3D SHAPE RESTORATION AND COMPARISON THROUGH SIMULTANEOUS REGISTRATION
同時位置合わせ手法を用いた三次元形状の復元と比較

by

Tomohito MASUDA
増田 智仁

A Master Thesis
修士論文

Submitted to
the Graduate School of Information Science and Technology
the University of Tokyo
on February 4, 2003
in Partial Fulfillment of the Requirements
for the Degree of Master of Information Science and Technology
in Computer Science

Thesis Supervisor: Katsushi IKEUCHI　池内 克史
Professor of Computer Science

ABSTRACT

Recently, the research on processing 3D objects (preservation, restoration and analysis) has been greatly advanced thanks to the affordable price of the accurate laser range finder. So far, an object shape was represented by 2D information (E.g. drawings and picture) or approximated 3D information (E.g. combination of primitive shapes for 3D shape modeling in CADs). However, 2D and approximated representation provide insufficient accurate information for the 3D shape analysis of the object.

Our objects of interest are 3D data of cultural assets obtained through a laser range finder. The 3D data reconstruction method Ikeuchi laboratory developed was adopted. In this thesis, we propose the method for an object shape restoration and analysis. Our proposed method is the extension of simultaneous registration of 3D data. In conventional simultaneous registration, the errors of 7-parameter pose and position are minimized. In our error function, We assume that the parametric function of the ideal object is available, and the errors are minimized on shape parameters as well. We assume that the parametric function of the ideal object is available. The shape of the actual object can be restored by our proposed error function. This thesis also describes the application of aligning method to visualize difference between similar objects.

論文要旨

近年、高精度な三次元計測が可能なレーザーレンジセンサーの入手が容易になったことにより、三次元データを用いた物体形状の保存、復元や修復、分析などの研究が可能となった。これまで物体形状は、絵や写真などの二次元的な情報や、三次元モデリング用の CAD によるプリミティブ形状の組み合わせによる近似的な三次元情報としてしか表現ができなかった。しかしながらこれらの表現形態は、物体形状を三次元的に分析するには不適切である。

我々の研究室では、レーザーレンジセンサーから得られた三次元データにより、文化財を対象とした実物体形状をコンピュータ上で再構成する手法を開発している。本論文では、これらの手法に基づいて作られた物体の三次元データを利用し、物体形状の復元、分析手法を提案する。提案する手法は、三次元データ間の同時位置合わせを拡張したものである。三次元画像同士の同時位置合わせ問題は位置姿勢の 7 パラメタによる誤差関数の最小化問題として帰着されるが、計測された三次元形状がパラメタ関数として表される場合、位置姿勢にこれらの形状パラメタを加えた誤差関数を設定することにより、物体形状の復元が可能となる。また、類似物体間の三次元データの位置合わせにより、物体形状の差異を可視化することが可能となる。本稿では、これらの手法と複数の文化財への適用例を示す。

Acknowledgements

First of all, I would like to express my gratitude to my supervisor, Professor Katsushi Ikeuchi for his advice and encouragement. I also appreciate all members in Ikeuchi laboratory for their help and providing comfortable environment for my research activities.

This thesis would not be possible without the kind collaboration from Nishino laboratory in the University Museum, University of Tokyo, and Kashihara Archaeological Institute. I would like to thank them for giving me chances for interesting and useful studies.

Contents

1 Introduction **6**

2 Registration of 3D Data **8**

2.1 Registration Strategies 8

2.1.1 Registration Order 8

2.1.2 Matching Unit 9

2.1.3 Error Metric 11

2.1.4 Overall 11

2.2 Robust Simultaneous Registration 11

2.2.1 Corresponding Point Search with Kd-tree 12

2.2.2 Minimization of Error Function 14

3 Effective Search with Kd-tree **17**

3.1 Basic Search Algorithm 17

3.2 Improving Searching Speed 20

3.3 Application to Registration 22

4 Extended Registration Formulation **25**

4.1 Parameter Estimation Formulation 25

4.2 Minimization of Error Function for Parameter Estimation 28

4.2.1 Conjugate Gradient Method 29

4.2.2 Line Minimization 30

4.3 Experiment 30

4.3.1 Revolution Surface of Catenary 32

4.3.2 Dini's Surface 35

4.3.3 Kuen's Surface 36

4.3.4 Inverse Function of the Elliptic Integral of the First Kind 36

5 Shape Difference Visualization **39**
5.1 Visualization Method . . . 39
5.2 Mathematical Model . . . 40
5.2.1 Revolution Surface of Catenary . . . 41
5.2.2 Dini's Surface . . . 41
5.2.3 Kuen's Surface . . . 41
5.2.4 Inverse Function of the Elliptic Integral of the First Kind . . . 43
5.3 Ancient Bronze Mirrors . . . 43
5.3.1 Mirror Analysis and Difference Visualization . . . 43
5.3.2 Application Result . . . 43

6 Evaluation **53**
6.1 Measurement Error of 3D Data . . . 53
6.2 Accuracy of Shape Parameter . . . 56
6.2.1 Influence of Measurement Error . . . 56
6.2.2 Influence of Initial Registration . . . 56
6.2.3 Influence of Initial Parameter . . . 59

7 Conclusion and Future Work **61**

A Quaternion for Rotation **66**
A.1 Convenient Quaternion Representation for Rotation . . . 66
A.2 Quaternion Operation . . . 67
A.3 Rotation Representation by Quaternion . . . 68
A.4 Jacobian of Rotation Matrix with Respect to Quaternions . . . 70

B Derivation of Partial Derivatives **75**
B.1 Revolution Surface of Catenary . . . 75
B.1.1 Numerical Formula . . . 75
B.1.2 Preliminary . . . 75
B.1.3 Partial Derivative . . . 76
B.2 Dini's Surface . . . 77
B.2.1 Numerical Formula . . . 77
B.2.2 Partial Derivative . . . 77
B.3 Kuen's Surface . . . 77
B.3.1 Numerical Formula . . . 77
B.4 Inverse Function of the Elliptic Integral of the First Kind . . . 77
B.4.1 Numerical Formula . . . 77

List of Figures

2.1 Example of registration failure in sequential order 9
2.2 Correspondence using 3D point 10
2.3 Kd-tree construction and search in 2D space 13

3.1 Example of kd-tree search 18
3.2 Example of the Bounds-Overlap-Threshold 22
3.3 Relationship between the neighbor distance and the number of examined records in the basic search 23
3.4 Relationship between the neighbor distance and the number of examined records in the improved search 24

4.1 The incorrect convergence of measured data in considering the ideal data . . . 28
4.2 Effective and ineffective gradient search 29
4.3 Line minimization 31
4.4 The convergent characteristic of the parametric data. 33
4.5 Mathematical models 34
4.6 Ideal data of mathematical models 38

5.1 Signed distance calculation 40
5.2 Shape difference of revolution surface of catenary 41
5.3 Shape difference of Dini's surface 42
5.4 Shape difference of Kuen's surface 42
5.5 Shape difference of inverse function of the elliptic integral of the first kind . . 44
5.6 Ancient Chinese bronze mirrors 45
5.7 Identical mirrors 46
5.8 Shape difference including the global one 47
5.9 Shape difference without the global one 48
5.10 Illustration of the deteriorating pattern 49

5.11 Example of cross section at the global difference (1) 50
5.12 Example of cross section at the global difference (2) 51
5.13 Example of cross section at the global difference (3) 52

6.1 System to evaluate measurement error 54
6.2 Measured error of tele lens 54
6.3 Measured error of middle lens 55
6.4 Measured error of wide lens 55
6.5 The maximum and the minimum of estimated parameter a 57
6.6 The maximum and the minimum of estimated parameter b 57
6.7 The maximum and the minimum of estimated parameter l 58
6.8 Model used in the experiment and its 3D coordinate 58

A.1 Rotation for quaternion representaion 69

List of Tables

6.1 Estimation result for shape parameters according to initial registration error . 59
6.2 Estimating result for noisy data when initial parameters was not correctly set . 60

Chapter 1

Introduction

Various techniques on the computer graphics have been widely applied for the entertainment purpose. Recently, CG research on actual object modeling has gained a wide interest thanks to the accurately 3D data from the laser range finder. The acquired data are undergone registration and merging to form the highly accurate model[1] [2] [3]. The accurate models are desirable for various application including industrial investigation. Particularly in the studies of the precious assets, for example, a cultural treasure, actual object is often unavailable due to its fragibility.

In the shape analysis of cultural assets, the pictures from multiple viewpoints are used in order to analyze the shape difference between objects. Large constructions, such as building, are restored by the combination of the primitive shapes for 3D shape modeling in CADs. Such 2D or approximated 3D representations does not provide the accurate information. Better representation for actual objects is available through the digital data in the computer graphics.

The main objectives for this study is to measure the accuracy of "mathematical models". The mathematical model is the plaster model imitating 3D surface of the predefined numerical formula. The numerical formula contains parameters which affect the shape of 3D surface. The models used in our experiment were made in Germany at the beginning of the 20th century, and they are currently preserved at Graduate School of Mathematical Sciences, University of Tokyo. It is acclaimed that they accurately represent numerical calculation, Support documentation to this claim is scare, and there is no well-known method to measure the accuracy of the model.

We investigated the accuracy of these mathematical models. However, parameters in the numerical formula of the particular model are unknown. It is necessary to estimate the parameters from the model reconstructed by the data of the laser range sensor, and to compare the measured data with the computed data under the estimated parameter.

In order to put the pair of 3D data into the same coordinate and to precisely adjust their pose and position, registration (alignment) method is generally used. In registration, 3D data are translated and/or rotated to minimize the distance between the pair of 3D data. There are seven parameters in this operation. three parameters accounts for the translation and the other four are the quaternion representation for rotation. But to measure the accuracy of the mathematical models, parameters in the formula, which have influence on the model shape, have to be estimated in addition to the seven parameters. In our method, all parameters are simultaneously estimated by minimizing the extended error function which includes the shape parameters as well as seven motion parameters.

Nishino and Ikeuchi [4] developed the simultaneous registration method for seven parameters. In this thesis, we extend their registration method to align the 3D data according to the error function of arbitrary parameters. In case of the registration of mathematical models, the parameter depending on their shape is adjusted in the same time as the pose and position parameters. After the error function converges, the measured and estimated data is exactly fit. We can then observe and visualize the differences between the actual model and the estimated model.

In Chapter 2, introduce various registration strategies and describe in detail the registration method developed by Nishino and Ikeuchi. In order to improve the registration speech, more efficient kd-tree search is necessary. BOT test is added into the convention BOB test to determine the searching space. Details of classical and our proposed kd-tree search are found in Chapter 3. In Chapter 4, we describe the extended registration formulation. In conventional registration, models are fitted according to the pose and position parameter. In the extended registration, the fitting parameters are not limited to pose and position ones. The result of extended registration is applied to determine the shape difference between the actual and ideal objects. Chapter 5 describes the calculation of the shape difference, and its application in the archaeology. In Chapter 6, we evaluate the accuracy of the extended registration method. Our extended registration is studied for its accuracy in estimating parameters according to the measurement error, the initial registration error and the initial input parameters. The thesis is closed with the discussion and conclusion in Chapter 7.

Chapter 2

Registration of 3D Data

In this chapter, we survey the algorithms for registration of 3D data based on Iterative Closest Point algorithm (ICP). ICP algorithm is widely applied in registration method. Registration methods are developed according to various demanding. We classify the algorithms according to the registration order, matching unit, and error metric.

Our main objective is to estimate shape parameters. More detail is provided for the strategy appropriate for this purpose.

2.1 Registration Strategies

ICP algorithm is widely applied to solve the problem of registration of 3D data [5] [6] [7]. In this algorithm, correspondences for match point between two 3D data are searched, sum of error between correspondences is minimized in order to match two data into the same coordinate, pose and position. The minimization is iterated until the sum of error converges. In the registration methods based on ICP algorithm, the following topics need to be considered.

- registration order
- matching unit
- error metric

2.1.1 Registration Order

In the registration of multiple sets of 3D data, the order of registering the data set affects the convergence of the final result. In the direct algorithm, the alignment is repetitively performed until all data sets were aligned [8]. The sequential strategy has low computation cost because

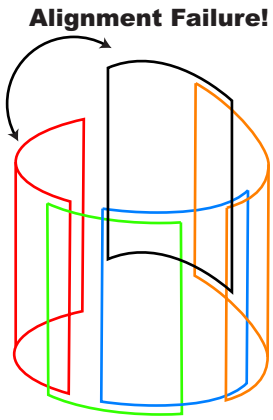


Figure 2.1: Illustration of registration failure in sequential strategy. The pair of red and green data are first aligned. Green and blue data are next aligned. The data are sequentially aligned. The final alignment is between orange and black data. However, the accumulated error prevented the black and red data from the correct alignment.

only two data sets are handled per one registration. However, it is susceptible to aligning failure because the alignment errors are accumulated in each step, which leads to the wrong convergence (Figure 2.1).

In contrast to sequential strategy, the simultaneous strategy aligns all the data at once. More accurate registration is acquired at the cost of higher computing load. The registration error is distributed instead of conglomerated to same data sets.

2.1.2 Matching Unit

In calculating the difference between data sets, we need to determine the corresponding of the point on each data set. [9] [10] use the geometric property of 3D data. This strategy works well on the assumption that one-to-one correspondence can be obtained for all feature points and the correspondences are unchanged. Accurate registration cannot be achieved, otherwise.

More accurate registration is feasible by using 3D point because more plausible correspondences can be selected again and again until the data are converged [5] [11]. This strategy provides a number of correspondence (Figure 2.2).

Corresponding point is usually the nearest neighbor point in 3D space. Besides nearest neighbor search, Normal-shooting [12] is widely applied to find correspondences because of its fast convergence, but the result is less accurate. Projection correspondence is another popular method. It depends on the projection direction, but much faster convergence is achieved

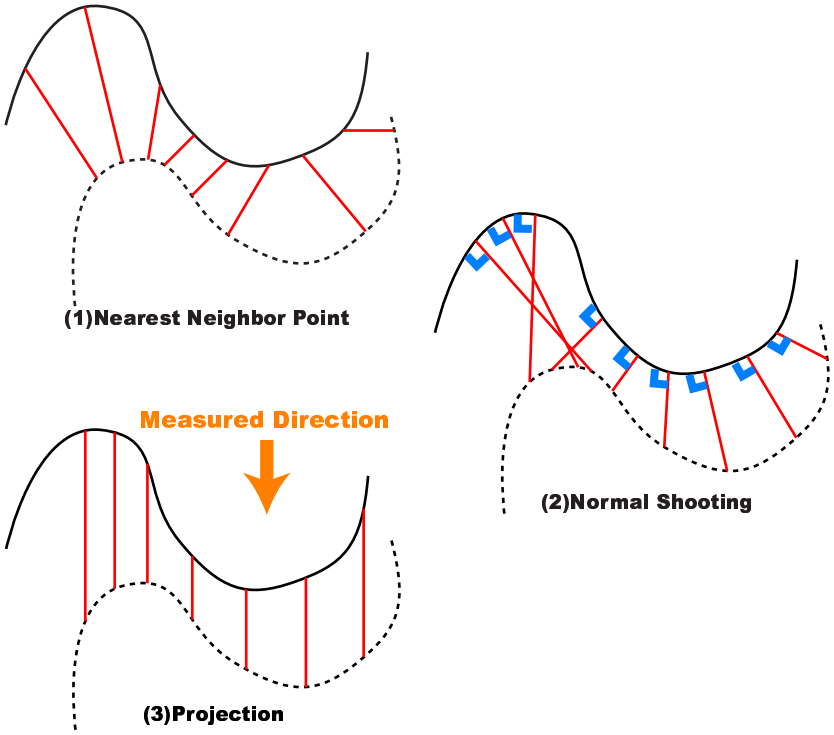


Figure 2.2: Correspondence using 3D point. Solid and dotted line depict the 3D data sets. Red line shows the correspondence between the 3D data on solid and dotted line. The solid line is then aligned to the dotted one according to the correspondence between the data.

by the rendering hardware [13] [14].

2.1.3 Error Metric

The registration solution gives the minimum sum of error between corresponding points. Registration results greatly relied on how the error function is defined. The error is usually defined by the distance between corresponding points. The distance may be calculated between a point and its nearest neighbor point on the other surface [12] [14]. Additional information, such as color [15], can be included into the function.

2.1.4 Overall

We need to align 3D data as accurately as possible in order to estimate shape parameter and analyze their shape; that is, the accurate convergence is taken precedence over the quick one. As shown in [16], it is best to regard the closest point as the corresponding point, because this strategy leads to the precise 3D data convergence. Noise of 3D data has to be considered in order to achieve the accurate correspondence between the data. Data obtained by the laser range sensor are noisy, features such as normal cannot be estimated from the point cloud with acceptable reliability. To cope with erroneous measurement, the simultaneous and point-based strategies were applied and the point-to-point distance is defined as error metric. Further details of our selected strategies are explained in the following section.

2.2 Robust Simultaneous Registration

In this study, we adopted the alignment method proposed by Nishino and Ikeuchi [4]. In their method, the nearest neighbor points are aligned in the way that the sum of point-to-point distance is minimized. the above section. The pseudo programming codes are as follows.

SimultaneousAlignment

input: $MeasuredData \quad D = \{d_i | i = 0, 1, ..., n\}$
input: $InitialPosition \quad P = \{p_i = (R_i, t_i) | i = 0, 1, ..., n\}$
output: $AlignedPosition \quad P' = \{p'_i = (R'_i, t'_i) | i = 0, 1, ..., n\}$
local: $KDTree = \{kdt_i | i = 0, 1, ..., n\}$

repeat
  $KDTree \leftarrow MakeKDTree(P, D)$
  for all $i = 0, 1, ..., n$ **do**
    $KDTree' \leftarrow KDTree - \{kdt_i\}$
    $p'_i \leftarrow MinimizeErrorFunction(p_i, d_i, KDTree')$
  end for
  $P \leftarrow ChangePosition(P', P)$
until ErrorFunctionConverged
$P' \leftarrow P$

We denote d_i, p_i and p'_i ($i = 0, 1, ..., n$) respectively by the set of 3D points in the ith range image, the initial position matrix of d_i in each iteration and the motion matrix of d_i added to p_i. For every i, every data in d_i as located by p_i are converted to the kd-tree structure in function *MakeKDTree*. Function *MinimizeErrorFunction* estimates the motion matrix (of d_i) from the initial position (p_i) by aligning d_i to other range images (d_j ; $j \neq i$). Function *ChangePosition* multiplies the estimated motion matrix (p'_i) by the initial position matrix (p_i) on all data (D) for the initial position in the subsequence iteration. The following sections describe the above algorithms in detail.

2.2.1 Corresponding Point Search with Kd-tree

In [4], the nearest neighbor point is regarded as the corresponding point. When multiple measured data are presented, it is not recommended to find the correspondence in a round robin way. The calculation amount of the round robin is very high and equals to:

$$\sum_i (n_i \times \sum_{j \neq i} n_j), \tag{2.1}$$

where n_i is the number of point in the ith measured data.

Effective corresponding point search is accomplished via kd-tree search [17] [18]. Kd-tree search algorithm is the extension of the binary search in 2-D space to the arbitrary k dimensional space. Kd-tree is constructed by dividing the elements at the median on the axis where

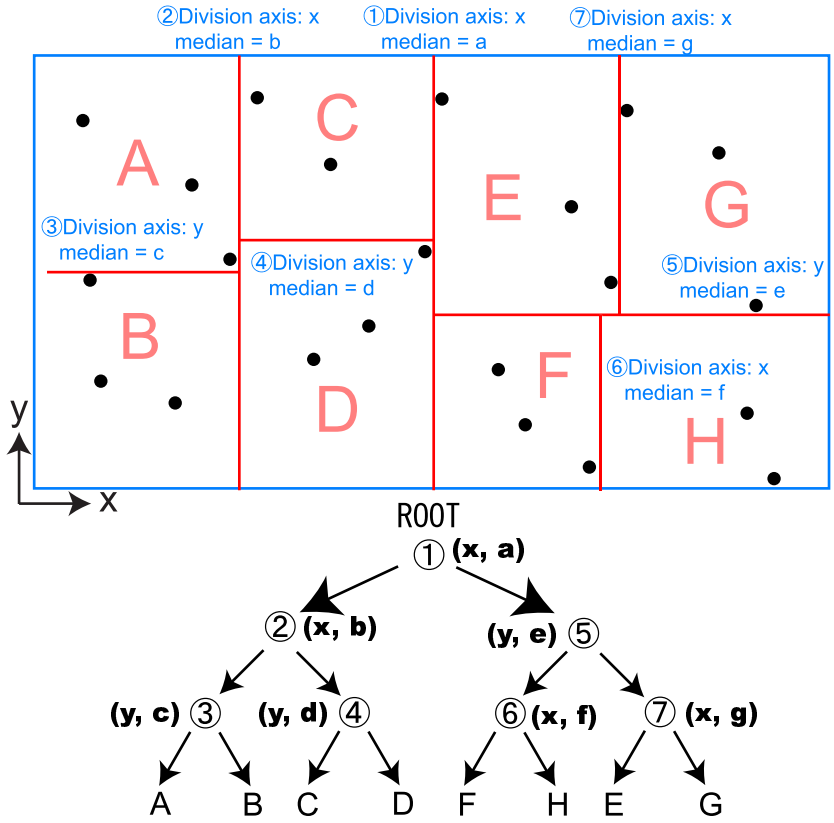


Figure 2.3: Kd-tree construction and search in 2D space. Points are divided in the above order. Leaf node has the coordinates of points, while the others have the information of the axis and the value where the data are divided. points divided.

the elements have the highest variance (Figure 2.3). The division of median is repeated until the number of data in each leaf node is less than the threshold. The depth of kd-tree made from N elements is $\log N$. By applying kd-tree search, the calculation amount of the corresponding point search is reduced to:

$$\sum_i (n_i \times \sum_{j \neq i} \log n_j). \tag{2.2}$$

Nevertheless, the classical kd-tree search is slow with heavy computing load. To speed up the search, we propose the search with limited searching space. Further detail of kd-tree search, both classical and our proposed kd-tree search, are described in Chapter 3.

2.2.2 Minimization of Error Function

Direct Error Function

The alignment is performed by rotating and translating the measured data. The movement is determined such that the total distance between the corresponding points is minimized. The error function (f) is thus defined as:

$$f(\vec{t}, \mathbf{R}) = \sum_{i,j} \|\mathbf{R}\vec{x_i} + \vec{t} - \vec{y_{ji}}\|^2, \tag{2.3}$$

where $\vec{t}$ translation vector,
$\mathbf{R}$ rotation matrix,
$\vec{x_i}$ ith point in the data set of interest,
$\vec{y_{ji}}$ the corresponding point of x_i in the jth measured data.

Even though 3×3 rotation matrix is easy for the vector or matrix computation, it has non-linear characteristic (Equation 2.4):

$$\mathbf{R}\mathbf{R}^T = \mathbf{I} \quad and \quad |\mathbf{R}| = 1 \tag{2.4}$$

The quaternion representation for rotation is more suitable for the error minimization. In our error function the quaternion representation is applied. Advantages and details of quaternion are in Appendix A.

Let $\mathbf{R}(q)$ be a rotation matrix corresponding to the quaternion q, Equation (2.3) is rewritten as follows:

$$f(\vec{t}, q) = f(\vec{t}, \mathbf{R}(q)) = \sum_{i,j} \|\mathbf{R}(q)\vec{x_i} + \vec{t} - \vec{y_{ji}}\|^2 \tag{2.5}$$

Outlier Elimination

In the direct error function, noise leads to the imprecise registration of 3D data, because the exact correspondences between the noisy data in the initial step is unavailable. The imprecise correspondences must be removed before registration. Thresholding is often used to eliminate the correspondences. The threshold value can be determined as a proportion of the standard deviation σ of the errors in the data. Typically, it is set to greater than or equal to 3. This is the simplest, but unreliable method because the elimination is affected by the binary classification of the threshold value. Better elimination is obtained by M-estimation, since probability distribution of the error is considered. M-estimation maximized the probability by minimizing a function of the form

$$E(z) = \sum_i \rho(z_i),$$

where $\rho(z)$ is an arbitrary function of the errors z_i in the data set. The M-estimator is the maximum-likelihood estimator of the probability distribution P equivalent to $E(z)$.

We can find the parameters $\overrightarrow{p}$ that minimize E by taking the derivative of E with respect to $\overrightarrow{p}$ and setting it to 0.

$$\frac{\partial E}{\partial \overrightarrow{p}} = \sum_i \frac{\partial \rho}{\partial z_i} \cdot \frac{\partial z_i}{\partial \overrightarrow{p}} = \sum_i w(z_i) z_i \frac{\partial z_i}{\partial \overrightarrow{p}} = 0, \tag{2.6}$$

$$where \quad w(z) = \frac{1}{z}\frac{\partial \rho}{\partial z}$$

In [4], the Lorentz function is used as the M-estimator because it yields the best result as written in [19].

Summary on Minimization

We have reviewed the minimization of the error function for 3D data registration. The problem is summed up to the minimization of:

$$E(\overrightarrow{p}) = \frac{1}{N(M-1)} \sum_i^N \sum_j^M \rho(z_{ij}(\overrightarrow{p})), \tag{2.7}$$

$$where \qquad \overrightarrow{p} = \quad (\overrightarrow{t}, q), \tag{2.8}$$

$$z_{ij}(\overrightarrow{p}) = \quad \|\mathbf{R}(q)\overrightarrow{x_i} + \overrightarrow{t} - \overrightarrow{y_{ji}}\|^2, \tag{2.9}$$

$$\rho(z_{ij}(\overrightarrow{p})) = \quad \log(1 + \frac{1}{2} z_{ij}(\overrightarrow{p})^2), \tag{2.10}$$

N = the number of data point,
M = the number of measured data

The (negative) gradient of quaternion at an identity quaternion q_I is obtained by Equation (2.11).

$$\left.\frac{\partial(\mathbf{R}(q)\overrightarrow{x_i})}{\partial q}\right|_{q_I} = 2\mathbf{C}(\overrightarrow{x_i})^T \tag{2.11}$$

Detail of the calculation is described in Appendix A.

Finally, the motion gradient is acquired by the following equation.

$$\begin{aligned}
\frac{\partial z_{ij}(\overrightarrow{p})}{\partial \overrightarrow{p}} &= 2(\mathbf{R}(q)\overrightarrow{x_i} + \overrightarrow{t} - \overrightarrow{y_{ji}}) \frac{\partial(\mathbf{R}(q)\overrightarrow{x_i} + \overrightarrow{t} - \overrightarrow{y_{ji}})}{\partial \overrightarrow{p}} \Bigg|_{q_I} \\
&= \begin{bmatrix} 2(\overrightarrow{x_i} + \overrightarrow{t} - \overrightarrow{y_{ji}}) \\ 4\mathbf{C}(\overrightarrow{x_i})^T(\overrightarrow{x_i} + \overrightarrow{t} - \overrightarrow{y_{ji}}) \end{bmatrix} \\
&= \begin{bmatrix} 2(\overrightarrow{x_i} + \overrightarrow{t} - \overrightarrow{y_{ji}}) \\ -4\overrightarrow{x_i} \times (\overrightarrow{t} - \overrightarrow{y_{ji}}) \end{bmatrix}
\end{aligned} \tag{2.12}$$

After the actual gradient is obtained, it is conjugated to the gradient of the motion in the previous iteration. Motion vector $(\overrightarrow{p})$ is acquired by the conjugate gradient method and line minimization with golden section search. Details of conjugate gradient method and line minimization with golden section search are found in Chapter 4.

Chapter 3

Effective Search with Kd-tree

3.1 Basic Search Algorithm

Our problem is to find the nearest neighbor point for a query point, p. In tree searching algorithm, we start at the root node and traverse down to the leaf node that contains the query point. In Figure 3.1-(1), node A contains p, and we compute the distances from p to the records of A. At first glance, the nearest neighbor search is simple and fast by searching within A; however, the searching is not enough. As shown in Figure 3.1-(2), N' is the nearest neighbor of p in node A, but the true nearest neighbor is N.

It is necessary to search a neighbor node to get the correct nearest neighbor. Instead of checking all the neighbor nodes, neighbor nodes is checked if it satisfies the following Bounds-Overlap-Ball (BOB):

$$d > d_B, \tag{3.1}$$

where d and d_B are the distance from the query point $\overrightarrow{p}$ to the boundary of A and B, respectively.

In this method, the nearest neighbor of p in A is searched. The distance to the nearest neighbor is then compared with the distance of p to the boundary of A to all the neighbor nodes. Only nodes that are closer to p than d are searched for the nearest neighbor. In Figure 3.1, node B, C and D are closer to p than d, so all of them are examined. The basic algorithm is in the following.

If a k-d tree contains n records, the depth of the tree is $O(\log_2 n)$. Because of this, Friedman et al.[17] asserted that the computational cost of classical kd-tree search for the nearest neighbor is $O(\log_2 n)$. However, the asserted computational cost is acquired only when the leaf node that contains the query is examined and all other branches can be pruned by the BOB test.

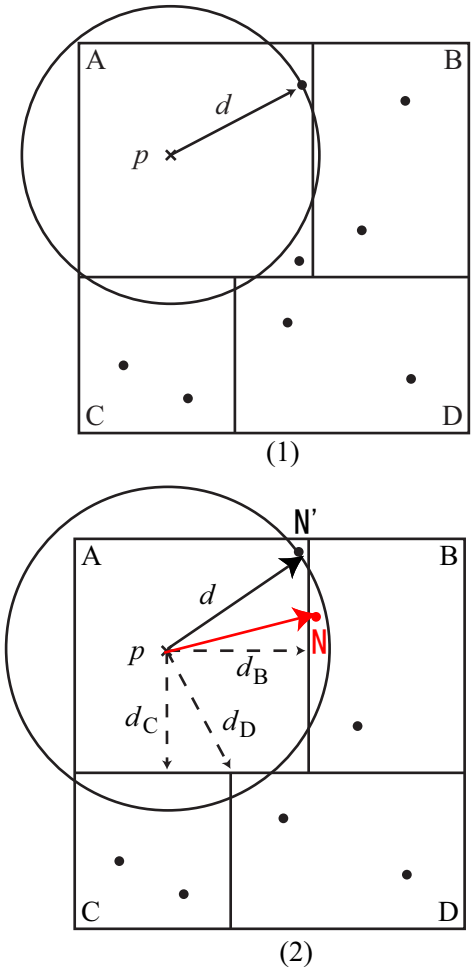


Figure 3.1: Example of the BOB test in kd-tree search. In (1), the point in A nearest to p, is not guaranteed to be the nearest neighbor. (2) indicates that all neighbor nodes to A must be examined to find the correct nearest neighbor.

BasicSearch

input: Node N
input: QueryPoint p
output: NearestNeighborPoint p_n
local: TemporalNeighborPoint p_{tn}

```
if N is leaf node then
  p_tn ← FindNearestNeigborPoint(N)
  if p_tn is nearer to p than p_n then
    p_n ← p_tn
  end if
else if N is leaf node then
  if p is inside leftson(N) then
    Search(leftson(N))
    if d > d_rightson(N) then
      Search(rightson(N))
    end if
  else if p is inside rightson(N) then
    Search(rightson(N))
    if d > d_leftson(N) then
      Search(leftson(N))
    end if
  end if
end if
```

In the worst case, such as Figure 3.1-(2), no branch can be pruned and the computational cost becomes $O(n)$[1]. In particular, when the distance d from the query to the nearest neighbor is large in comparison with the distribution of records in the k-d tree, almost all boundaries will lie inside the ball, and all must be examined.

3.2 Improving Searching Speed

As mentioned in the previous section, the computational cost of kd-tree searching may exceed the acclaimed ($\log_2 n$) when the nearest neighbor in each leaf are far away. However, if the high accuracy of the nearest neighbor which is very far from a query is not necessary, we can limit the search to area near the query. The computational cost is then reduced by introducing the Bounds-Overlap-Threshold (BOT) test to the searching algorithm.

When it is assumed that the approximated position of the nearest neighbor, further than δ, is sufficient for effective processing, the searching ball in Figure 3.1 can be reduced to the dashed ball in Figure 3.2. Whether node B is to be examined or not is then decided by the following criterion.

$$\delta > d_B. \tag{3.2}$$

In BOT test, when node B satisfies the above criterion and BOB test, node B can not be pruned and needs to be examined. If node B fail to satisfy the criteria in either/both BOB or/and BOT test node B is pruned.

In Figure 3.2, Node B and D fail in the BOT test, so they are not examined. On the other hand, node C still need to be examined since $d > d_C$ and $\delta > d_C$.

In case that $d \leq \delta$, the algorithm is completely the same as the one without the BOT test. If $d > \delta$, the improved algorithm may not find the correct nearest neighbor. The distance $d_{correct}$ from the query to the correct nearest neighbor is:

$$\delta < d_N < d_{correct} \leq d, \tag{3.3}$$

where d_N is the smallest distance from the query to the boundary of node N, which is larger than δ.

If the maximum distance of two points in a k-d tree is $2D$, the volume of the hypersphere, which contains all points of a k-d tree, is proportional to D^k. Similarly, the volume of the hypersphere of radius δ is proportional to δ^k. Therefore, if the points in a k-d tree have uniform distribution, our new method reduces the searching cost of the worst case from $O(n)$ to $O((\frac{\delta}{D})^k n)$.

The search algorithm with BOT test is constructed by a recursive function as shown below.

[1]This is a rough estimation. A more accurate estimation is $O(\sum_{k=0}^{\log_2 n} 2^k)$

ImprovedSearch

input: Node N
input: QueryPoint p
output: NearestNeighborPoint p_n
local: TemporalNeighborPoint p_{tn}

if N is leaf node **then**
  $p_{tn} \leftarrow$ FindNearestNeigborPoint(N)
  if p_{tn} is nearer to p than p_n **then**
    $p_n \leftarrow p_{tn}$
  end if
else if N is not a leaf node **then**
  if p is inside leftson(N) **then**
    Search(leftson(N))
    if $d > d_{\text{rightson}(N)}$ $\wedge$ $\delta > d_{\text{rightson}(N)}$ **then**
      Search(rightson(N))
    end if
  else if p is inside rightson(N) **then**
    Search(rightson(N))
    if $d > d_{\text{leftson}(N)}$ $\wedge$ $\delta > d_{\text{leftson}(N)}$ **then**
      Search(leftson(N))
    end if
  end if
end if

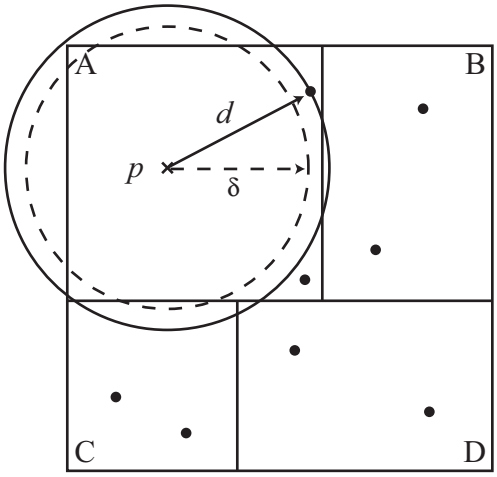


Figure 3.2: The Bounds-Overlap-Threshold (BOT) test compares a threshold δ and distances d_B, d_C and d_D. We examine nodes that satisfy both the BOB and BOT tests. In this case, we examine C only. B and D are pruned by the BOT test.

N, p and d are the ambiguous node, query point and distance from p to the current nearest neighbor, respectively. The ambiguous node is being checked for the nearest neighbor of p. Function rightson(N) and leftson(N) return right and left sons of node N, respectively. $d_{\text{rightson}(N)}$ / $d_{\text{leftson}(N)}$ are the distance from the query to the boundary of right/left son of N. The difference from the classical algorithm is high lighted by gray boxes.

3.3 Application to Registration

In aligning range images, incorrect correspondences can be removed by thresholding or M-estimation [19] [20] [21]. To compute the posture of range images, the corresponding points farther than the threshold distance are negligible. In our experiment, based on the error distribution of the laser range finder, the threshold δ was set to 1.0cm. Figure 3.3 shows the distribution of the number of leaves examined during the search for a nearest neighbor point in the aligning application.

When we use the classical searching algorithm, the number of records examined grows according to the distance from the queries. On the other hand, when the BOT test is applied, the number of records examined is drastically reduced in the case that the distance from the query is larger than d (Figure 3.4). The total numbers of records examined are 1842640 and 470300 without and with the BOT test, respectively. The computational cost of searching for

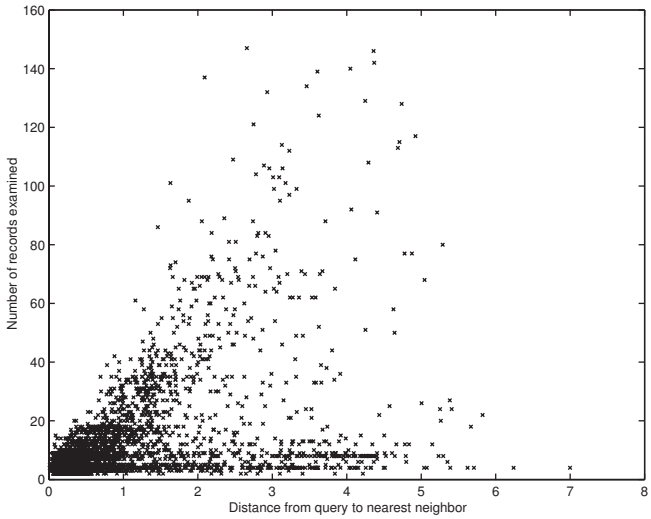


Figure 3.3: The graph shows the relationship between the distance from a query to the nearest neighbor and the number of records examined, when the classical search algorithm is applied to align range images.

nearest neighbor points is approximately 25.5% of the one by the classical search algorithm.

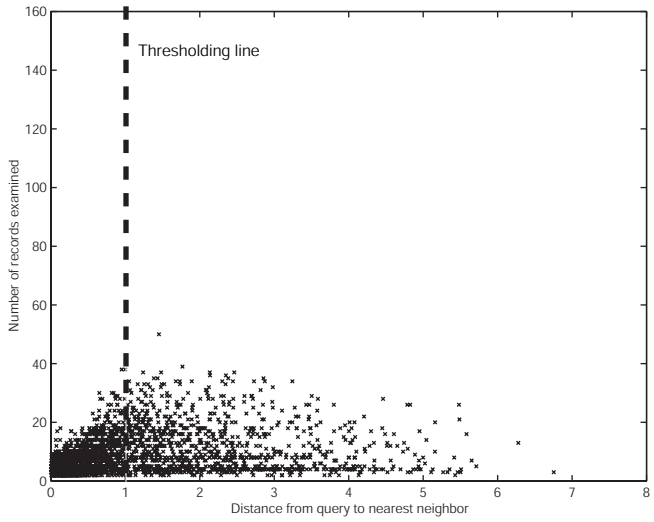


Figure 3.4: The graph shows the between the distance from a query to the nearest neighbor and the number of records examined, when the BOT test is applied to the searching algorithm for aligning range images.

Chapter 4

Extended Registration Formulation

In this chapter, we describe the parameter estimation of 3D data. Estimated parameters includes the ones which affect their shape appearance, in addition to 7 parameters of the pose and position of the conventional alignment.

4.1 Parameter Estimation Formulation

This chapter considers the solution to our main goal requoted here. We assume that the object of interest can be represented by some mathematical formula. Parameters in the formula affects the overall shape. Our goal is to estimate the parameters. By comparing the measured data with the ideal data computed by its corresponding formula and parameter, the registration and shape (parameter) matching problem is solved. If the pose and position can be registered between the measured data and the ideal data, shape matching can be considered as the error minimization problem. Moreover, 7 parameters of pose and position, and the shape parameters in the mathematical formula are simultaneously acquired.

We extend the parameter estimation of the existing registration formulation to include the shape parameter. The error function in Equation (2.5) is adapted to include the fitting of the ideal data.

$$f(\vec{p}) = \sum_{i,j} \|\mathbf{R}(q)\overrightarrow{g(\vec{k})}_i + \vec{t} - \vec{y_{ji}}\|^2, \tag{4.1}$$

where $\vec{p} = (\vec{t}, q, \vec{k})$

$\overrightarrow{g(\vec{k})_i}$ *i*th point in some ideal data computed from the formula whose parameters are $\vec{k}$

Therefore, $z_{ij}(\vec{p})$ in Equation (2.9) is transformed into:

$$z_{ij}(\vec{p}) = \|\mathbf{R}(q)\overrightarrow{g(\vec{k})_i} + \vec{t} - \vec{y_{ji}}\|^2 \tag{4.2}$$

And the motion gradient is obtained by:

$$\frac{\partial z_{ij}(\vec{p})}{\partial \vec{p}} = 2(\mathbf{R}(q)\overrightarrow{g(\vec{k})_i} + \vec{t} - \vec{y_{ji}}) \frac{\partial(\mathbf{R}(q)\overrightarrow{g(\vec{k})_i} + \vec{t} - \vec{y_{ji}})}{\partial \vec{p}}\Bigg|_{q_I}$$

$$= \begin{bmatrix} 2(\overrightarrow{g(\vec{k})_i} + \vec{t} - \vec{y_{ji}}) \\ -4\overrightarrow{g(\vec{k})_i} \times (\vec{t} - \vec{y_{ji}}) \\ 2(\overrightarrow{g(\vec{k})_i} + \vec{t} - \vec{y_{ji}}) \frac{\partial(\overrightarrow{g(\vec{k})_i})}{\partial \vec{k}} \end{bmatrix} \tag{4.3}$$

Then, the pseudo code of the calculation is as follows:

SimultaneousAlignmentWithParameterEstimation

input: $MeasuredData \quad D = \{d_i | i = 0, 1, ..., n\} \quad and \quad IdealData \quad d_c$

input: $InitialPosition \quad P = \{p_i = (R_i, t_i) | i = 0, 1, ..., n\}$

input: $InitialPositionAndParameterOfIdealData \quad p_c = (R_c, t_c, k)$

output: $AlignedPosition \quad P' = \{p'_i = (R'_i, t'_i) | i = 0, 1, ..., n\}$

output: $AlignedPositionAndEstimatedParameterOfIdealData \quad p_c' = (R_c', t_c', k')$

local: $KDTree = \{kdt_i | i = 0, 1, ..., n\}$

repeat

$KDTree \leftarrow MakeKDTree(P, D)$

for all $i = 0, 1, ..., n$ **do**

$KDTree' \leftarrow KDTree - \{kdt_i\}$

$p'_i \leftarrow MinimizeErrorFunction(p_i, d_i, KDTree')$

end for

$p_c' \leftarrow MinimizeErrorFunctionWithParameterEstimation(p_c, d_c, KDTree)$

$P \leftarrow ChangePosition(P', P)$

$p_c \leftarrow ChangePositionAndParameter(p_c', p_c)$

until ErrorFunctionConverged

$P' \leftarrow P$

$p_c' \leftarrow p_c$

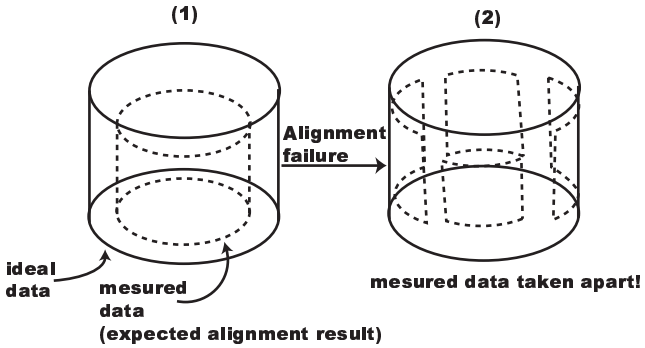


Figure 4.1: The incorrect convergence of measured data as the result of the incorrect ideal data. In (1), measured data is expected to be converged to the dotted line. (2) showed the converging result. The measured data are taken apart to partially match the incorrect ideal data.

The difference from the basic algorithm (Chapter 2) is high lighted by gray boxes. To the algorithm, we add the set of 3D points as computed by the numerical formula with the initial parameter (d_c), its initial position matrix (p_c) and its initial estimation of the motion matrix (p_c'). Function *MinimizeErrorFunctionWithParameterEstimation* determines the expected motion matrix (of d_c) in the subsequent iteration from the initial position and parameter (p_c) according to the entire sets of measured data (D). Function *ChangePositionAndParameter* multiplies the expected motion matrix (R_c', t_c') by the initial position matrix (R_c, t_c) and update the initial parameter (k) to the estimated parameter (k').

Simultaneous registration is applied in our experiment. Because the algorithm registrates every parameter simultaneously, registration of the measured data and fitting of the ideal data in the same time. It is unnecessary to supply the registrated measured data. The registration of the measured data is similar to the one in conventional algorithm. Note that the ideal data have no effect on the expected motion matrices of measured data. This is to avoid the incorrect converging due to the incorrect shape parameters of the ideal data (Figure 4.1). So the kd-tree of the ideal data is not required.

4.2 Minimization of Error Function for Parameter Estimation

Error function for parameter estimation is minimized by the steepest gradient. However, steepest gradient method is not always efficient. In this study, the conjugate gradient method is

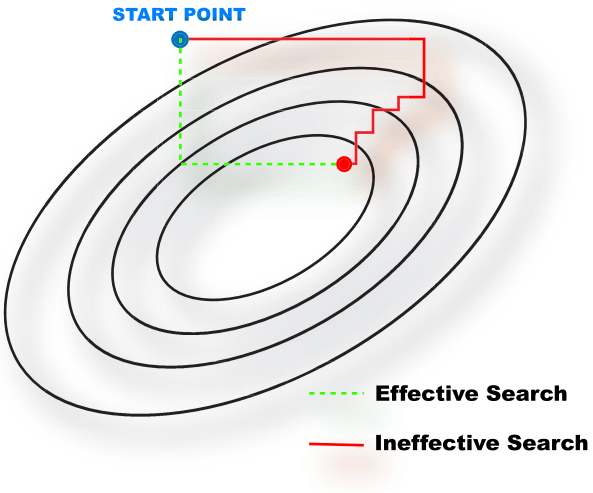


Figure 4.2: Illustration of an effective and ineffective gradient search. In ineffective search the gradient obtained in the current step is guaranteed to be orthogonal to the one in the previous and following step. However, many step have the gradient in the same direction. In effective search, these gradient should be summed up and estimated within one step.

applied. After obtaining the conjugate gradient, we have to determine the norm of motion vector. This determination is performed by line minimization. This section explains the conjugate gradient and line minimization method.

4.2.1 Conjugate Gradient Method

In the previous, the minimization applies the steepest gradient method. Though the gradient in the current iteration is orthogonal to the one in the previous step, the same gradient is taken repeatedly before the minimization is completed.

The conjugate gradient method has been proposed to avoid the above inefficiency. In this method, we use the gradient conjugated to the former gradient, instead of the steepest gradient. The usage of the gradient conjugated to the former gradient guarantees that it is a conjugated to any other gradients.

In our implementation, we use Fletcher-Reeves method and Polak-Ribiere method [22] [23] [24]. These two methods are closely related and have the well-found algebraic property. Define $\mathbf{A}\,\overrightarrow{g_i}$ and $\overrightarrow{h_i}$ as the positive constant symmetry matrix of $n \times n$, the steepest and the conjugate gradient in ith minimization step, respectively. The two types of gradient vector are

defined as follows:

$$\overrightarrow{g_{i+1}} = \overrightarrow{g_i} - \lambda_i \mathbf{A}\overrightarrow{h_i}, \quad \overrightarrow{h_{i+1}} = \overrightarrow{g_{i+1}} + \gamma_i \overrightarrow{h_i} \tag{4.4}$$

λ_i and γ_i are determined such that

$$\overrightarrow{g_{i+1}} \cdot \overrightarrow{g_i} = 0, \quad \overrightarrow{h_{i+1}} \cdot \mathbf{A} \cdot \overrightarrow{h_i} = 0. \tag{4.5}$$

From Equation (4.4) and (4.5), λ_i and γ_i can be calculated as:

$$\begin{aligned} if \quad \overrightarrow{g_i} \cdot \mathbf{A}\overrightarrow{h_i} \neq 0 \quad \overrightarrow{h_i} \cdot \mathbf{A}\overrightarrow{h_i} \neq 0, \quad \lambda_i &= \frac{\overrightarrow{g_i} \cdot \overrightarrow{g_i}}{\overrightarrow{g_i} \cdot \mathbf{A}\overrightarrow{h_i}}, \quad \gamma_i = -\frac{\overrightarrow{g_{i+1}} \cdot \mathbf{A}\overrightarrow{h_i}}{\overrightarrow{h_i} \cdot \mathbf{A}\overrightarrow{h_i}} \\ otherwise, \qquad \lambda_i &= 0, \quad \gamma_i = 0 \end{aligned} \tag{4.6}$$

As a result, we can obtain the following equation.

$$\overrightarrow{g_i} \cdot \overrightarrow{g_j} = 0, \quad \overrightarrow{h_i} \cdot \mathbf{A}\overrightarrow{h_j} = 0 \quad (i \neq j) \tag{4.7}$$

The first equation implies that $\overrightarrow{g_i}$ is orthogonal to other $\overrightarrow{g_j}$ and the second implies the orthogonality among $\overrightarrow{h_i}$ each $\overrightarrow{h}$ is conjugated to the former $\overrightarrow{h_i}$.

From (4.4) and (4.7), γ_i and λ_i can be found by:

$$\gamma_i = \frac{\overrightarrow{g_{i+1}} \cdot \overrightarrow{g_{i+1}}}{\overrightarrow{g_i} \cdot \overrightarrow{g_i}} = \frac{(\overrightarrow{g_{i+1}} - \overrightarrow{g_i}) \cdot \overrightarrow{g_{i+1}}}{\overrightarrow{g_i} \cdot \overrightarrow{g_i}} \tag{4.8}$$

$$\lambda_i = \frac{\overrightarrow{g_i} \cdot \overrightarrow{h_i}}{\overrightarrow{h_i} \cdot \mathbf{A}\overrightarrow{h_i}} \tag{4.9}$$

4.2.2 Line Minimization

Line minimization is usually used in the gradient-based optimization. Many techniques have been proposed for this minimization. We use a combination of golden-ratio bracketing (golden section search) and parabolic fits. Figure 4.3 illustrate our procedure.

4.3 Experiment

In this section, we attempted to estimate the shape parameters of the mathematical models. The corresponding formula to the tested mathematical models (Figure 4.5) were assumed to be available. Note that the scale parameter, *l*, appears in every surface. Detail of the function and its derivative are described in Appendix B.

The model were measured by MINOLTA VIVID 900. MINOLTA VIVID 900 is the laser range finder and obtains the depth information by the triangulation method. Its accuracy is

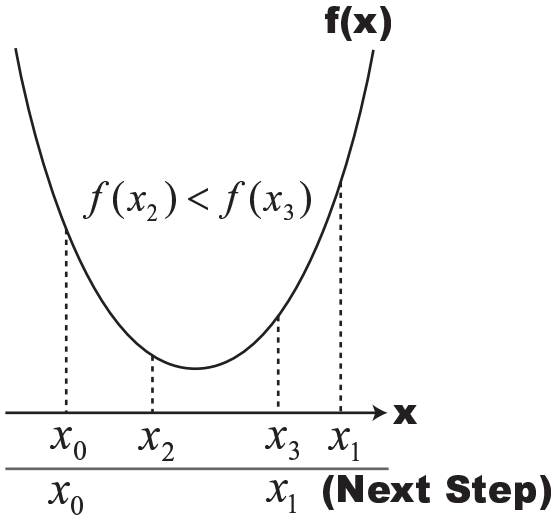


Figure 4.3: Illustration for line minimization. x and $f(x)$ depict the motion vector and the error function respectively. When the minima lies between x_0 and x_1, $x_2(= x_0 + \frac{\tau-1}{\tau}(x_1 - x_0))$ and $x_3(= x_0 + \frac{1}{\tau}(x_1 - x_0))$ are calculated (τ depicts a golden ratio ($= \frac{1+\sqrt{5}}{2}$)). By comparing $f(x_2)$ and $f(x_3)$, the position of the minima can be founded. If $f(x_2) < f(x_3)$, the minimum lies between x_2 and x_3, so in the subsequent search, x_3 is set at x_1; otherwise, the minimum is between x_2 and x_1, so x_2 is set at x_0. The figure shows the minimum is between x_0 and x_3, so x_1 in the subsequent iteration is set at x_3.

evaluated in Chapter 6. Initial data registration was manually performed via GUI. The initial shape parameters were also manually estimated. The number of iteration in all tests was set at 5. Figure 4.4 shows the convergent characteristic of our algorithm on the revolution surface of catenary. The figure indicates that the ideal model was modified to fit the actual one. The modification was performed via the shape parameter.

In this section, only the estimated parameters were shown. The application of the estimated parameter to measure the accuracy of the mathematical models is presented in Chapter 5. The accuracy of our estimation is evaluated in Chapter 6.

The accuracy of the estimated parameters is also explained in the Chapter 6, though we show only the obtained parameters in order to certify that shape parameters are estimated from these data in this section. Moreover, the shape difference between the measured data and the computed data is visualized in the Chapter 5.

4.3.1 Revolution Surface of Catenary

This surface is generated by rotating a 2D catenary and shown in Figure 4.6-(1). The surface by revolution always has the azimuthal symmetry. Besides scale parameter (l), there are 2 parameters (a, b) involved in the appearance of the revolutional surfaces.

Numerical Formula

$$for \quad {}^{\exists}a, {}^{\exists}b, {}^{\exists}l \quad (0 < b \leq a),$$

$$X(u,v) = (l\phi(v)\cos u, l\phi(v)\sin u, l\psi(v)), \tag{4.10}$$

$$where \quad 0 \leq u \leq 2\pi, \quad -a \cdot \sinh^{-1}\left(\frac{a}{b}\right) \leq v \leq a \cdot \sinh^{-1}\left(\frac{a}{b}\right)$$

$$\phi(v) = b\cosh\left(\frac{v}{a}\right), \quad \psi(v) = \int_0^v \sqrt{1 - \frac{b^2}{a^2}\sinh^{-1}\left(\frac{t}{a}\right)}dt \tag{4.11}$$

Partial Derivatives with respect to Shape Parameter

$$\frac{\partial X}{\partial a} = \left((l\cos u)\frac{\partial \phi}{\partial a}, (l\sin u)\frac{\partial \phi}{\partial a}, l\frac{\partial \psi}{\partial a}\right), \tag{4.12}$$

where

$$\frac{\partial \phi}{\partial a} = -\frac{bv}{a^2}\sinh\frac{v}{a} \tag{4.13}$$

and

$$\frac{\partial \psi}{\partial a} = \frac{vb^2}{2a^3}\left(-\frac{1}{2a^2}\sinh^{-1}\frac{v}{a} + \frac{1}{\sqrt{v^2+a^2}}\right) \tag{4.14}$$

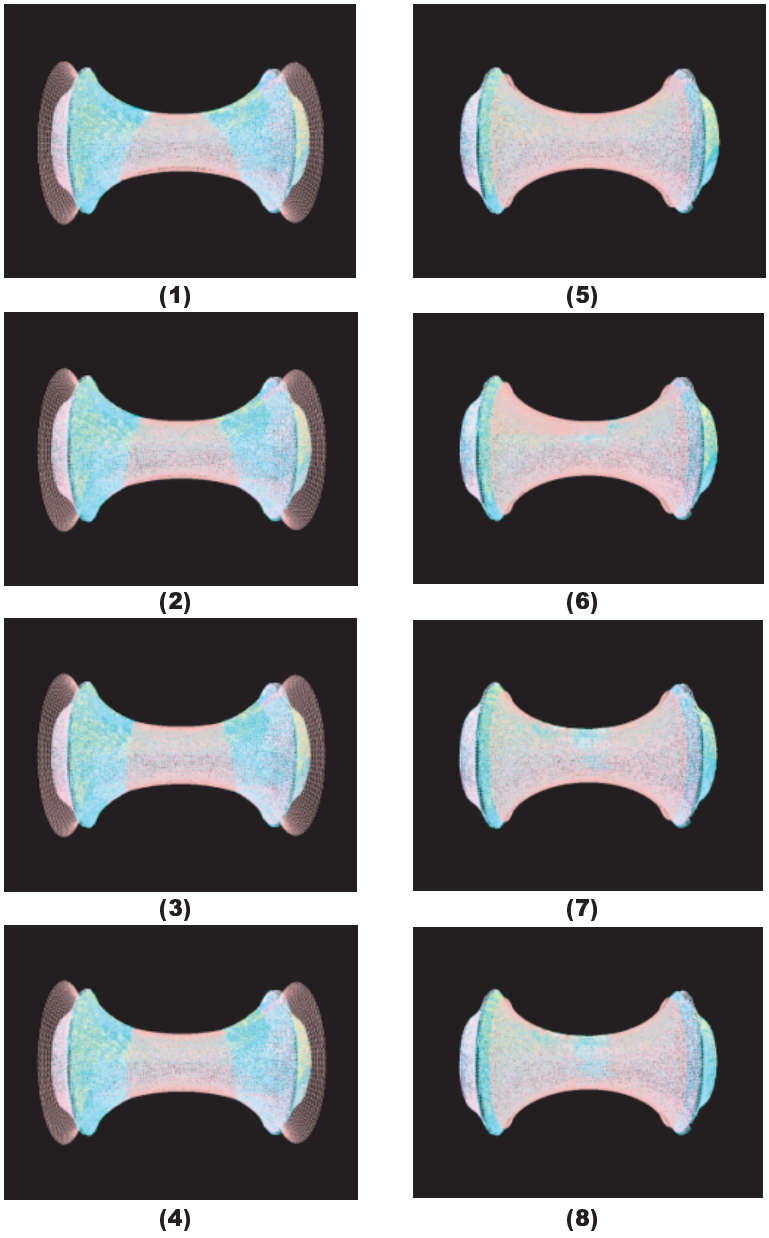


Figure 4.4: The convergent characteristic of the parametric data. (1)-(8) shows the order of convergence to the data of the revolution surface of catenary.

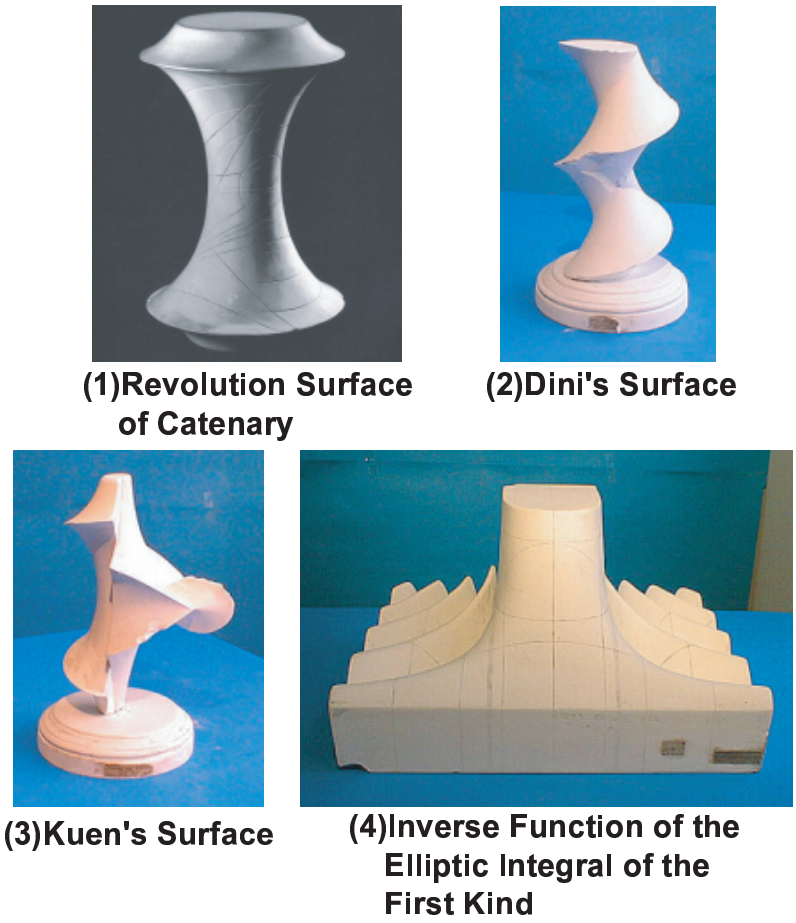


Figure 4.5: Mathematical models used in our experiment. (1), (2) and (3) have the constant negative curvature on all points of the model surface.

$$\frac{\partial X}{\partial b} = \left((l\cos u)\frac{\partial \phi}{\partial b}, (l\sin u)\frac{\partial \phi}{\partial b}, l\frac{\partial \psi}{\partial b}\right), \tag{4.15}$$

where

$$\frac{\partial \phi}{\partial b} = \cosh\frac{v}{a} \tag{4.16}$$

and

$$\frac{\partial \psi}{\partial b} = -\frac{b}{a^2}\sinh^{-1}\frac{v}{a} \tag{4.17}$$

$$\frac{\partial X}{\partial l} = \left(\frac{\cos t}{\cosh s}, \frac{\sin t}{\cosh s}, s - \tanh s + bt\right). \tag{4.18}$$

Estimated Parameter

$$a = 0.0568, \quad b = 0.0237, \quad l = 0.996$$

4.3.2 Dini's Surface

This surface is generated by spirally rotating a 2D tractrix and shown in Figure 4.6-(2). This surface has 2 parameters controlling its appearance. They are b and l.

Numerical Formula

$$for \quad {}^{\exists}b, {}^{\exists}l,$$

$$X(s,t) = \left(\frac{l\cos t}{\cosh s}, \frac{l\sin t}{\cosh s}, l(s - \tanh s + bt)\right), \tag{4.19}$$

$$where \quad |s| \le a \quad and \quad 0 \le t \le \theta$$

Partial Derivatives with respect to Shape Parameter

$$\frac{\partial X}{\partial b} = (0, 0, lt) \tag{4.20}$$

$$\frac{\partial X}{\partial l} = \left(\frac{\cos t}{\cosh s}, \frac{\sin t}{\cosh s}, s - \tanh s + bt\right), \tag{4.21}$$

Estimated Parameter

$$a = 0.309, \quad l = 0.0569$$

4.3.3 Kuen's Surface

Kuen's surface was first discovered by T. Kuen in 1884. This surface has several self intersections (indicated by the arrowed in Figure 4.6-(3)). Kuen's surface is a specialized case of Enneper's surface, and is characterized by two sets of lines or geodesics, where one of the geodesics has spherical curvature, while the other has planar curvature. Kuen's surface has a property known as Joachimstal's curvature, where all of the lines of planar curvature pass through a common axis; furthermore, all spherical curvature lines have midpoints lying on the same axis. Kuen's surface has the fixed appearance. Its size is controlled by only the scale parameter (l).

Numerical Formula

$$for \quad {}^{\exists}l,$$

$$X(u,v) = \left(l\frac{2(\cos u + u\sin u)\sin v}{1+u^2\sin^2 v}, l\frac{2(\sin u - u\cos u)\sin v}{1+u^2\sin^2 v}, l\left(\log\left(\tan\frac{v}{2}\right) + \frac{2\cos v}{1+u^2\sin^2 v}\right)\right), \tag{4.22}$$

$$where \quad \theta \le v \le \pi - \theta \quad and \quad \theta > 0$$

Partial Derivatives with respect to Shape Parameter

$$\frac{\partial X}{\partial l} = \left(\frac{2(\cos u + u\sin u)\sin v}{1+u^2\sin^2 v}, \frac{2(\sin u - u\cos u)\sin v}{1+u^2\sin^2 v}, \log\left(\tan\frac{v}{2}\right) + \frac{2\cos v}{1+u^2\sin^2 v}\right) \tag{4.23}$$

Estimated Parameter

$$l = 0.0491$$

4.3.4 Inverse Function of the Elliptic Integral of the First Kind

This function, defined as jacobi amplitude, is the inverse of the following function:

$$u = \int_0^{\phi} \frac{dt}{\sqrt{1-k^2\sin^2 t}} \tag{4.24}$$

Only the scale parameter (l) needs to be estimated to determine its actual appearance.

Numerical Formula

$$for \quad {}^{\exists}l,$$

$$X(\phi, k) = \left(l\phi, lk, l \int_0^{\phi} \frac{dt}{\sqrt{1 - k^2 \sin^2 t}} \right) \tag{4.25}$$

Partial Derivatives with respect to Shape Parameter

$$\frac{\partial X}{\partial l} = \left(\phi, k, \int_0^{\phi} \frac{dt}{\sqrt{1 - k^2 \sin^2 t}} \right) \tag{4.26}$$

Estimated Parameter

$$l = 0.0434$$

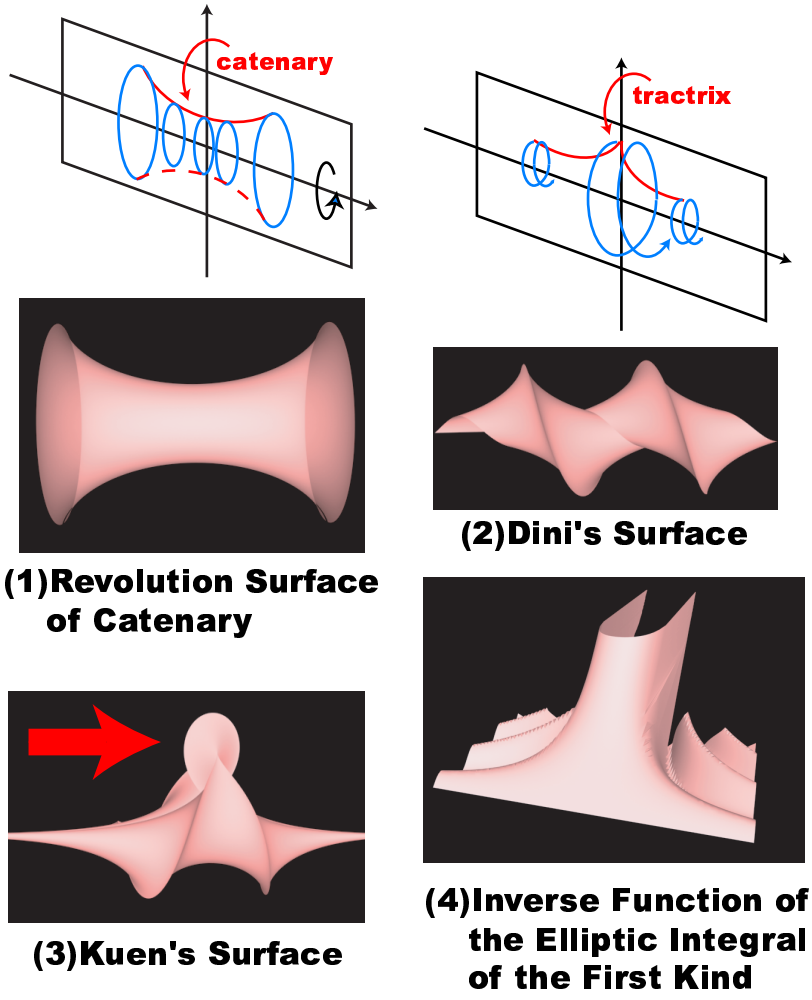


Figure 4.6: Ideal data of mathematical models.

Chapter 5

Shape Difference Visualization

In this chapter, we describe the visualization for difference between the measured data and the ideal data computed from its numerical formula. The shape parameters are estimated in the method described in the previous chapter. Although the algorithm is limited by the presence of the mathematical models, the limitation is not critical in the field of CAD, CAGD and CAM. To visualize the difference, we first define the "difference value". We then show the application to mathematical models [25], and introduce the application to ancient bronze mirrors.

5.1 Visualization Method

As a preparation of the shape difference visualization, the data of compared objects are aligned into the common coordinate.

Then we have to define the value representing the shape difference. As shown in Figure 5.1, the position of one object is fixed and this object is defined as a base object. The other object is defined as a check object and is aligned to the same pose and position of the base object. The check and base object is respectively shown in Figure 5.1 as the solid and dotted line. Difference between objects is the signed distance between the point on the check object and its corresponding point on the base object. A point in the check object has its nearest neighbor on the base object as its corresponding point. The shape difference is represented in mathematical form as:

$$d = sign(\overrightarrow{n(v)} \cdot \overrightarrow{vv_c}) \times |\overrightarrow{vv_c}|, \tag{5.1}$$

where v point on check object,
$\overrightarrow{n(v)}$ normal vector of v,
v_c corresponding point of v.

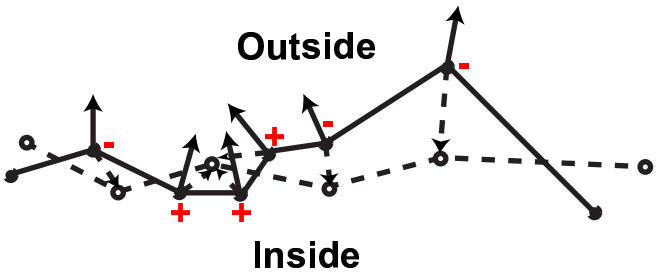


Figure 5.1: Signed distance calculation. The signed distance is inner product of the normal of the point on the check object and the vector from the point to the corresponding point on the base object.

and

$$sign(a) = \begin{cases} -1 & if \quad a < 0 \\ 1 & if \quad a \geq 0 \end{cases}$$

d is calculated on all points in the check object. The normal vector for each point is defined as the normalized vector of the sum of the normal vector of the plane that the point is the member. This value allows us to recognize their relative convex or concave shape difference to the check object. Positive d indicates that the base object is more convex; otherwise it is more concave.

5.2 Mathematical Model

In Chapter 4, we estimated the shape parameters of the mathematical models. Therefore we can evaluate the manufacturing accuracy of each model by visualizing the shape difference between the measured data of the model and the ideal data constructed according to the estimated parameters.

In this section, the shape difference between the actual and ideal models is observed. In each mathematical model, the ideal data is set as a base object. In each figure, green depicts no difference. Red and blue depicts that the actual object is more convex and concave, respectively. In Figures 5.2 and 5.3, the signed distance of the magnitude more than 1.5 *mm* is regarded as error, and in Figures 5.4 and 5.5, the difference of more than 4*mm* is regarded as error.

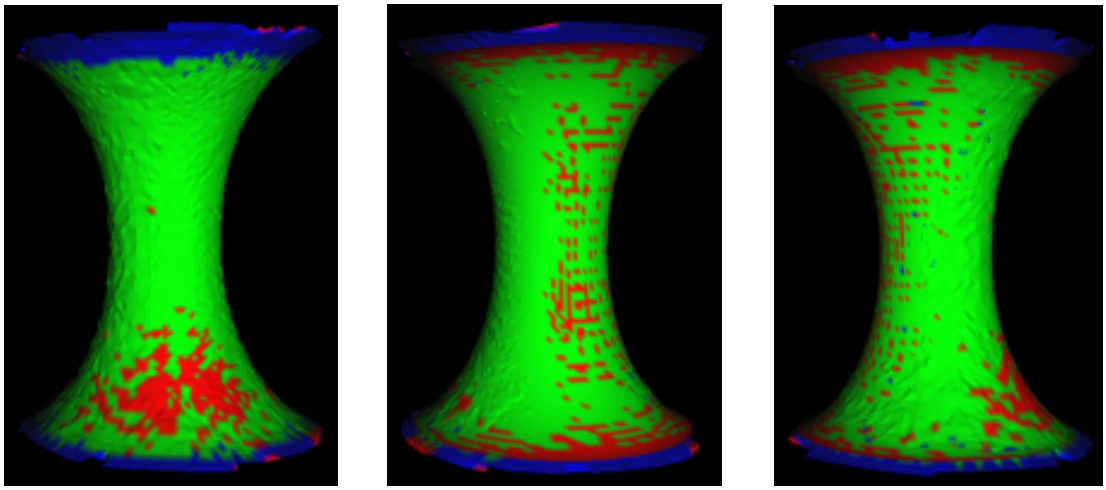


Figure 5.2: Shape difference of revolution surface of catenary

5.2.1 Revolution Surface of Catenary

Figure 5.2 indicates that the middle part of this model had almost no shape difference to the ideal model. Manufacturing error was concentrated on the upper and belower part (blue area).

5.2.2 Dini's Surface

Figure 5.3 shows the part generated at the minimum and maximum of the catenary curve contains the shape difference. This model contains the chipped part, and we could not determine whether the chipped points were created during or after the manufacture. In this experiment, we regarded them as the shape difference caused during the manufacture.

5.2.3 Kuen's Surface

Our experiment indicated that the difference between the mathematical model of Kuen's surface and its corresponding ideal model was widely spread (Figure 5.4). However, the difference of more than 4 *mm* is not visually distinguishable.

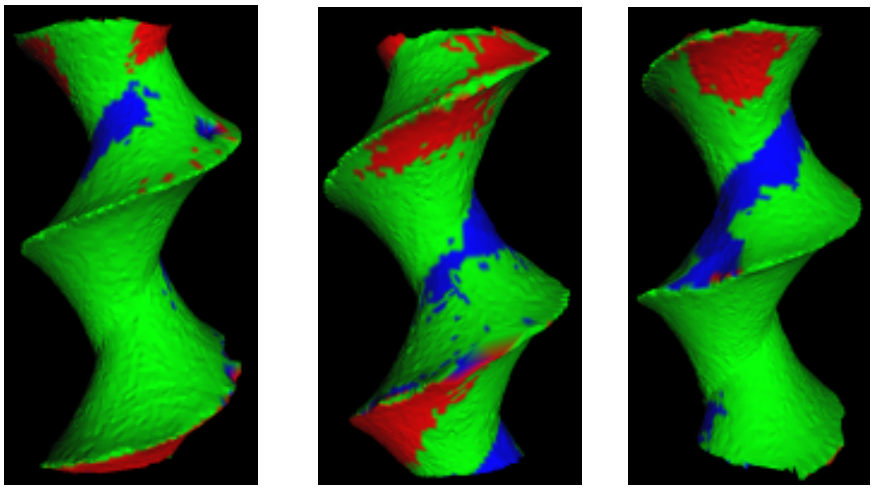


Figure 5.3: Shape difference of Dini's surface

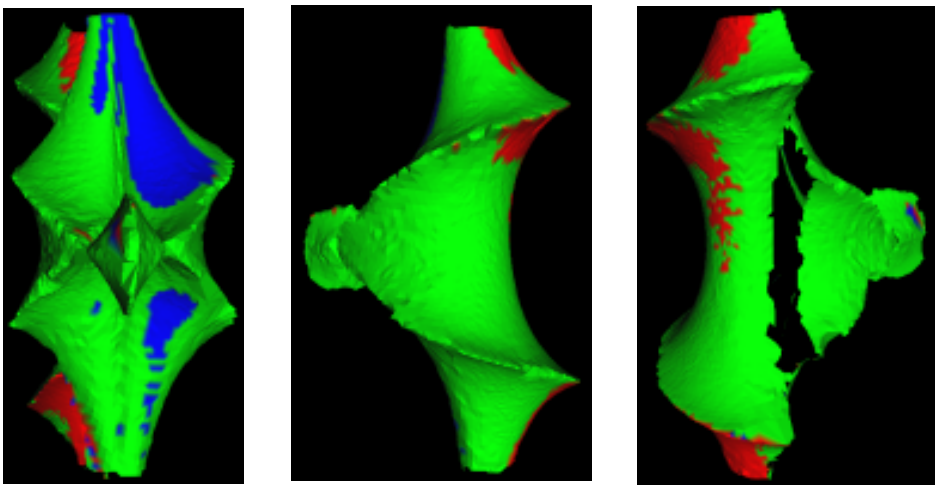


Figure 5.4: Shape difference of Kuen's surface

5.2.4 Inverse Function of the Elliptic Integral of the First Kind

5.5 shows that difference to the ideal model was found in lots of area. In contrast to Kuen's surface, the difference can be observed visually. In the figure, area with the distinct shape difference lies between the arrows.

5.3 Ancient Bronze Mirrors

Ancient Chinese bronze mirrors have the immortal and beast pattern in their reverse side with protruding rim of triangular cross section (Figure 5.6).

The mirrors are intensely studied because it may lead to the location of Yamatai state, the oldest documental state in Japan. Some archaeologists propose that Yamatai state governed the regions, where these mirrors were found. In this section, we describe how to analyze the mirrors and the usefulness of the visualization of the difference technique between the mirrors' shape. We then show the application results.

5.3.1 Mirror Analysis and Difference Visualization

The ancient bronze mirrors, with the same decoration pattern, were excavated from the various areas. These mirrors are cast from the same mold, or the mold duplicated from the similar mirrors. The former is defined as "individual mirrors", and the latter "identical mirrors". Individual and identical mirrors can be classified as "sibling mirrors". So far, archaeologists have paid attention to the classification of sibling mirrors according to the decoration patterns so far [26], they propose that the classification of individual and identical mirrors implies the order of the manufacture. The local difference of the decoration pattern indicated the duration of usage of the mold or its duplication mold. Continuous usages causes the unique spreading fissure or cracks of the decoration pattern are reflected in their decoration pattern.

So far, archaeologists have analyzed such shape difference by visually observing the original mirrors. The practice requires a huge amount of time and effort. If the candidates caused during the manufacturing process are automatically detected, mirror analysis can be performed much faster. So we developed the detection method of shape difference through 3D data on computer graphics [27]. An additional advantages of the 3D data is that the object will not receive the further damage.

5.3.2 Application Result

We performed the shape difference visualization to the five individual mirrors. They are distinguished by the following name: Dchoh01, Dchoh02, Dhebo02-63, Dnisik-75 and Dsamida08

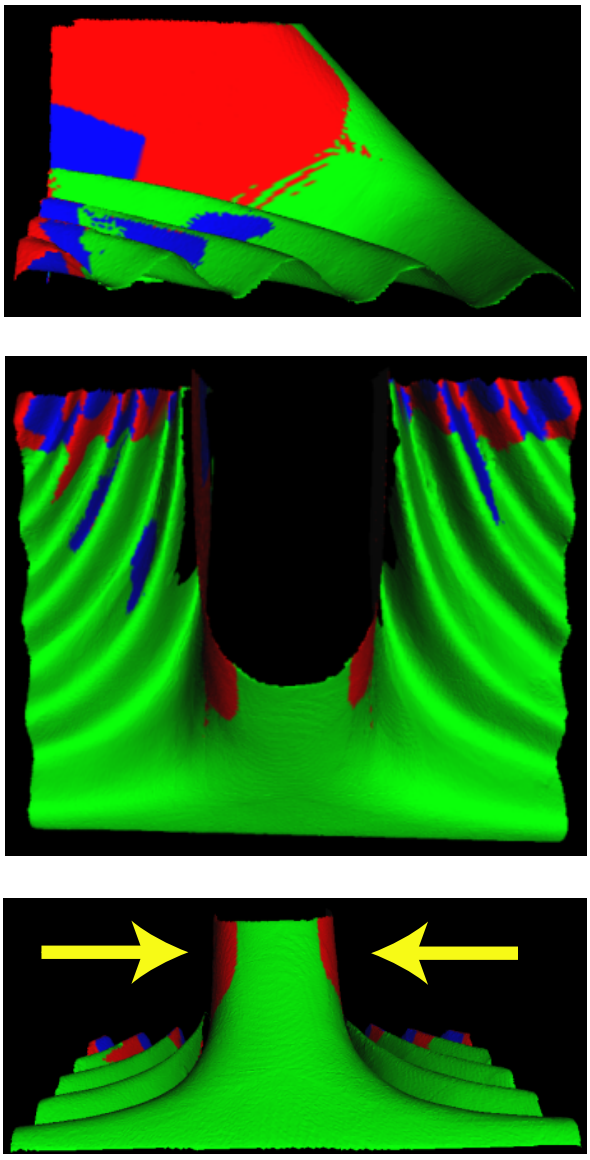


Figure 5.5: Shape difference of inverse function of the elliptic integral of the first kind

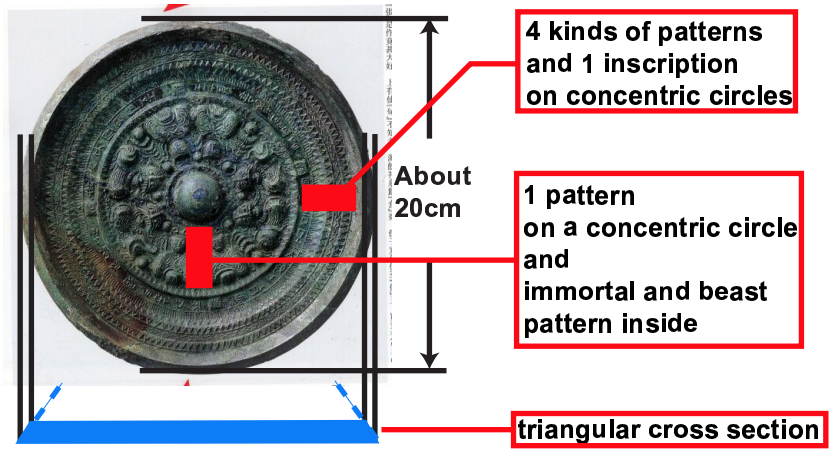


Figure 5.6: Ancient Chinese bronze mirror

(Figure 5.7).

The shape differences between Dsamida08 to the other bronze mirrors were observed, because Dsamida08 was not the entire mirror, but merely a combination of the fractured parts. So the observation of the five mirrors was possible only on the available parts of Dsamida08.

According to the algorithm in Section 5.1, Dsamida08 and the other mirrors are aligned in the same coordinate. The corresponding points are extracted for all points in any pair of mirrors in order to calculate the difference value.

In our experiment, the difference of less than 0.3*mm* was ignored. In Figure 5.8, the green, red and blue areas indicates no difference, convex shaped difference and concave shaped difference, respectively. Remark that the first comparison result contained the global differences. Global difference is the result of mirror bending, partial sinking, or internal cracking. It is caused by time and usage, and is unrelated to the sibling relationship. It should be rejected.

Areas containing the global difference were realigned to visualize the local difference. Local differences after the realignment and those found in the first comparison result were integrated and shown in Figure 5.9.

Figure 5.10 demonstrates the deteriorating pattern of the mold upon usages. Convex shape on a mirror corresponds to the concave part of the mold. This part tends to be worn out when the mold is used; therefore, corresponding part of latter manufactured mirrors will be more convex.

We further observed the local difference inside the circles in Figure 5.9. Figures 5.11, 5.12 and 5.13 respectively show the cross sections of the part marked by the yellow, blue and white

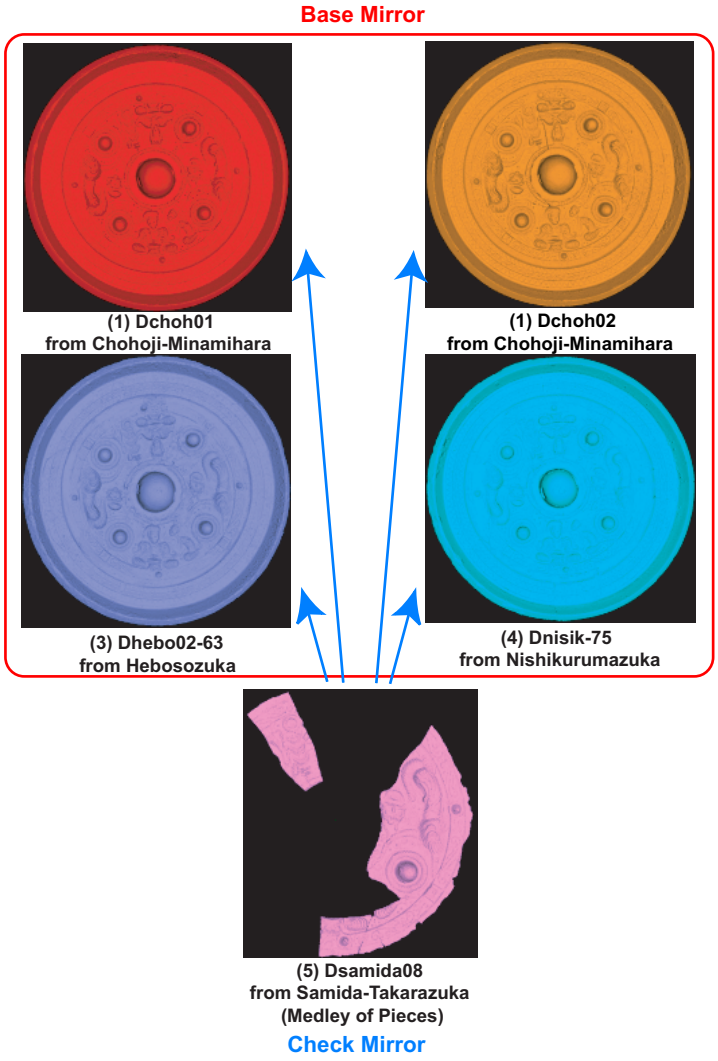


Figure 5.7: Identical mirrors. Dsamida08 is set as a check mirror, and the other mirrors are set as its base mirrors.

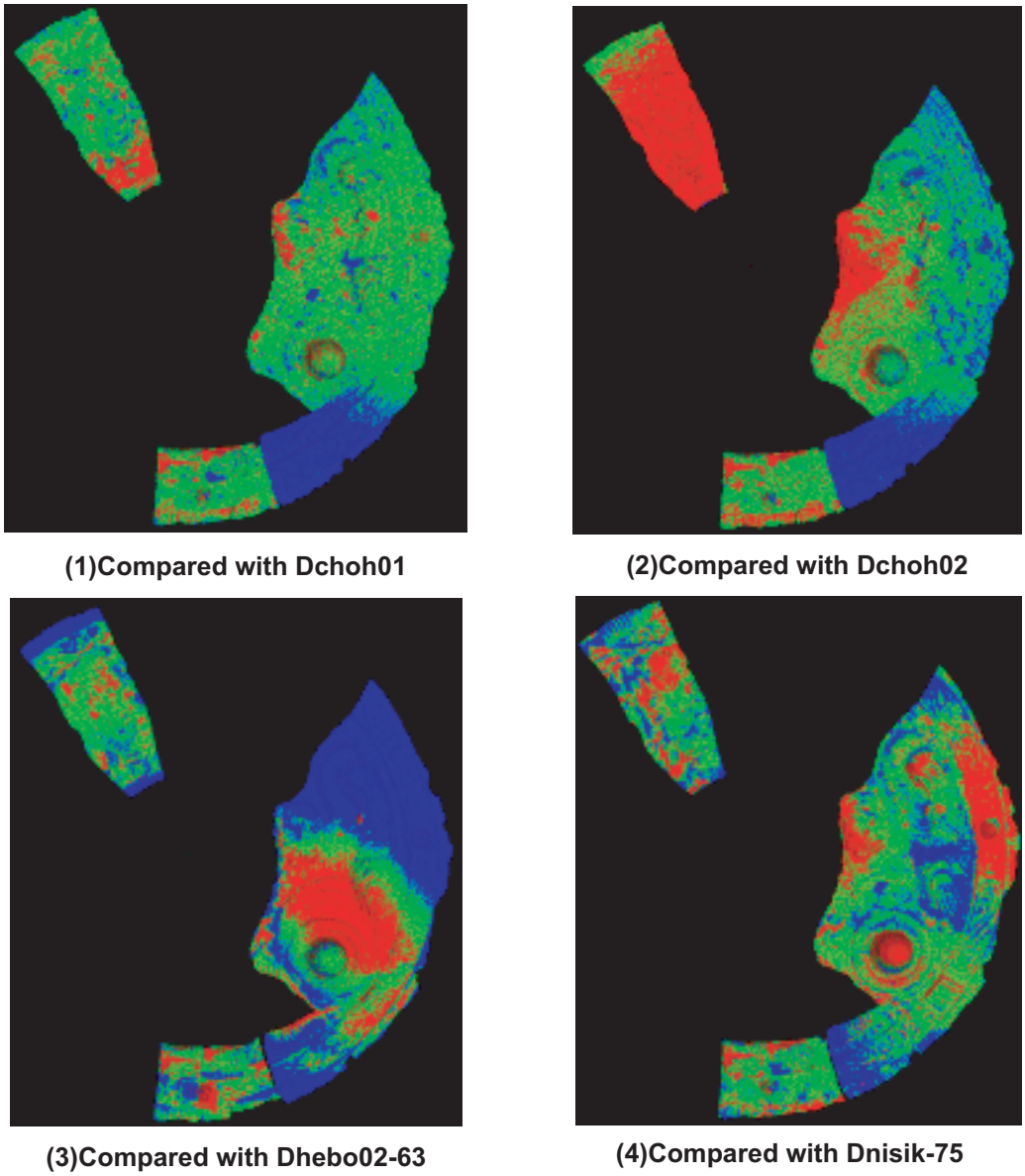


Figure 5.8: Shape differences of Dsamida08 compared to the rest four mirrors. Global differences are included in this stage.

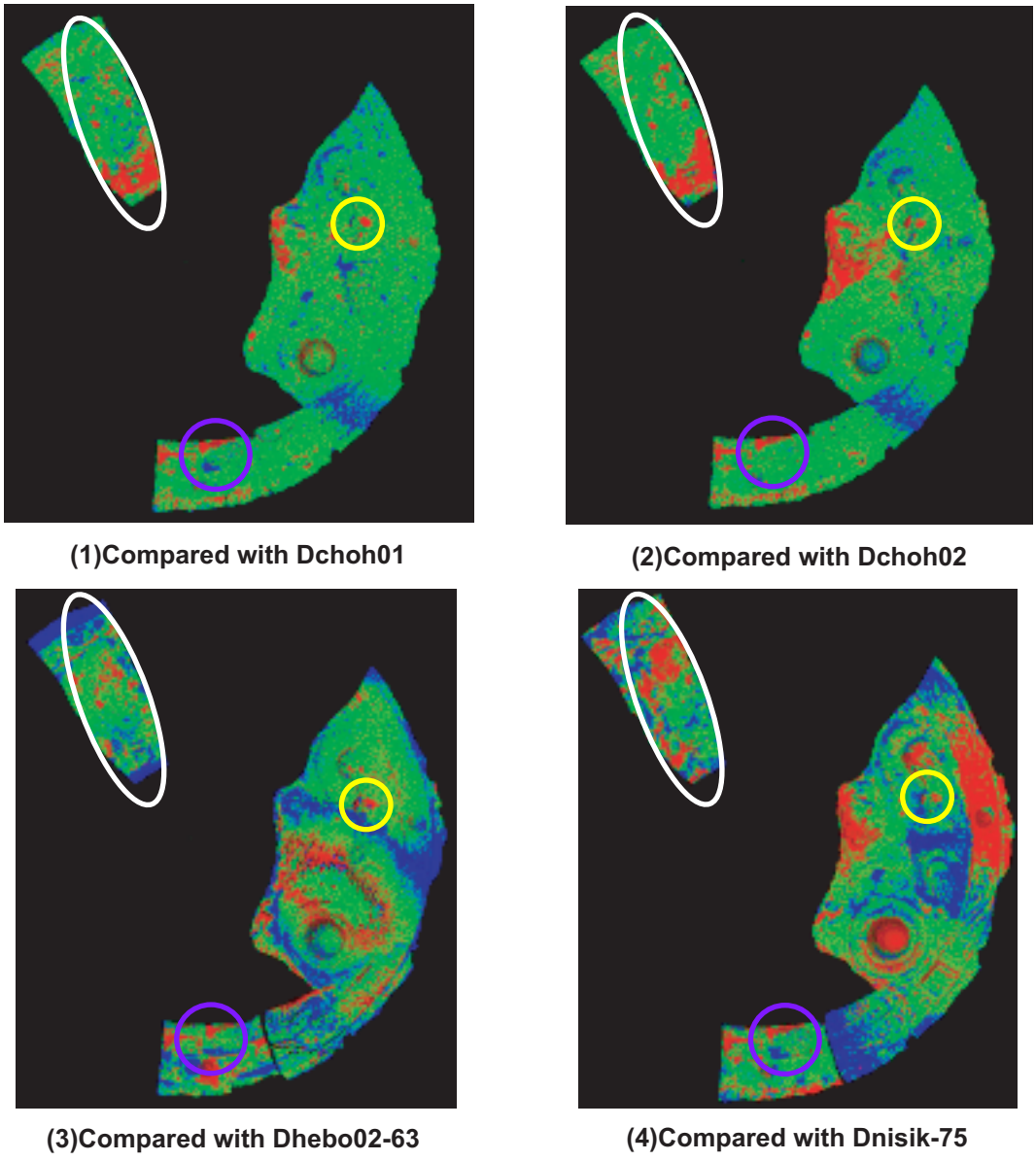


Figure 5.9: Local Shape differences to Dsamida08 without global differences.

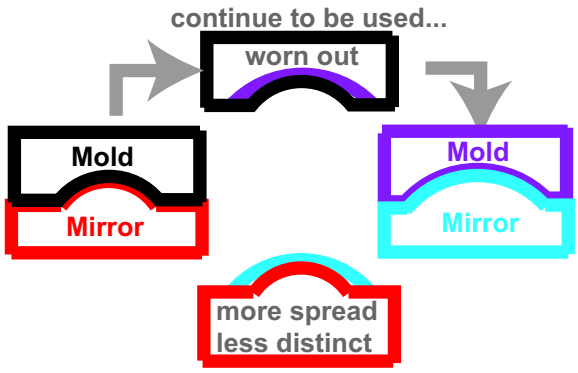


Figure 5.10: The deteriorating pattern of the mold by continuous usage.

circles in five mirrors. In Figure 5.11, the arrowed indicated the shape caused by the chipped mold. By observing the growth of this chip on Dchoh01, Dnisik-75 and Dsamida08, the order of manufacturation could be found as Dsamida08, Dchoh01 and Dnisik-75. In Figure 5.12, the concave shape indicated by the arrow implied the manufacturing order as Dsamida08, Dchoh02, Dchoh01, Dhebo02-63 and Dnisik-75. In the same manner, the manufacturing order according to the local difference in Figure 5.13 was evaluated and the result was consistent with those from Figures 5.11 and 5.12. Thus, we conclude that Dsamida08, Dchoh02, Dchoh01, Dhebo02-63 and Dnisik-75 were made in this order.

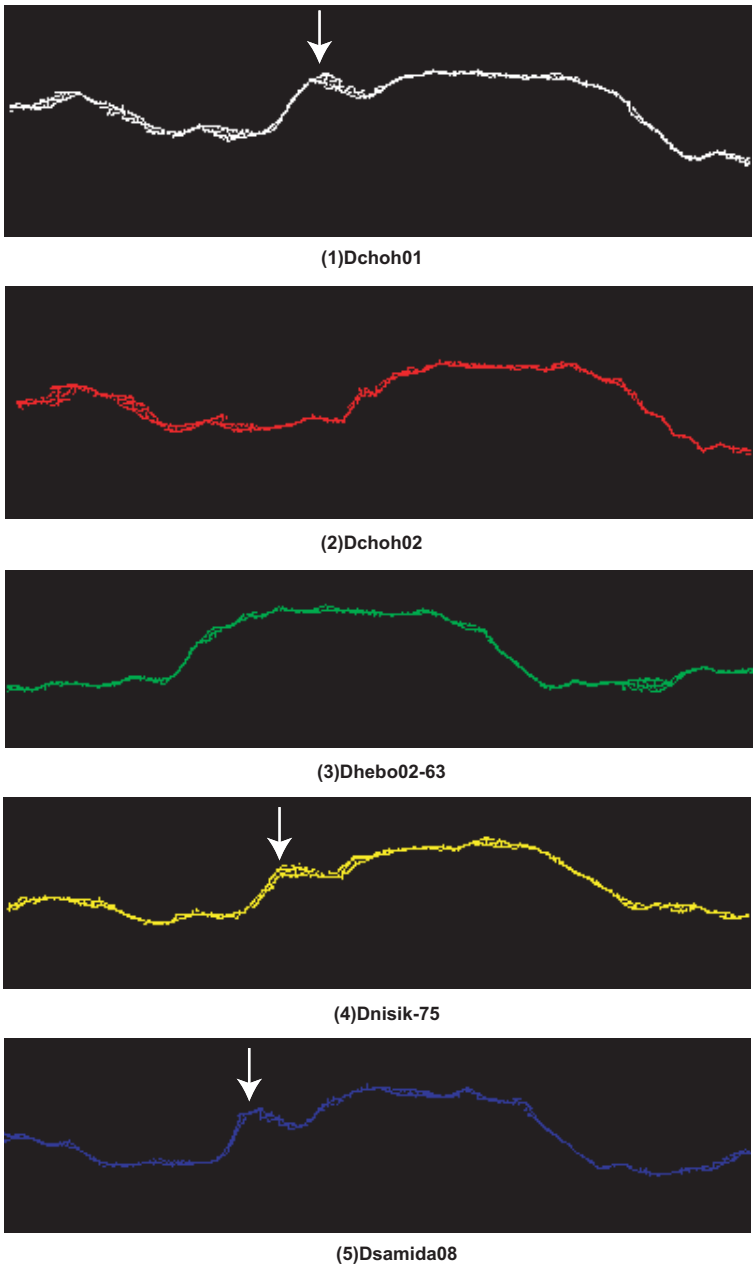
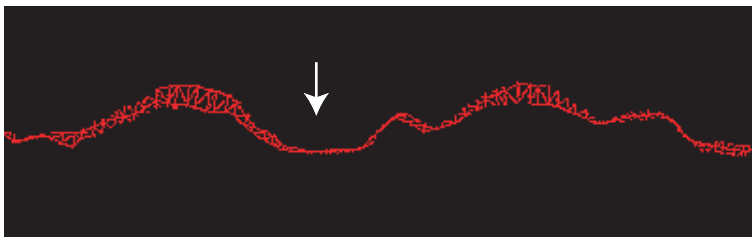
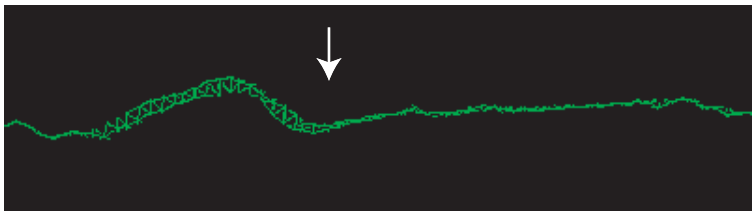


Figure 5.11: Cross section of the mirror inside the yellow circle in Figure 5.9

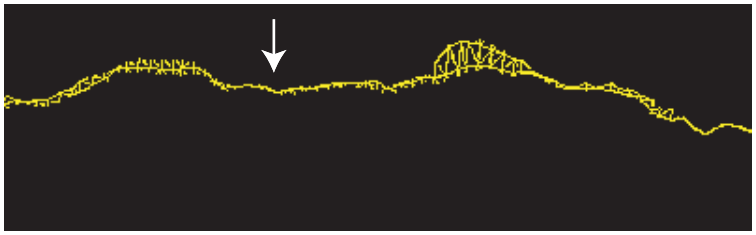
(1)Dchoh01



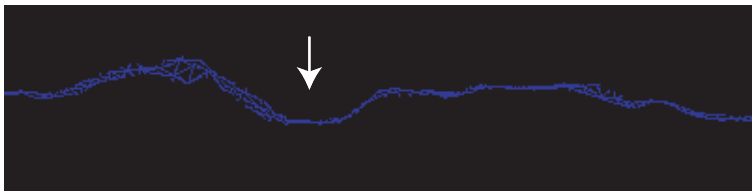
(2)Dchoh02



(3)Dhebo02-63



(4)Dnisik-75



(5)Dsamida08

Figure 5.12: Cross section of each mirror inside the purple circle in Figure 5.9

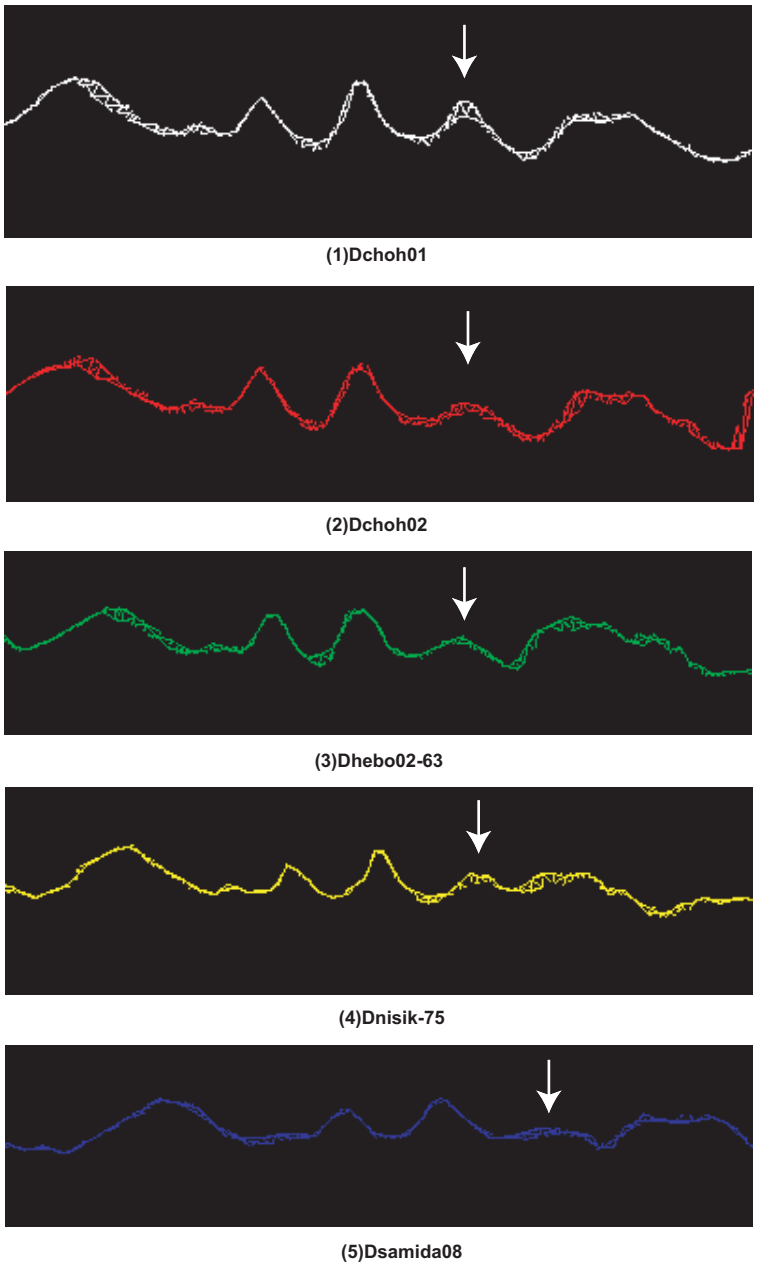


Figure 5.13: Cross section of each mirror inside the white circle in Figure 5.9

Chapter 6

Evaluation

In Chapter 4, the shape parameters are estimated. The estimation is affected by various kinds of error: measurement error, initial registration error and error in initially input parameter. In this chapter, we measure the estimation errors in each process, and evaluate the accuracy of the estimated parameters.

6.1 Measurement Error of 3D Data

The first error in the estimation is the result of the measurement error from the range finder. In our study, MINOLTA VIVID 900 laser range finder was used to measure the actual objects. It has three kinds of lens: tele, middle and wide. Tele lens gives highly accurate data, but its effective scanning area is narrow (10×10 cm^2 at 1 m from the laser range finder). Opposite to tele lens, wide lens can scan much wide area (1×1 m at 3 m^2), but has lower accuracy. Middle lens has the intermediate property of tele and wide lens.

The measurement noise depends on lots of factors. In this study, we investigate the relationship of measurement noise to the pose of the actual object. In this experiment, the data of the white and lusterless plane at the different poses were obtained. The system is shown in Figure 6.1. The white board was set on the turn table. The pose was changed by rotating and translating the turn table. We denote l and θ by the distance from the white board to the laser range finder and the angle between the normal of white board to the ray of the laser. That is, the pose was controlled by l and θ. Five data were obtained at each pose. Principal component analysis (PCA) was applied to analyze the point set. The plane containing the point set was estimated and the standard deviation of plane data was obtained. The change of the standard deviation of the plane data according to each type of lens are shown in Figures 6.2, 6.3 and 6.4.

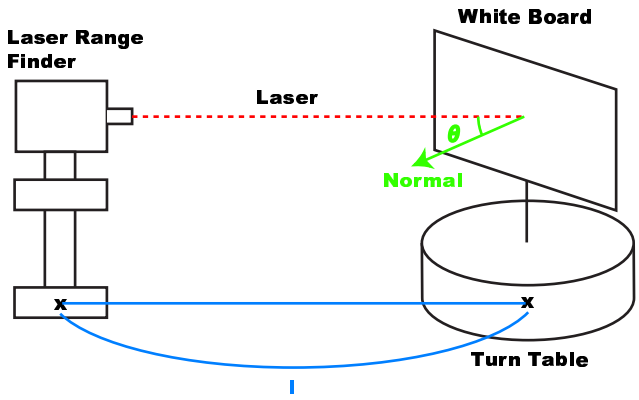


Figure 6.1: System to evaluate measurement error. The white board is set on the turn table, and can be rotated around the vertical axis. In this system, the incident angle and the distance from the white board to the laser range finder are changeable

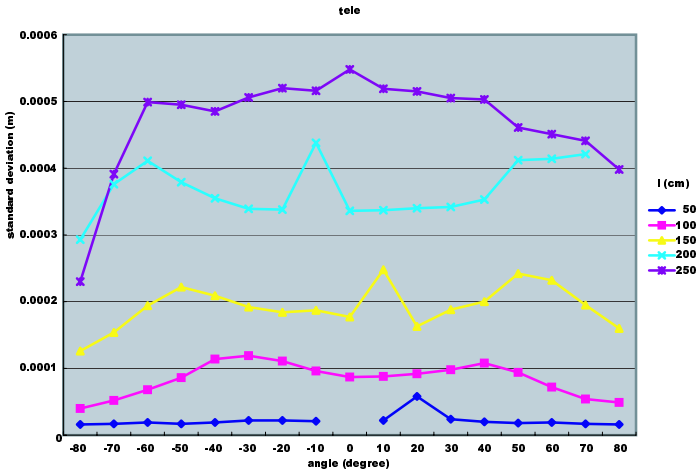


Figure 6.2: Graph showing the relationship between incident angle and standard deviation, when tele lens was applied. Each curve showed the data at different distance from white board to the finder. Standard deviation of the data is approximately $0.1mm$ within $1m$ distance. The abnormal peaks are caused by the condition of white board, such as dust.

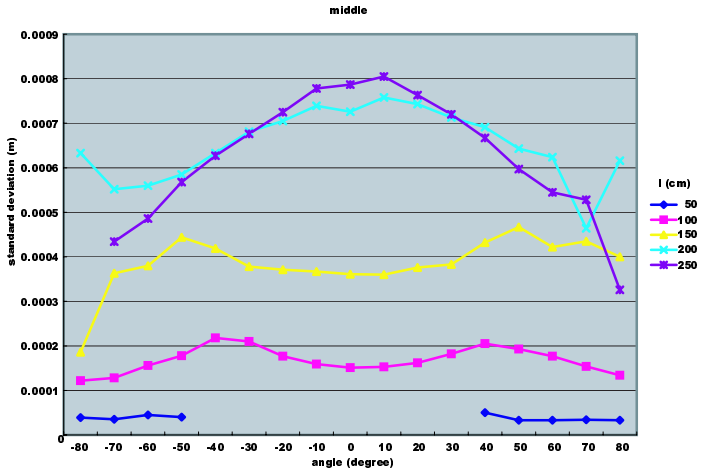


Figure 6.3: Graph showing the relationship between incident angle and standard deviation, when middle lens was applied. Each curve showed the data at different distance from white board to the finder. Similar change of the standard deviation are observed within 2 *m* distance.

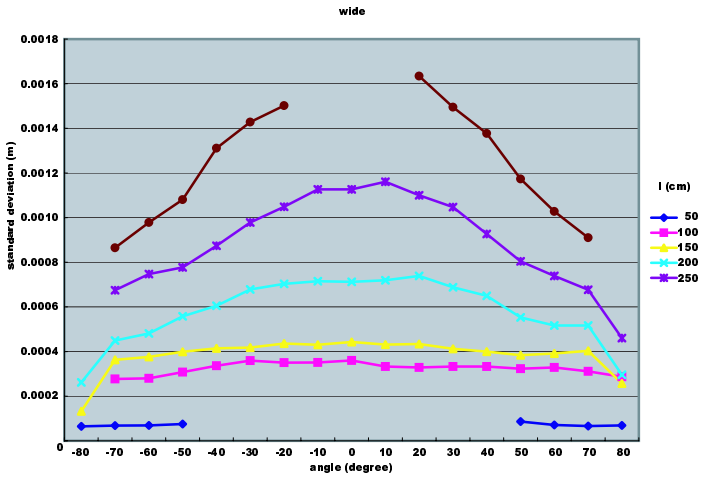


Figure 6.4: Graph showing the relationship between incident angle and standard deviation, when wide lens was applied. Each curve showed the data at different distance from white board to the finder. The data within 0.5 *m* could not be obtained, and there were large amount of noise when the distance exceeded 2 *m*.

6.2 Accuracy of Shape Parameter

In this experiment, we investigated the accuracy of estimated parameter. Parameters were manually defined to construct the ideal data. The estimated parameters were referred as the true parameters. Noise with the standard deviation observed in the previous section was added to the ideal data. We then estimated the parameters from the noisy data. The initial parameters in the estimation were subtly different than the true parameters. Since the correct parameters are known, the accuracy of the estimation can be measured.

In this experiment, we use the model of revolution surface of catenary. The shape parameters were set as $a = 0.05$, $b = 0.02$ and $l = 1.00$. Gaussian noise was added into the ideal data to form the noisy data.

6.2.1 Influence of Measurement Error

We investigated the influence of the measurement noise. Noise with different standard deviation was added into the ideal data. Standard deviation used in this experiment are 0.01, 0.1, 1.0, 10.0. Ten noisy data were created for each standard deviation. These noisy data were regarded as the data measured in practice. The initial pose, position and parameter of the virtual "measured data" were the same as the one of the computed (ideal) data. The number of iteration was set at 15. Figures 6.5, 6.6 and 6.7 show the maximum and the minimum of estimated parameter, a, b and l, respectively.

Effect of the noise standard deviation to the estimated parameters were similar in all parameters. The larger the standard deviation was, the larger range of the maximum and minimum parameter was. Noise added in this experiment was far higher than the observed noise. Even though noise with the standard deviation of 0.01 was added, the error in our results was low. In previous section, the maximum standard deviation of the measurement noise in MINOLTA VIVID 900 was detected at less than 0.002, the result indicated the robustness of our estimation method against the sensor noise.

6.2.2 Influence of Initial Registration

In this section, we investigate the influence of the initial pose and position. In the same manner as the previous section, we added the noise to the ideal data. The noisy data were regarded as the data obtained in practice. The standard deviation of the noise (σ) was set to 0.0004 according to the observed measurement error in our previous experiment. The initial value of every parameter was set at the true one. The coordinate of the model is shown in Figure 6.8

Effects of translation and rotation were investigated separately. For translation, noisy data were translated 0.01, 0.02 and 0.03 m along x or z axis, respectively. For rotation, three noisy

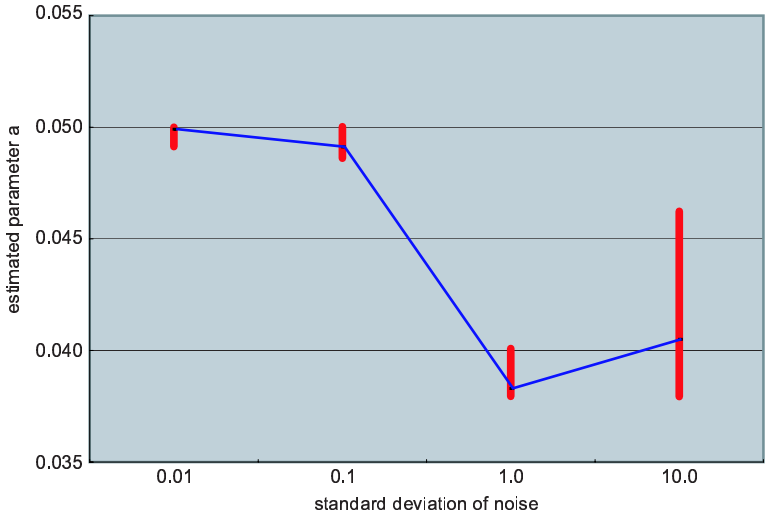


Figure 6.5: The maximum and the minimum of estimated parameter *a*.

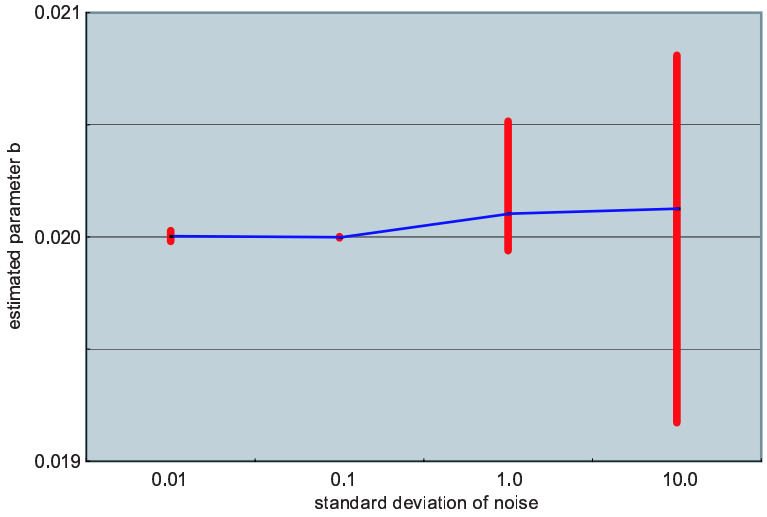


Figure 6.6: The maximum and the minimum of estimated parameter *b*.

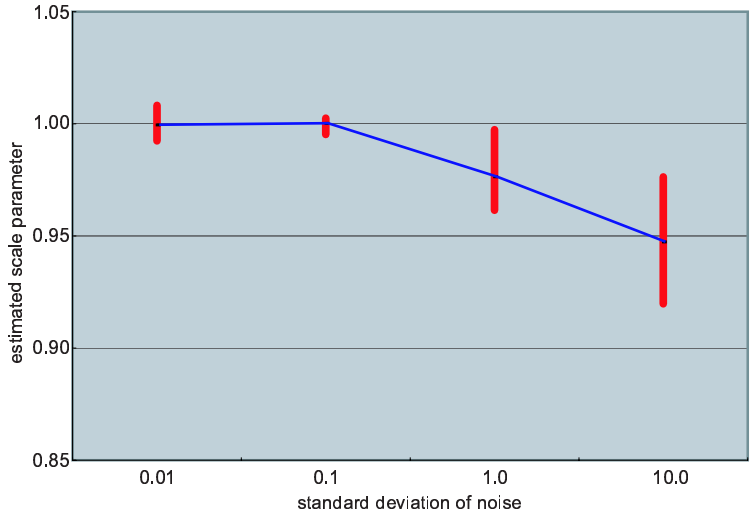


Figure 6.7: The maximum and the minimum of estimated parameter *l*.

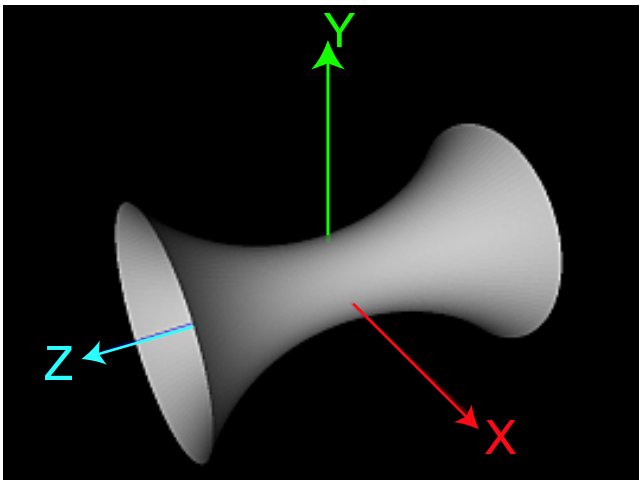


Figure 6.8: Model used in the experiment and its 3D coordinate.

0.01 *m* along x axis	0.02 *m* along x axis	0.03 *m* along x axis
a = 0.0500	a = 0.0483	a = 0.0478
b = 0.0188	b = 0.0188	b = 0.0192
l = 0.9685	l = 0.9464	l = 1.0072
0.01 *m* along z axis	0.02 *m* along z axis	0.03 *m* along z axis
a = 0.0458	a = 0.0457	a = 0.0451
b = 0.0204	b = 0.0218	b = 0.0219
l = 0.9474	l = 0.8909	l = 0.9004
10 degrees around x axis	20 degrees around x axis	30 degrees around x axis
a = 0.0493	a = 0.0492	a = 0.0471
b = 0.0197	b = 0.0208	b = 0.0208
l = 0.9403	l = 0.9403	l = 0.9624

Table 6.1: Estimation result for shape parameters according to initial registration error.

data were rotated 10, 20 and 30 degrees around x axis. Since revolutional surface of catenary has the x, y and z symmetry, these translation and rotation were sufficient for the evaluation. Results of estimation were shown in Table 6.1. In the table, "0.01 *m* along x axis" and "10 degrees around x axis" mean the noisy data is translated 0.01 *m* along x axis and rotated 10 degrees around x axis, respectively.

Z axis for this surface is the direction of expanding/contracting. The manipulation on z axis results in the ambiguity in the parameters. In contrast, manipulation on x axis is not directly related to the expansion and contraction. The translation in z axis simultaneously affects the pose/position and parameters. As the result indicated, 15 iteration was not sufficient for good estimation for the translation on z axis, while it was sufficient for the translation on and rotation around x axis. 45 iteration was required for the good estimation for model undergone z axis translation.

6.2.3 Influence of Initial Parameter

We investigate the influence of initial value for the estimation. The virtual "measured data" were made in the same manner as the previous sections, and the standard deviation of the noise (σ) was set to 0.0004. The initial pose and position of the "measured data" were the same as the one of the ideal data. Effect of initial value of each parameter was studied, so only one parameter had its initial value changed from the true value at a time. The amount of change was α, where α varied between parameter. Table 6.2 shows the estimation result.

a = 0.07	a = 0.06	a = 0.04	a = 0.03
a = 0.0535	a = 0.0509	a = 0.0491	a = 0.0482
b = 0.0210	b = 0.0224	b = 0.0185	b = 0.0184
l = 0.9703	l = 0.9824	l = 0.9487	l = 0.9061
b = 0.03	b = 0.025	b = 0.015	b = 0.01
a = 0.0560	a = 0.0560	a = 0.0440	a = 0.0440
b = 0.0209	b = 0.0204	b = 0.0196	b = 0.0191
l = 0.8709	l = 0.8713	l = 1.1202	l = 0.8710
l = 1.30	l = 1.15	l = 0.85	l = 0.70
a = 0.0483	a = 0.0474	a = 0.0517	a = 0.0440
b = 0.0282	b = 0.0193	b = 0.0216	b = 0.0224
l = 1.0261	l = 1.0268	l = 0.9752	l = 0.9736

Table 6.2: Estimating result for noisy data when initial parameters was not correctly set.

The expression above each table depicts the initial input value for the parameter. For example, "$a = 0.07$" means parameter a is initially set to 0.07, while the other parameters were set at its true value.

Table 6.2 indicated that the effect of incorrect initial value on each parameter was not equal. In this experiment, parameter b gave the highest effect.

Chapter 7

Conclusion and Future Work

In this thesis, we proposed the estimation method for the shape parameters to the object. It is required that the shape could be represented by some numerical formula. We first introduced the various strategies of the registration methods, and describe their advantage and disadvantage. We selected the simultaneous and point-based strategies as our base method because we extend the registration of pose and position parameters in conventional registration to the registration of pose, position and shape parameter.

Kd-tree is applied in the registration algorithm. To improve the speed of the registration, we develop the novel method for pruning trees in searching. Experiments showed that the number of the visited leaf was greatly reduced and indicated the decreasing computation cost.

From the estimated shape parameters, the ideal 3D model of the object can be constructed. Difference to the ideal model was visualized. In this thesis, difference of the "mathematical models" were investigated. Then we described the application of the difference visualization in archaeology.

Finally, we evaluated the estimating accuracy of the shape parameter. The experiment showed that the accuracy of our parameter estimation is not so affected by the measurement noise as by the initial pose, position and parameter.

We plan to apply our proposed technique to the geometrical field. For example, we would like to convert the measured model into the combination of CAD primitive models. The measured data is segmented to the combination of CAD primitives, and the model parameters of each primitive are estimated. The model then be represented by groups of primitive parameter instead of point cloud, so the data for a model is greatly reduced. The advantage is more distinct when the model contains of the large amount of points. Normal processing in CG is unable to deal with such large data. Compared to point cloud, CAD primitives and parameters are much more smaller and can be effectively handle. Thus, the research of fitting objects to

CAD primitive paves the way for increasing versatility of “Modeling from Reality”.

References

[1] D. Miyazaki, T. Oishi, T. Nishikawa, R. Sagawa, K. Nishino, T. Tomomatsu, Y. Yakase, and K. Ikeuchi. The great buddha project: Modelling cultural heritage through observation. In *Proceedings of the 6th International Conference on Virtual Systems and Multi-Media*, pages 138–145, October 2000.

[2] M. Levoy et al. The digital michelangelo project: 3d scanning of large statues. In *ACM SIGGRAPH Proceedings*, pages 131–144, July 2001.

[3] F. Bernardini and H. Rushmeier. The 3d model acquisition pipeline. In *Eurographics 2000 State of the Art Report (STAR)*, August 2000.

[4] K. Nishino and K. Ikeuchi. Robust simultaneous registration of multiple range images. In *Proceedings of the 5th Asian Conference on Computer Vision*, volume 2, pages 455–461, January 2002.

[5] 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):239–256, February 1992.

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

[7] T. Masuda and N. Yokota. A robust method for registration and segmentation of multiple range images. *Computer Vision and Image Understanding*, 61(3):295–307, 1995.

[8] G. Turk and M. Levoy. Zipped polygon meshes from range images. In *ACM SIGGRAPH Proceedings*, pages 311–318, July 1994.

[9] A.E. Johnson and M. Herbert. Surface matching for object recognition in complex 3-dimensional scenes. *Image and Vision Computing*, 16(9/10):635–651, July 1998.

[10] K. Higuchi, M. Herbert, and K. Ikeuchi. Building 3-d models from unregistered range images. In *Graphical Models and Image Processing*, volume 57, pages 315–333, July 1995.

[11] David Simon. *Fast and Accurate Shape-Based Registration*. PhD thesis, School of Computer Science, Carnegie Mellon University, 1996.

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

[13] G. Blais and M. Levine. Registering multiview range data to create 3d computer objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(8):820–824, 1995.

[14] P. Neugebauer. Geometrical cloning of 3d objects via simultaneous registration of multiple range images. In *Proceedings of International Conference on Shape Modeling and Application*, pages 130–139, March 1997.

[15] A.E. Johnson and S. Kang. Registration and integration of textured 3-d data. In *Proceedings of International Conference on 3D Digital Imaging and Modeling*, pages 234–241, May 1997.

[16] Szymon Rusinkiewicz and Marc Levoy. Efficient varinats of the icp algorithm. In *Proceedings of the 3rd International Conference on 3D Digital Imaging and Modeling*, pages 145–152, May 2001.

[17] J. L. Bentley J. H. Friedman and R. A. Finkel. An algorithm for finding best matches in logarithmic expected time. *ACM Transactions on Mathmatical Software*, 3(3), September 1977.

[18] J.L. Bentley. K-d trees for semidynamic point sets. In *Proceedings of the 6th Annual Symposium on Computational Geometry*, pages 187–197, 1990.

[19] Mark D. Wheeler. *Automatic Modeling and Localization for Object Recognition*. PhD thesis, School of Computer Science, Carnegie Mellon University, 1996.

[20] T. Masuda, K. Sakaue, and N. Yokota. Registration and integration of multiple range images for 3-d model construction. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 879–883, June 1996.

[21] K. Pulli. Multiview registration for large data sets. In *Proceedings of the 2nd International Conference on 3D Digital Imaging and Modeling*, pages 160–168, 1999 October.

[22] E. Polak. *Computational Methods in Optimization*. New York: Academic Press, 1971.

[23] David A.H. Jacobs. *The State of the Art in Numerical Analysis*. London: Academic Press, 1977.

[24] J. Stoer and R. Bulirsch. *Introduction to Numerical Analysis*. New York: Springer-Verlag, 1980.

[25] T. Masuda, R. Kurazume, M. Ikemizu, Y. Nishino, and K. Ikeuchi. Simultaneous parameter estimation by using simultaneous alignment (in japanese). In *Proceedings of Meeting on Image Recognition and Understanding*, volume 1, pages 171–176, July 2002.

[26] M. Misaki. Theoretical studies on categorize ancient copper mirrors to base on digital shape measurement (in japanese). Master's thesis, Department of Information Processing, Graduate School of Information Science, Nara Institute of Science and Technology, 1999.

[27] T. Masuda, S. Imazu, T. Furuya, K. Kawakami, and K. Ikeuchi. Shape difference visualization for old copper mirrors through 3d range images. In *Proceedings of The Eighth International Conference on Virtual Systems and Multimedia (VSMM 2002)*, pages 942–951, September 2002.

Appendix A

Quaternion for Rotation

In this Appendix, we explain the quaternion for the representation of the object rotation in detail. First we discuss the convenience of the quaternion. Next, we show its basic operations and the relationship between quaternion and rotation. Finally, we derive the differential of the 3×3 rotation matrix with respect to the quaternion parameters.

A.1 Convenient Quaternion Representation for Rotation

When the point $\overrightarrow{x_i}$ is moved to its corresponding point $\overrightarrow{y_i}$ in 3D space, we can compute the rotation matrix **R** and translation vector $\overrightarrow{t}$ as follows.

$$\mathbf{R}\overrightarrow{x_i} + \overrightarrow{t} = \overrightarrow{y_i} \tag{A.1}$$

This equation is linear, but the rotation matrix **R** has the non-linear constraints on the 9 elements because the rotation in 3D space is represented by only 3 independent variables.

Some methods have been developed in order to solve this constraint problem. One of them is the quaternion representation. The quaternion is composed of one scaler and three-element vector part according to the following equation:

$$q = (\overrightarrow{n}, s), \tag{A.2}$$

$$where \quad \overrightarrow{n} = (u, v, w)$$

Intuitively speaking, this representation means the rotation in proportion to the magnitude of s around the vector $\overrightarrow{n}$.

The advantages of this rotational representation are:

- Constraint is simplified by normalizing four parameters of the quaternion.

- Quaternion operations can be formulated from the ones of the matrix and vector.
- Inverse rotation quaternion is easily obtained by negating the scalar part.
- Quaternion representation is easily converted into axis-angle representation.
- Rotation matrix is easily obtained from the quaternion.
- Rotation is in the closed form through the quaternion when the corresponding between 3D point sets is taken.
- The differential of the rotation matrix can be solved with respect to quaternion parameters, so the quaternion is suitable to the iterative gradient search for 3D alignment solution.

A.2 Quaternion Operation

In this section, we define the basic operations of the quaternion. The sum, difference and product are respectively defined according to the following equations:

$$q_1 + q_2 = (\overrightarrow{n_1} + \overrightarrow{n_2}, s_1 + s_2) \tag{A.3}$$

$$q_1 - q_2 = (\overrightarrow{n_1} - \overrightarrow{n_2}, s_1 - s_2) \tag{A.4}$$

$$q_1 q_2 = (s_1\overrightarrow{n_2} + s_2\overrightarrow{n_1} + \overrightarrow{n_1} \times \overrightarrow{n_2}, s_1 s_2 - \overrightarrow{n_1} \cdot \overrightarrow{n_2}) \tag{A.5}$$

$$\textit{where} \quad q_1 = (\overrightarrow{n_1}, s_1), \quad q_2 = (\overrightarrow{n_2}, s_2)$$

Consider the following calculation:

$$qq' = (\overrightarrow{0}, s^2 + \overrightarrow{n} \cdot \overrightarrow{n}) \tag{A.6}$$

$$\textit{where} \quad q = (\overrightarrow{n}, s), \quad q' = (-\overrightarrow{n}, s).$$

Rearrange the above equation to:

$$\frac{(qq')}{s^2 + \overrightarrow{n} \cdot \overrightarrow{n}} = (\overrightarrow{0}, 1) \tag{A.7}$$

The right handed side of the above equation ($q_I = (\overrightarrow{0}, 1)$) is defined as identity quaternion since it satisfies the following relation.

$$q_I q = q q_I = q. \tag{A.8}$$

Hence, the inverse quaternion of q (q_{-1}) is defined as:

$$q^{-1} = \frac{q'}{s^2 + \vec{n} \cdot \vec{n}} \tag{A.9}$$

$$where \quad q = (\vec{n}, s) \quad q' = (-\vec{n}, s)$$

and the norm and the absolute of q are respectively as:

$$\|q\| = s^2 + \vec{n} \cdot \vec{n} \tag{A.10}$$

$$|q| = \sqrt{\|q\|} \tag{A.11}$$

The quaternion normalization is division by its absolute. Normalized quaternion is defined as a unit quaternion.

A.3 Rotation Representation by Quaternion

Consider the case when the point $\vec{p}$ is rotated to $\vec{p'}$ by a unit quaternion $q = (\vec{n}, s)$. In quaternion space, the rotation is calculated as follows.

$$\begin{bmatrix} \vec{p'} \\ 0 \end{bmatrix} = q \begin{bmatrix} \vec{p} \\ 0 \end{bmatrix} \bar{q} \tag{A.12}$$

$[\vec{p}, 0]^T$ represents the vector $\vec{p}$ in the quaternion space. Define $q = (\vec{n}, s)$, which implies $\bar{q} = (-\vec{n}, s)$. The above equation can then be rewritten as follows.

$$\begin{bmatrix} \vec{p'} \\ 0 \end{bmatrix} = \begin{bmatrix} s^2\vec{p} + 2s\vec{n} \times \vec{p} + (\vec{n} \cdot \vec{p})\vec{n} \\ 0 \end{bmatrix} \tag{A.13}$$

By replacing $\vec{p}$, $\vec{p'}$ and $\vec{n}$ with (x, y, z), (x', y', z') and (u, v, w), respectively, the vector part of the above equation is:

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} s^2 + u^2 - w^2 - v^2 & 2(uv - sw) & 2(sv + uw) \\ 2(sw + uv) & s^2 + v^2 - u^2 - w^2 & 2(vw - su) \\ 2(uw - sv) & 2(su + vw) & s^2 + w^2 - v^2 - u^2 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} \tag{A.14}$$

In geometrical aspect, the relationship between $\vec{p}$ and $\vec{p'}$ is according to the following equation (See Figure A.1 for detail of calculations)

$$\vec{p'} = (\vec{p} \cdot \vec{m})\vec{m} + (\vec{p} - (\vec{p} \cdot \vec{m})\vec{m}) \cos\phi + \frac{\vec{m} \times \vec{p}}{|\vec{m} \times \vec{p}|} |\vec{p} - (\vec{p} \cdot \vec{m})\vec{n}| \sin\phi, \tag{A.15}$$

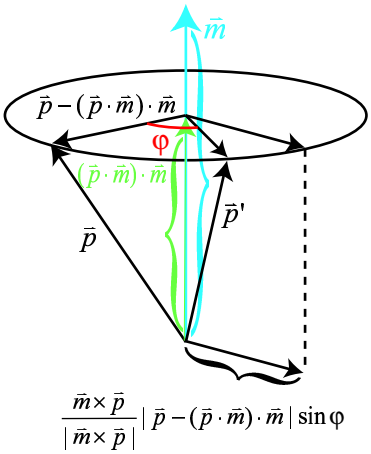


Figure A.1: Rotation for quaternion representation.

where $\vec{m}$ and ϕ are a normalized rotation axis and the angle between $\vec{p}$ and $\vec{p'}$ around $\vec{m}$, respectively.

Because

$$|\vec{m}\times\vec{p}| = |\vec{p}-(\vec{p}\cdot\vec{m})\vec{m}|, \tag{A.16}$$

the above equation is transformed into:

$$\vec{p'} = (\vec{p}\cdot\vec{m})\vec{m} + (\vec{p}-(\vec{p}\cdot\vec{m})\vec{m})\cos\phi + \vec{m}\times\vec{p}\sin\phi \tag{A.17}$$

By replacing $\vec{p}$, $\vec{p'}$ and $\vec{m}$ with (x,y,z), (x',y',z') and (m_x,m_y,m_z), respectively, the vector part of the above equation is:

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} =$$

$$\begin{bmatrix} (1-\cos\phi)m_x{}^2+\cos\phi & (1-\cos\phi)m_xm_y-(\sin\phi)m_z & (1-\cos\phi)m_zm_x+(\sin\phi)m_y \\ (1-\cos\phi)m_xm_y+(\sin\phi)m_z & (1-\cos\phi)m_y{}^2+\cos\phi & (1-\cos\phi)m_ym_z-(\sin\phi)m_x \\ (1-\cos\phi)m_zm_x-(\sin\phi)m_y & (1-\cos\phi)m_ym_z+(\sin\phi)m_x & (1-\cos\phi)m_z{}^2+\cos\phi \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} \tag{A.18}$$

Because q is the unit quaternion,

$$s^2+u^2+v^2+w^2 = 1. \tag{A.19}$$

Applying the above constraint, we compare the rotation matrix in equation (A.14) with the one in equation (A.18), and obtain the following relationship,

$$\begin{cases} u &= m_x \sin\frac{\phi}{2} \\ v &= m_y \sin\frac{\phi}{2} \\ w &= m_z \sin\frac{\phi}{2} \\ s &= \cos\frac{\phi}{2} \end{cases} \tag{A.20}$$

Hence, in order to rotate the point $\overrightarrow{p}$ for ϕ radian around the normalized axis $\overrightarrow{m}$, the rotation quaternion is set to

$$q = ((\sin\frac{\phi}{2})\overrightarrow{m}, \cos\frac{\phi}{2}), \tag{A.21}$$

then the quaternion calculation is performed like this.

$$q \begin{bmatrix} \overrightarrow{m} \\ 0 \end{bmatrix} \overline{q} \tag{A.22}$$

$$where \quad \overline{q} = (-(\sin\frac{\phi}{2})\overrightarrow{m}, \cos\frac{\phi}{2})$$

A.4 Jacobian of Rotation Matrix with Respect to Quaternions

The Jacobian of rotation matrix with respect to quaternion parameters is easier calculated with some special condition. In this section, the conditions are formulated step by step to acquire the Jacobian.

Let $\mathbf{R}(q_u)$ be the rotation matrix of a unit (normalized) quaternion q_u, Equation (A.1) can be rewritten as follows.

$$\overrightarrow{p'}_i = \mathbf{R}(q_u)\overrightarrow{p_i} + \overrightarrow{t} \tag{A.23}$$

In the derivative of the above equation with respect to quaternion parameters, we consider only the first term in the because the derivatives of second term is zero.

As the first step of this differential process, we should remove the normalization constraint of a quaternion. That is, we separate the normalization operation from the rotation matrix $\mathbf{R}(q_u)$) as follows.

$$\mathbf{R}(q_u) = \frac{1}{||q||}\mathbf{R}(q) \tag{A.24}$$

In $\mathbf{R}(q)$, quaternion parameters do not have to be normalized because the multiplication of $\mathbf{R}(q)$ by $\frac{1}{||q||}$ is the normalization operation. As mentioned earlier, it is convenient when obtaining the derivative under the special condition.

The rotation from a 3D point $\overrightarrow{p}$ to $\overrightarrow{p'}$ according to a unit quaternion q_u of rotation can be represented by the function f which takes the two argument, q_u and $\overrightarrow{p}$ as follows.

$$\begin{aligned} \overrightarrow{p'} &= \mathbf{R}(q_u)\overrightarrow{p} \\ &= f(q_u, \overrightarrow{p}) \end{aligned} \tag{A.25}$$

Let the unit quaternion $q_{u_{opt}}$ be the optimum quaternion in the current iteration of conjugate gradient method. The optimum quaternion in the subsequent iteration is estimated according to:

$$\frac{\partial f(q_u, \overrightarrow{p})}{\partial q}, \Big|_{q_{u_{opt}}} \tag{A.26}$$

where $\frac{\partial f(q_u, \overrightarrow{p})}{\partial q}\Big|_{q_c}$ is $\frac{\partial f(q_u, \overrightarrow{p})}{\partial q}$ at q_c.

It is difficult to obtain the differential value at arbitrary quaternions. However, we can easily acquire the differential value at an identity quaternion q_I. Hence, we first rotate $\overrightarrow{p}$ in the way that $q_{u_{opt}}$ is transformed into the identity quaternion, q_I. That is, Equation (A.25) can be rewritten as:

$$\mathbf{R}(q_{u_{opt}})\overrightarrow{p} = f(q_I, \mathbf{R}(q_{u_{opt}})\overrightarrow{p}). \tag{A.27}$$

By the above equation, the derivative at the arbitrary quaternion is transformed into the derivative at the identity quaternion:

$$\frac{\partial f(q_u, \mathbf{R}(q_{u_{opt}})\overrightarrow{p})}{\partial q}\Big|_{q_I}, \tag{A.28}$$

Let $\mathbf{R}(q_{u_{opt}})\overrightarrow{p} = \overrightarrow{p_{opt}}$. Then the term in (A.28) can be rewritten as follows.

$$\frac{\partial f(q_u, \overrightarrow{p_{opt}})}{\partial q}\Big|_{q_I} \tag{A.29}$$

From the above equation and Equation (A.25),

$$\begin{aligned} \frac{\partial f(q_u, \overrightarrow{p_{opt}})}{\partial q}\Big|_{q_I} &= \frac{\partial(\mathbf{R}(q_u)\overrightarrow{p_{opt}})}{\partial q}\Big|_{q_I} \\ &= \frac{\partial \mathbf{R}(q_u)}{\partial q}\Big|_{q_I} \overrightarrow{p_{opt}} + \mathbf{R}(q_u)\left(\frac{\partial \overrightarrow{p_{opt}}}{\partial q}\Big|_{q_I}\right) \\ &= \frac{\partial \mathbf{R}(q_u)}{\partial q}\Big|_{q_I} \overrightarrow{p_{opt}} \end{aligned} \tag{A.30}$$

because

$$\mathbf{R}(q_u)\left(\frac{\partial \overrightarrow{p_{opt}}}{\partial q}\Big|_{q_I}\right) = \begin{bmatrix} 0 & 0 & 0 & 0 \end{bmatrix}. \tag{A.31}$$

Replace Equation (A.24) into Equation (A.30).

$$\begin{aligned}\left.\frac{\partial \mathbf{R}(q_u)}{\partial q}\right|_{q_I} &= \left.\frac{\partial}{\partial q}\left(\frac{1}{\|q\|}\mathbf{R}(q)\right)\right|_{q_I} \\ &= \left.\frac{\partial}{\partial q}\left(\frac{1}{\|q\|}\right)\right|_{q_I} \mathbf{R}(q_I) + \left(\frac{1}{\|q_I\|}\right)\left(\left.\frac{\partial}{\partial q}\mathbf{R}(q)\right|_{q_I}\right)\end{aligned} \tag{A.32}$$

Since

$$\mathbf{R}(q_I) = \mathbf{I}, \quad \frac{1}{\|q_I\|} = 1,$$

Equation (A.32) is transformed into:

$$\left.\frac{\partial \mathbf{R}(q_u)}{\partial q}\right|_{q_I} = \left.\frac{\partial}{\partial q}\left(\frac{1}{\|q\|}\right)\right|_{q_I} \mathbf{I} + \left.\frac{\partial \mathbf{R}(q)}{\partial q}\right|_{q_I}. \tag{A.33}$$

From the definition of the quaternion $q = \begin{pmatrix} u & v & w & s \end{pmatrix}$, the first term in the right handed side of Equation (A.33) can be rewritten as:

$$\begin{aligned}\left.\frac{\partial}{\partial q}\left(\frac{1}{\|q\|}\right)\right|_{q_I}\mathbf{I} &= \left.\left(\frac{1}{\|q\|}\left(-\frac{(\|q\|)'}{\partial q}\right)\right)\right|_{q_I}\mathbf{I} \\ &= \left(\frac{1}{\|q_I\|^2}\right)\left(\left[-\frac{(\|q\|)'}{\partial u} - \frac{(\|q\|)'}{\partial v} - \frac{(\|q\|)'}{\partial w} - \frac{(\|q\|)'}{\partial s}\right]\Big|_{q_I}\right)\mathbf{I} \\ &= \left[\begin{array}{cccc} -2u & -2v & -2w & -2s \end{array}\right]\Big|_{q_I}\mathbf{I} \\ &= \left[\begin{array}{cccc} 0 & 0 & 0 & -2 \end{array}\right]\mathbf{I} \\ &= \left[\left[\begin{array}{ccc} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{array}\right]\left[\begin{array}{ccc} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{array}\right]\left[\begin{array}{ccc} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{array}\right]\left[\begin{array}{ccc} -2 & 0 & 0 \\ 0 & -2 & 0 \\ 0 & 0 & -2 \end{array}\right]\right]\end{aligned} \tag{A.34}$$

By replacing $\mathbf{R}(q)$ with Equation (A.14) the second term in the right handed side of Equation (A.33) can be evaluated as follows.

$$\begin{aligned}\left.\frac{\partial \mathbf{R}(q)}{\partial q}\right|_{q_I} &= \left[\begin{array}{cccc} \frac{\partial \mathbf{R}(q)}{\partial u} & \frac{\partial \mathbf{R}(q)}{\partial v} & \frac{\partial \mathbf{R}(q)}{\partial w} & \frac{\partial \mathbf{R}(q)}{\partial s} \end{array}\right]\Big|_{q_I} \\ &= \left[\left[\begin{array}{ccc} 2u & 2v & 2w \\ 2v & -2u & -2s \\ 2w & 2s & -2u \end{array}\right]\left[\begin{array}{ccc} -2v & 2u & 2s \\ 2u & 2v & 2w \\ -2s & 2w & -2v \end{array}\right]\left[\begin{array}{ccc} -2w & -2s & 2u \\ 2s & -2w & 2v \\ 2u & 2v & 2w \end{array}\right]\left[\begin{array}{ccc} 2s & -2w & 2v \\ 2w & 2s & -2u \\ -2w & 2u & 2s \end{array}\right]\right]\Big|_{q_I} \\ &= \left[\left[\begin{array}{ccc} 0 & 0 & 0 \\ 0 & 0 & -2 \\ 0 & 2 & 0 \end{array}\right]\left[\begin{array}{ccc} 0 & 0 & 2 \\ 0 & 0 & 0 \\ -2 & 0 & 0 \end{array}\right]\left[\begin{array}{ccc} 0 & -2 & 0 \\ 2 & 0 & 0 \\ 0 & 0 & 0 \end{array}\right]\left[\begin{array}{ccc} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 2 \end{array}\right]\right]\end{aligned} \tag{A.35}$$

From Equation (A.34) and (A.35), the solution of Equation (A.33) can be found as:

$$\left.\frac{\partial \mathbf{R}(q_u)}{\partial q}\right|_{q_I} = \left[\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -2 \\ 0 & 2 & 0 \end{bmatrix} \begin{bmatrix} 0 & 0 & 2 \\ 0 & 0 & 0 \\ -2 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 & -2 & 0 \\ 2 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \right] \tag{A.36}$$

Note that the scaler term of the quaternion disappears in case that the derivative is evaluated at q_I. To calculate the (negative) gradient, we have to estimate only the 3 parameters of its vector part.

By replacing Equation A.36 into Equation (A.30), we have the following relationship: differential as follows.

$$\begin{aligned}
&\left.\frac{\partial f(q_u, \mathbf{R}(q_{u_{opt}})\overrightarrow{p})}{\partial q}\right|_{q_I} \\
= &\left.\frac{\partial f(q_u, \overrightarrow{p_{opt}})}{\partial q}\right|_{q_I} \\
= &\left.\frac{\partial \mathbf{R}(q_u)}{\partial q}\right|_{q_I} \overrightarrow{p_{opt}} \\
= &\left[\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -2 \\ 0 & 2 & 0 \end{bmatrix} \begin{bmatrix} 0 & 0 & 2 \\ 0 & 0 & 0 \\ -2 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 & -2 & 0 \\ 2 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \right] \begin{bmatrix} x \\ y \\ z \end{bmatrix} \\
= &\left[\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -2 \\ 0 & 2 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} \begin{bmatrix} 0 & 0 & 2 \\ 0 & 0 & 0 \\ -2 & 0 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} \begin{bmatrix} 0 & -2 & 0 \\ 2 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} \right] \\
= &\begin{bmatrix} 0 & 2z_{opt} & -2y_{opt} & 0 \\ -2z_{opt} & 0 & 2x_{opt} & 0 \\ 2y_{opt} & -2x_{opt} & 0 & 0 \end{bmatrix},
\end{aligned} \tag{A.37}$$

$$where \quad \overrightarrow{p_{opt}} = \begin{bmatrix} x_{opt} \\ y_{opt} \\ z_{opt} \end{bmatrix}$$

When ignored the last column, the 4×3 matrix in the right handed side of Equation (A.37) is the skew-symmetric matrix of $\overrightarrow{p_{opt}}$ multiplied by -2. Skew-symmetric matrix is a useful expression, for example, for a cross product of two vectors:

$$\overrightarrow{a} \times \overrightarrow{b} = \begin{bmatrix} a_y b_z - a_z b_y \\ a_z b_x - b_z a_x \\ a_x b_y - b_x a_y \end{bmatrix} = \begin{bmatrix} 0 & -a_z & a_y \\ a_z & 0 & -a_x \\ -a_y & a_x & 0 \end{bmatrix} \begin{bmatrix} b_x \\ b_y \\ b_z \end{bmatrix} = \mathbf{C}(\overrightarrow{a})\overrightarrow{b}, \tag{A.38}$$

$$where \quad \overrightarrow{a} = \begin{bmatrix} a_x \\ a_y \\ a_z \end{bmatrix} \quad \overrightarrow{b} = \begin{bmatrix} b_x \\ b_y \\ b_z \end{bmatrix}$$

Considering $-\mathbf{C}(\overrightarrow{a}) = \mathbf{C}(\overrightarrow{a})^T$, Equation (A.37) is finally transformed into:

$$\left.\frac{\partial f(q_u, \mathbf{R}(q_{u_{opt}})\overrightarrow{p})}{\partial q}\right|_{q_I} = 2\mathbf{C}(\overrightarrow{p_{opt}})^T. \tag{A.39}$$

After obtaining the differential value at q_I, that is, the rotation axis $\lambda q_v = (\lambda u, \lambda v, \lambda w, 0)(u^2 + v^2 + w^2 = 1)$ in quaternion representation (negative gradient of the quaternion's vector part in line minimization process), we need the rotation amount around the determined axis. If the expected rotation quaternion is denoted by q', in the following equation

$$q' = q_I + \lambda q_v = (\lambda u, \lambda v, \lambda w, 1), \tag{A.40}$$

the scaler part q_s is calculated by normalization of the above quaternion.

$$q_s = \lambda^2 + 1 \tag{A.41}$$

Then we can obtain the rotation amount ϕ from the above equation and the equation (A.21) as follows.

$$\phi = 2\arccos\left(\frac{1}{\lambda^2 + 1}\right) \tag{A.42}$$

Appendix B

Derivation of Partial Derivatives

In this chapter, we show the derivation of partial derivatives of the numerical formula represented by each mathematical models.

B.1 Revolution Surface of Catenary

B.1.1 Numerical Formula

$$X(u,v) = (l\phi(v)\cos u, l\phi(v)\sin u, l\psi(v)), \tag{B.1}$$

$$where \quad \phi(v) = b\cosh\left(\frac{v}{a}\right), \quad \psi(v) = \int_0^v \sqrt{1 - \frac{b^2}{a^2}\sinh^{-1}\left(\frac{t}{a}\right)}dt$$

$$parameter \quad a, \quad b, \quad l$$

B.1.2 Preliminary

From

$$\sinh x = \frac{e^x - e^{-x}}{2}, \tag{B.2}$$

the following equation is obtained.

$$\sinh^{-1}x = \log(x + \sqrt{x^2+1}) \tag{B.3}$$

$$\therefore \frac{\partial}{\partial x}(\sinh^{-1}x) = \frac{1}{\sqrt{x^2+1}} \tag{B.4}$$

B.1.3 Partial Derivative

Partial Derivative with respect to Parameter a

$$\frac{\partial X}{\partial a} = \left((l\cos u)\frac{\partial \phi}{\partial a}, (l\sin u)\frac{\partial \phi}{\partial a}, l\frac{\partial \psi}{\partial a}\right) \tag{B.5}$$

where

$$\frac{\partial \phi}{\partial a} = -\frac{bv}{a^2}\sinh\frac{v}{a} \tag{B.6}$$

$$\begin{aligned}
\frac{\partial \psi}{\partial a} &= \left(\sqrt{1-\frac{b^2}{a^2}\sinh^{-1}\frac{v}{a}}\right)\frac{b^2\frac{\partial}{\partial a}\left(-\frac{1}{a^2}\sinh^{-1}\frac{v}{a}\right)}{2\sqrt{1-\frac{b^2}{a^2}\sinh^{-1}\frac{v}{a}}} \\
&= \frac{b^2}{2}\frac{\partial}{\partial a}\left(-\frac{1}{a^2}\sinh^{-1}\frac{v}{a}\right) \\
&= \frac{b^2}{2}\left(\frac{1}{2a^3}\sinh^{-1}\frac{v}{a} - \frac{1}{a^2}\frac{1}{\sqrt{\frac{v^2}{a^2}+1}}\right)\left(-\frac{v}{a^2}\right) \\
&= \frac{vb^2}{2a^3}\left(-\frac{1}{2a^2}\sinh^{-1}\frac{v}{a} + \frac{1}{\sqrt{v^2+a^2}}\right)
\end{aligned} \tag{B.7}$$

Partial Derivative with respect to Parameter b

$$\frac{\partial X}{\partial b} = \left((l\cos u)\frac{\partial \phi}{\partial b}, (l\sin u)\frac{\partial \phi}{\partial b}, l\frac{\partial \psi}{\partial b}\right) \tag{B.8}$$

where

$$\frac{\partial \phi}{\partial b} = \cosh\frac{v}{a} \tag{B.9}$$

$$\begin{aligned}
\frac{\partial \psi}{\partial b} &= \left(\sqrt{1-\frac{b^2}{a^2}\sinh^{-1}\frac{v}{a}}\right)\frac{\frac{-2b}{a^2}\sinh^{-1}\frac{v}{a}}{2\sqrt{1-\frac{b^2}{a^2}\sinh^{-1}\frac{v}{a}}} \\
&= -\frac{b}{a^2}\sinh^{-1}\frac{v}{a}
\end{aligned} \tag{B.10}$$

Partial Derivative with respect to Parameter l

$$\frac{\partial X}{\partial l} = (\phi(v)\cos u, \phi(v)\sin u, \psi(v)) \tag{B.11}$$

B.2 Dini's Surface

B.2.1 Numerical Formula

$$X(s,t) = \underset{\substack{\textit{parameter} \quad b, \quad l}}{\left(\frac{l\cos t}{\cosh s}, \frac{l\sin t}{\cosh s}, l(s - \tanh s + bt)\right)}, \tag{B.12}$$

B.2.2 Partial Derivative

Partial Derivative with respect to b

$$\frac{\partial X}{\partial b} = (0, 0, lt) \tag{B.13}$$

Partial Derivative with respect to l

$$\frac{\partial X}{\partial l} = \left(\frac{\cos t}{\cosh s}, \frac{\sin t}{\cosh s}, s - \tanh s + bt\right), \tag{B.14}$$

B.3 Kuen's Surface

B.3.1 Numerical Formula

$$X(u,v) = \underset{\substack{\textit{parameter} \quad l}}{\left(l\frac{2(\cos u + u\sin u)\sin v}{1 + u^2\sin^2 v}, l\frac{2(\sin u - u\cos u)\sin v}{1 + u^2\sin^2 v}, l\left(\log\left(\tan\frac{v}{2}\right) + \frac{2\cos v}{1 + u^2\sin^2 v}\right)\right)} \tag{B.15}$$

Partial Derivative with respect to l

$$\frac{\partial X}{\partial l} = \underset{\substack{\textit{parameter} \quad l}}{\left(\frac{2(\cos u + u\sin u)\sin v}{1 + u^2\sin^2 v}, \frac{2(\sin u - u\cos u)\sin v}{1 + u^2\sin^2 v}, \log\left(\tan\frac{v}{2}\right) + \frac{2\cos v}{1 + u^2\sin^2 v}\right)} \tag{B.16}$$

B.4 Inverse Function of the Elliptic Integral of the First Kind

B.4.1 Numerical Formula

$$X(\phi,k) = \underset{\substack{\textit{parameter} \quad l}}{\left(l\phi, lk, l\int_0^{\phi}\frac{dt}{\sqrt{1 - k^2\sin^2 t}}\right)} \tag{B.17}$$

Partial Derivative with respect to l

$$\frac{\partial X}{\partial l} = \left(\phi, k, \int_0^{\phi} \frac{dt}{\sqrt{1 - k^2 \sin^2 t}} \right) \tag{B.18}$$