3D Modeling and Its Evaluation for Large Scale Tunnel Using on Vehicle Laser Range Sensor

車載レンジセンサを用いた長大トンネルの３次元モデリングとその評価

Liang Xue
亮　薛

A Master Dissertation

Submitted to the Graduate School of
the University of Tokyo

東京大学
The University of Tokyo

in Partial Fulfillment of the Requirements
for the Degree of
Master of Information Science and Technology

February 2012

Thesis Supervisor: Katsushi IKEUCHI　池内克史

ABSTRACT

Precise and accurate 3D model of a road structure is the basic information which can be utilized for various purposes, such as safety measures, 3D navigation, driving simulation, and reference data for autonomous driving. Generally, the 3D model of a tunnel where GPS signal is not available has been constructed based on positioning using gyro sensor, however, error accumulation becomes a considerable problem in the case of long tunnels. We propose a method to obtain geometrically optimal whole structure of a tunnel:(1) Acquire a set of partial structures by static scanning using a laser range sensor, which is mounted on the roof of a survey vehicle, and align them by 3D matching using edge feature.(2) Fix the absolute position of the data for both ends of the tunnel by GPS and align the rest data again. By applying this method, we succeeded to create the 3D digital model of Kanaya Tunnel in New Tomei Expressway, whose length is 4.6 km.

For the comparison and evaluation of modeled tunnel, we present a system to evaluate the quality of the modeled tunnel comparing with the 2D CAD blueprint. This system can be described as following steps:(1)Extract the boundaries both from CAD blueprint and modeled tunnel to a 2D plane.(2)Transform the coordinate of modeled tunnel to the same coordinate as CAD blueprint.(3)Calculate the closest distance between the modeled tunnel boundaries and CAD boundaries.(4)Calculate the middle line of two data boundaries, and calculate the curvatures of both middle line, finally compare the curvature graph with the closest distance graph.

We succeed creating a dense model of actual tunnel with a length of 4.6 km by geometric process. For the error evaluation, we did the curvature calculation and closest distance calculation. The closest distance of our modeled tunnel with 2D CAD data is about 50 centimeters on the west side, about 20 centimeters on the east side and about 4 meters around the big curve in the middle of tunnel. From this result ,we have three assumptions about the error factors, which can be described as follows, the uncertainty in central part, the rotation error around the big curve during the local alignment process, and the GPS positioning errors accumulated in the simultaneous alignment. From curvature analysis, we can see that the largest error is at the central part, and the error rate is about 0.1 percent, also the GPS points should not be fully fixed in global alignment. About our future work, we plan to increase our modeling accuracy and reduce errors at three parts. For data acquisition, we want to increase more density of our raw data, for local alignment, we want to use more other features besides edges for local alignment to reduces accumulated errors, lastly, for global alignment, we want to allow some freedom for GPS points to move slightly.

Acknowledgements

First and foremost, my utmost gratitude to my Supervisor, Prof. Katsushi Ikeuchi, for his patience guidance and supervision. His high standard in researching left a deep impression on me. During each meeting, he pointed the problems very soon in my report and led me to think how to find the reason of problems instead of what to do. He has made available his support in a number of ways.

It gives me great pleasure in acknowledging the support and help of Dr. Shintaro Ono, Without his patience guidance and persistent helping this dissertation would not have been possible.

I would like to thank Dr. Atsuhiko Banno and Prof. Takeshi Oishi, they gave me a lot of helps and advises on both my research and this dissertation.

I am also grateful to Dr. Bo Zheng, he gave me a lot of advises when I have troubles on my research.

Many thanks to those who found time to direct me overcome difficulties and to lead me back to the right direction of my research. During these people, especially i would like to give my grateful to Yoshihiro Sato and Zhi Peng Wang. Special thanks to Boun Vinh Lu, for having a deep conversion about how to live a meaningful life and to make me view this world positively. I am also indebted to Keiko Motoki, Yoshiko Matsuura, Mikiko Yamaha and other members in the Computer Vision Laboratory at the University of Tokyo, for helping me so much in various ways.

I would like to thank my family for their support: my parents and my sister, for believing and inspiring me that I could overcome any kinds of difficulties, they let me know i am not alone here.

Lastly, I offer my regards and blessings to all of those who supported me in any respect during my master courses and the completion of this dissertation.

February 2012
Liang Xue

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background

Precise and accurate 3D model of a road structure is a basic information which can be utilized for various purposes, such as safety measure and its verification, 3D navigation, driving and noise simulation, and reference data for autonomous driving in the future, etc. Especially a tunnel zone is dark and narrow, and gives psychological suppression to drivers. Once an accident occurs in a tunnel, it tends to take long time for rescue and traffic control, therefore sufficient safety measure is required.

Nowadays, 3D modeling of urban structure is actively done generally based on aerial survey and vehicle survey by self position, and in some cases by matching sensing data with prior information such as maps and landmark databases. However, things drastically change in the case of tunnels, where GPS and aerial survey is completely unavailable.

The most popular solution for this is to obtain the self position by filtering measurement values of on-vehicle gyro sensor and speedometer [1], and quite a few cases is in practical use [2]. However, since this kind of method is essentially based on an integral of local measurement, accumulation of error becomes a considerable problem in a tunnel of kilometer-order length.

Although other kind method to use on-vehicle camera and apply image processing such as factorization is known [5], it is not easy to stably and precisely extract and track feature points in a dark environment, and the error accumulation is inevitable.

Another problem is, while using laser range sensor mounted on the survey car as a data surveying and modeling device, the most tradition method is surveying and modeling while driving such as what Mitsubishi MMS do, in this situation, they can only obtain a sparse source model, with less detail information.

Also, using prior information as [7] [8] is not reasonable, since it is essentially difficult to construct prior map and landmark database, and to uniquely match a series of quite similar scenes inside a tunnel.

In the case of this paper, we give priority to density and accuracy of the model rather than efficiency, with reasonable quantity of human work. We assume that a set of dense and accurate scans in a tunnel are obtained by static surveys and they are aligned by geometric processing without using external devices as a gyro sensor and a speedometer, and finally they are corrected and geometrically optimized using global information, where the accumulated errors are dispersed. The global corrective information can be given by GPS at both end of the tunnel.

Also, a CAD drawing for the tunnel construction can be considered to give global accuracy to some extent, however, it should be noticed that the actual structure of the tunnel is not assured to be same as the CAD drawing. One reason is that it only shows the most outer part in a cross section of the tunnel and does not show the inner wall area with some thickness, and will differ from the scanned data. Moreover, even at the stage of construction, a tunnel shield machine can not strictly follow the drawing, since the position of the machine digging forward in a tunnel is surveyed by using a total station behind the machine, and the machine can not suddenly change the heading. It can be naturally said that no one knows the true drawing or the 3D model of the tunnel inside.

1.2 Related Works and Problems

Aerial survey is always being used for topography surveying or historical remains. It is always used along with prior information to survey and model the historical remains. Especially for ruined city or building, a prior information is necessary. Sabry F and others from National Research Council of Canada have ever created hundreds of models from sites all over the world by using the prior information and aerial scanning method[4]. They firstly construct the basics shape such as columns and blocks from high-resolution digital images as prior information, then they used laser scanner to obtain the geometric detail from an aircraft and used the on-board GPS to record the aircraft trajectory, finally they integrated the prior information with the scanned geometric detail model. But for our object tunnel, neither the prior information nor the aircraft with GPS can be used. So we need to find out another way to do the modeling.

Mitsubishi Electric MMS system is a survey car system using laser range sensor for modeling city and road, it is using GPS as their positioning device in the usual modeling situation, they also choose to use a gyroscope sensor as the positioning device when they want to build the model inside a tunnel, the MMS has been used for evaluate the robustness of tunnel by the Transport and Tourism Kinki Regional Development Bureau [2], the gyroscope sensor can not supply a very precise position for the survey, so they can only evaluate the robustness of the tunnel, not including very accurate position information. Another example of MMS application is, the survey car mounted with MMS is driving while the survey and modeling is proceeding, it has been used for update the 2D road maps to 3D road maps by the Toyonaka city of Oosaka[3], because the scan speed limitation of laser scanner, they can only get a sparse model with less detail.

Our system overcame and avoided the problems existing in the above examples, we will discuss our system in next section.

1.3 System Overview

In this paper, we present an actual case to construct a 3D model of large-scale tunnel, where the structure can be considered as geometrically ″optimal″ and the error is dispersed, with reasonable quantity of human work.

The structure of our paper is described as follows: Chapter 2 aims to build a locally accurate raw model. We firstly extracted tunnel edges as alignment features, aligned each pieces of the acquired structure into a integrated tunnel by pair-wise alignment algorithm. Chapter 3 targets to reduce the accumulated errors that generated in the previous step and globally revise the tunnel structure by using simultaneous alignment algorithm with GPS constraint. Chapter 4 firstly introduces the object tunnel and data acquisition equipments, then describes the survey method we proposed, also, an algorithm to fill the holes on the tunnel ground where the survey car originally stopped is presented. The former three Chapters are the modeling part of our paper, modeling procedure is subdivided into three main parts as shown in Figure 1.1.

After tunnel modeling, we evaluate our modeling result comparing with the tunnel CAD blueprint. To do this work, in Chapter 5, fist we give the detail introduction of 2D CAD data received from Central-Nexco Expressway of Japan. Then we describe the detail procedure for boundary extraction both from 2D CAD data and our Modeled tunnel. Next we introduce how to do the coordinate transformation in order to put the two boundary data into a same coordinate. Lastly we use the k-d tree algorithm to do the closest distance search and calculate the curvatures of extracted boundaries for accuracy evaluation. Readers

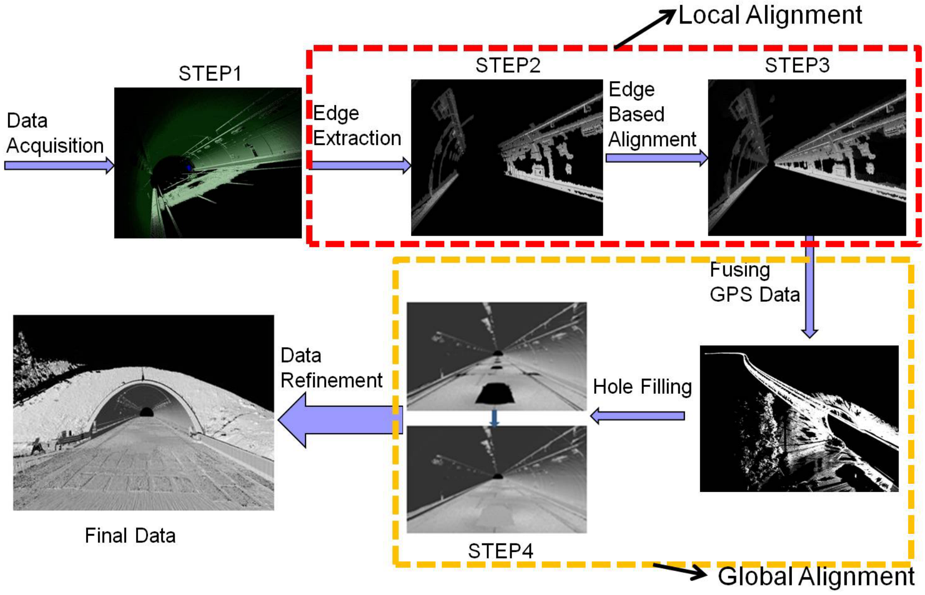

Figure 1.1: Modeling Procedure

can see the final comparison result in the last section of Chapter 5. Figure 1.2 is the process flow of the comparison and evaluation chapter.

In the conclusions chapter, firstly we discuss the possible reasons of why the difference occurs between our modeled tunnel and CAD blueprint. then for our future work, we plan to use three methods to increase the accuracy of of work in data acquisition, local alignment and global alignment respectively.

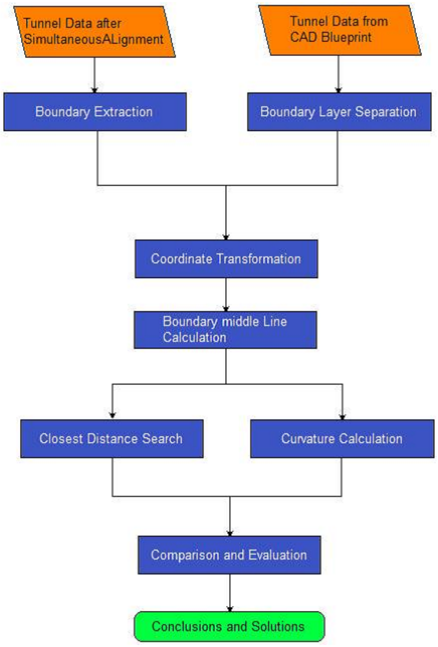
Tunnel Data after SimultaneousALignment
Tunnel Data from CAD Blueprint
Boundary Extraction
Boundary Layer Separation
Coordinate Transformation
Boundary middle Line Calculation
Closest Distance Search
Curvature Calculation
Comparison and Evaluation
Conclusions and Solutions

Figure 1.2: Process Flow of Comparison and Evaluation

Chapter 2

Local Alignment

Since we scanned the tunnel by using Stop and Go method, the scanned tunnel data consisting of many segments along with their own coordinate origin, we must align these segments into an integrated one. However, most parts of the tunnel are similar in geometry as a cylinder, especially in the axis direction, which will possibly lead to mismatching. Besides, the point cloud data of a long tunnel tends to be quite huge, which will obviously affect the processing speed to align them. So we extracted geometric features from each data, and used it for the pair-wise alignment.

In this chapter, the first section introduces some existing method for pair-wise alignment, section two describes the edge based extraction we proposed, the final section introduces the pair-wise alignment by using the extracted edges in previous section.

2.1 Sequential Alignment

The sequential alignment aims to build the homology between two or more polymer structures based on the three dimensional conformation and their shape [41]. This method is widely used in two areas. First is bioinformatics, for DNA,RNA or protein identification. The other is for 3D range image alignment.

For range image alignment, sequential method focus on two range images one time, and align one range image to another. After finish aligning one pair, a new pair including one of former range images becomes the next alignment object pair. By using this method, each new pair coordinate is based on the former aligned image coordinate. This alignment work proceeding repetitively until all range images are aligned.

To align two similar range images, many methods have been proposed. We can use two range images one time and estimate the relative position for the two images by calculating. The most famous two method is the ICP(Iterative Closest Point) method proposed by Besl [15] and the method proposed by Chen [16]. It is often used to reconstruct 2D or 3D surfaces from different scans. The ICP algorithm aims to find the closest corresponding points in two range images, in order to minimize the nearest distance between the two closest points, calculate the transformation matrix of the range images, by looping calculating the transformation matrix, the position relationship can be calculated. In the other side, Chen calculates the normal line of mesh, to minimize the distance between point and the mesh, Chen try to find the transformation relationship. Besides, there are other methods which also using "projecting along view line direction" method to find the corresponding point in other range image such as G.Blais proposed in [17]. We must notice that ICP sometimes have mismatching problem, and it is easily influenced by noises in range images, to solve these two shortages, random sampling and LMEDS(Least Median Squares Estimation) methods were proposed [24], this two methods has a higher robustness than ICP.

Because the sequential alignment tends to align two images one time, it has a low computational complexity and need less memory, so this method is widely implemented by range image alignment, we also used the sequential alignment method for our raw data alignment work this time.

2.2 Edge Extraction

Actually without extracting the features of tunnel, we can also align each segment into an integrated tunnel by using pair-wise alignment, but there are two problems if we align the tunnel directly. (1)The similar features between two nearby segments are not easy to distinguish, so the mismatching problems may occur sometimes. Before we exact the edges as alignment features, we actually align the tunnel for attempting, but the result was not good, shifting and distortion happened after mismatching. Figure 2.1 shows an example of mismatching. (2)The second problem is, the original data size is too huge after we scanned, the total size of the 142 scanned tunnel is about 100GB, so one scan data is about 800MB, it requires a considerable system memory and CPU to implement the alignment process smoothly.

To reduce mismatching and redundant data for the alignment, we extracted edges as features. In fact, edges on the tunnel wall are not the only feature for alignment, other objects can also act as features such as scavenging ports, emergency lamp, traffic lane, etc. In this paper, for convenience, we extracted edges as our alignment features. In this case, only the surrounding area of 3D edges will be used for the alignment. After this step, the matching point pair for alignment will be reduced obviously, so the alignment can also become more efficient. The extraction procedure is shown below:

- Extract 3D edges from the scanned data (Figure 2.2(a)).
- Map the 3D edges onto a virtual sphere whose center is at the sensor position (Figure 2.2(b)).
- Re-map the expanded edge back to the original place and extract the surrounding area of edges (Figure 2.2(c)).

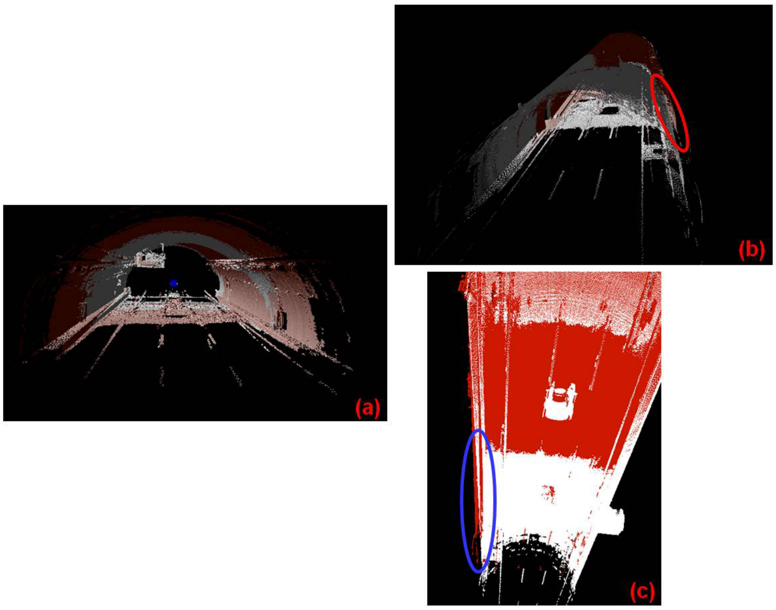

Figure 2.1: An Example of Misalignment. (a)Inside View (b)Side View (c)Bottom View

The output data of range sensor are the collection of 3D points of the object surface, and these continuous points can be considered as many small meshes. We firstly find the normal line of each mesh and then calculate the difference between each two nearby meshes. If the result is larger than a threshold, the two meshes can be considered as 3D edges: the highlighted line in Figure 2.2(a).

However, some places on the ground were also mis-extracted as edges, and because some edges on the wall were too thin or too sparse to discriminate, they can not be recognized easily. In order to correctly extract edges and to ensure the corresponding pair, we need to do the edge expansion process as Figure 2.2(b).

We do not expand the edges on the surface of meshes directly. In the first step, we mapped the edges onto a sphere surface. Then we split the surface of sphere into regular icosahedron, and subdivided them into 5,242,880 triangle patches Figure 2.3, find the nearby pair patches

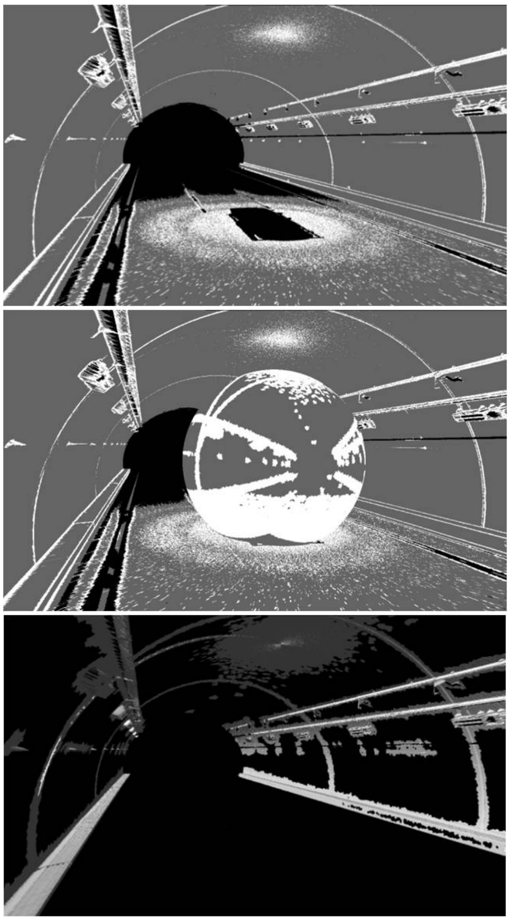

Figure 2.2: (a)Edge Detection (b)Edge Mapping (c)Remap Edges

mapped with edges.

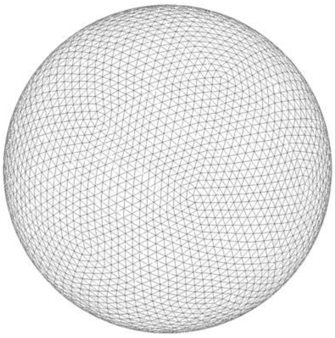

Figure 2.3: Virtual Sphere

Lastly, the expanded surrounding patches of edge should be remapped back to the original data. This process is done by the rendering function of OpenGL, which can be finished in a high speed. Note that the data directly under the sensor has high density, even if small difference occurs when surveying, the direction of normal line will also be changed, then the detection of edge will also become error. In order to avoid this error, we calculated the distance between road and eliminated the road if it is under a certain threshold. After these steps only edges were extracted and the data size also has been reduced to around 10 percent of original data. Figure 2.2(c) shows the result feature after edge extraction.

After edge extraction, the most obvious thing is, the source data size was reduce to 10 percent of the original data from 100GB to about 10GB as shown in Figure 2.4. This time we can align the pair tunnel in the next section easily and speedy.

2.3 Edge Based Alignment

The range image taken by range sensor contains the distance information between object and laser sensor, so the origin of coordinate is at

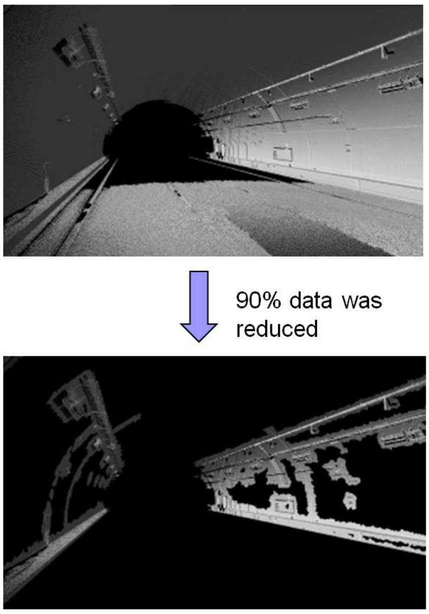

Figure 2.4: Data Size Reduced Obviously

the sensor position. In order to express the aligned multiple data, the coordinate relationship must be calculated and all the data coordinate should be transformed into a same coordinate system. To express all of the data into a same coordinate system, it is necessary to know the position of range sensor.

We used the fast simultaneous 3D-3D matching algorithm for the alignment that we have ever proposed in [9] [10]. Certainly we can find same features and corresponding points in the adjacent data for alignment feature. By doing this alignment work repetitively, we can get a whole tunnel in the same coordinate system. The procedure of the matching method is listed below.

- Set the nearby two range images as a pair.
  - Search the corresponding point of all peak point.
  - Calculate the difference between the corresponding point.

- In order to minimize the difference, calculate the rotation and translation between the range images.

- Apply the rotation and translation to each range image.

Below function means to minimize the Euclidean distance between the corresponding points. R means rotation matrix, $\overrightarrow{\mathrm{T}}$ means translation vector, $\overrightarrow{\mathrm{x_i}}$ and $\overrightarrow{\mathrm{y_i}}$ means points on each range image. The appropriate R and $\overrightarrow{\mathrm{T}}$ value can be solved by iterative numeric computation.

$$\min_{R,T} \sum_i |R\overrightarrow{\mathrm{x_i}} + \overrightarrow{\mathrm{T}} - \overrightarrow{\mathrm{y_i}}| \tag{2.1}$$

Figure 2.5 shows the schema of edge based alignment, we can see the pair tunnel were aligned well without apparent dislocation. By using this kind of method repetitively, we aligned all of the 140 scans into an integrated tunnel with a local alignment coordinate system.

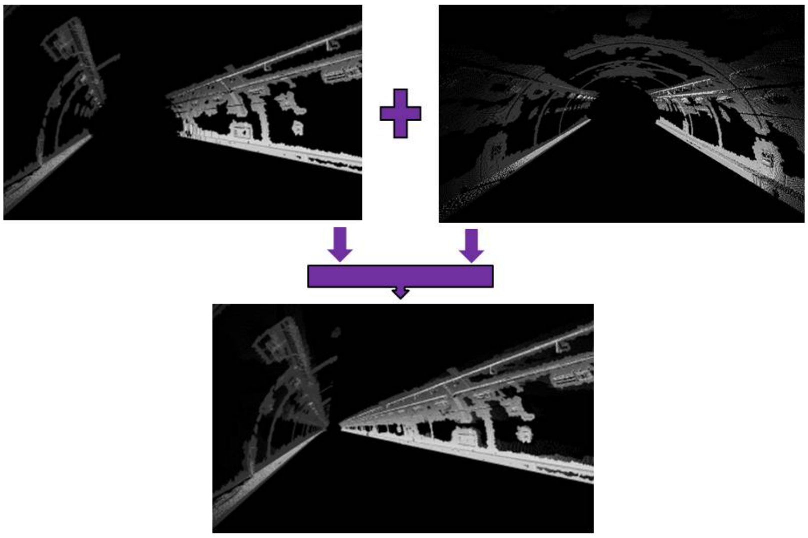

Figure 2.5: Edge Based Alignment

Chapter 3

Global Alignment

After the edge based alignment, we obtained a locally accurate raw tunnel model including the accumulated errors. We can see the aligned tunnel shape is slightly different with 2D CAD blueprint as shown in Figure 3.1. So we simultaneously aligned the tunnel with fused GPS data.

3.1 Simultaneous Alignment

3.1.1 Traditional Method

The traditional pair-wish alignment such as ICP [15] can align two range images at the same time, when there are more than two range images need to align, by using ICP, it can align each pair of pictures, do this pair by pair until all the range images have been aligned. But there is a problem, by using this kind of method, errors occur in pair-wise alignment and will be accumulated along with the increasing of image numbers. To avoid this problem, the most traditional method is, try to estimate the relative position relationship of all of the range images, and then align all of the range images simultaneously. Neugebauer try to search the corresponding points along view line direction, and find the distance of sum of squares between point to mesh from all of the range images, then regard this result as error, linearize this least squares method problem and get the final result [18]. Benjamaa proposed a method based on what Bergevin proposed [19], he separate the vertex along the normal line direction, and use multi-buffer for high speed pair-wise alignment [20]. Nishino used the M estimation and proposed a robust simultaneous alignment [21].

Though there are already many kinds of methods for simultaneous alignment, all of them have a common problem, the computational complexity is too huge. The ICP algorithm tends to search the closest point from all vertexes, suppose there are totally N vertexes in both range images, the computational complexity of corresponding point searching is $O(N^2)$. To speed up the searching, kd-tree method also has been proposed [37], in addition, to reduce the searching range, make nearby points as pre-cache based on kd-tree method also being proposed by D.Simon [23]. However, all the kd-tree based method has a computational complexity of $O(NlogN)$, it is still too huge for computing. A method has been proposed using multiple resolution and closest points searching, this method has a $O(N)$ level computational complexity [24], it is better than all the methods we mentioned previously, but it has a low speed of convergence, and it is difficult to use multi-resolution for huge amount of range images, which hold different resolutions. Another alignment method which has a $O(N)$ computational complexity is proposed by Blais as we mentioned above [17], the feature is to project points along view line direction, but it also has an obvious problem, that the parameters for projection must be calculated for each sensor or for each scan, it will take such long time and will also generate additional complexity for computing.

3.1.2 Adopted Simultaneous Alignment Method

After the edge-based alignment the ”locally accurate” raw tunnel model can be obtained, including the accumulated errors in global. Although [10] [11] can originally provide solution for the global optimization, the simultaneous alignment, it requires so-called loop-closure constraint to work properly, i.e. the model data at one side and the other side should have overlap and can be aligned as a pair, which is not the case of a tunnel. One idea for this problem is to give absolute position and posture to the data at both ends, and then apply the simultaneous alignment method. By using this idea, we aligned all of the range images of tunnel to one same coordinate system.

Figure 3.1(a) shows the top view of modeled tunnel after global alignment by using pair-wise method in last chapter, Figure 3.1(b) shows the top view of 2D CAD tunnel blueprint data which received from Central-NEXCO. Apparently, the modeled tunnel after local alignment is quite different with the CAD blueprint, the most possible reason is the accumulated errors. To correct the accumulated errors, we used simultaneous method to align the tunnel.

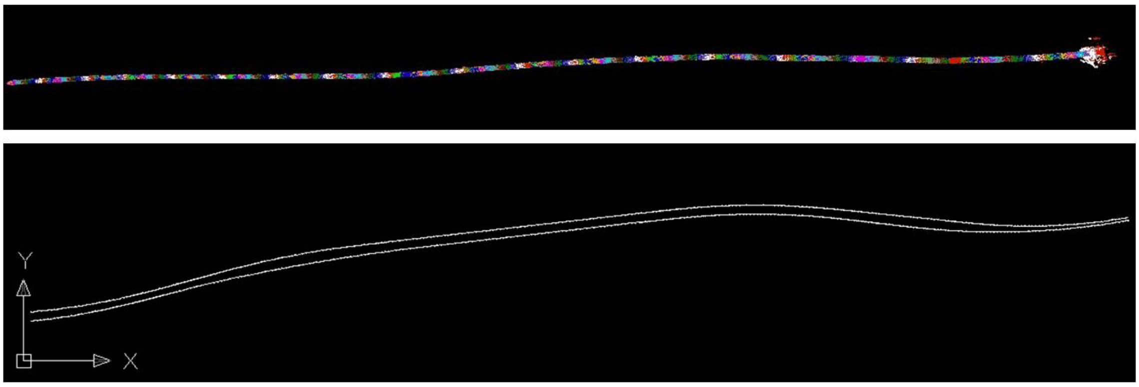

Figure 3.1: (a)Tunnel after Local Alignment (b)2D CAD Tunnel Blueprint

Fist we fix the two ends of tunnel, make it immovable, then we use our method to find the corresponding points on each pair of tunnel segment, different with traditional pair-wish such as ICP, we do not change the corresponding points to minimize the distance, we regard the corresponding points as the right points, then search the nearest distance between corresponding pair, and the most proper position of the tunnel, this method can not accumulate errors because we distribute errors in all tunnel segments when doing alignment. The schema for our method is shown in Figure 3.2, A to R represent each segment of the tunnel, exit and entrance are fixed.

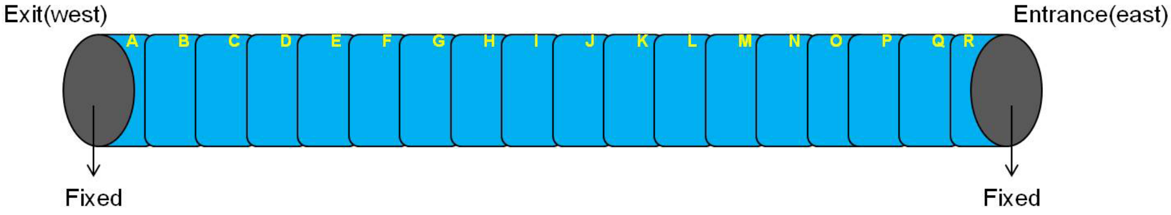

Figure 3.2: Simultaneous Alignment Schema

The basic idea of our alignment method is similar with ICP, which defines an error function, and estimate all of the range image positions, the main procedure is described as below.

- Implement the below calculation to all of the range images.
  - Search all of the corresponding points in B for all vertexes in A as Figure 3.3.
  - Calculate the errors between all corresponding points.
- To minimize the errors after calculation, calculate the transformation matrix.
- Loop first and second step until we get an acceptable result.

The corresponding points in the pair segments were searched before alignment in Figure 3.3, we did not used the closest point searching for its huge computational complexity, we adopted the mesh projection along view line direction method. We selected A,B as the pair segments, make vertexes a,b,c,d,e,f,g in A as original points, when we want to find the corresponding points in B, we make a virtual line of laser range sensor view and the vertex, extend the virtual line to the mesh in B, the intersection points a,b,c,d,e,f,g are the corresponding points we want.

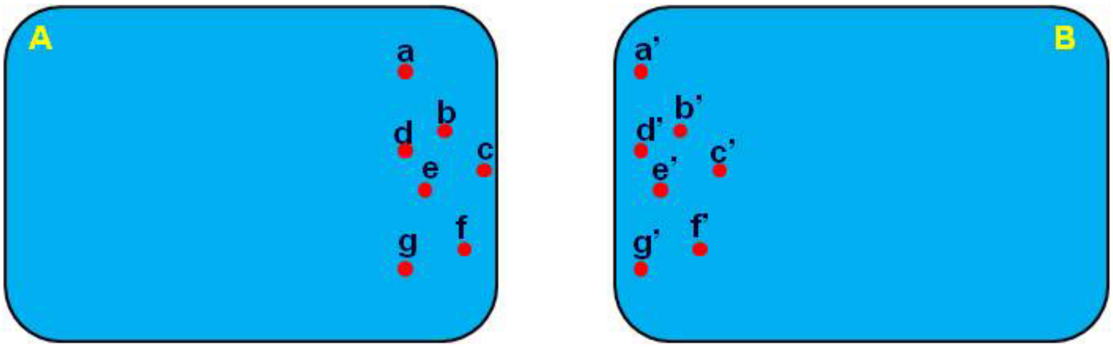

Figure 3.3: Corresponding Points in Pair Segment

To evaluate the errors during alignment, we assume vertex x in image A, y in image B, n_x is the normal line of x, n_y is the normal line of y, the error evaluation equation can be defined as follows

$$R_M n \cdot \{(R_s y + t_s) - (R_M x + t_M)\} \tag{3.1}$$

$$n = \frac{n_x + ny}{\left\| n_x + n_y \right\|} \tag{3.2}$$

Here, R_m, t_m is the rotation matrix and translation matrix of image A, R_s,t_s is the rotation matrix and translation matrix of image B. Extended to all the vertexes in image A and the corresponding points in image B, we can get the equation of square error ε^2.

$$\varepsilon^2 = \min_{R,t} \sum_{i \neq j,k} \{R_M n \cdot (R_s y + t_s) - (R_M x + t_M)\}^2 \tag{3.3}$$

The rotation matrix R and translation matrix t is defined as follows.

$$R = \begin{pmatrix} 1 & -c_3 & c_2 \\ c_3 & 1 & -c_1 \\ -c_2 & c_1 & 1 \end{pmatrix} \tag{3.4}$$

$$t = \begin{pmatrix} t_x & t_y & t_z \end{pmatrix} \tag{3.5}$$

Because Equation (3.3) is not linear, it is difficult to calculate, we linearize (3) to the equation as below.

$$\varepsilon^2 = \min_{\delta} \sum_{i \neq j,k} \left\| A_{ijk}\delta - s_{ijk} \right\|^2 \tag{3.6}$$

For the calculation of transformation matrix, we can calculate d based on method of least square in Equation (3.6). We can get the result of *d* by calculating the linear functions.

$$(\sum_{i \neq j,k} A_{ijk}^T A_{ijk})\delta = \sum_{i \neq j,k} A_{ijk}^T s_{ijk} \tag{3.7}$$

If we aligned the tunnel without fixing the ends, the final aligned result will become distort in all possible directions, for example, if we aligned the tunnel from west to east, the errors will be accumulated and the final result will also distort as Figure 3.4.

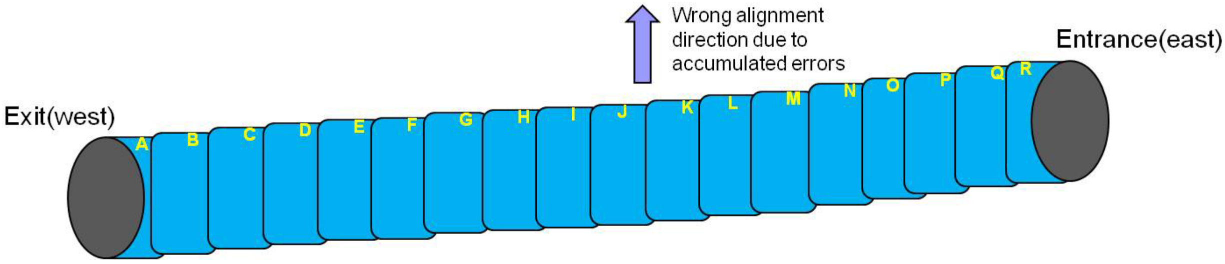

Figure 3.4: Wrong Alignment Result Due to Accumulated Errors

The alignment works took us 51 times, since we fitted the two ends and do the alignment in the middle part of tunnel, we can assume that the alignment scene is like a snake wringing its body but keep head and tail stable, its purpose is to find a most acceptable posture. So to our tunnel object, we can see the alignment procedure in Figure 3.5. Which is from a tunnel exit view line direction, if we can link all of the 51 pictures into one movie and speed up the view speed, we can see the tunnel end keep stable, and the body part keep wringing to find a most proper posture.

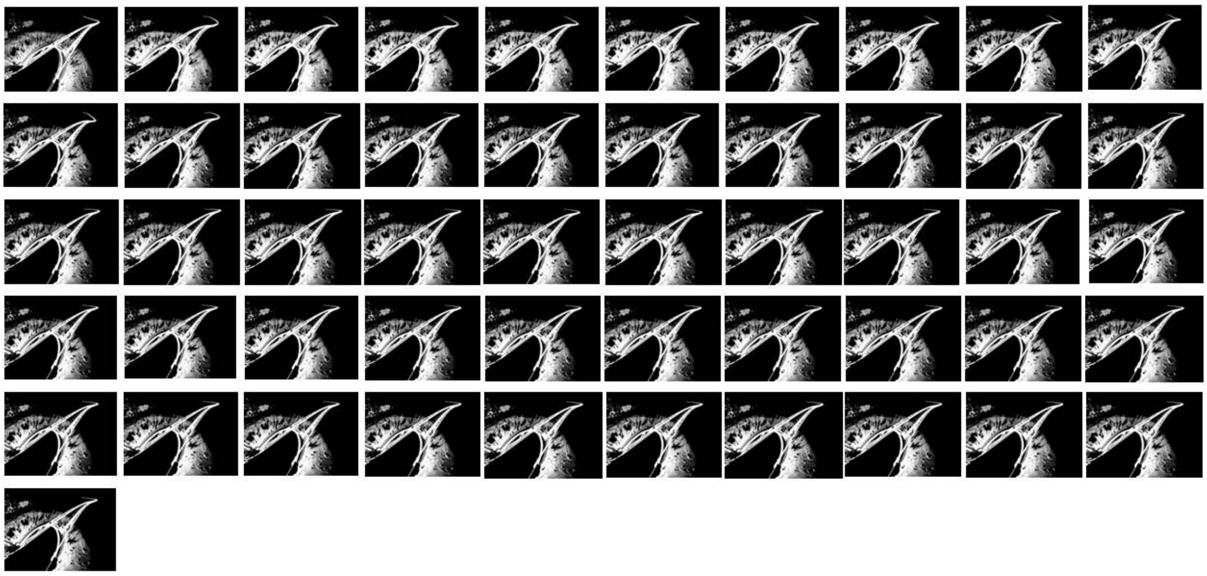

Figure 3.5: Simultaneous Alignment Process

Figure 3.6(a) is the initial posture of tunnel, and Figure 3.6(b) is the final posture of tunnel, it can be seen that the tunnel shape was changed obviously after simultaneous alignment.

3.2 Fusing GPS Data

For the absolute localization, we use high-accuracy GPS in this case. We put the GPS antenna at multiple places in both ends of the tunnel by turns, and record the positional information. At this point, the both ends of the model are fixed in absolute positions. Keeping

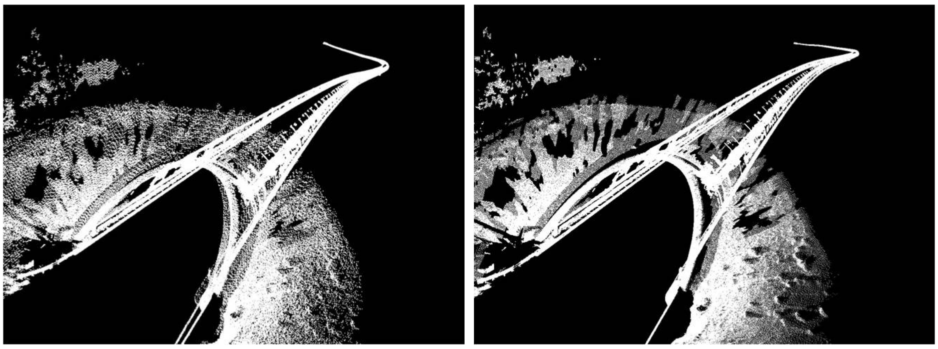

Figure 3.6: (a)Simultaneous Alignment Initial Posture (b)Simultaneous Alignment Final Posture

this constraint, we applied the simultaneous alignment and can get the model that can be considered as geometrically optimal in global. By stablizing the two ends, we minimized the errors through the alignment from all range images of the tunnel. The schema of setting GPS antenna is shown in Figure 3.7.

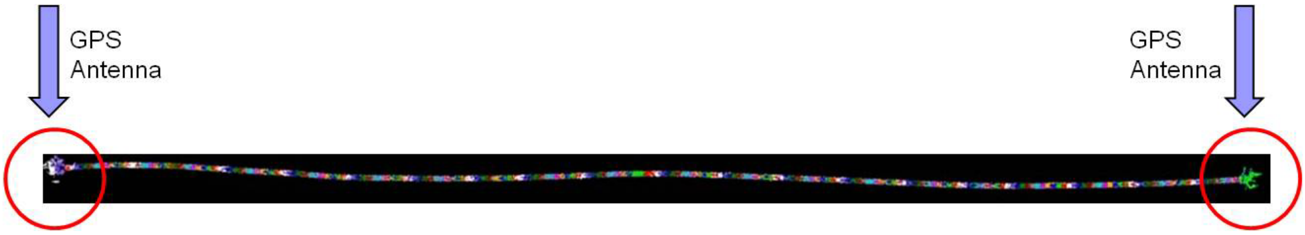

Figure 3.7: Set GPS Antennas in both Ends of Tunnel

For fusing GPS data with locally aligned data, we used Nikon Trimble GPS5700 [33]. We put the disc-shaped antenna at 14 places around both ends of the tunnel. In parallel with this, we scanned the scene geometry including the antenna by the Z+F Imager, and align the existing local-aligned data to the scanned data. Here, in order to find the position of GPS antenna in the range image which can not be clearly extracted because of the scanning resolution, we matched a simple 3D model of the antenna created by scanning the antenna with another precise sensor, VIVID 9i [34], in advance.

We adopted the pre-mentioned method to simultaneously align the raw tunnel model and reduce the error. Key point of this idea can be described as below.

- Set GPS antenna and get position information of both ends of tunnel.

- Use GPS information to correct tunnel coordinate and align the tunnel ends separately.

- Do simultaneous alignment under the condition that both ends are fixed After these three steps, we obtained the tunnel model with globally optimal geometry.

Figure 3.8 is the final result after simultaneous alignment from a top view, we can compare this graph with the CAD blueprint in Figure 3.1(b), we can see the aligned tunnel shape became similar with the CAD blueprint, which means we got a good result and the errors through alignment was distributed into every section of tunnel

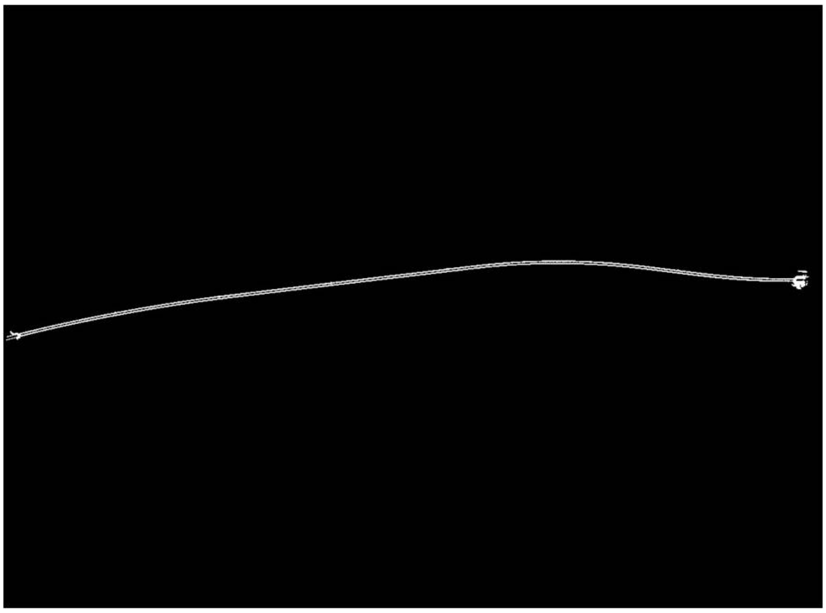

Figure 3.8: Global Alignment Final Result

Chapter 4

Modeling Experiment

For the target of 3D modeling, we selected Kanaya tunnel in Shin-Tomei Expressway, whose length is 4.6 km. Under the permission of the road administrator (Central Nippon Expressway Co. Ltd.) [39], we could make the experiment without other vehicles although the Shin-Tomei Expressway is not opened to public at the present moment.

For on-vehicle laser range scanner, we used an omni-directional laser sensor named Z+F Imager 5003. This sensor can make a scan for all directions, using phase difference detection method to get the data.

We proposed a ”Stop and Go” method to take the range images by the laser sensor . Then we got a huge amount of source data over 100GB. In the edge extraction step, we used 20 degree for the angle threshold value for edge extraction. After noise elimination and edge extraction, the data size reduced to about 10 GB. For fusing GPS data with locally aligned data, we scanned the scene geometry of two ends of the tunnel, where the GPS antenna is scanned together with the scene. The scanning at the ends of the tunnel is made for more than 10 times, changing the position of the GPS antenna. Lastly, we filled the hole on the ground where the survey car existing.

4.1 Survey Car System

We scanned the tunnel by a laser range sensor mounted on the top of a survey car as Figure 4.1, which we designed and assembled for scanning the urban scenery. Generally, the survey car can not only take range image by laser sensor, we also mounted a GPS and a Gyroscope for positioning, an omni direction camera for 360 degree image taking, these devices can be used in different situations when survey.

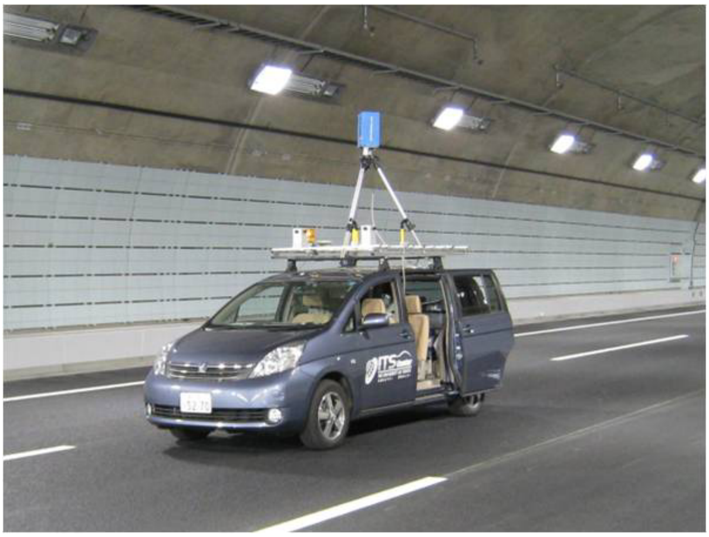

Figure 4.1: Survey Car We Used

The Z+F Imager 5003 [12] is the laser sensor we used this time for our experiment, it is a product of Swiss company named *Zoller + Frohlich*2, an image of this laser sensor is shown in Figure 4.2.

The sensor can scan 360 degree scenery by rotating both on vertical direction and horizontal direction. The specification detail of this scanner is shown in Table 4.1.

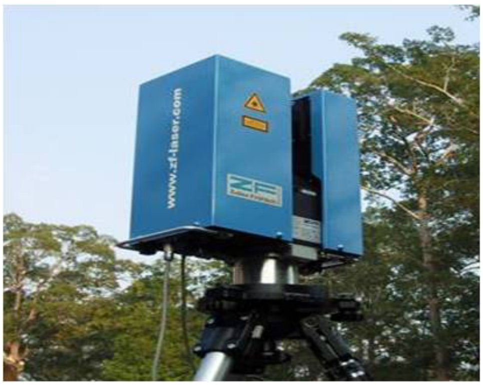

Figure 4.2: Z+F Imager 5003

Table 4.1: Specification of Laser Range Sensor Z+F Imager 5003

Ambiguity interval	53.5 m
Min. range	1.9 m
Resolution Range 16 Bit	1.0 mm/ lsb
Typical data acquisition rate	125000 $pixel / sec$
Linearity error	5 mm
Field of View (vertical, horizontal)	$310^\circ, 360^\circ$
Resolution(vertical, horizontal)	$0.018\ ^\circ, 0.01^\circ$
Accuracy(vertical, horizontal)	0.02 $^\circ$rms, 0.02°rms
Typ.scanning speed vertical	1500 rpm
Scanning time(middle mode)	100sec
Dimension	$30 \times 18 \times 50$cm
Weight	16 kg

4.2 Overview of Target Tunnel

The target tunnel for our research this time named *Shin − Tomei KanayaTunnel*, which is a part of Shin-Tomei Expressway [39], the expressway is located between the Shimata City and Kawagawa City of Shizuoka Ken. The Tomei Expressway is the provincial level expressway between Tokyo and Nagoya, it contains two traffic lanes, including the up-bound lane from Nagoya to Tokyo, and the down-bound lane from Tokyo to Nagoya. Generally, people called the down-bound expressway as Tomei Expressway, the up-bound expressway is called the Shin-Tomei Expressway. As you can see in Figure 4.3 the red line and black line stands for the Shin-Tomei Expressway and Tomei Expressway respectively. This time the object tunnel we experimented and modeled was a part of the Shin-Tomei Expressway. It has a length of 4.6km.

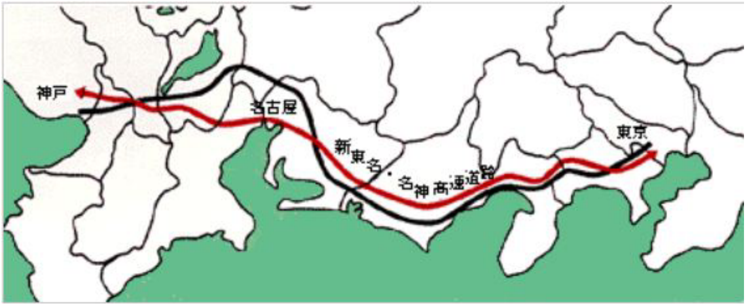

Figure 4.3: Shintomei Expressway Overview

The Kanaya Tunnel is located in the middle part of the Shin-Tomei Expressway [40]. It is a two-way tunnel the same as Shin-Tomei Expressway as you can see in Figure 4.4. In this paper we used the below one of Kanaya Tunnel for our modeling object. This tunnel has an inner diameter of 16.89m, the thickness of tunnel wall of each boundary is 0.5m.

The actual tunnel view can been seen below, Figure 4.5(a) is the entrance part of Kanaya Tunnel, Figure 4.5(b) is the exit part of Kanaya

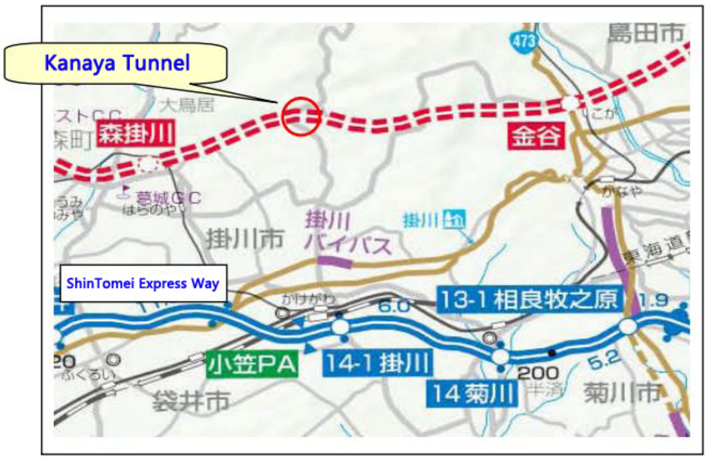

Figure 4.4: Kanaya Tunnel in Shintomei Expressway

Tunnel, Figure 4.5(c) is the scenery inside the tunnel.

4.3 Acquisition of Tunnel Data

The surveying activity took 6 hours/km. It took us 4 days to scan the tunnel. Finally, we obtained 240 range images with the total size in about 100 GB in the Standard PLY format [32].

In this paper, we proposed a method to survey the tunnel which we called ”Stop and Go” method. GPS Antenna based positioning can not be used inside the tunnel because of the thickness of the tunnel wall, the most general fungible device for GPS positioning is gyroscope, it can also positioning along with car driving. So why don’t we use the gyroscope as positioning device along with car driving? After taken the data we can use the position information to build the tunnel model directly, that is simply. There are two reasons why we didn’t use gyroscope as positioning device.

The first reason is the positioning precision problem, as we can see in Figure 4.6, outside the tunnel we can use a GPS Antenna as the positioning device, it has a good position precision since the GPS can

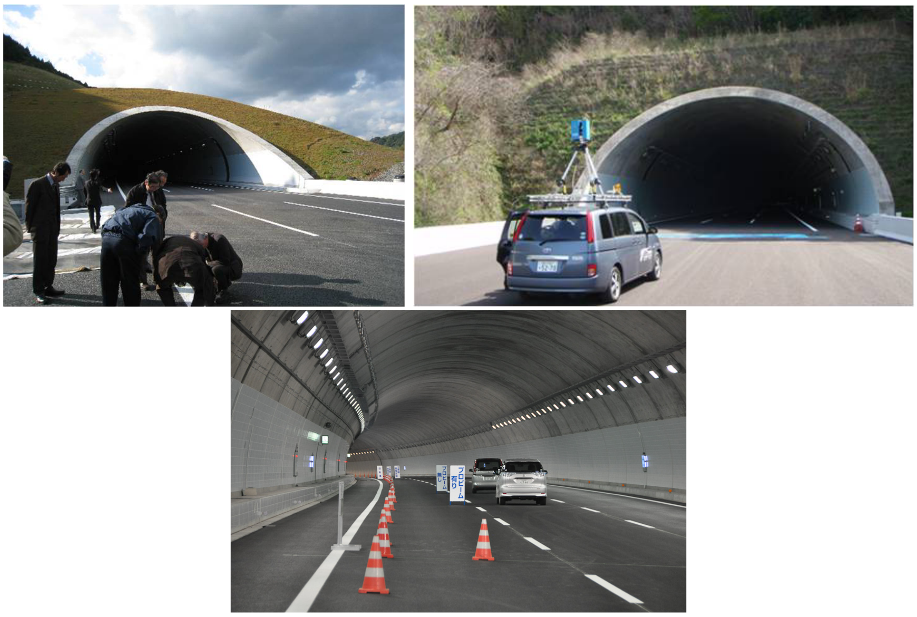

Figure 4.5: (a)Outside View near Entrance (b)Outside View near Exit (c)Inside View

correct it is position by using the satellites, but inside the tunnel things changed when we are using the gyroscope as the positioning device, because there is a angle difference d between the two moving directions after some distances, a slight position difference occurs, the gyroscope can not correct this position difference as GPS, so the position difference will become bigger and bigger [1].

The second reason is, generally it takes several minutes for making areal scan with enough resolution by current technology, when we take the range images along with car driving, though we can get a continuous modeled tunnel after not very long time, but when we check the range images contained in the model of tunnel, we can find the resolution of image is too low, in some places the points can even not express the details of the tunnel wall correctly.

As the two reasons we explained above, we proposed the ”Stop and Go” method to survey the tunnel. This method can be described

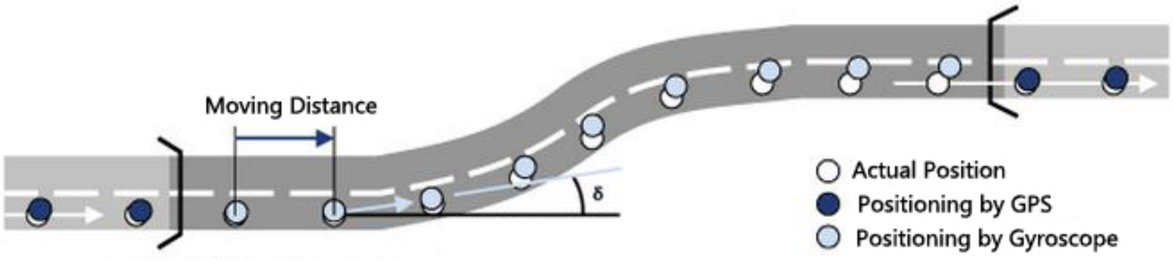

Figure 4.6: Positioning Comparison of GPS and Gryoscope

as follows: The scan is done with the vehicle stationary. In order to survey the whole tunnel, we need to move and survey multiple times. Keeping enough overlap to the next scan, the vehicle moves and stops repeatedly (approximately every 20 meters in the case of the experiment this time). The schema of our method is shown in Figure 4.7.

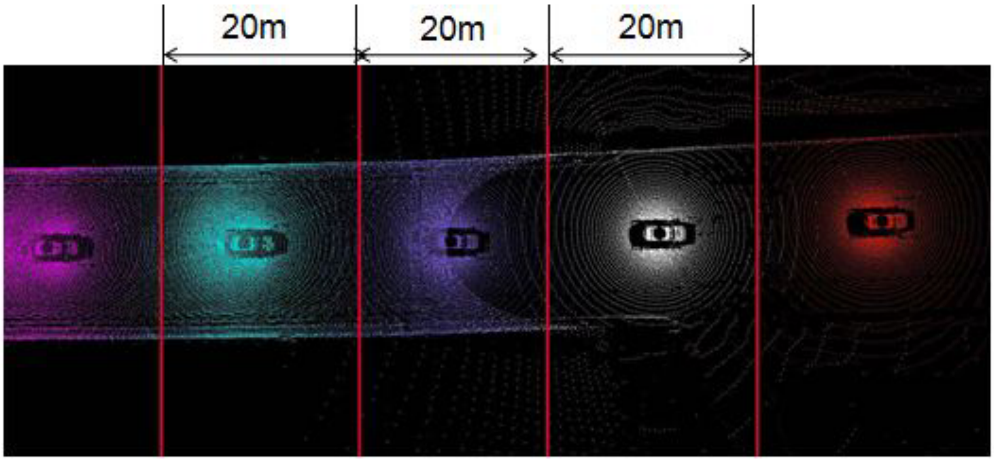

Figure 4.7: Positioning Comparison of GPS and Gryoscope

An example of the tunnel image taken by the laser sensor is shown in Figure 4.8. Here the survey car is also considered as noise, which can be simply eliminated by thresholding the distance from the laser sensor. The blue rectangle is the origin point of tunnel coordinate, it is also the position where the range sensor was placed.

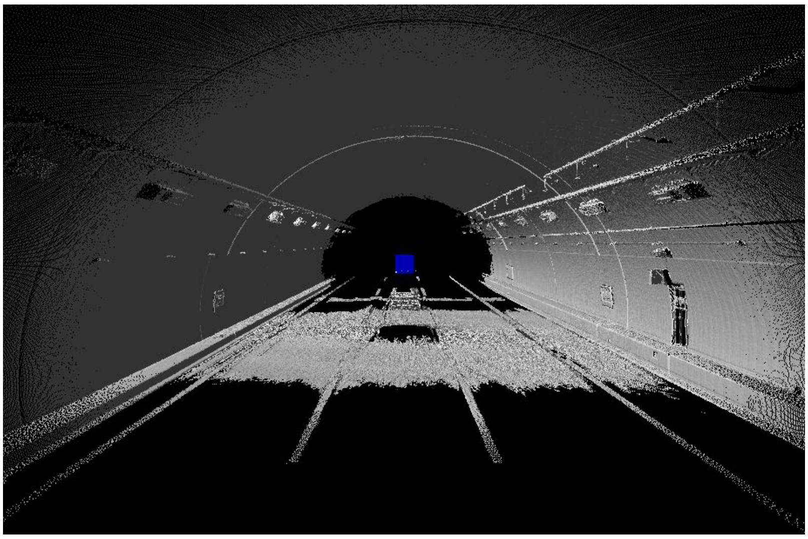

Figure 4.8: An Example of Scanned Kanaya Tunnel

4.4 Processing of GPS Antenna Data

For simultaneous alignment in Chapter 3 and coordinate transformation work in Chapter 5, we used the GPS antenna information as the basic reference.

To the simultaneous alignment, as we explained in Chapter 3, we set GPS antenna in both edges of the tunnel, recorded the latitude and longitude value, and did the simultaneous alignment by fitting two edges using GPS information.

To the coordinate transformation, we proposed a method to align the GPS antenna model with the antennas in tunnel edges. So the first step is to build an acceptable GPS antenna model.

Fist we scanned the GPS Antenna we used in tunnel model for 14 times from different point of view as Figure 4.9 shown. The laser scanner we used this time is *VIVID9i*, a non-contact 3D Digitizer made by *KONICA MINOLTA*

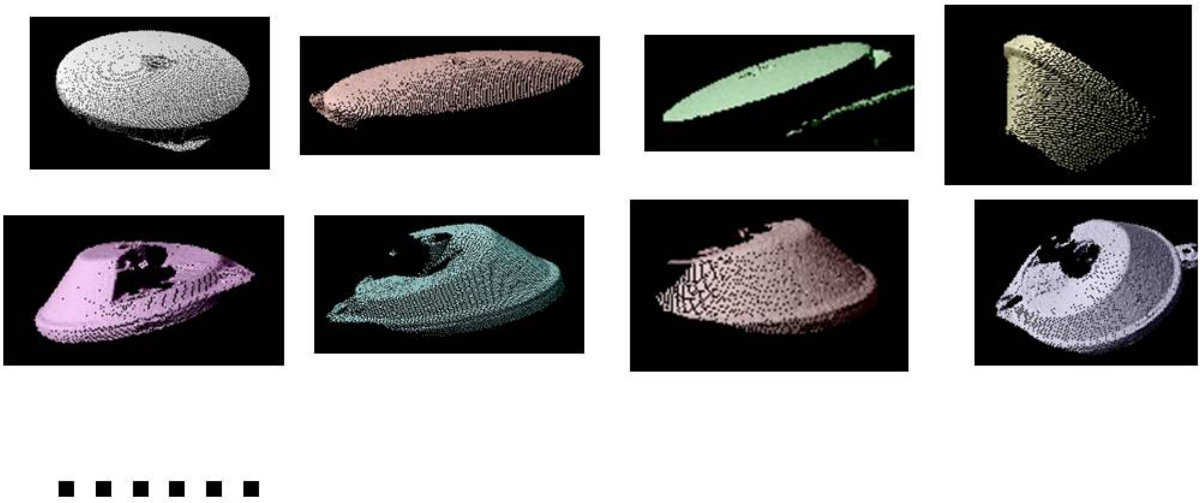

Figure 4.9: Scanned GPS Antennas from Different Point of View

Next we aligned these scanned GPS Antenna fragments into an integrated Antenna model with full details from different angles as Figure 4.10.

After alignment, the GPS antenna model can be utilized for the coordinate transformation in Chapter 5.

4.5 Filling the Deficient Regions

After we got the scanned tunnel, we found the survey car is also appeared in each segment, as we noticed before the laser sensor is mounted above the survey car, the scanning principle is laser pulse reflectance, so of course the survey scanned along with the tunnel scanning. When we do the local alignment in the chapter2, we considered the survey car as noise, thus before the alignment we must eliminate the car, so a deficient region appeared in each segment of tunnel after final global alignment. Figure 4.11(a) shows the deficient region after car eliminated.

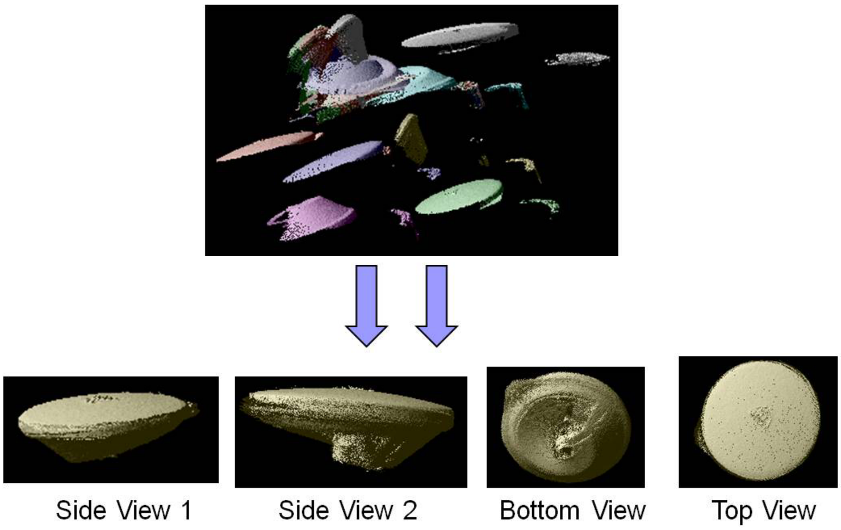

Figure 4.10: Align GPS Antenna Fragments

The final step of our work after global alignment is to build the perfect model of Kanaya Tunnel, so we must fill the deficient region after car elimination, besides, some other reasons described below caused the existence of deficiency when scanning, we proposed a method to solve these problem.

- Road directly under the vehicle can not scanned by laser sensor.
- Some parts of the road surface do not reflect laser enough. The deficiency regions mostly correspond to road, not just regular plane. So the regions around the deficiency parts region also need to be considered, gradient and curvature should also be interpolated smoothly. For filling the region, we re-sampled the scanned data so that we can assume a top-view depth image $\Omega(u, v)$ above the deficient area, here each pixel contains the distance to the surface $z(u, v)$. This depth image can be speedily generated by rendering function of OpenGL. In order to fill the deficiency in the whole image, we calculated z that minimize the cost function

$$\int_{\Omega}\left[\alpha(z-D)^2+(1-\alpha)\left[\frac{\partial^2 z}{\partial u^2}+\frac{\partial^2 z}{\partial v^2}\right]\right]d\Omega \tag{4.1}$$

$D(u,v)$ is the actual depth value by the scanning, and is a binary parameter representing that the scanned data exist (1) or not (0). The first term in the integral let z be near to the scanned data when it exists, and the second term of the integral let z be connected with surrounding data smoothly. The cost function is calculated repeatedly to minimize the value. Figure 4.11(b) shows the result of the filling.

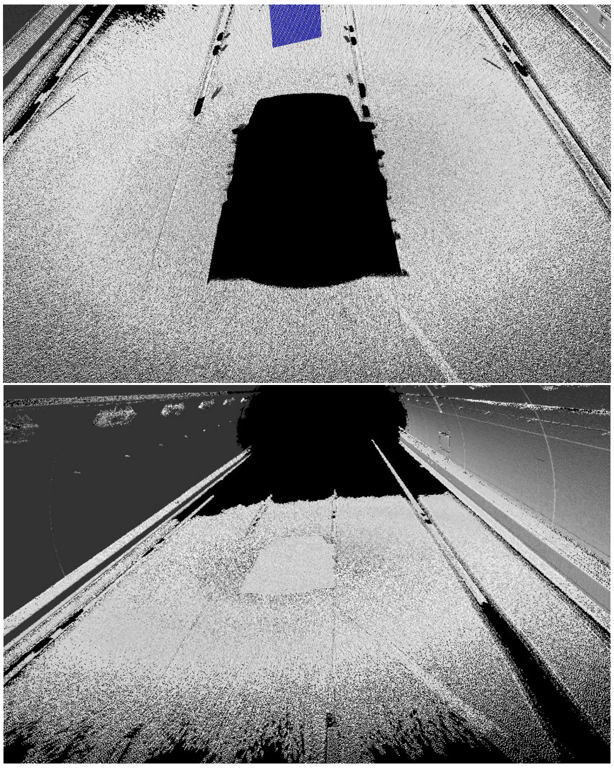

Figure 4.11: (a)Before Filling Deficient Region. (b)After Filling Deficient Region

4.6 Modeling Result

We plotted our modeling result into the Google Maps along with the actual Kanaya Tunnel, we can see our modeled tunnel fitted the actual tunnel well in an overall standpoint as shown in Figure 4.12.

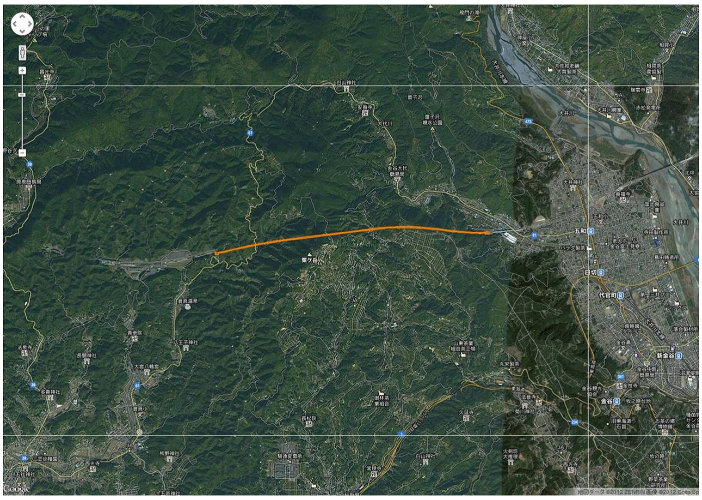

Figure 4.12: Modeled Tunnel in Google Maps

The final modeling result after hole filling is shown in Figure 4.13. The quality of the modeled tunnel is good enough for car navigation system, autonomous driving and so on. From this point of view, we did a good job. In next chapter we roughly compared the modeled tunnel and 2D CAD designing data, we checked the comparison result both from simple shape comparison and quantitative calculation level.

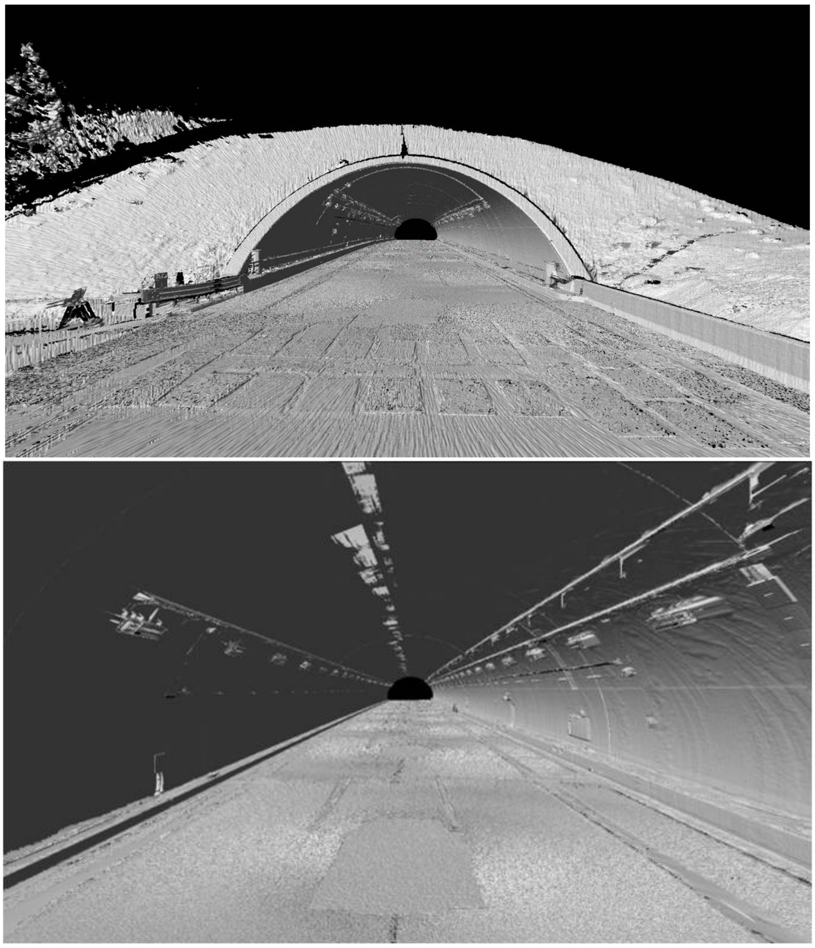

Figure 4.13: (a)Modeling Result Outside View. (b)Modeling Result Inside view

Chapter 5

Comparison and Evaluation

5.1 2D CAD Data

The 2D CAD data was originally received from Nexco(Nippon Expressway Limited), which is an privatization expressway management company. Nexco has three branches: East-Nexco, West-Nexco and Central-Nexco, whom are in charging for the expressway fairs in Eastern Japan, Western Japan and Central Japan respectively. The Central-Nexco designed and managing the Shin Toumei Kanaya Tunnel in Shizuoka of Japan.

The Shin Toumei Kanaya Tunnel CAD data was made up from two paralleled tunnels that we can see in Figure 5.1. As this time we just surveyed and modeled the second tunnel located on the bottom, from now on, the "Kanaya Tunnel CAD Data" represents the bottom tunnel.

The Kanaya Tunnel CAD data has a WGS(World Geodetic System) coordinate system [35] corresponding to the latitude and longitude information of real world. So theoretically we can find every part of the tunnel structure in the real world corresponding to the Kanaya Tunnel CAD data.

There are several layers in the 2D Kanaya Tunnel CAD blueprint,

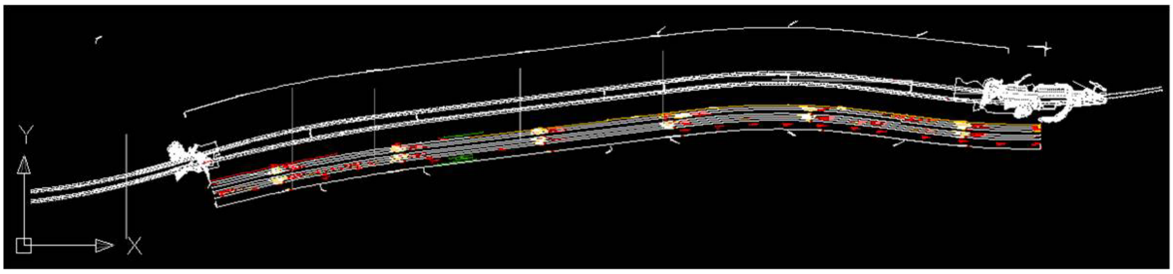

Figure 5.1: Original Kanaya Tunnel CAD Blueprint

including the entrance and exit bridge layer, the center line of tunnel layer , emergency devices layer, marking data layer, etc. Because we just want to compare and evaluate the shape between CAD data and Modeled data, we are only interested in the tunnel boundary, so this time we are concentrating the D-TUN layer of CAD date which stands for the boundary of Kanaya Tunnel CAD data Figure 5.2.

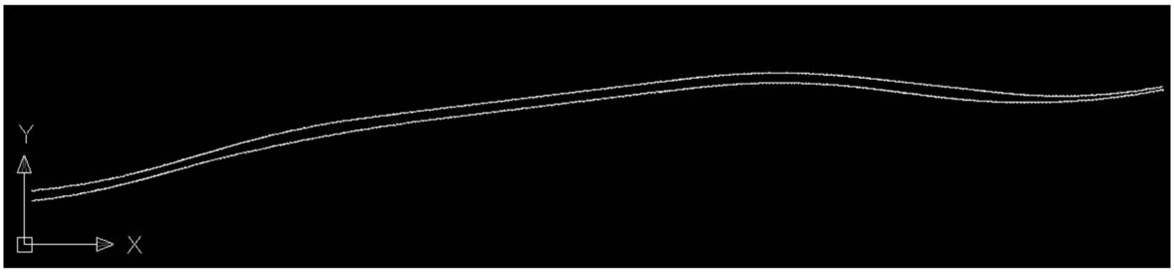

Figure 5.2: Overall View of Kanaya Tunnel CAD Data

Figure 5.3(a)(b) is the enlarged view of Entrance and Exit of Kanaya Tunnel CAD data, it is on the east side and west side geographically in the real world. The tunnel actually if from east to west, but this time we scanned tunnel from exit(west) to entrance(east), which is reverse with the tunnel forwarding direction.

The original Kanaya Tunnel CAD data format is .dwg, it is a standard AutoCAD format. This format includes various kinds of information such as block record, dim style, layer, font style, view method,

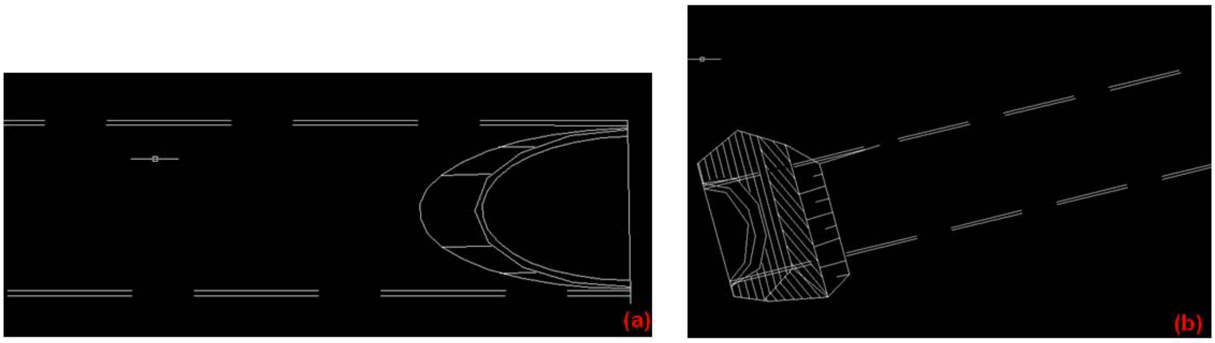

Figure 5.3: (a)Entrance of Kanaya Tunnel CAD Data(east)
(b)Exit of Kanaya Tunnel CAD Data(west)

section, latitude and longitude, etc. As we want to confirm the details of the CAD blueprint to decide which part of information should be exacted, we convert the *.dwg* format to *.dxf* format, which the encode type is ASCII. After we browsed the contents of *.dxf* of CAD data, we made an extraction program and extracted the D-TUN layer.

Theoretically, the CAD data is correct, however, due to some external reasons such as topography errors when building the tunnel, or tunnel designing specification has been changed. Nobody can guarantee that the CAD data is 100 percent correct corresponding with the real tunnel, after all the CAD data is just a designing chart. On the other hand, we can also not say that our modeling result is 100 percent accurate, there may be some accumulated errors during our process when modeling. As this reason, we need to figure out a kind of method to compare the CAD data and our modeled tunnel, and evaluate our modeling result with the CAD blueprint.

5.2 Making Comparison Data

The Comparison and Evaluation work needs two kinds of source data, one is Kanaya Tunnel CAD blueprint received from Central-

Nexco, the other is modeled data after global alignment step, but the two source data can not be compared directly, before that we must do some data handling works.

In Section 5.1 we will introduce that we exacted the boundaries of both Kanaya Tunnel CAD blueprint and Modeled data as the data source of comparison. So in the first part of Section 5.2 we will introduce how to extract the boundaries of Kanaya Tunnel CAD blueprint and Modeled data, in the second part of Second 5.2 we will introduce the coordinate transformation in order to put the two extracted boundaries into a same coordinate system, finally in the last part of Second 5.2 we would like to give a brief introduction about evaluating the errors by using $k - d$ *tree* closest point searching algorithm.

5.2.1 Boundary Extraction

The modeled tunnel after global alignment in Chapter 3 is actually 3D data with the coordinate (x, y, z). Because the Kanaya Tunnel CAD blueprint is 2D data with coordinate (x, y), we must convert the 3D modeled tunnel into 2D data to do the comparison and evaluation. Meanwhile, the 2D CAD blueprint contains several layers including the tunnel we are interested, we also extract the tunnel boundary layer in this section. The boundary extraction schema can be seen as Figure 5.4.

Because the Kanaya Tunnel CAD blueprint has a WGS(World Geodetic System) coordinate system as we mentioned previous, for boundary extraction, we just need to exact the coordinate of boundaries in $.dxf$ file, the D-TUN layer is the boundaries layer of tunnel, we programmed to extract the AcDbLine section from the $.dxf$ file representing the D-TUN layer. Here the AcDbLine section contents the 2D x and y coordinate of Boundaries. By using the Gnuplot tools, we plotted the extracted boundaries in a 2D plane with the same coordinate system as original CAD blueprint Figure 5.5.

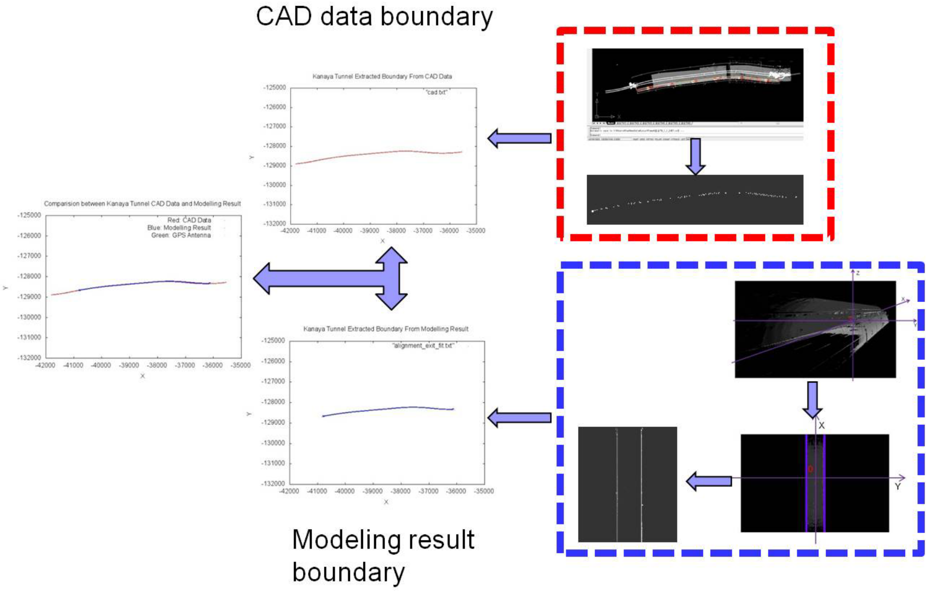
CAD data boundary
Kanaya Tunnel Extracted Boundary From CAD Data
"cad.txt"
Comparision between Kanaya Tunnel CAD Data and Modelling Result
Red: CAD Data
Blue: Modelling Result
Green: GPS Antenna
Kanaya Tunnel Extracted Boundary From Modelling Result
"alignment_ext_fit.txt"
X
Y
Z
Modeling result boundary

Figure 5.4: Boundary Extraction Procedure

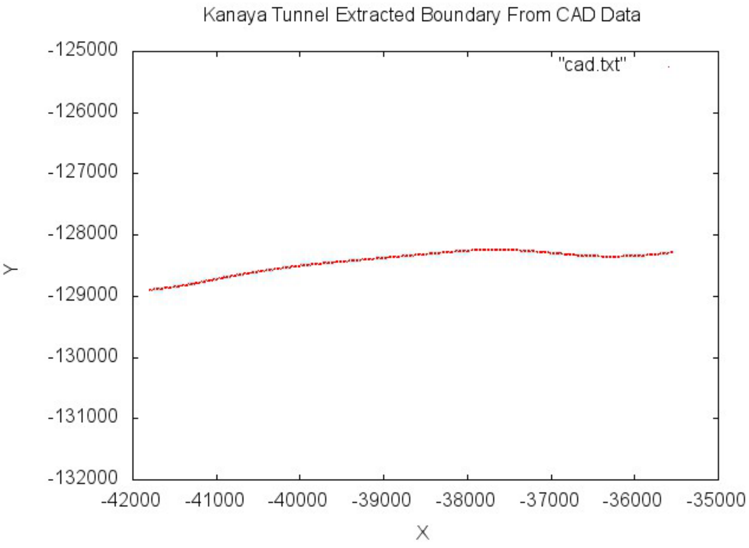

Figure 5.5: Kanaya Tunnel Extracted Boundary from CAD Data

To do the 3D-2D conversion, firstly we made a coordinate definition for our 3D extracted tunnel. The forwarding direction of tunnel is x coordinate, the diameter direction of the tunnel is y coordinate, from ground to the roof of the tunnel is z coordinate as shown in Figure 5.6.

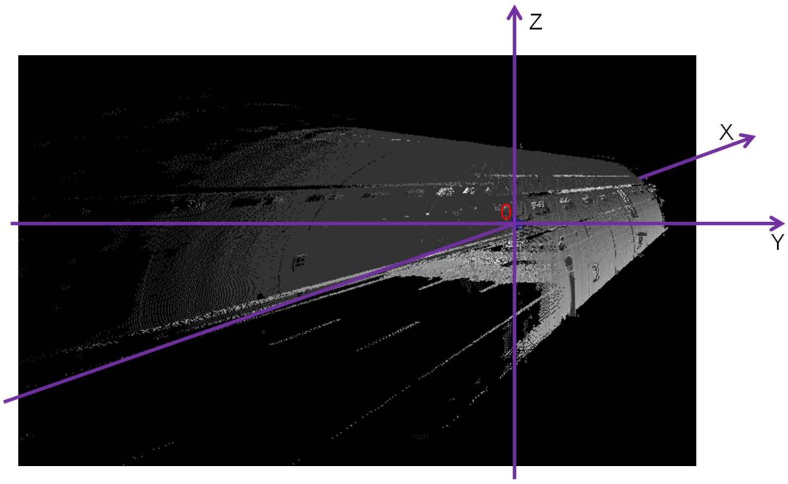

Figure 5.6: Coordinate System of Kanaya Tunnel(side view)

The blue rectangle zones in Figure 5.7 represents the range sensor mounted above the survey car, and the center of the sensor can be considered as the origin of coordinate as we can see in Figure 5.7, the sensor was about 1.86 meters high from ground. We can see that the diameter line of the tunnel is under the origin line of coordinate.

For extraction of boundaries, the most general thinking is to exact the continuous two boundaries directly from our modeled tunnel after global alignment, however, this method only works when the modeled tunnel is on a flat horizontal ground, the actual topography where tunnel was built on is not a standard flat horizontal ground, we can see the ground is up and down in the modeled tunnel from a overall stand point, if we still use threshold based method to extract boundaries directly from the model tunnel, because not all of the tunnel sections are on the same horizontal level, we will get two wrong tunnel boundaries as shown in Figure 5.8. We can see when we want to extract boundaries

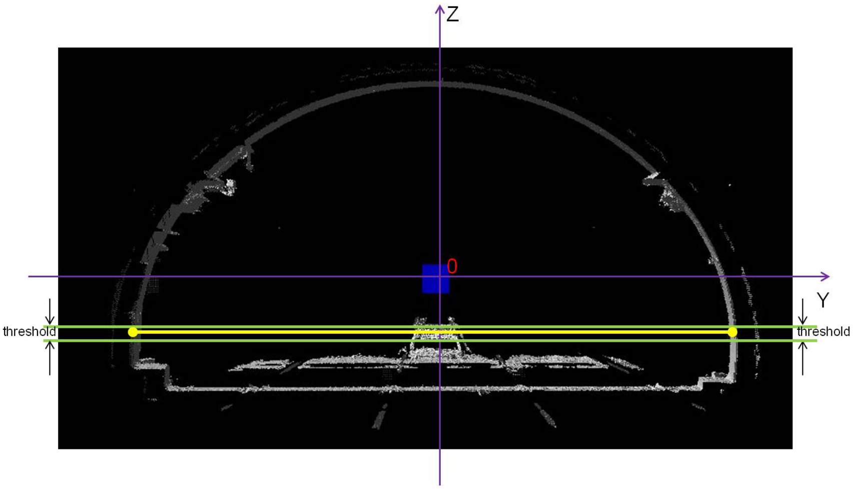

Figure 5.7: Coordinate System of Kanaya Tunnel(cross-section view)

based on the first section of tunnel from west to east, since the tunnel is forwarding to the east, the same level of threshold range will extract different height of tunnel wall, some sections even not being extracted.

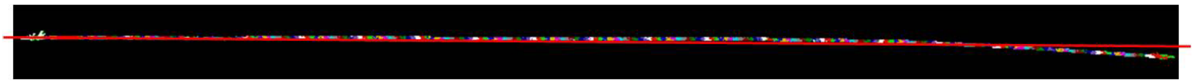

Figure 5.8: Boundaries Extracted Directly from Modeled Tunnel

We also had tried to cut the tunnel into several slices in horizontal direction, and tried to use each slices edges as boundaries for comparison. But after made these slices as raw data of boundary, we realized that the vertical transect of tunnel is more like a regular ellipse, not a rectangle, the two endpoints of the diameter are where the boundaries located in (yellow points in Figure 5.7, so we proposed a method to find the two endpoints of diameter from tunnel walls, and extracted the continuous endpoints as the boundaries of Kanaya Tunnel. We marked the boundaries by blue line under the top-view mode in Figure 5.9.

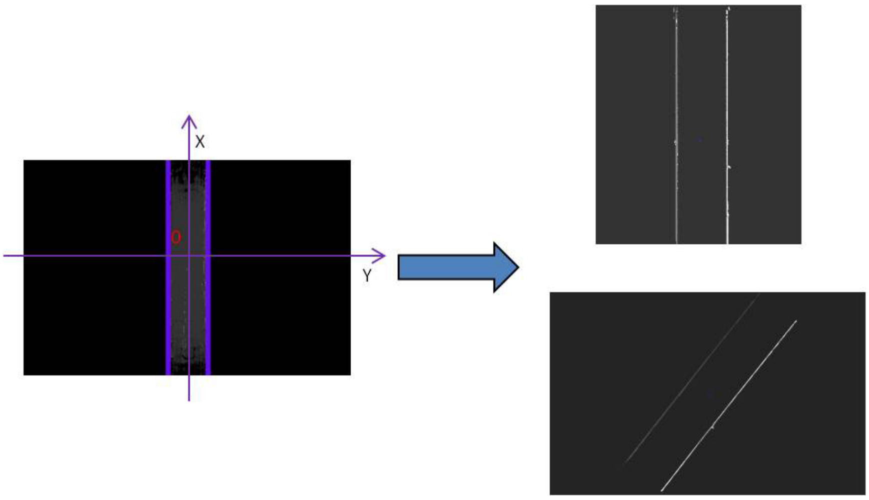

Figure 5.9: Top View of Boundary Extraction Schema

The method we used for boundary exaction from modeled tunnel is threshold based method. First we remove the rest part of the tunnel between boundaries manually (zones between the two blue lines in Figure 5.9. Because we define the forwarding direction of tunnel as x coordinate, and direction of diameter as y direction, so next step, we would find the tunnel wall between a bigger value and smaller value of y coordinate as boundary, and tried to find the two exact y ranges which representing the two boundaries of tunnel. We found that when we selected different z ranges as threshold, we can get different thickness of boundary. We tried many times to find out a proper threshold for boundaries. Finally, by using this method, we found that when z has a range of $(-1.65, -1.60)$, we can get the most proper points stands for the boundaries.

Figure 5.10 is the extracted boundaries from Kanaya Tunnel Modeled data.

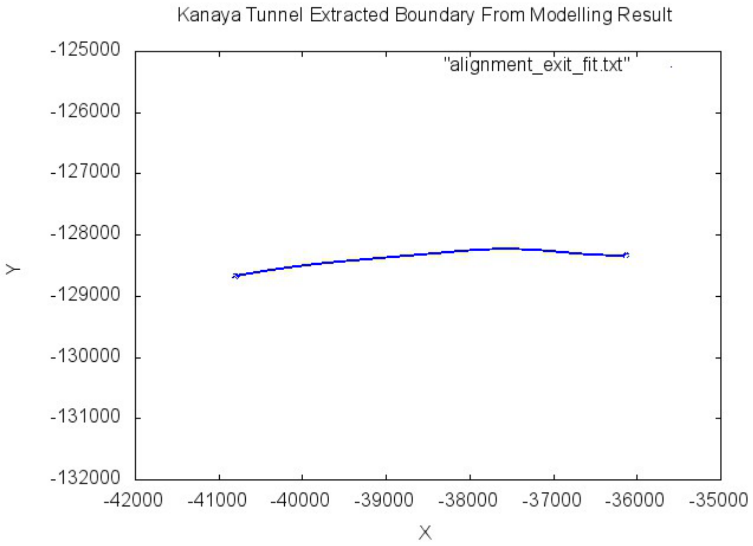

Figure 5.10: Extracted Boundaries of Modeled Data Plotted on 2D Plane

5.2.2 Coordinate Transformation

Actually we have four kinds of coordinate as we explained in the Appendix chapter:

- Original Coordinate(*O Coordinate*)
- Coordinate after Local Alignment(*L Coordinate*)
- Coordinate after Global Alignment(*G Coordinate*)
- WGS(World Geodetic System) Coordinate (*W Coordinate*)

Every piece of the tunnel taken by range sensor was in the *O coordinate*, every of them have their independent origin in the center of where the range sensor was placed.

After pair-wise alignment, all of the tunnel slices were linked together into a whole tunnel, and this time the tunnel itself has a separate coordinate, *L Coordinate*.

As we mentioned in Chapter 3, when we were doing the simultaneous alignment, because the data type of our tunnel points are float. The precision of float is 7 digitals, but as to the WGS, the coordinate points are too long to be represented by float type, the points must be represented by double type or else the result will become incorrect. If we use float to represent the coordinate in WGS, we will lose several digitals after decimal point, also the precision will decrease. But in the other hand, if we try to modify the simultaneous alignment algorithm to use double type, the long digitals of double need huge system memories to do the processing of data. To solve this contradiction, we defined another coordinate related with *W coordinate* but different with *L coordinate*. The first step is to apply an offset value to every point in our tunnel in *L coordinate*, then we do the simultaneous alignment by fitting the exit and entrance stable using GPS data, after that we got a new coordinate, we call this system as *G Coordinate*

The Kanaya Tunnel CAD data also has a coordinate corresponding to the real world latitude and longitude, we call this coordinate as *WCoordinate*. The formal name of this coordinate being used widely is World Geodetic System(WGS) as we mentioned before, which is also recognized as a standard used in geodesy, navigation and cartography, the WGS comprises standard coordinate frame for the earth, a reference surface for raw altitude data and a gravitational equipotential surface that defines the nominal sea level.

Since our method is to extract boundaries with an exact height range in each tunnel section's own coordinate, we proposed this method to extract boundaries from each tunnel section separately. After extracted the boundaries of each section of tunnel, we got the O coordinate based boundaries Figure 5.11. Next we need to do the coordinate transformation work to fit these extracted boundaries of tunnel into an integrated tunnel as the tunnel already aligned.

Next we plan to put our extracted boundaries into the tunnel coordinate the same as the coordinate after Global Alignment. Which we

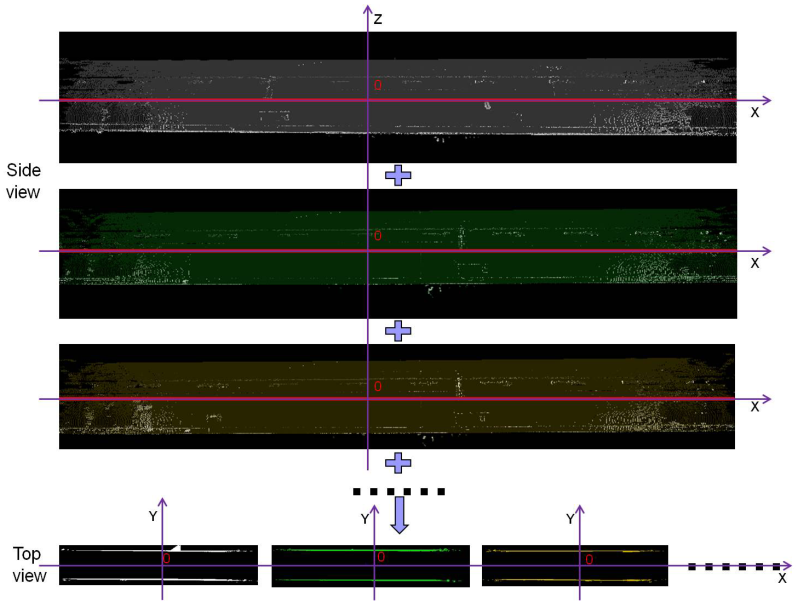

Figure 5.11: Boundary Extraction Schema in O Coordinate

can also considered as *O coordinate* to *G coordinate*. For the coordinate transformation this time, the head matrix information of both aligned tunnel and the extracted boundary data would be used for calculation. The standard data we used representing the tunnel is *.ply* format, this kind of data has a head matrix in the very first four lines of each file which indicating the relationship of position transformation. The principle for transformation can be simply described as follows. First, we read header matrix A, B in the aligned tunnel and not aligned tunnel, and write calculation result of $B^{-1} \cdot A$ to standard output, then we apply this calculation result matrix to each coordinate point in not aligned tunnel data, by applying this method, we can move every point in tunnel data from *O coordinate* to *G coordinate*. Because we stored extracted coordinate points in *.txt* format data, this format data does not have a head matrix, here we define unit matrix as A. The aligned tunnel data is *.ply* format, so we use the head matrix B of the aligned tunnel directly for the calculating formulation. Figure 5.12 is the processing flow chart from original *O coordinate* to *G coordinate* as aligned tunnel.

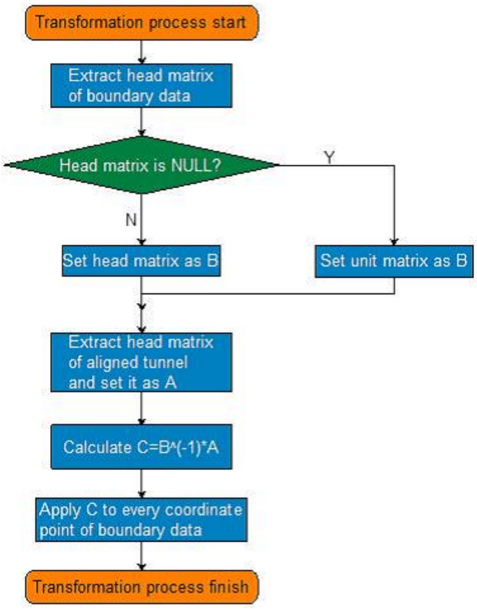
Transformation process start
Extract head matrix of boundary data
Head matrix is NULL?
Y
N
Set head matrix as B
Set unit matrix as B
Extract head matrix of aligned tunnel and set it as A
Calculate C=B^(-1)*A
Apply C to every coordinate point of boundary data
Transformation process finish

Figure 5.12: Transformation Process Flow from O Coordinate to L Coordinate

As we introduced once before, the reason why we defined G coordinate is, the x and y coordinate points in *W coordinate* are too long to be represented by float. In order to do the simultaneous alignment, we did a simple transformation work to shift *W coordinate* to new defined *G coordinate* by applying an offset value to all points of the tunnel. As a result, we also need to do some works to transform our extracted boundaries from *G coordinate* to *W coordinate*. We separated the procedure of preparation and transformation into several main steps:

- GPS Antenna model building.
- Scan tunnel entrance and exit with GPS Antenna placed.
- Align GPS Antenna model with scanned GPS Antenna in tunnel model.

- Calculate the average value of all the points consisting of each aligned GPS Antenna, then calculate the average value of all GPS Antennas.

- By calculating the d-value of result in step four and the coordinate value converted directly from GPS measured data, we can get the offset value for coordinate transformation.

The basic idea for coordinate transformation is to find the GPS Antenna average coordinate value in *G coordinate*, and calculate the difference between this average value and real coordinate value in *W coordinate*. Because it is nearly impossible to find out the coordinate points stand for a GPS Antenna from the scanned tunnel directly, we proposed a method to solve this problem.

We scanned the GPS antenna by *VIVID9i* laser scanner as explained in Chapter 4, and built the antenna model by using the several scanned segment data of GPS antenna. We will use the GPS antenna model later for coordinate transformation.

Actually we have been to Kanaya Tunnel to scan the entrance and exit model of tunnel with GPS Antenna included, we set GPS Antenna to 8 different places in tunnel entrance(west), and 6 different places in tunnel exit(east). Recorded the latitude and longitude value of GPS, and scanned the tunnel with GPS Antenna. Figure 5.13 is the scanned entrance and exit model with GPS Antenna placed. The red circle marked places are where GPS Antennas were placed.

This time we are using a 2-dimensional coordinate system for our experiment and evaluation work, so for the coordinate transformation, we just need a translation operation, other than a rotation operation. Therefore, we only need an offset value for calculation from one direction to the other direction. Because the antennas near the entrance(east) are not easy to distinguish, we selected the antennas near exit(west) as the base for calculating offset value. Fist we found 6 places where the antennas were placed and eliminated the other tunnel points as noises

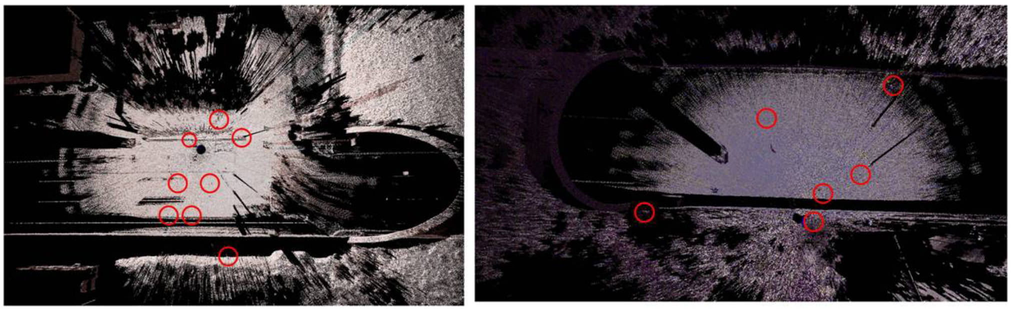

Figure 5.13: (a)GPS Antenna Placed in Tunnel Entrance(west)
(b)GPS Antenna Placed in Tunnel Exit(east)

as shown in Figure 5.14, then we aligned the modeled GPS Antennas with these extracted GPS antenna points, finally we read the coordinate point values of each antenna, calculated the average value of these points, then calculate the difference value between this average value with the coordinate values from recorded GPS latitude and longitude. We repeated this work for 6 times for 6 antennas and then we calculated the average value of the 6 difference values, the final result is the offset value we wanted.

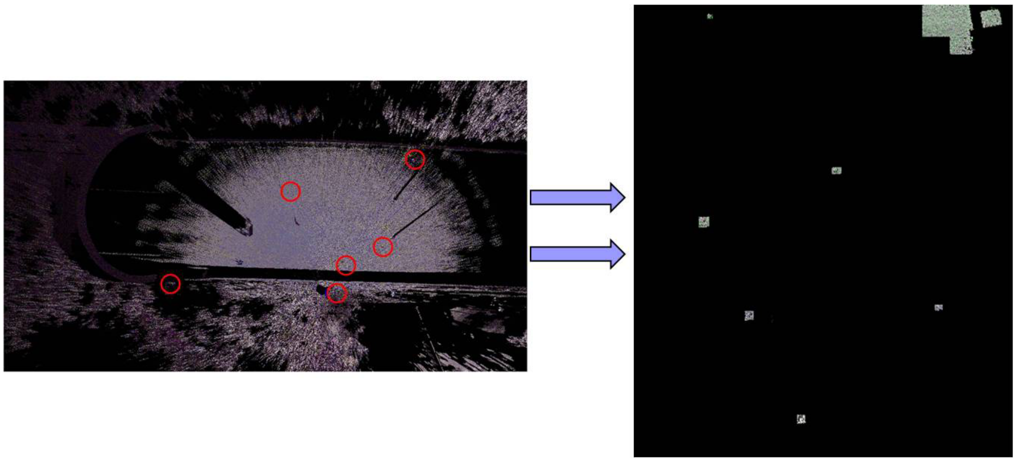

Figure 5.14: Extract the Six Antennas near Exit(west)

Since we should put the extracted boundaries and CAD data into the same coordinate system, the next step is to transform the *G coordinate* of boundaries to *W coordinate* by applying the calculated offset value. If we define (X,Y,Z) as the point in *W coordinate*, and (a,b,c) is the point in *G coordinate*, we can just plus the offset value to (a,b,c), then we got the (X,Y,Z) coordinate in *W coordinate*, this time we are using the 2-D plane, so we just need (X,Y) and (a,b), Equation 5.0 is the translation equation we used this time, the coordinate (-13900,-128200) is the offset value calculated from the previous steps.

$$\begin{pmatrix} X \\ Y \end{pmatrix} = \begin{pmatrix} a \\ b \end{pmatrix} + \begin{pmatrix} -13900 \\ -128200 \end{pmatrix} \tag{5.1}$$

5.2.3 Closest Point Search

As we have already done the boundary extraction and coordinate transformation works in 5.2.1 and 5.2.2, next work is to find a proper method to calculate the difference between 2D CAD blueprint and our extracted boundaries of modeled tunnel. In this paper we used the $k-d$ *tree* algorithm as the evaluation method.

$K-d$ tree [36] has the advantage that it is easy to build, and the algorithm is simple for nearest neighbor searching. Also $k-d$ tree is quite suitable for small dimensions. As this time we just has a 2-Dimension coordinate set (x,y), So this time we selected $k-d$ tree as our kernel algorithm for searching.

The $K-d$ in $K-d$ *tree* denotes the k dimensional, it is a derivative tree from binary-search tree especially used in high dimension searching. It can be created by splitting the data set iteratively. Splitting will stop when the leaf nodes has an small acceptable number of points [38].

OpenCV supplys some functions for $k-d$ *tree* algorithm, for our calculation program, we used these functions. The first step was building a matrix by using the *cvMat*() function, then defined the $k-d$ tree by *cvFeatureTree*(), final step was the closest distance searching by using the *OpenCV* function *cvFindFeatures*(). We plotted the searching result by defining the coordinate set as (x coordinate of tunnel data, calculated closest distance). The calculation results of modeled tunnel and CAD blueprint is shown later in next Section.

5.3 Comparison and Evaluation

5.3.1 General Comparison

For the comparison between extracted CAD blueprint boundaries and extracted model boundaries, we used two kinds of comparison methods, one is direct boundary shape compare, the other is to calculate the closest distance between the corresponding points between modeled tunnel and CAD blueprint tunnel boundaries, and plotted the distances into a 2D plane for the quantitative evaluation.

Figure 5.15 is the direct tunnel boundary shape comparison result. The red line is the boundaries from CAD blueprint, the blue line is the boundaries from modeled tunnel. It can be seen that the difference between the modeled tunnel with CAD blueprint is quite small from an overall standpoint. But from a local view point, things become different.

Because we translated the *G coordinate* from *W coordinate* from Exit(West) to Entrance(East), we can see the west part of boundaries were fitted well Figure 5.16(a). But when we turned to east part of tunnel, we found the two tunnel boundaries were not fitted well as Figure 5.16(b), modeled boundaries shifted from the CAD boundaries about one tunnel wide distance. The shifting started from the curvature of tunnel around coordinate $x = -38000$ as we can see in Figure 5.16(c). This is not what we expected , we will discuss the reasons and our evaluation methods in next section.

For the quantitative evaluation, because our interesting is to find why there is a big difference on the east side, so we are concerning the

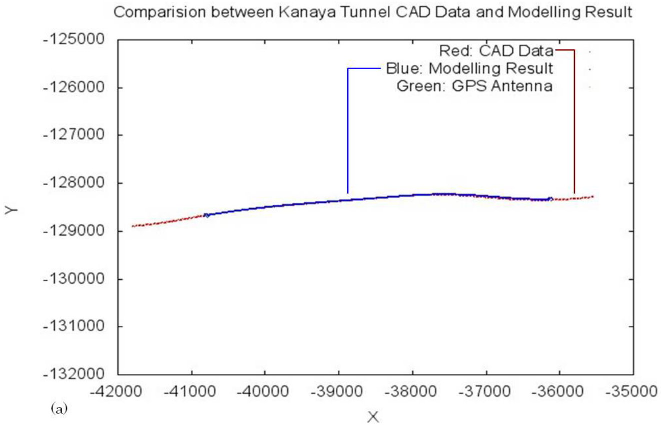

Figure 5.15: Plotted Comparison Result of CAD Data Boundaries and Modeled Data Boundaries

curvature of the whole tunnel and the derivation of closest distance to find out whether if the error increasing rate is the same as curvature. If it is, we can reach the conclusion that the big difference was initially started from the apparent big curve around $x = -38000$. To identify our assumption, first we calculated the curvature near every point of the boundary and plotted it out. Then we calculated the derivation of closest distance between the modeled tunnel and CAD blueprint boundaries and plotted it out, finally we compared the curvature graph with the derivation of closest distance and reach a conclusion.

5.3.2 First Time Evaluation for Modeled Tunnel

At very first we used the original exacted boundaries and CAD boundaries as input data, after we did the closest distance searching, we got a result as Figure 5.17.

We realized that this result has some problems,

- Actually the extracted CAD boundaries were longer than extracted modeled boundaries as you can see in Figure 5.15, so

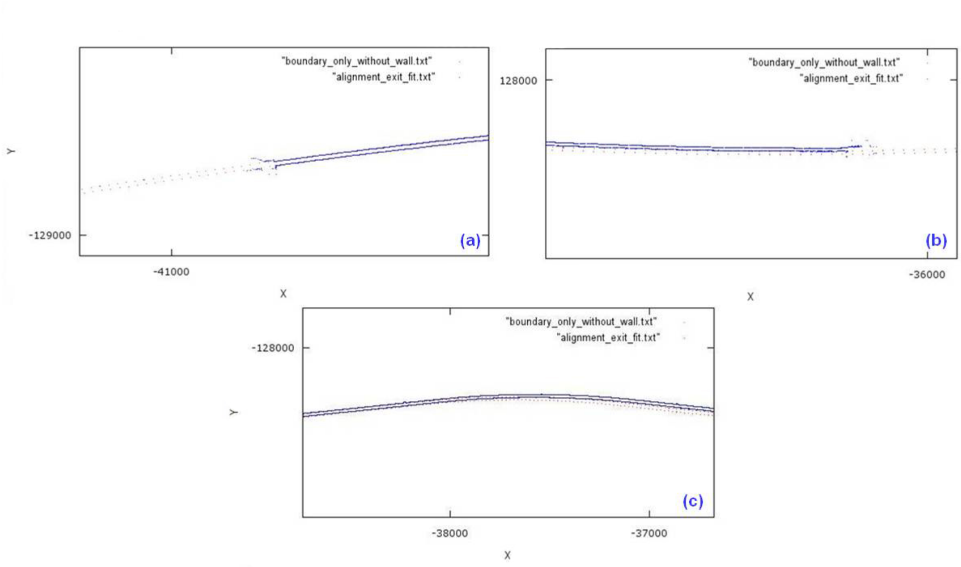

Figure 5.16: (a)View from Exit(west) (b)View from Entrance(east) (c)view near Big Curve

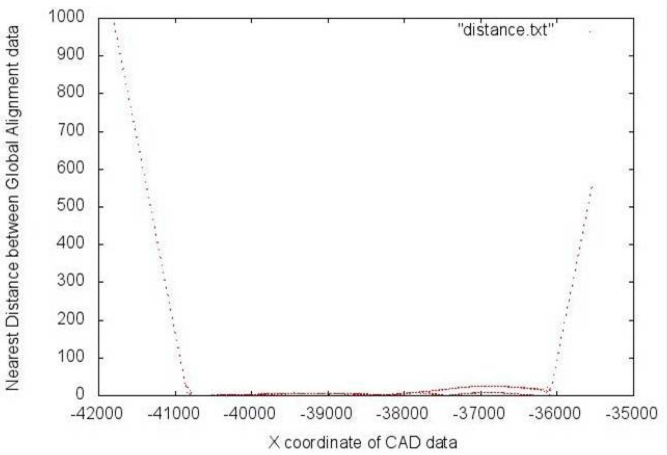

Figure 5.17: Closest Distance Search Result (first time)

the two biases in the closest distance searching chart stands for the redundant points distance, this two biases were actually not need to be calculated.

- We can see around coordinate $x = -37000$, there are two distance result, actually this two distance lines were the wrong results between wrong pair of boundaries.

In Figure 5.18, (1) and (2) pointing to the north and south two boundaries of CAD designing data, (3) and (4) pointing to the north and south two boundaries of modeled tunnel, originally, we should calculate the closest distance between (1) and (3), (2) and (4) respectively. But because we input the two tunnel boundaries directly as the source data of $k - d$ *tree* program, the calculated became wrong between (1) and (3), (2) and (3).

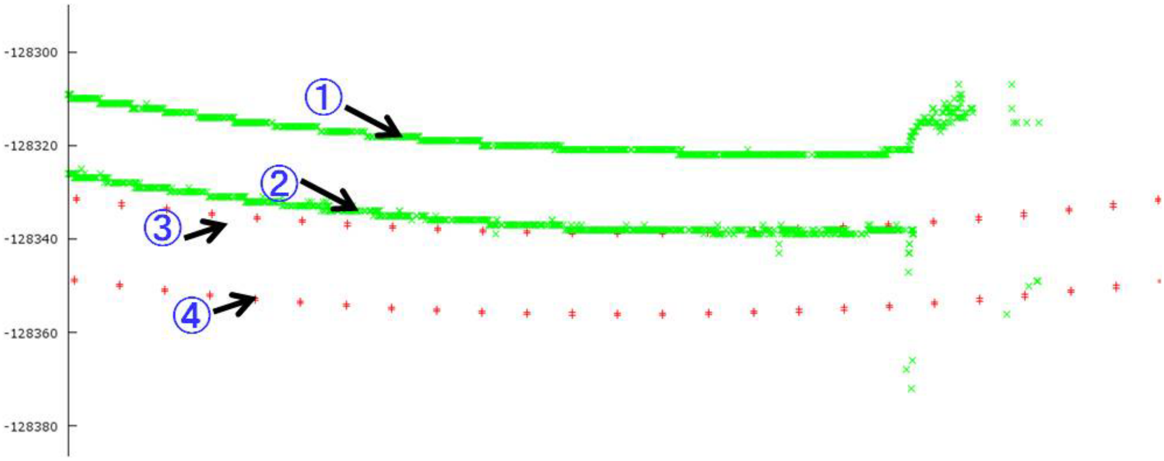

Figure 5.18: Closest Distance Search Corresponding Relationship

Due to these problems, we separated the north and south tunnel boundaries both from modeled tunnel and CAD blueprint, and calculated the closest distance respectively.

We separated the north boundary and south boundary by making a small *Matlab* program. The rough idea for the separation is by using the middle line of the north and south boundaries. Figure 5.19(a) is the separation schema from the boundaries of modeled tunnel and CAD data.

We numbered the separated boundaries of Modeled tunnel and CAD blueprint as *boundary*1 *boundary*2 *boundary*3 *boundary*4 from north to south. The next step is to compare *boundary*1 and *boundary*3, *boundary*2 and *boundary*4, we marked *boundary*1 as green, *boundary*2 as black, *boundary*3 as blue, *boundary*4 as red, the comparison schema are shown in Figure 5.20(a)(b). We can see the two north boundaries are overlapped in the west part near the exit, distance appears first time when there is a big curvature, then the difference become bigger, near the entrance on the east, the difference become slightly smaller again. The comparison graph between south boundaries is the same as the result of north boundaries.

By using the $k-d$ *tree* algorithm, we calculated the nearest distance between nearby points from north boundaries of CAD data and Modeled data. As you can see in below, x stands for the x *coordinate* of CAD data, y stands for the distance of nearby points from CAD blueprint boundary and Modeled data boundary. It can be seen that the trend of difference is the same as the shape comparison result. Near the west part about $x = -41000$, the difference is nearly 0, difference becomes bigger when the tunnel is forwarding to east, difference becomes smaller after the first small curvature, then it becomes bigger and bigger from the big curvature, and becomes slightly smaller near the east part of tunnel. The closest distance of south boundaries between CAD data and Modeled data is the same situation as the north boundaries as we can see in Figure 5.19(b).

From the closest distance chart, we found the result was not very good. We surmised two reasons for the big errors. The first reason is, our local alignment result may not accurate enough. The second reason is, there maybe some relationship with the rotation error near the big curvature when we did the global alignment.

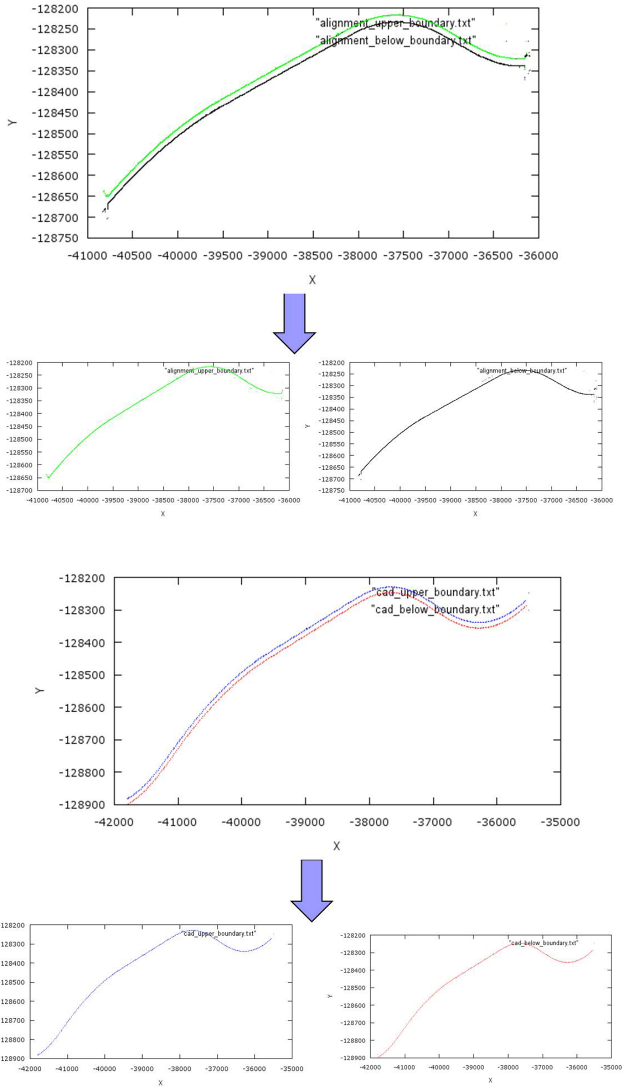

Figure 5.19: Boundary Separation Schema.(a)Modeled Data. (b)CAD Data

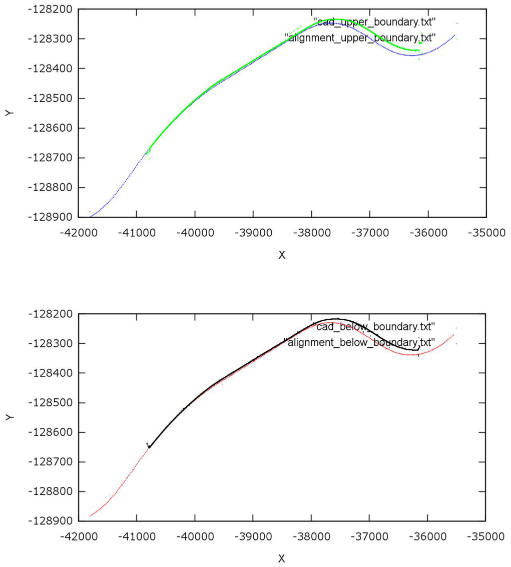

Figure 5.20: (a)North Boundary Comparison. (b)South Boundary Comparison

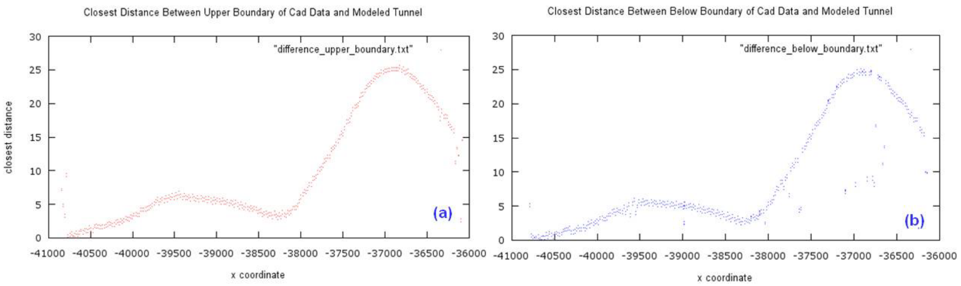

Figure 5.21: (a)Closest Distance of North Boundaries. (b)Closest Distance of South Boundaries

To figure out whether if the bad result was derived from the rotation error, we plan to calculate the curvatures of each point on the boundaries, because the CAD boundaries was not extracted exactly from the inner wall meanwhile the boundaries extracted from our modeled data was inner wall, so we want to use the middle of both CAD data and modeled data to reduce the second time error.

Middle line can be easily calculated by below steps by using north and south boundaries.

- Order the tunnel data from both north and south boundary.
- Select one point (X_n, Y_n) from ordered north boundary.
- By using closest distance searching method we used, search the nearest point (X_s, Y_s) in south boundary.
- Calculate $X_{middle} = \frac{X_n+X_s}{2}$, $Y_{middle} = \frac{Y_n+Y_s}{2}$.
- Loop step two to step three.

After calculating the middle, we can use it for curvature calculation. The curvature calculation equation can be described as follows.

$$k = \frac{d\alpha}{dx} = \frac{y''}{[1 + (y')^2]^{\frac{3}{2}}} \tag{5.2}$$

In order to calculate the curvature k, we need to find a proper function $f(x)$ which can approximately represents the tunnel center line. We used the cftool(Curvature Fitting Tool) in *Matlab* [45] to do this work, by selecting the fitting function, we tried many times using some traditional functions such as polynomial function, but the results were not good, finally we found the Fourier Expansion function was the most proper function for fitting our example data. We have also used different frequencies for fitting, from: $1 - to - 2$ frequency, to: $1 - to - 10$ frequency. We found that when we do the fitting for up to $1 - to - 8$ frequency, we can get a most robustness result comparing with the others. The Fourier expansion function we used is shown as follows

$$\begin{aligned} f(x) = {} & a0 + a1 \cdot \cos(x \cdot w) + b1 \cdot \sin(x \cdot w) + a2 \cdot \cos(2 \cdot x \cdot w) + b2 \cdot \sin(2 \cdot x \cdot w) \\ & + a3 \cdot \cos(3 \cdot x \cdot w) + b3 \cdot \sin(3 \cdot x \cdot w) + a4 \cdot \cos(4 \cdot x \cdot w) + b4 \cdot \sin(4 \cdot x \cdot w) \\ & + a5 \cdot \cos(5 \cdot x \cdot w) + b5 \cdot \sin(5 \cdot x \cdot w) + a6 \cdot \cos(6 \cdot x \cdot w) + b6 \cdot \sin(6 \cdot x \cdot w) \\ & + a7 \cdot \cos(7 \cdot x \cdot w) + b7 \cdot \sin(7 \cdot x \cdot w) + a8 \cdot \cos(8 \cdot x \cdot w) + b8 \cdot \sin(8 \cdot x \cdot w) \end{aligned} \tag{5.3}$$

Here, $a0$, $a1$, $a2$, $a3$, $a4$, $a5$, $a6$, $a7$, $a8$, $b1$,$b2$, $b3$, $b4$, $b5$, $b6$, $b7$, $b8$, and w are all constants which can be calculated directly when fitting.

For fitting accuracy evaluation, we also calculated all the R-Square(the coefficients of multiple determination) and RMSE(root mean squared errors) for each fitting frequency. For R-Square, a value closer to 1 indicates a better fit. Opposite with R-Square, for RMSE, a value closer to 0 indicates a better fit. We can see the fitting result of modeled tunnel and CAD blueprint in Figure 5.22(a)(b), the red parts are the final fitting result, blue parts are the original data

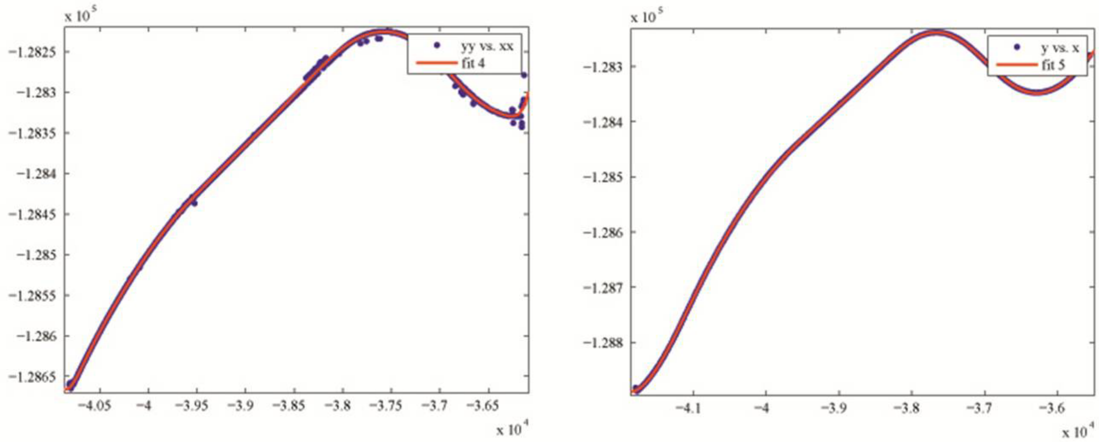

Figure 5.22: Fitting Result. (a)Modeled Data (b)CAD Data

Next we substituted Equation 5.3 to Equation 5.2, we got the final expression of K. By substitute constant w and all the (x,y) points from tunnel middle line to Equation 5.2, we can calculate the curvature k of every point. The curvature results are shown below in Figure 5.23.

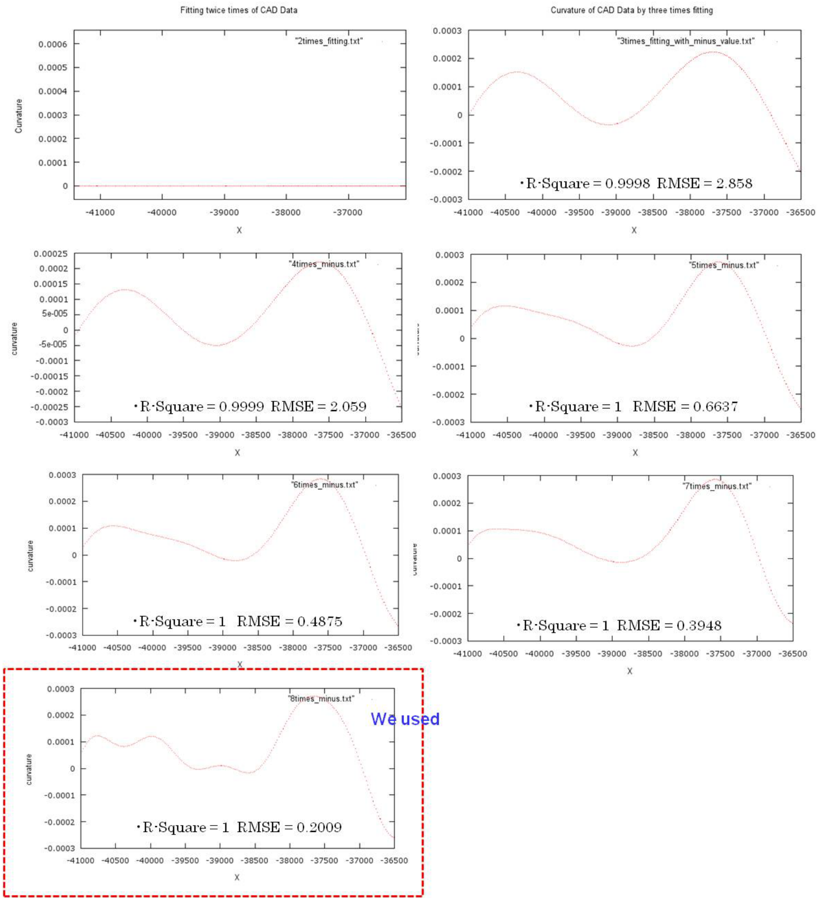

Figure 5.23: Fitting Results from $upto1-2Frequency$ to $upto1-8Frequency$

The up to 1-2 frequency result is totally different with others because the frequency is too low for a correct fitting. Besides, all of the fitting results over 1-8 frequency are over-fitting which have already been filtered from the results. From the curvature calculation result chart, we can see there are two obvious peaks in all of the curvature results, which is corresponding to the obvious curve in extracted boundary chart as Figure 5.24. Since is better to select the result with R-Square close to 1, RMSE close to 0, it can be seen that the 1-8 frequency result is the best one among all the fitting results.

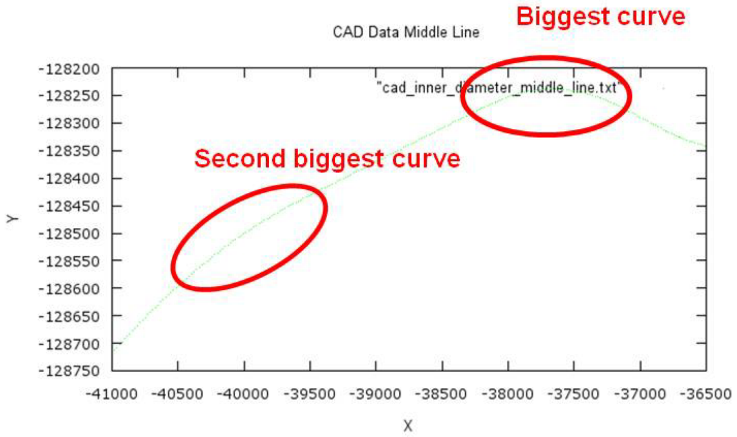

Figure 5.24: Curvatures in Extracted Boundaries

From the fitting result char, we surmise that the errors may increase around the curves, so next we would like to calculate the derivation of closest distance to compare with the curvature result, Figure 5.25 shows the derivation calculation from a closest distance result.

The process for derivation of closest distance is similar with curvature calculation, first we also need to find a proper fitting function, then we derivate the fitting function, and substituted all the closest distance values, we can get the derivation result. Figure 5.26(a)(b) shows the comparison between (a) $Upto1-3Frequency$ Fitting Curvature and (b)Derivation of Closest Distance.

Figure 5.27 shows the comparison between (a) $Upto1-8Frequency$ Fitting Curvature and (b)Derivation of Closest Distance.

It can be seen that the $upto1-frequency$ curvature is more similar with the derivation result, they both has a biggest curvature, and a smaller curvature region. From the comparison result, our below surmises can be verified more or less.

- Local alignment result is not such accurate.

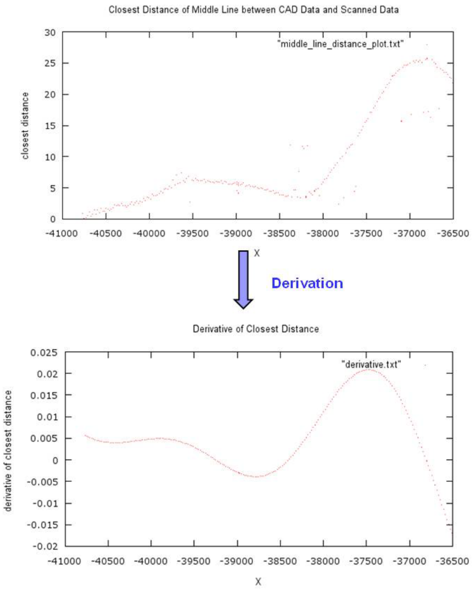

Figure 5.25: Derivation from Closest Distance

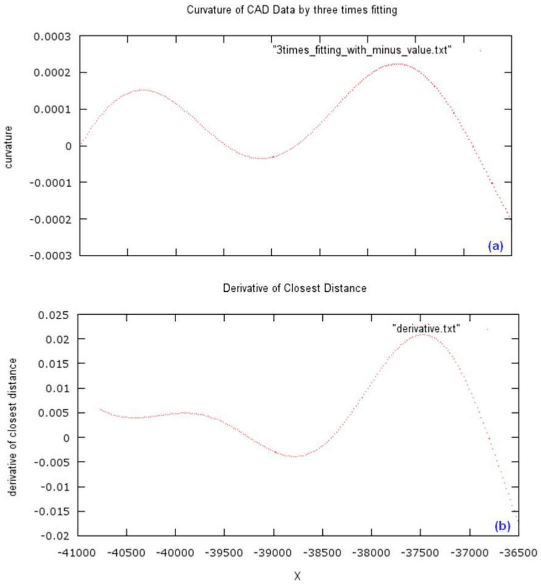

Figure 5.26: Comparison between (a)$Upto1 - 3Frequency$ Fitting Curvature (b)Derivation of Closest Distance

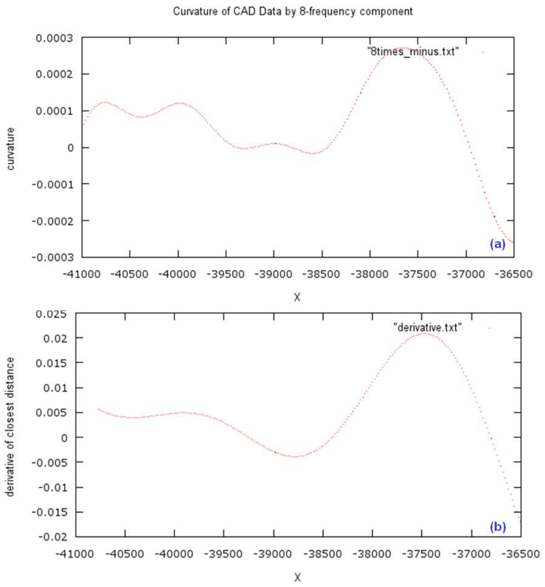

Figure 5.27: Comparison between (a)$Upto1 - 8Frequency$ Fitting Curvature (b)Derivation of Closest Distance

- Rotation errors existing.
- Errors increasing obviously around the biggest curve.

5.3.3 Evaluation for Modeled Tunnel after Error Correction

After the general comparison and evaluation, we aimed to find out the error reasons, so we checked every parts of our work but did not find any mistakes. However, when we converted the 2D CAD data into the WGS(World Geodetic System) just as a try, we found the converted result fitted our modeled tunnel and GPS data well. As a confirmation, we contacted the Central-Nexco and asked the Coordinate System they used, surprisingly, they told us it is true that they used the Japan Coordinate System which is well known as Tokyo Datum. This result is so unexpected and confused because the Japan Government has announced that all the constructors should use the WGS(World Geodetic System) instead of the Tokyo Datum since 2001. Actually we did not know the standard of the company at all, we just thought they must be used the WGS along with what government announced as law.

As we already found out where the mistake is, next step we converted the 2D CAD blueprint from the Tokyo Datum coordinate system to WGS, then we separated the the north and south boundary layer as what we did in the general comparison part. After calculating the middle line from the north and south boundary, we fitted the middle line curve and calculated the curvature again using the same principle by filtering the high frequency parts of curve using Fourier function, the fitting process is shown in Figure 5.28. Because the R-Square is better when it is close to 1, and the RMSE is better when it is close to 0, we selected the 1-8 frequency fitting result as our final curvature graph.

Next step, we calculated the closest distance between the our modeled tunnel and CAD data after coordinate system converting again by using k-d tree algorithm. We can see the distance result in Figure 5.29. It shows the largest distance is around the big curve which is about 4.5 meters, the distance on both sides of tunnel has reduced to 50 centimeters and 20 centimeters which are acceptable. Also we have done the derivation for closest distance result to observe the error increasing rate. Here, the derivation was also done based on 1-8 frequency Fourier function after fitting. We can see the closest distance and derivation result as Figure 5.30.

We compared the curvature graph and closest distance of the middle line between modeled tunnel and 2D CAD Data, the result is shown in Figure 5.31, we can see around the big curvature on x coordinate -38000, the error in closest distance is so big about 3.5 meters, we found the largest error in closest distance graph is 4.5 meters around x coordinate -37500 , we think this is because the rotation error around the big curve. We can see the comparison result of curvature and closest

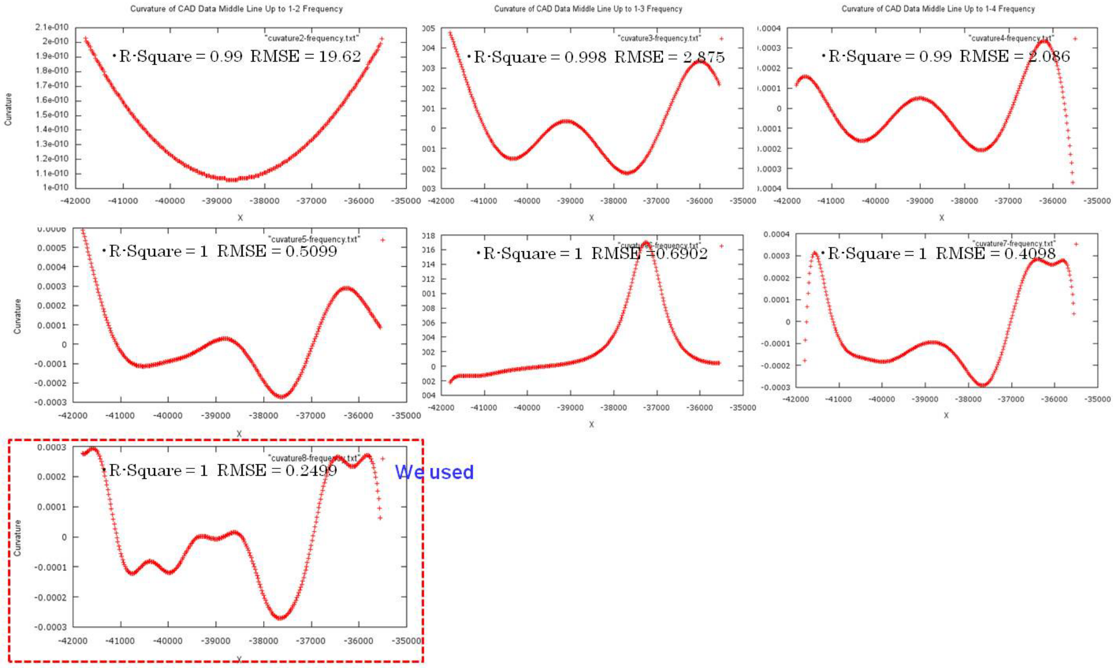

Figure 5.28: Fitting curvature result from:$Upto1-2Frequency$ to $Upto1-8Frequency$

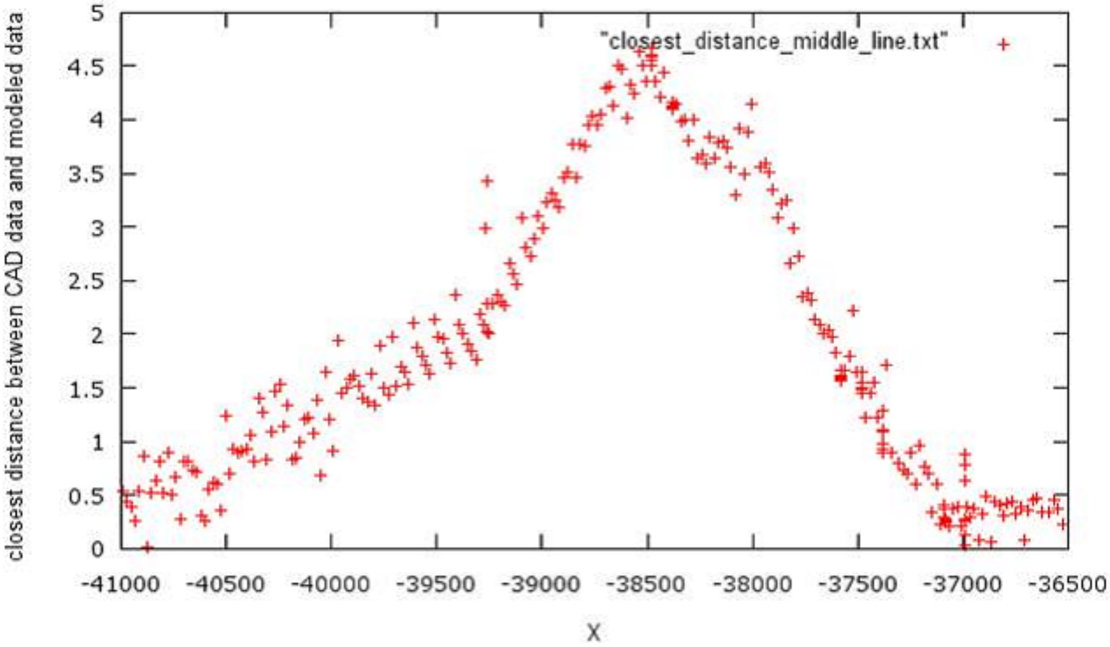

Figure 5.29: The Closest Distance between the middle Line of CAD Data and Our Modeled Tunnel

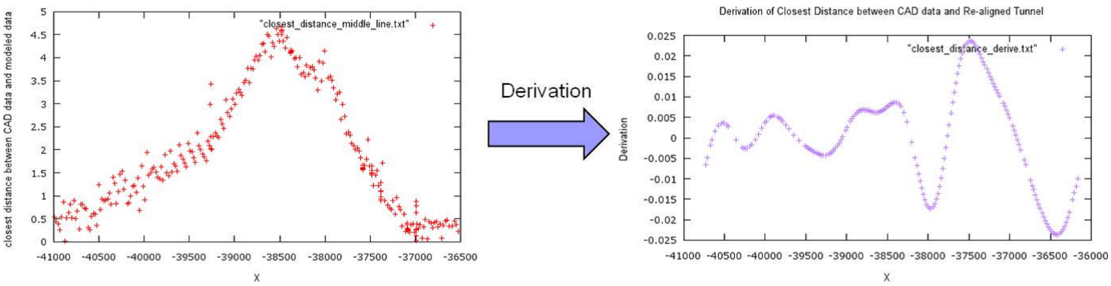

Figure 5.30: Closest Distance Graph and Its Derivation

distance in Figure 5.30.

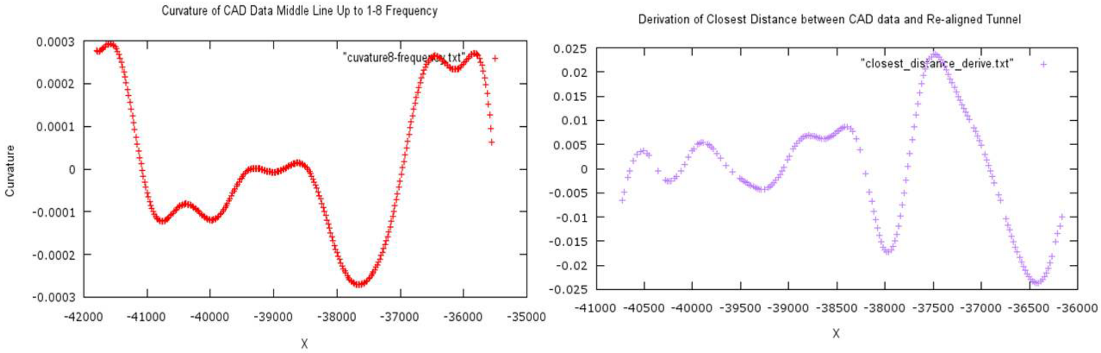

Figure 5.31: Comparison between Curvature Result and Derivation Result

5.4 Comparison Result

The biggest distance between CAD data boundary and Modeled data boundary is about 4.5 meters, it happens around x coordinate -38000, a little shifted from the big curve curve of x=-37500, we predict this is because the rotation error around the big curve when we are doing the simultaneous alignment. For the closest distance on the other parts, on the west side is around 50 centimeters, on the east side is around 20 centimeters. The errors of this two sides are acceptable we think. For the errors around the big curve, we surmise there are three possibilities, the uncertainty in central part, the rotation error may occurs during the local alignment around the big curve, and the GPS antenna itself has errors.

Chapter 6

Conclusions

6.1 Summary

In this paper, we proposed a method for 3D modeling of large-scale tunnel by using laser range sensor. As a result of our work, we succeed creating a dense model of actual tunnel with 4.6 km length by geometric process. For avoiding error accumulation and obtain geometrically optimal model, we made local and global alignment, i.e. (1) Acquire a set of partial structures by static scanning and align them by 3D matching using edge feature, and (2) Fix the absolute position of the data for both ends of the tunnel by GPS and align the rest data again. Applying this to the Kanaya Tunnel in the New Tomei Expressway, we could succeed to create its 3D model and confirm the effect of the global alignment process. Although this method is not efficient enough for modeling every road in Japan and in the world, it may be reasonable enough for modeling especially long and complex tunnels, or the tunnels with quite a few traffic accidents.

For preliminary evaluation, we compared the 3D modeled tunnel with the 2D CAD blueprint as a kind of reference data although it essentially do not correspond each other. To analyze and evaluate the difference, we presented a system which can be described as follows:(1)Extract the boundaries both from CAD blueprint and modeled tunnel to a 2D plane.(2)Transform the coordinate of extracted modeled tunnel boundaries to the same coordinate as CAD blueprint.(3)Calculate the closest distance between the modeled tunnel boundaries and CAD boundaries and do the derivation of the result curve.(4)Calculate the middle line of two data boundaries, and calculate the curvatures of both middle line, then compare the curvature graph with the closest distance graph. By doing the comparison, we analyzed the difference quantitatively and made two surmises about the most probable reasons for the errors:

- Rotation errors existing during local alignment process.
- GPS antenna itself has errors, and the errors accumulated in each step.

Though the comparison result shows our modeled tunnel has some difference around the big curve of tunnel comparing with CAD blueprint. From an overall standpoint, it was acceptable. Our tunnel model can still be used for many purposes such as driving simulation, 3D city for virtual reality and history heritage.

6.2 Future Work

Our future work can be divided into three sub-works. First, for data acquisition, we want to increase more density of the raw data taken by laser range sensor. Second, for local alignment, we plan to use more features besides edges for the local alignment to reduce error accumulation. Last, for global alignment, we want to allow some freedom for GPS points to move slightly, by doing this, even if the GPS antenna shift slightly, the position error will not be accumulated in the after steps.

List of Publications

Liang Xue, Shintaro Ono, Atsuhiko Banno, Takeshi Oishi, Yoshihiro Sato, Katsushi Ikeuchi, ”3D Modeling of Large Scale Tunnel Using On-Vehicle Laser Range Sensor”, *10th ITS Symposium, Dec. 2011.*

Liang Xue, Shintaro Ono, Atsuhiko Banno, Takeshi Oishi, Yoshihiro Sato, Katsushi Ikeuchi, ”Global 3D Modeling and its Evaluation for Large-Scale Highway Tunnel using Laser Range Sensor”, *19th ITS World Congress, Vienna, Austria, Oct.2012* (to be published)

References

[1] Ublox Product Guidance, ”Positioning of Gyroscope by Automotive Dead Reckoning” *http://www.u-blox.ch/ja/dead-reckoning.html*

[2] ”Mitsubishi Mobile Mapping System: High Precision GPS Mobile Survey System”, *http://www.mitsubishielectric.co.jp/pas/mms/*

[3] ”The Experiment and Verification for Topography Surveying by Using Mobile Mapping System(MMS)”, *http://wwweng.cs.shinshu-u.ac.jp/CIVIL/ARCH/TASS/tomilab/shiryou/22-3shiomi.pdf*

[4] Sabry F EI-Hakim, J Angelo Beraldin, Michel Picard, Guy Godin, ”Detailed 3D reconstruction of large-scale heritage sites with integrated techniques”, *IEEE Computer Graphics and Applications(2004)*, Volume:24, Issue:3, Pages:21-29.

[5] T. Anai, N. Fukaya, T. Sato, N. Yokoya, and N.Kochi, ”Exterior orientation method for video image sequences with considering RTK-GPS accuracy”, *Proc. 9th Int. Conf. on Optical 3-D MeasurementTechniques (Optical 3D 2009),* Vol.I, pp.231-240,July 2009.

[6] T. Taketomi, T. Sato, and N. Yokoya, ”Real-time camera position and posture estimation using a feature landmark database with priorities”, *Proc. 19th IAPR Int. Conf. on Pattern Recognition (ICPR2008),* Dec. 2008.

[7] Hiroyuki Uchiyama, Daisuke Deguchi, Tomokazu Takahashi, Ichiro Ide, Hiroshi Murase, ”Egolocalization using Streetscape Image Sequences from In-vehicle Cameras”, *IEEE Intelligent Vehicles Symposium,* pp.185-190, Jun. 2009.

[8] Masafumi NODA, Tomokazu TAKAHASHI, ”Egolocalization by Sequantial Matching of Road-surface in Aerial Image and In-vehicle Camera Images”, *MIRU2010, IS1-79, pp.1-8,* July 2010.

[9] Ono Shintaro,Sato Yoshihiro,Banno Atsuhiko,Tamaki Makoto,Ikeuchi Katsushi, ”Fast Modeling of Large-Scale and Complex Road Structure under No GPS Coverage”, *9th ITS symposium,* 2010.

[10] Takeshi Oishi, Atsushi Nakazawa, Ryo Kurazume, Katsushi Ikeuchi, ”Fast Simultaneous Alignment of Multiple Range Images Using Index Images”, *Proc. Fifth International Conference on 3-D Digital Imaging and Modeling (3DIM’05).*

[11] Takeshi Oishi, Atsushi Nakazaw, Katsushi Ikeuchi, ”Fast Simultaneous Alignment of Multiple Range Images Using Index Images”, *MIRU 2004.*

[12] Zoller + Frohlich GmbH, ”Technical Data Z+F IMAGER(R) 5003”, *http://www.zf-laser.com/*$Technische_{D}aten_{I}MAGER_{5}003_{E}.pdf(Jul.13,2011)$

[13] Stanford University, ”The Stanford 3D Scanning Repository”, *http://graphics.stanford.edu/data/, 3Dscanrep* (Jul. 13, 2011)

[14] Rusinkiewicz, S. Levoy,M, ”Efficient Variants of the ICP Algorithm”, *Third International Conference on 3D Digital Imaging and Modeling,* 2001(3DIM 2001).

[15] P.J.Besl and N.D.McKay, ”A method for registration of 3-D shapes”, *IEEE Transactions on Pattern Analysis and Machine Intelligence,* 14(2) 1992, 239-256.

[16] Y. Chen and G.Medioni, ”Object modeling by registration of multiple range images”, *Image and Vision Computing 10(3),* 1992, 145-155

[17] G.Blais and M.Levine, ”Registering Multi-view Range Data to Create 3D Computer Objects”, *IEEE Transactions on Pattern Analysis and Machine Intelligence,* Vol.17, No.8, 1995.

[18] P.J. Neugebauer, ”Reconstruction of Real-World Objects via Simultaneous Registration and Robust Combination of Multiple Range Images”, *International Journal of Shape Modeling,* 3(12):71-90, 1997.

[19] R.Bergevin, M.Soucy, H.Gagnon, and D.Laurendeau, ”To-wards a general multi-view registration technique”, *IEEE Transactions on Pattern Analysis and Machine Intelligence,* 18(5):540-547, May1996.

[20] R.Benjemaa and F.Schmitt, ”Fast global registration of 3d sampled surfaces using a multi-z-buffer technique”, *Conf. on Recent Advances in 3-D Digital Imaging and Modeling,* papers 113-120, May 1997.

[21] K.Nishino and K.Ikeuchi, ”Robust Simultaneous Registration of Multiple Range Images”, *Fifth Asian Conference on Computer Vision,* pp454-461, Jan, 2002.

[22] Z.Zhang, ”Iterative point matching for registration of free-form curves and surfaces”, *International Journal of Computer Vision,* 13(2):119-152, 1994.

[23] D.Simon, M.Hebert and T.Kanade, ”Real time 3-D pose estimation using a high-speed range sensor”, *In IEEE Intl. Conf. Robotics and Automation,* Pages 2235-2241, San Diego, California, May 8-13 1994.

[24] Jost.T and Hugli.H, ”A multi-resolution ICP with heuristic closest point search for fast and robust 3D registration of range images”, *Fourth International Conference on 3D Digital Imaging and Modeling(3DIM2003),* pages 427-433, Oct 2003.

[25] T. Oishi, R. Sagawa, A. Nakazawa, R. Kurazume and K. Ikeuchi, ”Parallel Alignment of a Large Number of Range Images”, *4th*

International Conference on 3-D Digital Imaging and Modeling, pp. 195-202 (2003)

[26] T. Anai, N. Fukaya, T. Sato, N. Yokoya, and N. Kochi, "Exterior orientation method for video image sequences with considering RTK-GPS accuracy", *Proc. 9th Int. Conf. on Optical 3-D Measurement Techniques,* Vol. I, pp. 231-240, July 2009.

[27] P.Perona and J. Malik, "Scale-Space and Edge Detection using Anisotropic Diffusion", *IEEE Transactions on Pattern Analysis and Machine Intelligence,* Vol. 12, No. 7, pp. 629-639 (1990)

[28] T. Taketomi, T. Sato, and N. Yokoya, "Real-time camera position and posture estimation using a feature landmark database with priorities", *Proc. 19th IAPR Int. Conf. on Pattern Recognition (ICPR2008),* Dec. 2008.

[29] Hiroyuki Uchiyama, Daisuke Deguchi, Tomokazu Takahashi, Ichiro Ide, Hiroshi Murase, "Egolocalization using Streetscape Image Sequences from In-vehicle Cameras", *IEEE Intelligent Vehicles Symposium,* pp.185-190, Jun. 2009.

[30] Masafumi NODA, Tomokazu TAKAHASHI, "Egolocalization by Sequantial Matching of Road-surface in Aerial Image and In-vehicle Camera Images", *MIRU2010,* IS1-79, pp.1-8, July 2010.

[31] Qingquan Li, Bijun Li, Yuguang Li, "3D Modeling and Visualization Based on Laser scanning", *Geographic Information Sciences 2000.*

[32] Ohmori, S. T. Horimatsu, M. Fujise, K. Tokuda. "Radio communication technologies for vehicle information systems", in L. Vlacic, M. Parent and F. Harashima (eds), *Intelligent Vehicle Technologies, Butterworth,* 2001.

[33] "Trimble 5700", *http://www.trimble.com/5700.shtml*

[34] "VIVID 9i Non-contact 3D Digitizer", *http://www.konicaminolta.com/instruments/products/3d/non-contact/vivid9i/*

[35] Wikipedia, "World Geodetic System", *http://en.wikipedia.org/wiki/World*$_{G}$*eodetic*$_{S}$*ystem*

[36] Wikipedia, "K-Dimensional Tree", *http://en.wikipedia.org/wiki/K-d*$_{t}$*ree*

[37] Anton Milev, "KD Tree- Searching in N-dimensions, PartI", *http://www.codeproject.com/Articles/18113/KD-Tree-Searching-in-N-dimensions-Part-I*

[38] Hatena.ne.jp, "Building kd-tree and the nearby points extraction", *http://d.hatena.ne.jp/mokehehe/20091010/kdtree*

[39] Central Nexco Expressway Project, "Shin-tome Expressway Kanaya Tunnel" *http://www.c-nexco.co.jp/corporate/project/pdf/kanayaTN.pdf*

[40] OKUMURA CORPORATION, "Daini Tomei Expressway Project Introduction", *http://www.okumuragumi.co.jp/experience/index01.html*

[41] Wikipedia, "Structural alignment" *http://en.wikipedia.org/wiki/Structural*$_{a}$*lignment*

[42] Wikipedia , "Iterative Closest Point" *http://en.wikipedia.org/wiki/Iterative*$_{C}$*losest*$_{P}$*oint*

[43] Geospatial Information Authority of Japan, "Introduction of The Japanese Geodetic System 2000", *http://www.gsi.go.jp/LAW/G2000-g2000-h3.htm*

[44] Tomohito Masuda, Yuichiro Hirota, Ko Nishino and Katsushi Ikeuchi, "Simultaneous Determination of Registration and Deformation Parameters among 3D Range Images", *Fifth International Conference on 3-D Digital Imaging and Modeling(3DIM'05)*

[45] "Curve Fitting Toolbox: For Use with MATLAB", *http://www.ctr.unican.es/asignaturas/instrumentacion*$_{5I}T/curvefit.pdf$

[46] P.Perona and J.Malik, "Scale-Space and Edge Detection using Anisotropic Diffusion", *IEEE Transactions on Pattern Analysis and Machine Intelligence,* Vol.12, No.7, pp.629-639(1990)

[47] J.Canny, "A computational approach to edge detection", *IEEE Trans. on Pattern Analysis and Machine Intelligence(PAMI),* Vol.8, No.6, pp.679-698,1986.

[48] C.Langis, M.Greenspan, G.Godin, "The parallel iterative closest point algorithm", *Proc. International Conference on 3D Digital Imaging and Modeling(3DIM01),* pp.195-202,2001.

[49] Charles Parker, Alan Fern, Prasad Tadepalli, "Gradient boosting for sequence alignment", *In AAAI'06: Proceedings of the 21st national conference on Artificial intelligence (2006),* pp. 452-457

[50] Dror Aiger, Niloy J. Mitra, Daniel Cohen-Or,"4-POINTS CONGRUENT SETS FOR ROBUST SURFACE REGISTRATION", *The 35th International Conference and Exhibition on Computer Graphics and Interactive Techniques(ACM SIGGRAPH 2008).*

[51] A.Banno, K.Ikeuchi, "Shape recovery of 3D data obtained from a moving range sensor by using image sequences", *Proc. of the Tenth IEEE International Conference on Computer Vision(ICCV'05),* pp.792-799,2005.

[52] M.SOUCY, D.LAURENDEAU, "A general surface approach to the integration of a set of range views", *IEEE Trans. On Pattern Analysis and Machine Intelligence archive,* Vol.17, No.4, pp.344-358,1995.

[53] L.Xue, S.Ono, A.Banno, T.Oishi, Y.Sato, K.Ikeuchi, ”3D Modeling of Large Scale Tunnel Using On-Vehicle Laser Range Sensor”, *10th ITS Symposium, Dec. 2011.*

[54] Atsuhiki BANNO, Shintaro ONO, Yoshihiro SATO, Liang XUE, Katsushi IKEUCHI, ”Accurate Digital Modeling of Inside Structure of Large-Scale Tunnels through 3D Scanning”, *Journal of Society of Automotive Engineers of Japan(Journal of JSAE).*

[55] Ikeuchi Katsushi, Oishi Takeshi, ”3D DIGITAL ARCHIVING”, *University of Tokyo Press,* pp61-66, 68-75,2010.