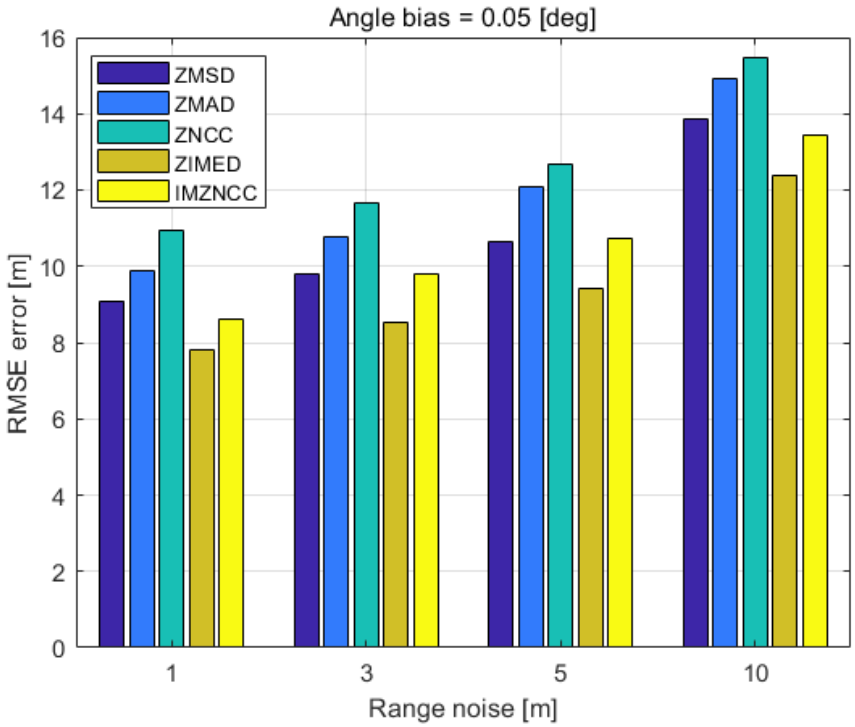
	RMSE									
$\delta\theta$	MSD	MAD	NCC	IMED	IMNCC	ZMSD	ZMAD	ZNCC	ZIMED	IMZNCC
0.05	34.27	34.69	17.15	36.70	16.18	13.85	14.91	15.45	**12.40**	13.45
0.15	45.46	47.27	21.26	49.47	19.69	14.88	15.79	18.51	**13.93**	15.12
0.3	90.60	90.98	51.11	94.23	47.60	**31.72**	36.39	53.17	36.66	44.02

(d) $\delta\rho = 10$m

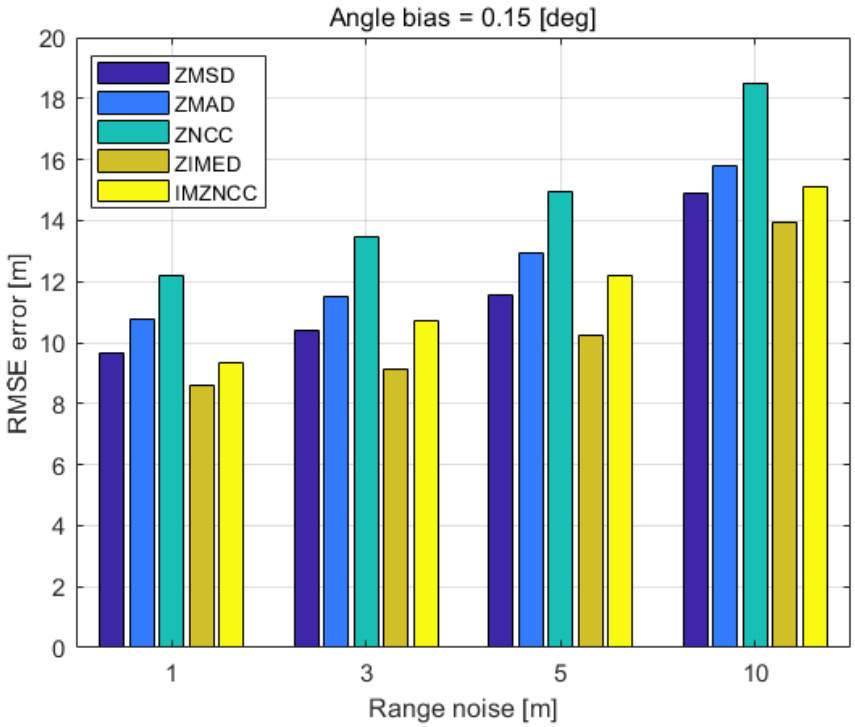
The performances of zero-mean CFs are better than normal functions, regardless of the error condition due to barometer bias. The trend of the table shows the effect of $\delta\theta$ is dominant to that of $\delta\rho$. For normal functions, for $\delta\rho$ from 1 to 5m, NCC is superior performance followed by IMNCC. In $\delta\rho = 10$, IMNCC shows better performances than NCC because the effect of white noise is weaker than consideration of neighbor pixels by $g_{i,j,i',j'}$. Overall, NCC and IMNCC has similar result when compared with other functions.

Comparison of RMSE error of zero-mean CFs is depicted in Fig. 4.3. Here we only display the result of zero-mean CFs, because they outperform normal functions as addressed. Among zero-mean CFs, ZIMED has the best performance for every conditions except for $\delta\rho = 10$ and $\delta\theta = 0.3$. From the result we can see that ZMSD is more robust to extreme errors than ZIMED. This is because Image-class CFs occasionally overrate the relationships between pixels even if the pixels are contaminated by severe noise. However, it is known that for moderate noise, Image-class functions show better result [16]. As conclusion, ZIMED is the best for overall, and ZMSD is good for high error levels.

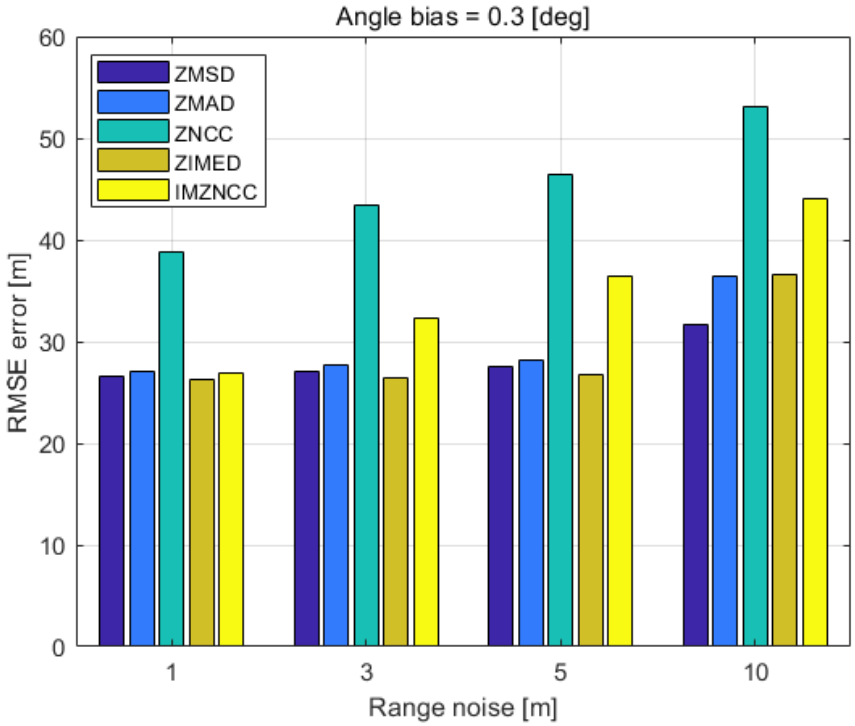
Under conditions of flight time 300 seconds, 30 times of Monte Carlo result is shown in Fig. 4.4 with the position RMSE and their standard deviation (1σ). Their respective conditions are $(\delta\rho, \delta\theta) = \{(1m, 0.05^\circ), (3m, 0.3^\circ), (10m, 0.3^\circ)\}$. ZIMED is used because it shows the best performance for overall. Initial error converges and the result is stable with a precision of 20, 30, 50m, even though the system only estimates and updates the position. Moreover, template matching in image-based navigation suffers from scale of template and illumination change, however in TMTRN, the effect of bias in profile comes from barometric altimeter, and it can be perfectly eliminated by zero-meaning. The result is not included here, but the performance of TMTRN is not affected by error in barometric altimeter.



(a) Angle bias : 0.05°

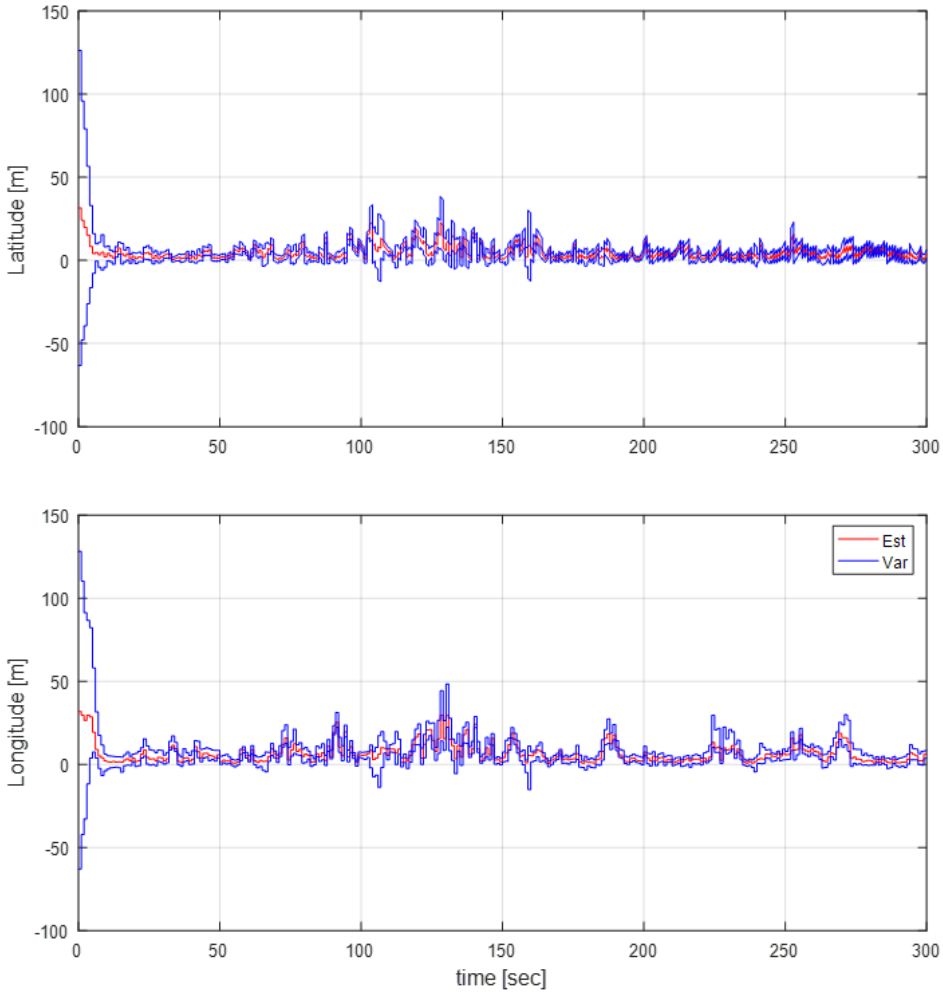


(b) Angle bias : 0.15°

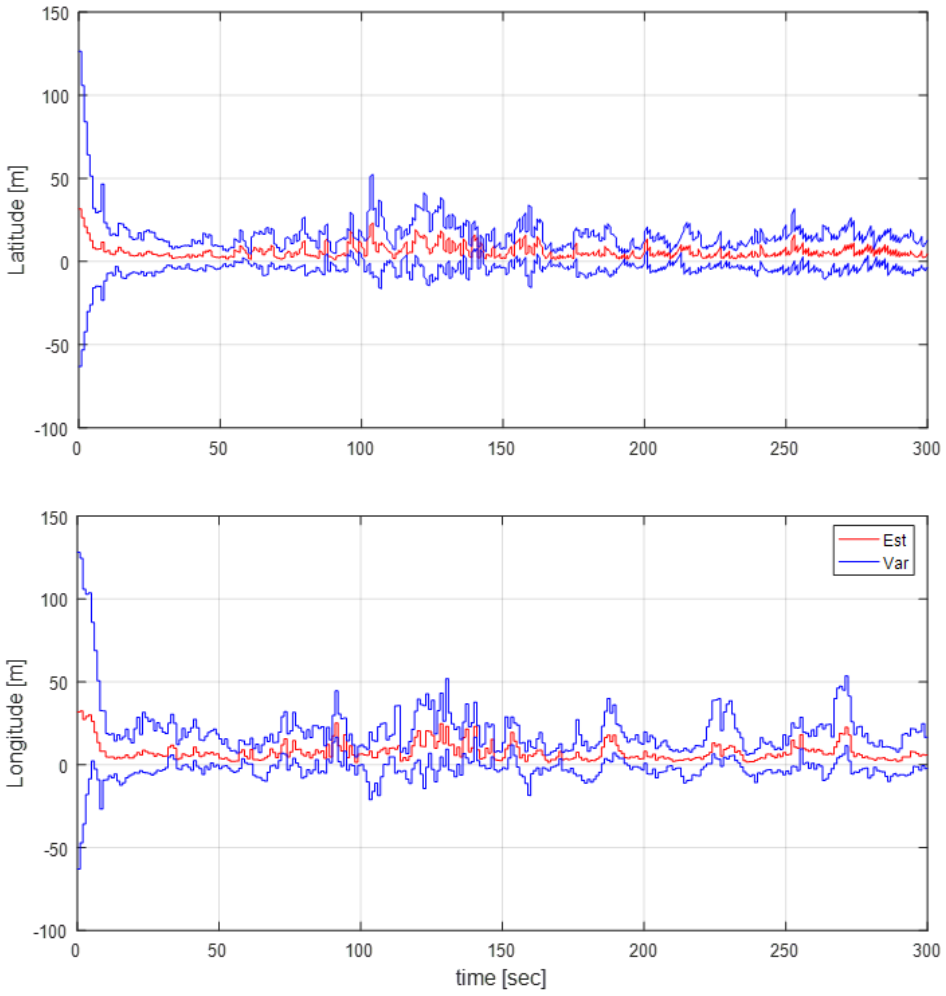


(c) Angle bias : 0.3°

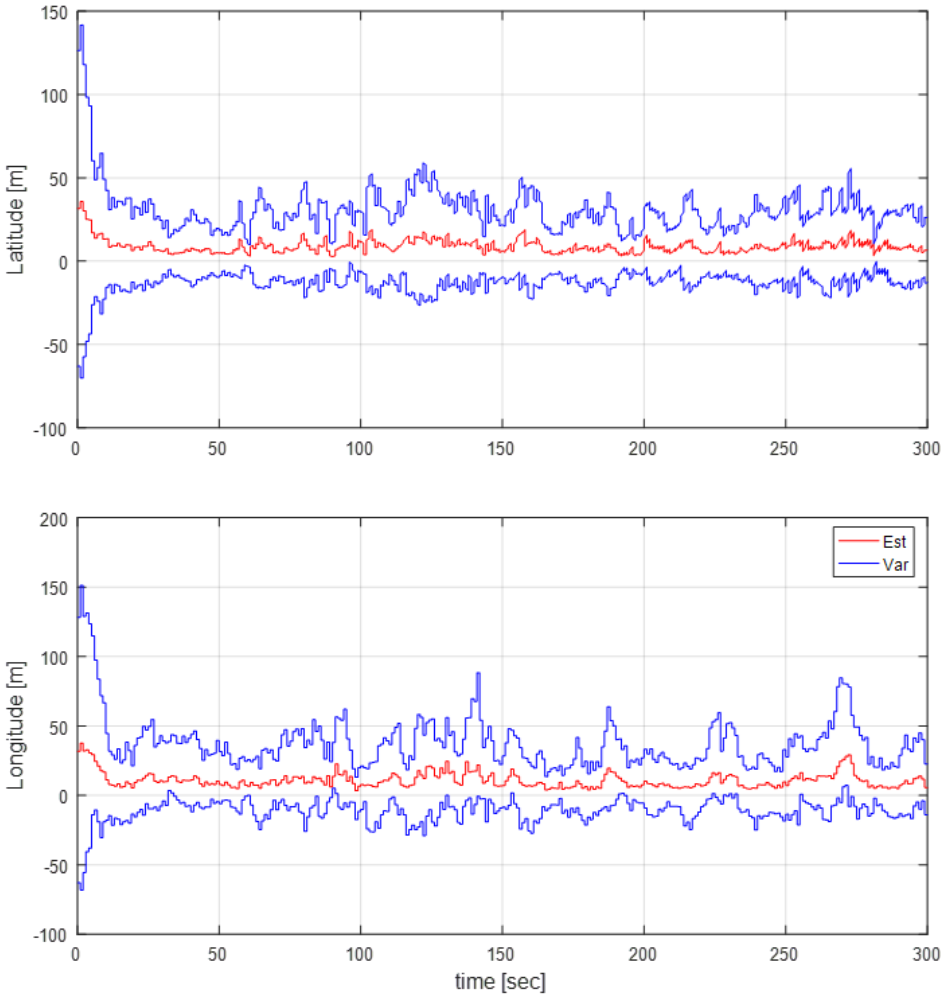
Figure 4.3: Matching results, RMSE of zero-mean correlation functions



(a) Range white noise : 1m, Angle bias : 0.05°



(b) Range white noise : 3m, Angle bias : 0.3°



(c) Range white noise : 10m, Angle bias : 0.3°

Figure 4.4: TRN RMSE results for ZIMED

ZIMED outperforms the other CFs for all conditions. IMED explores the spatial relationship between pixels by adopting the spatial Gaussian function. The terrain is continuous and composed of low-frequency wave shapes. If a shot is taken by flash LiDAR, that vicinal points will have similar vertical values. The spatial function gives the larger weight to the vicinal pixels when Euclidean distance is calculated, so the same principle is possible to applied in terrain PC matching, therefore, the matching performance can be enhanced.

However, it is controversial about the performance of IMNCC. IMNCC also adopts the spatial Gaussian function on its metric, but the performance is inferior to IMED. Many recent literatures have studied and commented on IMED about the metric itself and the applications, while IMNCC has almost no following researches.

Chapter 5

Conclusions and Future Works

5.1 Summary of the contribution

A LiDAR is able to measure multiple ranges with high accuracy and speed, but not many studies covered its application to TRN, especially for the flash LiDAR. In this thesis, a point cloud of the flash LiDAR is converted to 2D terrain profile by slant ranges, angles of incident, and a barometer altitude. The synthesized terrain profile is compared with candidates from DEM to find the aircraft's position.

This thesis presents three important remarks. Firstly, the flash LiDAR is integrated with TRN. Secondly, the integration of the flash LiDAR enables using of IMED and IMNCC, which have not been used for TRN. Conventional batch-TRN only builds 1D vector profile but the proposed algorithm's measured profile is a transformed point cloud, which is a 2D matrix profile. Thirdly, MSD, MAD, NCC, IMED, IMNCC, and their zero-mean versions are compared. By simulation, the robustness of CFs to range white noise and angle bias is studied by comparing TRN performances. The matching performance is improved as 13.86% for 1m of range noise and 0.05° of angle bias. For real application, we can assume range noise as centimeter level and small angle bias as less than 0.1°. As a conclusion, IMNCC could possibly show the best performance in real world.

Furthermore, the proposed TMTRN can be considered as sub-system of the larger TRN system, and TMTRN is able to provide the position estimation as a pseudo-measurement of the filter.

A drawback of this study is and Lambertian surface assumption of the terrain might be unrealistic. However, conventional LiDARs have millimeter or centimeter level of precision, and even when 10m white noise is applied IMED still shows improved result with the robustness. Therefore, IMED is possibly the best correlation functions for real PC matching task.

5.2 Future works

Future works of the thesis possibly will include two issues. First, more realistic simulation with the flash LiDAR is needed. Possible considerations are as following.

- Without Lambertian surface assumption, it needs to consider reflecting emitter and receiver geometry.
- The forest or vegetation would have different albedo from the ground.
- The effect of atmospheric condition needs to be considered for sensors using lasers.

BPTRN and other TRN applications have adopted filtering framework to estimate the state of the vehicle. Especially with Kalman filter, deriving the measurement covariance should be discussed.

Bibliography

[1] S. Woicke and E. Mooij, "Terrain relative navigation for planetary landing using stereo vision measurements obtained from hazard mapping," in *Advances in Aerospace Guidance, Navigation and Control*. Springer, 2018, pp. 731–751.

[2] R. Brunelli, *Template matching techniques in computer vision: theory and practice*. John Wiley & Sons, 2009.

[3] J. V. Carroll, "Vulnerability assessment of the us transportation infrastructure that relies on the global positioning system," *The Journal of Navigation*, vol. 56, no. 2, pp. 185–193, 2003.

[4] P. D. Groves, *Principles of GNSS, inertial, and multisensor integrated navigation systems*. Artech house, 2013.

[5] L. Hostetler and R. Andreas, "Nonlinear kalman filtering techniques for terrain-aided navigation," *IEEE Transactions on Automatic Control*, vol. 28, no. 3, pp. 315–323, 1983.

[6] M. Simandl, J. Královec, and T. Soderstrom, "Anticipative grid design in point-mass approach to nonlinear state estimation," *IEEE Transactions on Automatic Control*, vol. 47, no. 4, pp. 699–702, 2002.

[7] P. J. Klass, "New guidance technique being tested," *Aviation Week & Space Technology*, pp. 48–51, 1974.

[8] C. A. Baird and M. R. Abramson, "A comparison of several digital map-aided navigation techniques," in *PLANS'84- Position Location and Navigation Symposium, San Diego, CA*, 1984, pp. 286–293.

[9] J. P. Golden, "Terrain contour matching (tercom): a cruise missile guidance aid," in *Image processing for missile guidance*, vol. 238. International Society for Optics and Photonics, 1980, pp. 10–19.

[10] G. M. Siouris, *Missile guidance and control systems*. Springer Science & Business Media, 2004.

[11] L. G. Brown, "A survey of image registration techniques," *ACM computing surveys (CSUR)*, vol. 24, no. 4, pp. 325–376, 1992.

[12] P. Aschwanden and W. Guggenbuhl, "Experimental results from a comparative study on correlation-type registration algorithms," *Robust computer vision*, pp. 268–289, 1992.

[13] G. M. Williams, "Optimization of eyesafe avalanche photodiode lidar for automobile safety and autonomous navigation systems," *Optical Engineering*, vol. 56, no. 3, p. 031224, 2017.

[14] H. An, Y. Kwon, and Y. Jeong, "Monte carlo simulation of 3d flash lidar scheme for high-speed autonomous driving," in *Advanced Photonics 2017*, Optical Society of America. Optical Society of America, 2017, p. JTu4A.17.

[15] R. Stettner, "Compact 3d flash lidar video cameras and applications," in *Laser Radar Technology and Applications XV*, vol. 7684. International Society for Optics and Photonics, 2010, p. 768405.

[16] A. Nakhmani and A. Tannenbaum, "A new distance measure based on generalized image normalized cross-correlation for robust video tracking and image recognition," *Pattern recognition letters*, vol. 34, no. 3, pp. 315–321, 2013.

[17] B. M. Lee and J. H. Kwon, "Terrain referenced navigation simulation using area-based matching method and tercom," *Journal of the Korean Society of Surveying, Geodesy, Photogrammetry and Cartography*, vol. 28, pp. 73–82, 2010, (Korean).

[18] S. N. Borade, R. R. Deshmukh, and P. Shrishrimal, "Effect of distance measures on the performance of face recognition using principal component analysis," in *Intelligent Systems Technologies and Applications*. Springer, 2016, pp. 569–577.

[19] M. U. de Haag, A. Vadlamani, J. L. Campbell, and J. Dickman, "Application of laser range scanner based terrain referenced navigation systems for aircraft guidance," in *Electronic Design, Test and Applications, 2006. DELTA 2006. Third IEEE International Workshop on*. IEEE, 2006, pp. 6–pp.

[20] R. J. Handley, P. Groves, P. McNeil, and L. Dack, "Future terrain referenced navigation techniques exploiting sensor synergy," in *Proceedings of GNSS 2003*. The European Navigation Conference, 2003.

[21] A. Vadlamani and M. U. de Haag, "Flight test and simulation results of an integrated dual airborne laser scanner (dals)/ins navigator," in *Proc. of SPIE Vol*, vol. 7684, 2010, pp. 768 418–1.

[22] M. T. Leines, "Terrain referenced navigation using sift features in lidar range-based data," Air force institude of technology, Tech. Rep., 2014.

[23] A. E. Johnson and A. M. San Martin, "Motion estimation from laser ranging for autonomous comet landing," in *Robotics and Automation, 2000. Proceedings. ICRA'00. IEEE International Conference on*, vol. 1. IEEE, 2000, pp. 132–138.

[24] Y. Hwang and M. Tahk, "Terrain Referenced UAV Navigation with Lidar – a Comparison of Sequential Processing and Batch Processing Algorithms," *28th International Congress of the Aeronautical Sciences*, pp. 1–7, 2012.

[25] H. C. Jeon, Y. B. Park, and C. G. Park, "Robust performance of terrain referenced navigation using flash lidar," in *Position, Location and Navigation Symposium (PLANS), 2016 IEEE/ION*. IEEE, 2016, pp. 970–975.

[26] D. Lee, Y. Kim, and H. Bang, "Vision-based terrain referenced navigation for unmanned aerial vehicles using homography relationship," *Journal of Intelligent & Robotic Systems*, pp. 1–9, 2013.

[27] Y. Kim and H. Bang, "Vision-based navigation for unmanned aircraft using ground feature points and terrain elevation data," *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, vol. 232, no. 7, pp. 1334–1346, 2018.

[28] I. H. Witten, I. H. Witten, A. Moffat, T. C. Bell, and T. C. Bell, *Managing gigabytes: compressing and indexing documents and images*. Morgan Kaufmann, 1999.

[29] C. C. Aggarwal, A. Hinneburg, and D. A. Keim, "On the surprising behavior of distance metrics in high dimensional space," in *International conference on database theory*. Springer, 2001, pp. 420–434.

[30] P. Simard, Y. LeCun, and J. S. Denker, "Efficient pattern recognition using a new transformation distance," in *Advances in neural information processing systems*, 1993, pp. 50–58.

[31] N. Sudha *et al.*, "Robust hausdorff distance measure for face recognition," *Pattern Recognition*, vol. 40, no. 2, pp. 431–442, 2007.

[32] S. Santini and R. Jain, "Similarity measures," *IEEE Transactions on pattern analysis and machine Intelligence*, vol. 21, no. 9, pp. 871–883, 1999.

[33] D. W. Jacobs, D. Weinshall, and Y. Gdalyahu, "Classification with non-metric distances: Image retrieval and class representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 6, pp. 583–600, 2000.

[34] J. B. Tenenbaum, V. De Silva, and J. C. Langford, "A global geometric framework for nonlinear dimensionality reduction," *science*, vol. 290, no. 5500, pp. 2319–2323, 2000.

[35] S. T. Roweis and L. K. Saul, "Nonlinear dimensionality reduction by locally linear embedding," *science*, vol. 290, no. 5500, pp. 2323–2326, 2000.

[36] S.-D. Wei and S.-H. Lai, "Fast template matching based on normalized cross correlation with adaptive multilevel winner update," *IEEE Transactions on Image Processing*, vol. 17, no. 11, pp. 2227–2235, 2008.

[37] J. P. Lewis, "Fast normalized cross-correlation," in *Vision interface*, vol. 10, no. 1, 1995, pp. 120–123.

[38] L. Wang, Y. Zhang, and J. Feng, "On the euclidean distance of images," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 27, no. 8, pp. 1334–1339, 2005.

[39] Q. Gao, F. Gao, H. Zhang, X.-J. Hao, and X. Wang, "Two-dimensional maximum local variation based on image euclidean distance for face recognition," *IEEE Transactions on Image Processing*, vol. 22, no. 10, pp. 3807–3817, 2013.

[40] J. Li and B.-L. Lu, "An adaptive image euclidean distance," *Pattern Recognition*, vol. 42, no. 3, pp. 349–357, 2009.

[41] J. Metzger, O. Meister, G. F. Trommer, F. Tumbrägel, and B. Taddiken, "Adaptations of a comparison technique for terrain navigation," *Aerospace science and technology*, vol. 9, no. 6, pp. 553–560, 2005.

[42] P. M. Ku, Y. B. Park, and C. G. Park, "Improvement of batch trn using mean removal and two step search method for lunar lander," in *Control Conference (ASCC), 2015 10th Asian.* IEEE, 2015, pp. 1–6.

[43] T. G. Farr, P. A. Rosen, E. Caro, R. Crippen, R. Duren, S. Hensley, M. Kobrick, M. Paller, E. Rodriguez, L. Roth *et al.*, "The shuttle radar topography mission," *Reviews of geophysics*, vol. 45, no. 2, 2007.

국문초록

본 논문은 다양한 오류 유형 및 상관 함수에 따른 플래시 라이다를 사용한 템플릿 매칭 기반의 지형 참조 항법(TMTRN)의 성능을 비교하고 분석한다. 일반적인 일괄처리방식 지형참조항법은 일반적으로 레이더 고도계를 사용하고, 배치 프로파일을 지형 데이터베이스와 매칭시키기 위해 평균 제곱 차이(MSD), 평균 절대 차이(MAD)및 정규교차상관(NCC)를 사용한다. 레이더 대신 플래시 라이다를 사용하면, 시간에 따라 측정치를 모으는 과정이 필요하지 않아 한번에 프로파일을 생성할 수 있다. 일괄처리방식에서 사용하는 벡터 프로파일과 달리 플래시 라이다의 포인트 클라우드는 2D프로파일로 변환할 수 있다. 따라서 플래시 라이다를 사용하면 컴퓨터 비전 필드에서 사용되는 이미지 유클리디안 거리(IMED)및 이미지 정규교차상관(IMNCC)과 같은 새로운 상관 함수의 적용이 가능하다. 시뮬레이션 결과에 따르면, 이미지 유클리디안 거리가 오류에 가장 강건하다.

주요어: 서울대학교, 석사학위논문

학번: 2016-28891