2011

An online algorithm for matching noisy space curves with statistical error analysis

Hyuntae Na
Iowa State University

An online algorithm for matching noisy space curves with statistical error analysis

by

Hyuntae Na

A thesis submitted to the graduate faculty
in partial fulfillment of the requirements for the degree of
MASTER OF SCIENCE

Major: Computer Science

Program of Study Committee:
Yan-Bin Jia, Major Professor
Vasant Honavar
Namrata Vaswani

Iowa State University
Ames, Iowa
2011

TABLE OF CONTENTS

LIST OF FIGURES

ABSTRACT

In this thesis, we presents a new algorithm that finds the longest partial match between two space curves. The algorithm iteratively extends an initial matching portion of two curves within some tolerance over the matching quality. Each iteration adjusts the matching transformation (rotation, scale, and translation) to handle noisy data more robustly and to enlarge the matched portion. To control the matching accuracy, a statistical threshold is introduced to stop the iterative extension. Experiment shows that the algorithm has a comparable accuracy to that of the well known ICP algorithm [3] but its efficiency is improved by an order of magnitude. The algorithm has been demonstrated over synthetic and range data. Experiment shows that it adjusts well to noise distributions and performs effectively over curves of complex shapes.

CHAPTER 1. INTRODUCTION

Finding the matching segment of two data curves is one of the classical problems in pattern recognition, computer vision, and robotics. A range of recognition and localization tasks in these areas can be transformed into this problem or rely on its solution. In particular, reconstruction of an object from patches of data may be carried out by matching the segments of the boundary curves of these patches. Examples include assembly of an antique pot from a few broken fragments [18, 24], reconstruction of an entire document from the ripped pieces of a paper [42], building of 3-D shape of a scene by matching curves in images [35], or mosaicing of images by piecing together their edges [39]. Additionally, there are many studies to recognize an object by curve matching [2, 17, 19, 20, 25, 26, 29, 28]. The pose of a 3-D object can be estimated by matching a "characteristic" surface curve with the corresponding one on its model [9]. In such a case, the capability of handling partial matches and occlusions with noisy data [21, 33] is important because most observable data from sensing devices are noisy and may be occluded. Furthermore, the curve matching algorithm has been applied to various applications: vision-based robot control [23], hand-written signature and character recognitions [1, 22], and protein structure alignment [4].

We find the longest matching segment of two space curves which are similar: one curve can be superposed to the other by scaling, rotating and translating it. We assume that initial information is given: a small portion of the matching segment and a transformation superposing it onto the other curve. In this case, the longest matching segment can be obtained by gradually extending the initially given segment while adjusting its transformation, as long as its matching quality is tolerable. The extension can be performed using the ICP[1] algorithm in $O(n^2)$ time with allowing scaling the matching segment [41].

The goal of this research is to find an algorithm that gradually extends the matching segment to find its longest extension in such a case. Additionally, we hope to achieve a comparable accuracy to that of the ICP algorithm but is more efficient by achieving $O(n)$ time while aligning the two matching segments using the similarity transformation.

[1] It finds the optimal transformation (rotation and translation) superposing a set P of entire data points onto a model X by iteratively minimizing the sum of squared distances from the points to the model. The detail of the ICP algorithm will be provided in Appendix C.

1.1 Related Work

A number of approaches have been introduced for finding matching curve segments. One of them divides each curve to be matched into small sections for piecewise comparison to achieve efficiency by reducing redundancies [12, 16, 17, 31, 34, 40]. Here length and local properties such as curvature and torsion are utilized in the segmentation. However, the major drawback is that the behavior heavily depends on the outcome of segmentation. If the segmentation is not fine enough, a large portion of a matching segment could be neglected [16, 34]. A second drawback is that it allows two curves to match despite accumulating difference [34].

Segmentation often requires smoothing a data curve in the preprocessing to reduce the degradation of results due to noise [6, 14, 17, 31, 33, 40]. Especially, if a curve is given as a list of noisy points, which is common for range and tactile data, fitting with a differentiable curve like a B-spline is used to extract features such as length, curvature and torsion. However, it is difficult to achieve a certain level of accuracy given the possible distortion of the curve caused by over-smoothing. In particular, smoothing can blunt sharp corners of the curve.

One issue in finding the matching segment lies in what range of transformations should be considered. Some work has been done on registering 3-D sensory data under rigid body transformations (rotations and translations) [9, 10]. To attain accuracy, a certain amount of freedom in scaling should be allowed because it is hard to determine the scale of the observed data precisely output from sensing devices. Furthermore, in a case like object recognition, the expected range of scale may not be given. To deal with scaling, finding a similarity transformations (rotations, scales, and translations) has been studied [11, 41]. In [41], scalability is added to the ICP algorithm [3], which is well-known for registering 3-D objects under rigid body transformations. However, the method needs lower bounds for scales to avoid the situation of shrinking all data points into a point on the model.

In the following sections, we list the prior researches that find the longest common segment of two planar curves in Section 1.1.1, and two space curves in Section 1.1.2. Additionally, we summarize prior researches that register 3-D shapes under rigid body, similarity, and other transformations in Section 1.1.3.

1.1.1 Planar Curve Matching

B. Thomas et al. [34] proposed an algorithm matching two largely deformable planer curves using dynamic programming and their intrinsic formulation. In the algorithm, a table is filled with the cost of deforming an infinitesimal segment of one curve to that of the other using the intrinsic formulation, and dynamic programming is used to find the optimal path in the table which minimizes the cost aligning two curve by deforming one of them. The

algorithm allows recognizing largely deformed shapes like the hand written characters and the outlines of different animals in figures.

Arie Pikaz and Its'hak Dinstein [32] proposed an algorithm that finds the partial match of two curves. It finds two ends of the longest matching segments of the two curves using dynamic programming; in each iteration of the dynamic programming, it extends both ends of the matching segment as long as the total curvatures (and the ratios of increased lengths) of the segments are the same. The approach achieves the similarity transformation (rotations, scales, and translations), but is constrained to the planar curves because of using the total curvature.

H. J. Wolfson [38] utilized the string matching algorithm, the dynamic programming algorithm to find the longest common sequence of alphabet in two different strings, to find the longest common segment of two curves. To adjust the curve matching problem into the string matching problem, he first divides each curve into several equidistant curve segments, and then assigns to each segment the value of the local shape signature which is invariant to the translation and rotation like curvature. However, the method could be sensitive to the length of each segment and the value assigned to it.

Aristeidis Diplaros and Evangelos Milios [31] presented an approach to match distorted and possibly occluded shapes using the dynamic programing. The approach uses inflection points of a curve, which are invariant to translation, scaling, rotation and starting point; the dynamic programming of the approach merges the matched concave segments of two curves, which are determined in between the inflection points. However, the approach works only when one curve belongs to the other, or both curves are fully matched.

Minghui Xia and Bede Liu [39] proposed a method to match and align curves for accurate image registration by finding a super-curve. It is a single B-spline approximation to two point sequences to generate their best superposition under affine transformation. Contrast to other methods, their method uses raw data points instead of intermediate smoothed curves, to ensure the accuracy of its main algorithm. However, it has to find a partial match of two curves to feed into the main algorithm, in order to polish the accuracy of two aligned curves as the super-curve.

Farzin Mokhtarian and Alan Mackworth [27] presented the Curvature Scale Space (CSS) image which uniquely represents a curve with a limited number of data. To generate CSS image, their algorithm finds the list of inflection points on the curve that is smoothed by the Gaussian kernel with different width ρ, and then draws hat-shaped curves on the t-ρ image with the ρ value and the index t of the inflection points normalized by arc-length. Matching two different curves are performed by comparing the t-ρ images of them. In [37], Yue Wang and Eam Khwang Teoh reduced the matching error of the CSS image using B-spline fitting instead of the Gaussian kernel.

Dong Xu and Wenli Xu [40] presented a descriptor of an object contour using arc length and tangent orientation, in order to find a matching object in a database. The descriptor uses arc length variance function (ALVF) which measures the variation of arc length as the tangent angle changes; it is different to another function measuring the variation of tangent angle as the arc length increase. They recognized a query object in the database by comparing the correlation of two ALVFs of the query and an object contours in the database.

I. Cohen and I. Herlin [7] studied the method to match curves using geodesic paths. It finds the match of multiple curves by iteratively minimizing their geodesic distance error. Contrast to other methods, it allows the topological change of matching curves, and therefore was applied to track a cloud structure in a sequence of images depicting a tropical storm [8].

1.1.2 Space Curve Matching

B. Kamgar-Parsi and B. Kamgar-Parsi [16] proposed a closed form solution that determines the rigid body transformation (rotations and translations) to superpose one polyline onto another polyline. Its computational cost is proportional to the number of line segments connected in the polyline. Because it is designed to match two whole curves, if one curve is a part of the other, it sequentially shifts the small curve over the long curve to find the matching portion; it is not appropriate when both curves are partially matched.

Stan Z. Li [21] designed the similarity-invariant coordinate system (SICS), which describes the shape of a curve segment with few variables. They segmented a curve at its vertexes in order to build a list of SICS, and represented the curve as a form of the attributed relational graph (ARG) where nodes are defined using SICS. To find the longest common segment of two curves, they employed MAP-MRF (maximum a posteriori - Markov random fields) labeling method. Their method is robust against complex shape of curves, but may be sensitive to the result of the curve segmentation.

E. Kishon et al. [17] studied the method to recognize an object using a curve matching method. They built hash table in order to efficiently find curve segments similar to a query segment in their database; the key of the hash table is defined as the curvature-torsion based shape signature of a curve segment. To find the queried object in the database, they searched the curve that has the longest common sequence of shape signatures with a curve in the object.

T. Pajdla and L. Van Gool [30] proposed the semi-differential invariants of a curve to recognize an object using another curve matching method. The invariants of an arbitrary point on the curve are defined as one distance and two cosine angles by the comparison with a given reference point on the curve; the invariants of several curves on different object surfaces are stored in a hash table. To recognize an object with a queried curve, they listed candidates

of matching curves by searching invariants of the queried curve on the hash table and counting reference points of invariants; and then, they trimmed candidates and selected one curve with its recognized object using the least mean square motion estimation and the modified ICP (Iterative Closest Point) algorithm, respectively.

J. T. Schwartz and M. Sharir [33] presented a method that matches partially obscured and noisy curves in 2-D and 3-D. It samples a sequence of evenly spaced points on a smoothed curve. From two point sequences $\boldsymbol{p}_1, ..., \boldsymbol{p}_m$ and $\boldsymbol{q}_1, ..., \boldsymbol{q}_n$ which are sampled on two different curves, it finds an offset d and the Euclidean transformation T that superposes $\boldsymbol{p}_i$ onto $\boldsymbol{q}_{i+d}$, $1 \leq i \leq m \leq n$. It uses the fast Fourier transform to achieve $O(n \log n)$ time complexity while determining d and T that minimize the superposition error.

1.1.3 Transformations to Register 3-D Shapes

Faugeras and Hebert [9] presented an algorithm that registers 3-D sensory data under rigid body transformations (rotations and translations). They represented an object as a set of segmented data such as corner points, line segments, and surface patches; in their object representation, each data is stored as the feature values that compress the data information. For the object registration, they found the rigid body transformation that minimizes sum of squared differences of feature values under the transformation. The algorithm is robust to the occlusions and the initial positions of the two objects, but the performance of the algorithm may depend on the object segmentation.

P. Besl et al. [3] proposed the well known ICP (Iterative Closest Point) algorithm. They distinguished objects as two types: data shape and model shape. The data shape is a set of points, and the model shape is any of type; the closest point on the model shape to the data point is assumed to be determined. The algorithm finds the optimal ridge body transformation superposing the data shape onto the model shape by iteratively minimizing the sum of square distances from points of the superposed data shape to the model shape. The detail of this algorithm will be provided in Appendix C.

Yang Chen and Gérard Medioni [5] improved the computation speed of the ICP algorithm. They assumed that finding the closest point to the model shape at a data point is an optimization problem. Therefore, instead of solving the optimization problem, they approximated the result by finding the closest point to one tangent plane to the model shape, which is determined by the data point.

Shihui Ying et al. [41] incorporated a scale factor into the ICP algorithm [3] to achieve the similarity transformation (rotations, scales, and translations) to register the model and data shapes. However, it has the degenerate case that all data points are shrunk into a point on the

model to minimize the sum of square distances from points of the superposed data shape to the model shape, while iterating its procedure. To avoid the case, they also proposed a method to determine the allowed range of scale using the statistics of the data point distribution.

A. Gruen and D. Akca [11] presented a generalized method that finds any type of transformation by iteratively superposing two surfaces under the transformation. They demonstrated their method by finding a similarity transformation that superposes two surfaces of an object scanned in different orientations. To find the transformation by minimizing the superposition error of the two surfaces, they reformulated the Taylor expansion of the error as the linear combination of 7 transformation parameters, and then solved the reformulated error using a Gauss-Markoff estimation model. Additionally, they demonstrated matching a space curve onto a 3-D surface in a similar way.

J. Feldmar and N. Ayache [10] proposed a method that finds the local affine transformation to register free-form surfaces. The method determines the local affine transformation to superpose two surfaces using curvatures and principal frames by sequentially finding (1) the rigid body transformation and (2) the global affine transformation.

R. Szeliski [36] studied a method that matches 3-D surfaces with non-rigid deformations. They used the volumetric transformation to allow the deform of an object surface, and used the precomputed distance map using an octree spline in order to achieve the trade-off between the accuracy and the computing speed.

CHAPTER 2. ONLINE SPACE CURVE MATCHING ALGORITHM

In this section, we explain details of the algorithm that gradually extends the initial match of two curve segments to its longest match in $O(n)$ time while aligning the two matching segments using the similarity transformation. It also stops the extension to control the matching quality considering the noise distribution of raw data points.

The remainder of this thesis is organized as follows. Section 2.1 gives the overview of the algorithm. Sections 2.2, 2.3, and 2.4 describe details of iterative extension of a matching segment, accuracy handling, and similarity transformation, respectively. We will use the notation $d(\mathcal{S}, \boldsymbol{p})$, throughout this thesis, to denote the distance from a point $\boldsymbol{p}$ to a shape $\mathcal{S}$.

2.1 Overview

We investigate the problem of finding the longest matching segment of two space curves. One of the curves is referred as the "model curve" and the other as the "data curve". The model curve $\mathcal{M}$ can be of any type. The data curve $\mathcal{D}$ is a polyline joining n data points $\boldsymbol{p}_1, ..., \boldsymbol{p}_n$, subject to Gaussian noise. Denote by $\mathcal{D}_{k..l}$ the segment of $\mathcal{D}$ composed of the data points $\boldsymbol{p}_k, \boldsymbol{p}_{k+1}, ..., \boldsymbol{p}_l$. Suppose that they were originally sampled from the unknown points $\boldsymbol{r}_k, \boldsymbol{r}_{k+1}, ..., \boldsymbol{r}_l$ on the model curve, and then underwent some transformation. Figure 2.1 illustrates the relationship between the model curve $\mathcal{M}$ and the data points $\boldsymbol{p}_1, \boldsymbol{p}_2, ..., \boldsymbol{p}_n$.

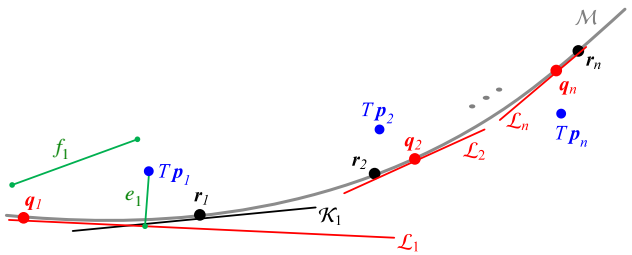


Figure 2.1: Model curve $\mathcal{M}$ and a transformed "data curve" $(T\boldsymbol{p}_1, T\boldsymbol{p}_2, ..., T\boldsymbol{p}_n)$. Each data point $\boldsymbol{p}_i$ is sampled from $\boldsymbol{r}_i$ which is unknown. The distance $d(\mathcal{M}, T\boldsymbol{p}_i)$ was originally estimated from tangent $\mathcal{L}_i$ to $\mathcal{M}$ at $\boldsymbol{q}_i$, $1 \le i \le n$.

We would like to measure the dissimilarity between the model curve $\mathcal{M}$ and the data segment $\mathcal{D}_{k..l}$ which is superposed onto it under some transformation. Suppose the real transformation T^* is known. The noise of sampling $\boldsymbol{r}_i$ is then $\boldsymbol{\xi}_i = \boldsymbol{r}_i - T^*\boldsymbol{p}_i$, $k \le i \le l$. However, it cannot be measured because the position $\boldsymbol{r}_i$ on the model curve is unknown. Here, we try to approximate the distance $\|\boldsymbol{\xi}_i\|$ using that of $T^*\boldsymbol{p}_i$ from the tangent line $\mathcal{K}_i$ at $\boldsymbol{r}_i$, except that this line is also unknown. Again, we approximate $\mathcal{K}_i$ using the tangent line $\overline{\mathcal{K}}_i$ to $\mathcal{M}$ at the closest point to $T^*\boldsymbol{p}_i$. Summarizing the above steps, the dissimilarity between $\mathcal{M}$ and $\mathcal{D}_{k..l}$ under a transformation T is then measured as $\sum_{i=k}^{l} d(\overline{\mathcal{K}}_i, T\boldsymbol{p}_i)^2$. If the sum is less than a pre-determined threshold ε, we deem $\mathcal{D}_{k..l}$ as a matching data segment under the transformation T. The threshold ε is to be chosen as the maximum value that $\sum_{i=k}^{l} d(\mathcal{K}_i, T\boldsymbol{p}_i)^2$ can have statistically, as will be explained in Section 2.3.

In a real situation, $\overline{\mathcal{K}}_i$ cannot be determined either because it depends on T^*, where $k \le i \le l$. Therefore, we approximate $\mathcal{K}_i$ using the tangent line $\mathcal{L}_i$ to $\mathcal{M}$ at some point $\boldsymbol{q}_i \in \mathcal{M}$; the point $\boldsymbol{q}_i$ will be explained soon. In Figure 2.1, $d(\mathcal{K}_1, T\boldsymbol{p}_i)$ is close to the distance e_1 from $T\boldsymbol{p}_1$ to a tangent line $\mathcal{L}_1$ obtained at $\boldsymbol{q}_1$. The *dissimilarity error*, which estimates $\sum_{i=k}^{l} d(\mathcal{K}_i, T^*\boldsymbol{p}_i)^2$, is defined as follows:

$$e(\langle k..l\rangle, T) = \sum_{i=k}^{l} d(\mathcal{L}_i, T\boldsymbol{p}_i)^2. \tag{2.1}$$

Since the data source $\boldsymbol{r}_i$, $k \le i \le l$, and the transformation T^* are unknown, $\mathcal{K}_i$ is best approximated by the tangent line $\mathcal{L}_i^*$ to $\mathcal{M}$ at the closest point $\boldsymbol{q}_i^* \in \mathcal{M}$ to $T\boldsymbol{p}_i$. This forces us to find the closest point $\boldsymbol{q}_i^*$ for all $k \le i \le l$ in order to evaluate the error $e(\langle k..l\rangle, T)$ whenever T changes. The ICP algorithm does this.

However, if the transformation T changes slightly, then $T\boldsymbol{p}_i$ and its new closest point on $\mathcal{M}$ would stay in the almost same places and its corresponding tangent line would be similar to $\mathcal{L}_i^*$ near $T\boldsymbol{p}_i$. Relaxing the "closest point" condition $\boldsymbol{q}_i^* = \operatorname{argmin}_{\boldsymbol{q}\in\mathcal{M}} \|\boldsymbol{q} - T\boldsymbol{p}_i\|^2$ saves the heavy computation because the tangent $\mathcal{L}_i^*$ can be reused. Define the *relaxed condition error* as follows:

$$f(\langle k..l\rangle, T) = \sum_{i=k}^{l} \|\boldsymbol{q}_i - T\boldsymbol{p}_i\|^2, \tag{2.2}$$

and postpone updating $\mathcal{L}_i$ until $f(\langle k..l\rangle, T)$ exceeds a pre-determined threshold ρ. Here ρ is selected to be the maximum value that $\sum_{i=k}^{l} \|\boldsymbol{q}_i - T\boldsymbol{p}_i\|^2$ is allowed. We will discuss it in Section 2.3.2.

Let $\mathcal{D}_{k..l}$ and $\mathcal{D}_{k..l+1}$ be the matching data segments to $\mathcal{M}$ under transformations T_l and T_{l+1}, respectively. The relaxed condition (2.2) allows reuse of tangents $\mathcal{L}_k, ..., \mathcal{L}_l$, as we extend $\mathcal{D}_{k..l}$ to $\mathcal{D}_{k..l+1}$ as long as $f(\langle k..l+1\rangle, T_{l+1}) < \rho$. In this case, the curve dissimilarity $e(\langle k..l+1\rangle, T_{l+1})$ and the relaxed condition $f(\langle k..l+1\rangle, T_{l+1})$ are obtained in $O(1)$ time from $e(\langle k..l\rangle, T_l)$ and

$f(\langle k..l\rangle, T_l)$. Additionally, the transformation T_{l+1} that minimizes the error (2.1) is polished from T_l in average $O(1)$ time. In a similar way, extension of $\mathcal{D}_{k..l}$ to $\mathcal{D}_{k-1..l}$ can be achieved in constant time. This allows us to find the longest matching data segment extended from $\mathcal{D}_{k..l}$ in $O(n)$ time. Figure 2.2 illustrates the scenario of iteratively extending the matching segment of model and data curves when the extension begins with the data segment $\mathcal{D}_{1..2}$ and the initial transformation T_0. In the figure, it is assumed that $f(\langle 1..i\rangle, T_{i-2}) < \rho$ for all $2 \le i \le 6$.

In the above method, it is assumed that the initial information be given: the initial matching data segment $\mathcal{D}_{k..l}$ and the transformation T to superpose it onto its corresponding segment of $\mathcal{M}$. If the prior knowledge of the longest matching segment of $\mathcal{M}$ and $\mathcal{D}$ is given, the initial information will be selected using the prior knowledge. However, when it is not available, the initial information can be obtained using two arbitrary *vertices*[2] $\boldsymbol{v}$ and $\boldsymbol{v}'$ on $\mathcal{M}$ and $\mathcal{D}$, respectively, as follows:

- Select the starting point $\boldsymbol{s}$ and ending point $\boldsymbol{e}$ of the initial model segment in $\mathcal{M}$ such that two line segments $\overline{\boldsymbol{sv}}$ and $\overline{\boldsymbol{ve}}$ have the same length and the total curvature of the arc $\widehat{\boldsymbol{se}}$ is a constant, like say, $\pi/3$.
- Similarly, select the starting $\boldsymbol{p}_k$ and ending $\boldsymbol{p}_l$ points of the initial data segment $\mathcal{D}_{k..l}$ in $\mathcal{D}$.
- Determine the initial similarity transformation T that superposes $\boldsymbol{p}_k$ (and $\boldsymbol{p}_l$) onto $\boldsymbol{s}$ (and $\boldsymbol{e}$) and places $\boldsymbol{v}'$ on the plane constructed from the points $\boldsymbol{s}, \boldsymbol{v}$ and $\boldsymbol{e}$.

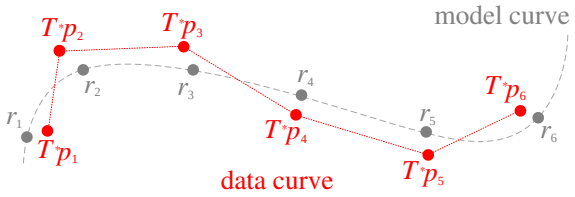
The algorithm finds the longest matching extension of $\mathcal{D}_{k..l}$ with the initial transformation T. If $\mathcal{D}$ and $\mathcal{M}$ have n and m vertices, respectively, there are $\binom{n}{m}$ combinations of initial matches. The longest common segment of $\mathcal{D}$ and $\mathcal{M}$ is selected from their extensions.

2.2 Iterative Extension of a Matching Segment

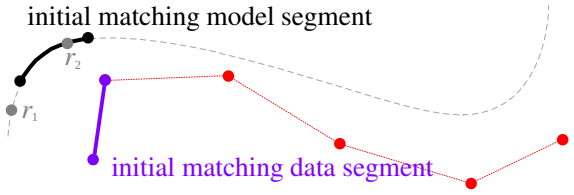
The previous section overviews iterative extension of the matching segment of two curves. The following is the procedure that does so in linear time. It starts with an initial matching data segment $\mathcal{D}_{k_0..l_0}$ and a transformation T_0:

1. Determine $\mathcal{L}_{k_0}, ..., \mathcal{L}_{l_0}$, $e(\langle k_0..l_0\rangle, T_0)$, and $f(\langle k_0..l_0\rangle, T_0)$.
2. Let $\boldsymbol{p}_k, ..., \boldsymbol{p}_l$ and T be the current matching data points and transformation, respectively. Select $\boldsymbol{p}_i \in \{\boldsymbol{p}_{k-1}, \boldsymbol{p}_{l+1}\}$ minimizing the error $d(\mathcal{L}_i, T\boldsymbol{p}_i)^2$, and get a tangent line $\mathcal{L}_i$ to $\mathcal{M}$ at the closest point $\boldsymbol{q}_i$ to $T\boldsymbol{p}_i$.

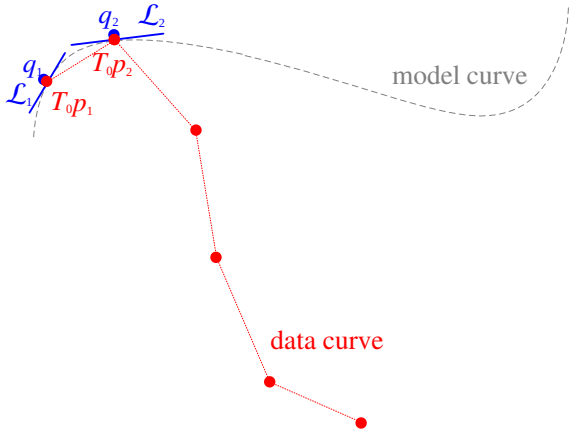
[2] A vertex of a curve is a point of where the curvature is a local maximum or minimum.



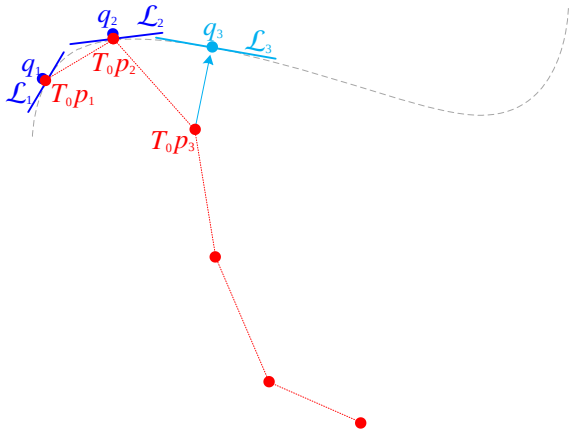
(a) The superposition of the data curve onto the model curve under the real transformation T^*. Each data point $\boldsymbol{p}_i$ is sampled at $\boldsymbol{r}_i$, $1 \le i \le 6$.



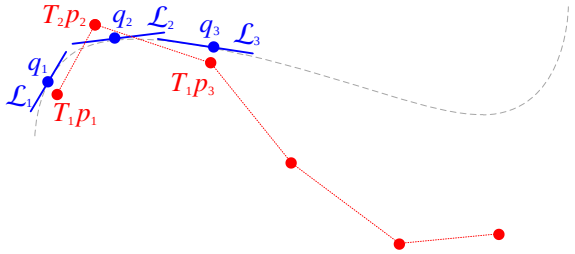
(b) Initial matching information. Initial matching model (and data) segment is highlighted on the model (and data) curve.



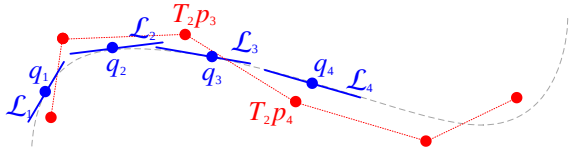
(c) Initial configuration, and two tangents $\mathcal{L}_1$ and $\mathcal{L}_2$ for their corresponding data points $\boldsymbol{p}_1$ and $\boldsymbol{p}_2$, respectively. Tangent $\mathcal{L}_i$ is obtained at point $\boldsymbol{q}_i \in \mathcal{M}$, $1 \le i \le 2$.



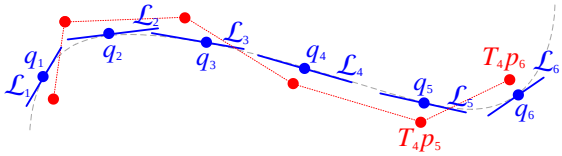
(d) New tangent $\mathcal{L}_3$ for the extending data point $\boldsymbol{p}_3$. Point $\boldsymbol{q}_3 \in \mathcal{M}$ is the closest to $T_0\boldsymbol{p}_3$.



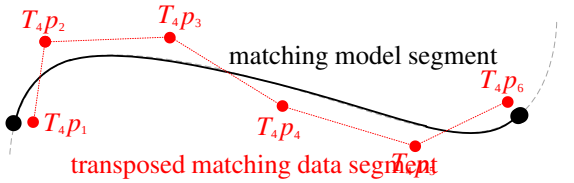
(e) Updated transformation T_1 that minimizes dissimilarity error $e(\langle 1..3\rangle, T_1)$



(f) Updated transformation T_2 minimizing $e(\langle 1..4\rangle, T_2)$



(g) Updated transformation T_4 minimizing $e(\langle 1..6\rangle, T_4)$



(h) Final result of the longest matching segment. The matching model segment is highlighted as the bold black line.

Figure 2.2: Scenario of iteratively extending matching segment. The data point $\boldsymbol{p}_i$, $1 \le i \le 6$, underwent the real transformation T^* and its source $\boldsymbol{r}_i$ are shown in (a); the initial matching model and data segments are highlighted in (b); the data segment is superposed onto the model curve with the initial matching information in (c); the tangent $\mathcal{L}_3$ to the model curve for the extending data points $\boldsymbol{p}_3$ is shown in (d) as the intermediate extending step; the results of extending the matching segment with $\boldsymbol{p}_3$, $\boldsymbol{p}_4$, and $\boldsymbol{p}_6$ are displayed in (e)-(g); and in the final match (f) the matching model segment is highlighted. Here it is assumed that $f(\langle 1..i\rangle, i-2) < \rho$ for all $2 \le i \le 6$ in (c)-(h).

3. Let k' and l' the min and max of $\{k, l, i\}$, respectively. Determine T' that minimizes $e(\langle k'..l'\rangle, T')$.

4. If $e(\langle k'..l'\rangle, T') > \varepsilon$ or $f(\langle k'..l'\rangle, T') > \rho$, then update $\mathcal{L}_{k'}, ..., \mathcal{L}_{l'}$ and T'.

5. If $e(\langle k'..l'\rangle, T') > \varepsilon$, stop and report $\mathcal{D}_{k..l}$ as the longest matching extension of $\mathcal{D}_{k_0..l_0}$ with T_0.

6. Otherwise, go back to step 2.

We will see that each iteration step takes constant time in the following sections except updating $\mathcal{L}_{k'}, ..., \mathcal{L}_{l'}$ in step 4. Section 2.2.1 describes in detail how to evaluate the errors $e(\langle k'..l'\rangle, T')$ and $f(\langle k'..l'\rangle, T')$ using $e(\langle k..l\rangle, T)$ and $f(\langle k..l\rangle, T)$. Sections 2.2.2 and 2.2.3 present an iterative procedure to polish a similarity transformation T' from T, formally introduced in Section 2.4 which minimizes $e(\langle k'..l'\rangle, T')$. Section 2.3 discusses the statistical analysis, on which this approach is based, and the choice of tolerances ε and ρ.

2.2.1 Error Update

Here, we denote $\boldsymbol{p}_n$ the nth extended point of the matching data segment of $\mathcal{D}$, $\mathcal{L}_n : \boldsymbol{a}_n t + \boldsymbol{q}_n$ the tangent line to $\mathcal{M}$ at $\boldsymbol{q}_n \in \mathcal{M}$ that corresponds to $\boldsymbol{p}_n$, and T_n the transformation matrix that minimizes $\sum_{i=1}^{n} d(\mathcal{L}_i, T_n \boldsymbol{p}_i)^2$. Here $\boldsymbol{p}_n$ and $\boldsymbol{q}_n$ are represented in homogeneous coordinates of the form $(x, y, z, 1)^t$, the tangent direction $\boldsymbol{a}_n$ is $(x, y, z, 0)^t$ with $\|\boldsymbol{a}_n\| = 1$, and T_n is a 4×4 transformation matrix whose last row is $(0, 0, 0, 1)$.

Denote the minimum curve similarity error (2.1) after the addition of $\boldsymbol{p}_n$ as $e_n = \sum_{i=1}^{n} d(\mathcal{L}_i, T_n \boldsymbol{p}_i)^2$. Notice that the squared distance between a point $\boldsymbol{p}_i$ and a line $\mathcal{L}_i : \boldsymbol{a}_i t + \boldsymbol{q}_i$ is $d(\mathcal{L}_i, \boldsymbol{p}_i)^2 = (\boldsymbol{q}_i - \boldsymbol{p}_i)^t (I - \boldsymbol{a}_i \boldsymbol{a}_i^t)(\boldsymbol{q}_i - \boldsymbol{p}_i)$, where I is the 4×4 identity matrix. Hence,

$$e_n = \sum_{i=1}^{n} (\boldsymbol{q}_i - T_n \boldsymbol{p}_i)^t (I - \boldsymbol{a}_i \boldsymbol{a}_i^t)(\boldsymbol{q}_i - T_n \boldsymbol{p}_i). \tag{2.3}$$

Write the above error in terms of that over the first $n - 1$ points:

$$\begin{aligned} e_n &= e_{n-1} + d(\mathcal{L}_n, T_{n-1}\boldsymbol{p}_n)^2 + \sum_{i=1}^{n} \Big(d(\mathcal{L}_i, T_n \boldsymbol{p}_i)^2 - d(\mathcal{L}_i, T_{n-1}\boldsymbol{p}_i)^2 \Big) \\ &= e_{n-1} + d(\mathcal{L}_n, T_{n-1}\boldsymbol{p}_n)^2 + \sum_{i=1}^{n} \Big(2 \operatorname{tr}\Big((I - \boldsymbol{a}_i \boldsymbol{a}_i^t) \boldsymbol{q}_i \boldsymbol{p}_i^t (T_{n-1} - T_n)^t \Big) \\ &\qquad + \operatorname{tr}\Big(\boldsymbol{p}_i \boldsymbol{p}_i^t T_n^t T_n \Big) - \operatorname{tr}\Big(\boldsymbol{p}_i \boldsymbol{p}_i^t T_{n-1}^t T_{n-1} \Big) - \operatorname{tr}(\boldsymbol{a}_i \boldsymbol{p}_i^t T_n^t)^2 + \operatorname{tr}(\boldsymbol{a}_i \boldsymbol{p}_i^t T_{n-1}^t)^2 \Big). \end{aligned} \tag{2.4}$$

It is obvious that $\boldsymbol{a}^t\boldsymbol{b} = \text{tr}(\boldsymbol{a}\boldsymbol{b}^t)$. Denote by $\text{vec}\,(A)$ the vector version[3] of a square matrix A. The trace $\text{tr}(AB^t)$ is equal to the inner product $\text{vec}\,(A) \cdot \text{vec}\,(B)$. In a similar way, $\text{tr}(AB^t)^2 = \text{vec}\,(A \otimes A) \cdot \text{vec}\,(B \otimes B)$, where $\otimes$ denotes the Kronecker product[4]. Using those notations, the second and fourth summands in the last summation in (2.4) can be rewritten as below:

$$\begin{aligned}
\sum_{i=1}^{n} \text{tr}\Big(\boldsymbol{p}_i\boldsymbol{p}_i^t \cdot T_n^t T_n\Big) &= \left(\sum_{i=1}^{n} \text{vec}\,(\boldsymbol{p}_i\boldsymbol{p}_i^t)\right) \cdot \text{vec}\,(T_n^t T_n), \\
\sum_{i=1}^{n} \text{tr}\Big(\boldsymbol{a}_i\boldsymbol{p}_i^t \cdot T_n^t\Big)^2 &= \left(\sum_{i=1}^{n} \text{vec}\,(\boldsymbol{a}_i\boldsymbol{p}_i^t \otimes \boldsymbol{a}_i\boldsymbol{p}_i^t)\right) \cdot \text{vec}\,(T_n \otimes T_n).
\end{aligned}$$

Similarly, the transformations T_n and T_{n-1} can be separated from other terms in e_n, yielding a recurrence form:

$$\begin{aligned}
e_n = e_{n-1} + d(\mathcal{L}_n, T_{n-1}\boldsymbol{p}_n)^2 &+ 2 \cdot \boldsymbol{\alpha}_n \cdot \text{vec}\,(T_{n-1} - T_n) \\
&+ \boldsymbol{\beta}_n \cdot \Big(\text{vec}\,(T_n^t T_n) - \text{vec}\,(T_{n-1}^t T_{n-1})\Big) \\
&+ \boldsymbol{\gamma}_n \cdot \Big(\text{vec}\,(T_{n-1} \otimes T_{n-1}) - \text{vec}\,(T_n \otimes T_n)\Big),
\end{aligned} \tag{2.5}$$

where

$$\begin{aligned}
\boldsymbol{\alpha}_n &= \boldsymbol{\alpha}_{n-1} + \text{vec}\,((I - \boldsymbol{a}_n\boldsymbol{a}_n^t)\boldsymbol{q}_n\boldsymbol{p}_n^t), \\
\boldsymbol{\beta}_n &= \boldsymbol{\beta}_{n-1} + \text{vec}\,(\boldsymbol{p}_n\boldsymbol{p}_n^t), \\
\boldsymbol{\gamma}_n &= \boldsymbol{\gamma}_{n-1} + \text{vec}\,(\boldsymbol{a}_n\boldsymbol{p}_n^t \otimes \boldsymbol{a}_n\boldsymbol{p}_n^t).
\end{aligned} \tag{2.6}$$

Note that the Kronecker products in (2.5) and (2.6) are independent of the number of points and can therefore be carried out in constant time. Therefore, $\boldsymbol{\alpha}_n, \boldsymbol{\beta}_n, \boldsymbol{\gamma}_n$ and e_n can be evaluated in $O(1)$ time from $\boldsymbol{p}_n$ and $\mathcal{L}_n : \boldsymbol{a}_n t + \boldsymbol{q}_n$.

[3]We denote $\text{vec}\,(A) = (a_{11}, a_{21}, a_{31}, ..., a_{n1}, a_{12}, a_{22}, ..., a_{nn})^t$ when

$$A = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{pmatrix}$$

[4]The Kronecker product $A \otimes B$ of an $m \times n$ matrix A and an $k \times l$ matrix B is an $km \times ln$ matrix, such that:

$$A \otimes B = \begin{pmatrix} a_{11}B & a_{12}B & \dots & a_{1n}B \\ a_{21}B & a_{22}B & \dots & a_{2n}B \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1}B & a_{m2}B & \dots & a_{mn}B \end{pmatrix}$$

We rewrite the relaxed condition error (2.2) as $f_n = \sum_{i=1}^{n} \|\boldsymbol{q}_i - T_n \boldsymbol{p}_i\|^2$, and formulate its computation as a recurrence:

$$\begin{aligned} f_n &= \eta_n - 2 \cdot \boldsymbol{\delta}_n \cdot \mathrm{vec}\,(T_n) + \boldsymbol{\beta}_n \cdot \mathrm{vec}\,(T_n^t T_n), \\ \eta_n &= \eta_{n-1} + \boldsymbol{q}_n^t \boldsymbol{q}_n, \\ \boldsymbol{\delta}_n &= \boldsymbol{\delta}_{n-1} + \mathrm{vec}\,(\boldsymbol{q}_n \boldsymbol{p}_n^t). \end{aligned} \tag{2.7}$$

Here η_n, $\boldsymbol{\delta}_n$ and f_n can be obtained in $O(1)$ with $\boldsymbol{p}_n$ and $\mathcal{L}_n$.

2.2.2 Update of Transformation Data

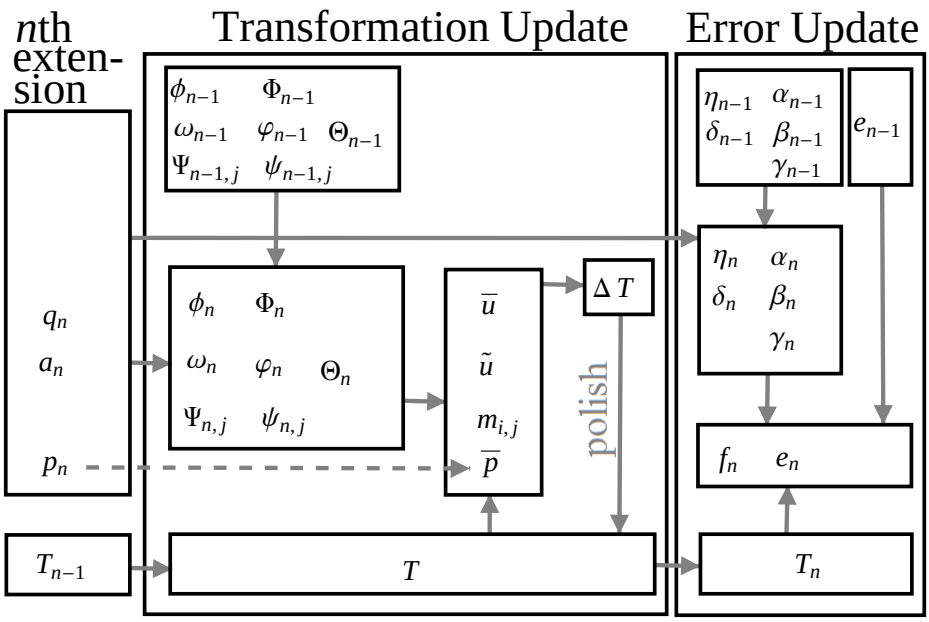


Figure 2.3: The transformation T_n is iteratively polished initially from T_{n-1} with the intermediate data ($\bar{\boldsymbol{u}}$, $\tilde{\boldsymbol{u}}$, $m_{i,j}$, and $\bar{\boldsymbol{p}}$), and then the errors e_n and f_n are determined using the updated transformation T_n. In the figure, terms $\boldsymbol{\phi}_n$, Φ_n, ω_n, $\boldsymbol{\varphi}_n$, Θ_n, Ψ_n, $\boldsymbol{\psi}_n$, $\boldsymbol{\alpha}_n$, $\boldsymbol{\beta}_n$, $\boldsymbol{\gamma}_n$, η_n, and $\boldsymbol{\delta}_n$ are defined as recurrence forms.

In step 5 of the procedure in the beginning of Section 2.2, it is assumed that the error e_n in (2.5) is minimized by the similarity transformation T_n. We here find its scale, rotation, and translation. For this, we initialize a transformation T as T_{n-1}, and repeatedly polish T until the variation of the error $\sum_{i=1}^{n} d(\mathcal{L}_i, T\boldsymbol{p}_i)^2$ vanishes. Finally, T_n is set as T. Figure 2.3 illustrates the procedure: (1) polish the transformation T_n initially from T_{n-1}, which minimizes $\sum_{i=1}^{n} d(\mathcal{L}_i, T_n \boldsymbol{p}_i)^2$, and then (2) update errors e_n and f_n described in Section 2.2.1. We will see that each iteration takes constant time. Since the repetition takes the constant number of iterations on average, we consider that T_n can be obtained in $O(1)$. In this section, we define several recurrences; they will be used in the following section to polish T.

Denote by $\boldsymbol{u}_i$ the closest point to $T\boldsymbol{p}_i$ on the tangent line $\mathcal{L}_i : \boldsymbol{a}_i t + \boldsymbol{q}_i$, $1 \le i \le n$. It can be written to single out T as follows:

$$\begin{aligned} \boldsymbol{u}_i &= \boldsymbol{q}_i + \boldsymbol{a}_i\left(\boldsymbol{a}_i \cdot (T\boldsymbol{p}_i - \boldsymbol{q}_i)\right) \\ &= (I - \boldsymbol{a}_i\boldsymbol{a}_i^t)\boldsymbol{q}_i + \left(\boldsymbol{p}_i^t \otimes (\boldsymbol{a}_i\boldsymbol{a}_i^t)\right) \cdot \mathrm{vec}\,(T). \end{aligned} \tag{2.8}$$

Decomposing T from $\boldsymbol{a}_i$, $\boldsymbol{q}_i$, and $\boldsymbol{p}_i$ in the above equation makes it possible to find an update T' of the transformation T to minimize a pointwise superposition error $\sum_{i=1}^{n} \|\boldsymbol{u}_i - T'T\boldsymbol{p}_i\|^2$ without estimating the closest points $\boldsymbol{u}_1, ..., \boldsymbol{u}_n$ to $T\boldsymbol{p}_1, ..., T\boldsymbol{p}_n$, respectively.

Let $\bar{\boldsymbol{u}}$ and $\bar{\boldsymbol{p}}$ be the centroids of $\{\boldsymbol{u}_1, ..., \boldsymbol{u}_n\}$ and $\{\boldsymbol{p}_1, ..., \boldsymbol{p}_n\}$, respectively. The transformation T' can be obtained from a scalar $\tilde{u} = \sum_{i=1}^{n} \|\boldsymbol{u}_i - \bar{\boldsymbol{u}}\|^2$, vectors $\bar{\boldsymbol{p}}$ and $\bar{\boldsymbol{u}}$, and a matrix

$$M = \sum_{i=1}^{n} (T\boldsymbol{p}_i - T\bar{\boldsymbol{p}})(\boldsymbol{u}_i - \bar{\boldsymbol{u}})^t. \tag{2.9}$$

Now, the remainder of this section describes how to update M, $\bar{\boldsymbol{p}}$, $\bar{\boldsymbol{u}}$ and $\tilde{u}$ in $O(1)$ time.

To simplify the notation, denote $\boldsymbol{v}_i = (I - \boldsymbol{a}_i\boldsymbol{a}_i^t)\boldsymbol{q}_i$ and $E_i = \boldsymbol{p}_i^t \otimes (\boldsymbol{a}_i\boldsymbol{a}_i^t)$. It is trivial to define $\bar{\boldsymbol{p}}$ as a recurrence. We reformulate $\bar{\boldsymbol{u}}$, $\tilde{u}$ and M using $\boldsymbol{v}_i$ and E_i. First, $\bar{\boldsymbol{u}}$ can be obtained using recurrences as follows:

$$\begin{aligned} \bar{\boldsymbol{u}} &= \frac{1}{n}\sum_{i=1}^{n} \boldsymbol{v}_i + \frac{1}{n}\sum_{i=1}^{n} E_i \cdot \mathrm{vec}\,(T) \\ &= \frac{1}{n}\boldsymbol{\phi}_n + \frac{1}{n}\Phi_n \cdot \mathrm{vec}\,(T), \end{aligned} \tag{2.10}$$

where

$$\begin{aligned} \boldsymbol{\phi}_n &= \boldsymbol{\phi}_{n-1} + \boldsymbol{v}_n, \\ \Phi_n &= \Phi_{n-1} + E_n. \end{aligned} \tag{2.11}$$

Next, $\tilde{u}$ can be rewritten:

$$\begin{aligned} \tilde{u} &= \sum_{i=1}^{n} \boldsymbol{u}_i^t\boldsymbol{u}_i - n \cdot \bar{\boldsymbol{u}}^t\bar{\boldsymbol{u}} \\ &= \omega_n \;+\; 2\,\boldsymbol{\varphi}_n^t\,\mathrm{vec}\,(T) \;+\; \mathrm{vec}\,(T)^t\,\Theta_n\,\mathrm{vec}\,(T) \;-\; n \cdot \bar{\boldsymbol{u}}^t\bar{\boldsymbol{u}}, \end{aligned} \tag{2.12}$$

where

$$\begin{aligned} \omega_n &= \omega_{n-1} + \boldsymbol{v}_n^t\boldsymbol{v}_n, \\ \boldsymbol{\varphi}_n &= \boldsymbol{\varphi}_{n-1} + \boldsymbol{v}_n^t E_n, \\ \Theta_n &= \Theta_{n-1} + E_n^t E_n. \end{aligned} \tag{2.13}$$

Finally, we derive the entry $m_{i,j}$ which lies in the ith row and jth column of M. Denote by $A_{k,*}$ the kth row vector of a matrix A. Note that the kth entries of the vectors $T\boldsymbol{p}_i$ and $\boldsymbol{u}_i$ are written as follows:

$$\begin{aligned}(T\boldsymbol{p}_i)_k &= T_{k,*} \cdot \boldsymbol{p}_i, \\ (\boldsymbol{u}_i)_k &= (\boldsymbol{v}_i)_k + (E_i)_{k,*} \cdot \mathrm{vec}\,(T).\end{aligned}$$

The kth entries of $T\bar{\boldsymbol{p}}$ and $\bar{\boldsymbol{u}}$ can be written in similar forms. Now, we have

$$\begin{aligned}m_{i,j} &= \sum_{k=1}^{n} \left((T\boldsymbol{p}_k)_i \cdot (\boldsymbol{u}_k)_j \right) - n \cdot (T\bar{\boldsymbol{p}})_i \cdot \bar{\boldsymbol{u}}_j \\ &= T_{i,*} \left(\Psi_{n,j} - \bar{\boldsymbol{p}} \left(\Phi_n\right)_{j,*} \right) \mathrm{vec}\,(T) + T_{i,*} \left(\boldsymbol{\psi}_{n,j} - \bar{\boldsymbol{p}} \left(\boldsymbol{\phi}_n\right)_j \right), \end{aligned} \tag{2.14}$$

where

$$\begin{aligned}\Psi_{n,j} &= \Psi_{n-1,j} + \boldsymbol{p}_n \left(E_n\right)_{j,*}, \\ \boldsymbol{\psi}_{n,j} &= \boldsymbol{\psi}_{n-1,j} + \boldsymbol{p}_n \left(\boldsymbol{v}_n\right)_j. \end{aligned} \tag{2.15}$$

Notice that terms $\boldsymbol{\phi}_n$, Φ_n, ω_n, $\boldsymbol{\varphi}_n$, Θ_n, $\Psi_{n,j}$ and $\boldsymbol{\psi}_{n,j}$ in (2.11), (2.13) and (2.15) are updated when the point $\boldsymbol{p}_n$ and the tangent line $\mathcal{L}_n$ are considered. Then, $\bar{\boldsymbol{u}}$, $\tilde{u}$ and M can be obtained in $O(1)$ time using the terms and the transformation T independently from $\boldsymbol{p}_i$ and $\mathcal{L}_i$, $1 \le i \le n$.

2.2.3 Transformation Update

Using M, $\bar{\boldsymbol{p}}$, $\bar{\boldsymbol{u}}$ and $\tilde{u}$, we can find the transformation T^*, composed of scale s^*, rotation R^*, and translation $\boldsymbol{b}^*$, that minimizes the pointwise superposition error $\sum_{i=1}^{n} \|\boldsymbol{u}_i - T^* T \boldsymbol{p}_i\|^2$. The following briefly lists the steps of computations:

$$\begin{aligned}R^* &= \operatorname*{argmax}_R \sum_{i=1}^{n} R(T\boldsymbol{p}_i - T\bar{\boldsymbol{p}}) \cdot (\boldsymbol{u}_i - \bar{\boldsymbol{u}}); \\ \hat{s} &= \frac{\sum_{i=1}^{n} \|\boldsymbol{u}_i - \bar{\boldsymbol{u}}\|^2}{\sum_{i=1}^{n} R^*(T\boldsymbol{p}_i - T\bar{\boldsymbol{p}}) \cdot (\boldsymbol{u}_i - \bar{\boldsymbol{u}})}; \\ s^* &= \begin{cases} \hat{s}, & \text{if } s_{\min} \le \hat{s} \le s_{\max}; \\ s_{\min}, & \text{if } \left|\frac{1}{\hat{s}} - \frac{1}{s_{\min}}\right| < \left|\frac{1}{\hat{s}} - \frac{1}{s_{\max}}\right|; \\ s_{\max}, & \text{otherwise}; \end{cases} \\ \boldsymbol{b}^* &= \bar{\boldsymbol{u}} - s^* R^* T \bar{\boldsymbol{p}}, \end{aligned} \tag{2.16}$$

where $s_{\min}$ and $s_{\max}$ are the maximum and minimum scales under prior knowledge. The above procedure minimizes $\sum_{i=1}^{n} \|\boldsymbol{u}_i - T^* T \boldsymbol{p}_i\|^2$ as well as extends the arc length of a matching model segment. Its detail will be presented in Section 2.4.

The rotation matrix R^* is obtained using unit quaternions [13] and principal component analysis (PCA) [15]. The summation $\sum_{i=1}^{n} R(T\boldsymbol{p}_i - T\bar{\boldsymbol{p}}) \cdot (\boldsymbol{u}_i - \bar{\boldsymbol{u}})$ can be reformulated using a matrix $\tilde{M}$ which is generated from $m_{i,j}$ in (2.14). The matrix $\tilde{M}$ helps decompose the unit rotation quaternion q from the summation. The scale $\hat{s}$ can be obtained from $\tilde{M}$ in detail as follows:

$$\begin{aligned}
\tilde{M} &= \begin{pmatrix} m_{1,1}+m_{2,2}+m_{3,3} & m_{2,3}-m_{3,2} & m_{3,1}-m_{1,3} & m_{1,2}-m_{2,1} \\ m_{2,3}-m_{3,2} & m_{1,1}-m_{2,2}-m_{3,3} & m_{2,1}+m_{1,2} & m_{3,1}+m_{1,3} \\ m_{3,1}-m_{1,3} & m_{2,1}+m_{1,2} & m_{2,2}-m_{3,3}-m_{1,1} & m_{3,2}+m_{2,3} \\ m_{1,2}-m_{2,1} & m_{3,1}+m_{1,3} & m_{3,2}+m_{2,3} & m_{3,3}-m_{1,1}-m_{2,2} \end{pmatrix}; \\
q^* &= \arg\max_{q, \|q\|=1} (q^t \ \tilde{M} \ q); \\
\hat{s} &= \frac{\tilde{u}}{q^{*t} \ \tilde{M} \ q^*}.
\end{aligned} \tag{2.17}$$

Here q^* is the unit eigenvector of the matrix $\tilde{M}$ that corresponds to the maximum eigenvalue. The rotation matrix R^* is determined by q^* [13].

The transformation T_n that minimizes the error e_n in (2.5) can be obtained as follows:

1. Initialize T as T_{n-1}.
2. Get $\bar{\boldsymbol{u}}$, $\tilde{u}$, and M using (2.10), (2.12), and (2.14).
3. Polish T using $\langle s^*, R^*, \boldsymbol{b}^* \rangle$ which are obtained from $\bar{\boldsymbol{u}}$, $\tilde{u}$, and M using (2.16) and (2.17).
4. Repeat steps 2-3 until $\sum_{i=1}^{n} d(\mathcal{L}_i, T\boldsymbol{p}_i)^2$ no longer varies, then set T_n as T.

The error $\sum_{i=1}^{n} d(\mathcal{L}_i, T\boldsymbol{p}_i)^2$ in step 4 is obtained in a similar way to (2.5). All terms are estimated in homogeneous coordinates, except b^* in Cartesian coordinates because R^* is a 3×3 rotation matrix[5] in (2.16). Notice that each repetition takes constant time.

In simulation, the error $\sum_{i=1}^{n} d(\mathcal{L}_i, T\boldsymbol{p}_i)^2$ no longer varies within the average of 5.6 iterations. Therefore, we consider the optimal transformation T_n is obtained in constant time.

[5]The 3×3 rotation matrix R^* can be determined from the unit rotation quaternion $q^* = (r, i, j, k)^t$ as follows:

$$R^* = \begin{pmatrix} r^2+i^2-j^2-k^2 & 2(ij-rk) & 2(ik+rj) \\ 2(ij+rk) & r^2-i^2+j^2-k^2 & 2(jk-ri) \\ 2(ik-rj) & 2(jk+ri) & r^2-i^2-j^2+k^2 \end{pmatrix}.$$

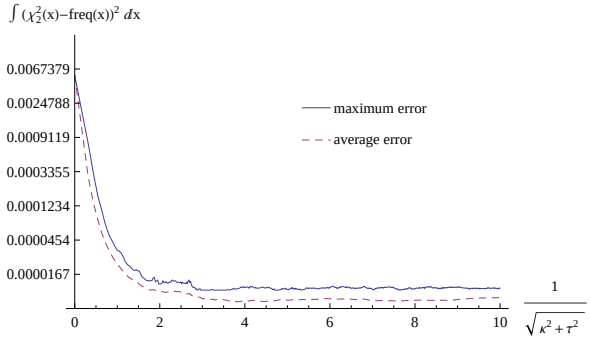


Figure 2.4: Plot of dissimilarity between the chi-square χ^2_2 distribution and the frequency distribution of squared distance from a noise $\boldsymbol{\xi}$ to the Frenet frame approximation $\mathcal{F}_{\kappa,\tau}$.

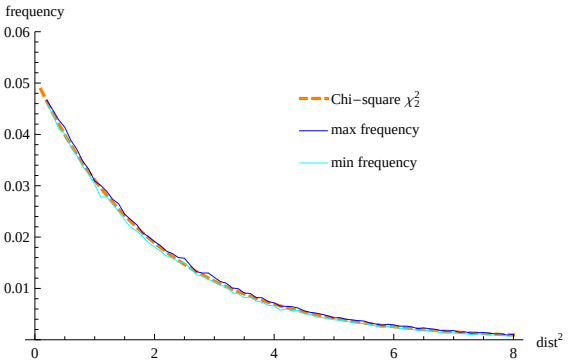


Figure 2.5: Overlay of two distributions as an expansion of the statistics in Figure 2.4 where $1/\sqrt{\kappa^2+\tau^2} = 3$.

2.3 Noise Analysis

Recall the procedure described in the beginning of Section 2.2 that iteratively extends the matching segment of the model curve $\mathcal{M}$ and the data curves $\mathcal{D}$. In the procedure, we stop the extension of the matching data segment or update the tangent line $\mathcal{L}_i$, $k \le i \le l$, when the dissimilarity error $e(\langle k..l\rangle, T)$ of (2.1) or the relaxed condition error $f(\langle k..l\rangle, T)$ of (2.2) is greater than the thresholds ε or ρ, respectively. Here we look at how the thresholds ε and ρ are defined. First, we will examine the distribution of the squared distance between noise point and an approximation of an arbitrary curve in the following section, to determine the statistics of the dissimilarity error $e(\langle k..l\rangle, T)$. Then, we will define the thresholds ε and ρ.

2.3.1 Analysis of Squared Distance of Noisy Point

Figure 2.4 and 2.5 summarizes the simulation results based on the Frenet frame approximation $\mathcal{F}_{\kappa,\tau} : \{t, \frac{\kappa}{2}t^2, \frac{\kappa\tau}{6}t^3\}$, the best quadratic approximation of a space curve at a point on

[6]Let $x_1, x_2, \ldots, x_k$ be identically independent random variable following a normal distribution $N(0,1)$. The sum of the sequence follows the *chi-square distribution* χ^2_k with k degrees of freedom:

$$\sum_{i=1}^{k} x_i^2 \sim \chi^2_k.$$

[7]Let $f(x;k)$ be the probabilistic density function of χ^2_k, and $g(x;\kappa,\tau)$ the frequency of "$d(\mathcal{F}_{\kappa,\tau},\boldsymbol{\xi})^2 = x$" where $\boldsymbol{\xi}$ is a Gaussian noise with mean $\mu = 0$ and variance $\sigma^2 = 1$. The dissimilarity measure for comparing the chi-square distribution χ^2_2 and the frequency distribution of $d(\mathcal{F}_{\kappa,\tau},\boldsymbol{\xi})^2$ is:

$$\int_0^\infty (f(x;2) - g(x;\kappa,\tau))^2\, dx.$$

the curve using curvature κ and torsion τ. Let $\boldsymbol{\xi}$ be random noise which follows a normal distribution $N(0,1)$. Recall that $d(\mathcal{F}_{\kappa,\tau}, \boldsymbol{\xi})$ is the distance from $\boldsymbol{\xi}$ to the Frenet frame approximation $\mathcal{F}_{\kappa,\tau}$. Figure 2.4 plots the dissimilarity[7] between the chi-square distribution[6] χ^2_2 and the frequency distribution of $d(\mathcal{F}_{\kappa,\tau}, \boldsymbol{\xi})^2$ as $1/\sqrt{\kappa^2+\tau^2}$ changes. Here χ^2_2 is the distribution of the squared distance from $\boldsymbol{\xi}$ to x-axis. In the figure, the frequency distribution approximates to χ^2_2 as $\sqrt{\kappa^2+\tau^2}$ decreases. It is because the tangent line to $\mathcal{F}_{\kappa,\tau}$ at the closest point to $\boldsymbol{\xi}$ approximates x-axis. Figure 2.5 gives another perspective of the dissimilarity by expanding the statistics when $1/\sqrt{\kappa^2+\tau^2} = 3$. It overlays the probability density function (PDF) of the chi-square χ^2_2 distribution and the frequency of the squared distance $d(\mathcal{F}_{\kappa,\tau}, \boldsymbol{\xi})^2$. In the figure, the maximum and minimum frequencies, as the curvature κ and torsion τ vary under the condition $1/\sqrt{\kappa^2+\tau^2} = 3$, are displayed as two solid lines.

The simulation is related to the noise distribution of the data point that is sampled from a point on the model curve. Recall the assumption in Section 2.1 that each data point $\boldsymbol{p}_i$ on the data segment $\mathcal{D}_{k..l}$ was sampled from a source $\boldsymbol{r}_i$ on the model curve $\mathcal{M}$ under Gaussian noise and underwent some transformation T^{-1}, where $k \le i \le l$. The transformed data point $T\boldsymbol{p}_i$, the model curve $\mathcal{M}$, the source $\boldsymbol{r}_i$, and the tangent line $\mathcal{K}_i$ at $\boldsymbol{r}_i$ correspond to the terms $\boldsymbol{\xi}$, $\mathcal{F}_{\kappa,\tau}$, the origin, and the x-axis of the Frenet frame, respectively. The simulation implies that, when the data curve $\mathcal{D}_{k..l}$ is superposed onto its source $\mathcal{M}$ under the transformation T, the frequency distribution of $d(\mathcal{M}, T\boldsymbol{p}_i)$ can be considered as that of $d(\mathcal{K}_i, T\boldsymbol{p}_i)$, $k \le i \le l$, in a certain condition because those distributions are similar. Here we further consider that the tangent line $\mathcal{K}_i$ is approximated by another tangent line to $\mathcal{M}$ at the closest point to $T\boldsymbol{p}_i$.

2.3.2 Tolerance

Recall that the dissimilarity error $e(\langle k..l\rangle, T)$ of (2.1) is the approximation of the dissimilarity $\sum_{i=k}^{l} d(\mathcal{K}_i, T\boldsymbol{p}_i)^2$ between $\mathcal{M}$ and $\mathcal{D}_{k..l}$ under the transformation T. We know from Section 2.3.1 that $d(\mathcal{K}_i, T\boldsymbol{p}_i)^2$ follows χ^2_2, for all $k \le i \le l$. By definition, the sum $\sum_{i=k}^{l} d(\mathcal{K}_i, T\boldsymbol{p}_i)^2$ follows the chi-square distribution $\chi^2_{2(l-k+1)}$ with $2(l-k+1)$ degrees of freedom, when the noise $\boldsymbol{\xi}_i$ of the data point $\boldsymbol{p}_i$ follows a normal distribution $N(0,1)$, $k \le i \le l$.

Let us generalize the noise variance σ^2 of the data points. Denote by $f_{\sigma^2}(x;k)$ the probability density function of the chi-square χ^2_k distribution with k degrees of freedom and the noise variance σ^2. Let $n = l-k+1$. We let $\Pr(e \mid n, \sigma^2)$ be the probability that $\sum_{i=k}^{l} d(\mathcal{K}_i, T\boldsymbol{p}_i)^2$ has value e:

$$\Pr(e \mid n, \sigma^2) = c\, f_{\sigma^2}(e; 2n), \tag{2.18}$$

where c is a normalizing constant to make it a probability density function. We define a threshold that determines the statistical equivalence of $\mathcal{D}_{k..l}$ and $\mathcal{M}$, using the significance

level α meaning that $\sum_{i=k}^{l} d(\mathcal{K}_i, T\boldsymbol{p}_i)^2$ unlikely has value e by chance, as

$$\begin{aligned}
\varepsilon(n, \sigma^2) &= \text{root } e \text{ of } \left[\left(\int_e^{\infty} \Pr(h \mid n, \sigma^2)\, dh\right) - \alpha\right] \\
&= \text{root } e \text{ of } \left[\frac{\Gamma\left(n,\ e/2\sigma^2\right)}{\Gamma(n)} - \alpha\right] \\
&= 2\sigma^2 \cdot \left(\text{ root } e \text{ of } \left[\Gamma\left(n, e\right) - \alpha\Gamma(n)\right]\right) \\
&= \sigma^2 \cdot \varepsilon(n, 1),
\end{aligned} \tag{2.19}$$

where "root e of $[f(e)]$" is a root value e of an arbitrary function $f(e)$, and $\Gamma(x)$ and $\Gamma(x, y)$ are the gamma function and the upper incomplete gamma function[8], respectively. In the above equation, it is obvious that there exists a unique threshold $\varepsilon(n, \sigma^2)$ given n and σ^2 because the integration term $\int_0^e \Pr(h \mid n, \sigma^2)\, dh$ is a monotonically increasing function of e.

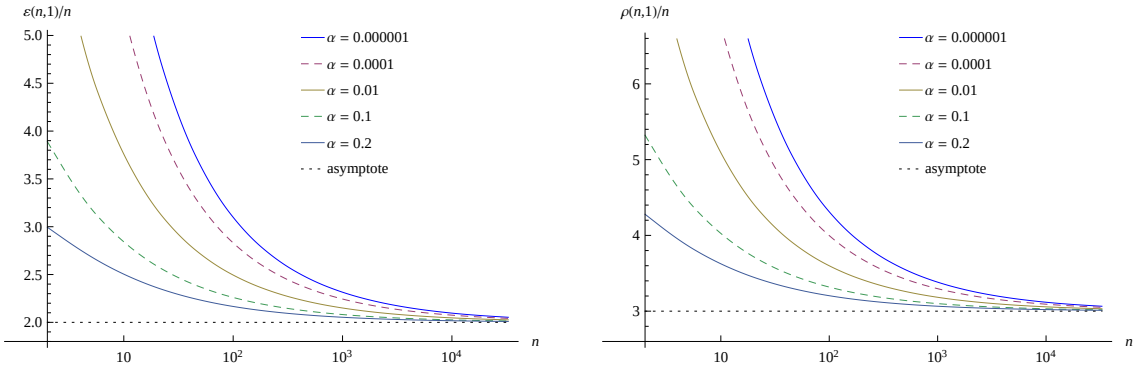
Similarly, the threshold ρ of the relaxed condition error (2.2) can be defined. Recall that $f(\langle k..l\rangle, T)$ is the relaxed condition error $\sum_{i=k}^{l} \|\boldsymbol{q}_i - T\boldsymbol{p}_i\|^2$, and $\boldsymbol{r}_i$ is the source of $\boldsymbol{p}_i$. For the threshold ρ, we assume that $\boldsymbol{q}_i$ approximates to $\boldsymbol{r}_i$; therefore, $\mathcal{L}_i$ approximates to $\mathcal{K}_i$. Now, each summand $\|\boldsymbol{q}_i - T\boldsymbol{p}_i\|^2$ follows the chi-square distribution χ^2_3. Similar to $\varepsilon(n, \sigma^2)$, the threshold that determines the statistical accuracy of approximating $\mathcal{K}_i$ by $\mathcal{L}_i$ is defined as follows:

$$\begin{aligned}
\rho(n, \sigma^2) &= \text{root } e \text{ of } \left[\left(\int_e^{\infty} cf_{\sigma^2}(h; 3n)\, dh\right) - \alpha\right] \\
&= \text{root } e \text{ of } \left[\frac{\Gamma\left(1.5\ n,\ e/2\sigma^2\right)}{\Gamma(1.5\ n)} - \alpha\right] \\
&= 2\sigma^2 \cdot \left(\text{ root } e \text{ of } \left[\Gamma\left(1.5\ n,\ e\right) - \alpha\Gamma(1.5\ n)\right]\right) \\
&= \sigma^2 \cdot \rho(n, 1)
\end{aligned} \tag{2.20}$$

The thresholds $\varepsilon(n, \sigma^2)$ and $\rho(n, \sigma^2)$ can be obtained from $\varepsilon(n, 1)$ and $\rho(n, 1)$ which can be cached in advance, respectively. For both thresholds, we use $\alpha = 0.000001$ so very few matching scenarios are excluded.

For a given noise variance $\sigma^2 = 1$, the threshold functions (2.19) and (2.20) are monotonically decreasing, have only one root for any value of n, and converge to $2n$ and $3n$ as n increases, respectively. Figure 2.6(a) plots the dependence of the threshold distribution $\frac{1}{n}\varepsilon(n, \sigma^2)$ on n and α when $\sigma^2 = 1$; Figure 2.6(b) plots that of $\frac{1}{n}\rho(n, \sigma^2)$. Each threshold curve corresponds to a single significance level α, which ranges over 0.000001 to 0.2. Additionally, the figures show that the thresholds have to increase as the significance level α decreases in order to include more statistical cases.

[8]The gamma function $\Gamma(s)$ is an extension of factorial function $(s-1)!$ to real and complex number, which is expressed as an integration with interval $[0, \infty)$. The upper incomplete gamma function $\Gamma(s, x)$ is a gamma function with integration interval $[x, \infty)$.



(a) (b)

Figure 2.6: Thresholds $\frac{1}{n}\varepsilon(n,1)$ and $\frac{1}{n}\rho(n,1)$ versus the length n of a data segment. Each tolerance curve in (a) and (b) corresponds to a single significance level α, which ranges over 0.000001 to 0.2.

2.4 Transformation Minimizing Superposition Error

In Section 2.2.3, we described a procedure (2.16) that finds the scale s^*, the rotation R^*, and the translation $\boldsymbol{b}^*$. Here we look at how the procedure is made.

Let two point sets $W = \{\boldsymbol{w}_k, ..., \boldsymbol{w}_l\}$ and $U = \{\boldsymbol{u}_k, ..., \boldsymbol{u}_l\}$ be given. Points $\boldsymbol{w}_i$ and $\boldsymbol{u}_i$, $k \le i \le l$, represent a transformed data point $T\boldsymbol{p}_i$ and a point on a line $\mathcal{L}_i$ that is closest to $T\boldsymbol{p}_i$, respectively. Denote by ζ the arc length of the matching model segment whose starting and ending points are the closest to $T\boldsymbol{p}_k$ and $T\boldsymbol{p}_l$, respectively. Assume that $\boldsymbol{w}_i$ has noise ξ_i following the normal distribution $N(0, \sigma^2)$. Here we find the similarity transformation T^* (rotation R^*, scale s^*, and translation $\boldsymbol{b}^*$) that minimizes a pointwise superposition error $\sum_{i=k}^{l} \|\boldsymbol{u}_i - T^*\boldsymbol{w}_i\|^2$ and enlarges the arc length ζ of the matching model segment.

Define $\Pr(s, R, \boldsymbol{b}, \zeta \mid s_{\text{pri}})$ as the probability of moving $\boldsymbol{w}_i$ to $\boldsymbol{u}_i$, $k \le i \le l$, under a scale s given prior knowledge s_{pri}, a rotation R, and a translation $\boldsymbol{b}$, and enlarging the arc length ζ. The probability can be decomposed according to the dependencies among the variables as follows:

$$\Pr(s, R, \boldsymbol{b}, \zeta \mid s_{\text{pri}}) = \Pr(s, R, \boldsymbol{b}) \cdot \Pr(\zeta \mid s) \cdot \Pr(s \mid s_{\text{pri}}) / \Pr(s). \tag{2.21}$$

Let us explain each probability component. Denote by $\boldsymbol{a}_k^*$ (and $\boldsymbol{a}_l^*$) the unit tangent vector to $\mathcal{M}$ at the closest point to $T\boldsymbol{p}_k$ (and $T\boldsymbol{p}_l$). The increase of the model segment length ζ under scale s is defined, using $\boldsymbol{a}_k^*$, $\boldsymbol{a}_l^*$, and the centroid $\bar{\boldsymbol{w}}$ of the point set $W = \{\boldsymbol{w}_k, \ldots, \boldsymbol{w}_l\}$, as

$$\Delta\zeta = (s-1)\Big(((\bar{\boldsymbol{w}} - \boldsymbol{w}_k) \cdot \boldsymbol{a}_k^*) - ((\bar{\boldsymbol{w}} - \boldsymbol{w}_l) \cdot \boldsymbol{a}_l^*)\Big). \tag{2.22}$$

The detail of deriving (2.22) will be provided in Appendix A. The second factor in (2.22) can be considered as a constant because it is independent of the scale s. Since $\Delta\zeta$ is proportional

to s, we define the probability $\Pr(\zeta \mid s)$ of extending the model segment under scale s as

$$\Pr(\zeta \mid s) = c_1 s^2. \tag{2.23}$$

The constant c_1 is chosen such that $\int_0^{s_{\max}} \Pr(\zeta \mid s) ds = 1$, assuming that there is an upper bound $s_{\max}$ for s.

The transformation error $g(s, R, \boldsymbol{b})$ between two point sets W and U is defined as follows:

$$g(s, R, \boldsymbol{b}) = \sum_{i=k}^{l} \|sR(\boldsymbol{w}_i - \boldsymbol{\xi}_i) + \boldsymbol{b} - \boldsymbol{u}_i\|^2. \tag{2.24}$$

The noises in the data points can be shown to cancel altogether. We define the probability $\Pr(s, R, \boldsymbol{b})$ of the similarity between the two point sets as follows:

$$\Pr(s, R, \boldsymbol{b}) = \frac{c_2}{g(s, R, \boldsymbol{b})/(l - k + 1)}, \tag{2.25}$$

where the constant c_2 is a normalization factor for $\Pr(s, R, \boldsymbol{b})$ under the assumption that there is an upper bound $e_{\max}$ of $e(s, R, \boldsymbol{b})$.

The probability $\Pr(s \mid s_{\text{pri}})$ of a scale s given the prior knowledge $s_{\text{pri}} = \langle s_{\min}, s_{\max} \rangle$ of the maximum and minimum scales is defined as a filtering function:

$$\Pr(s \mid s_{\text{pri}}) = \begin{cases} \frac{1}{s_{\max} - s_{\min}}, & \text{if } s_{\min} \le s \le s_{\max}, \\ 0, & \text{otherwise.} \end{cases} \tag{2.26}$$

The last probability term $\Pr(s)$ in (2.21) is a constant for a given scale s.

Finding the optimal transformation turns into a minimization problem as follows:

$$\begin{aligned} \langle s^*, R^*, \boldsymbol{b}^* \rangle &= \operatorname*{argmax}_{s,R,\boldsymbol{b}} \Pr(s, R, \boldsymbol{b}, \zeta \mid s_{\text{pri}}) \\ &= \operatorname*{argmin}_{s,R,\boldsymbol{b}} \frac{g(s, R, \boldsymbol{b})}{s^2} \cdot \frac{1}{\Pr(s \mid s_{\text{pri}})}. \end{aligned} \tag{2.27}$$

In the above, we need only minimize the first factor $g(s, R, \boldsymbol{b})/s^2$ because $\Pr(s \mid s_{\text{pri}})$ is a filtering function which bounds the scale s to be inside the allowing range $[s_{\min}, s_{\max}]$. Rewrite the problem as

$$\begin{aligned} \langle \hat{s}, \hat{R}, \hat{\boldsymbol{b}} \rangle &= \operatorname*{argmin}_{s,R,\boldsymbol{b}} \frac{1}{s^2} g(s, R, \boldsymbol{b}) \\ &= \operatorname*{argmin}_{s,R,\boldsymbol{b}} \sum_{i=k}^{l} \left\| R(\boldsymbol{w}_i - \boldsymbol{\xi}_i) + \frac{\boldsymbol{b}}{s} - \frac{\boldsymbol{u}_i}{s} \right\|^2. \end{aligned} \tag{2.28}$$

Because the noise value $\boldsymbol{\xi}_i$ following the normal distribution $N(0, \sigma^2)$ is undeterminable, we find $\langle \hat{s}, \hat{R}, \hat{\boldsymbol{b}} \rangle$ that minimizes the average of the summation in (2.28). Recall that $\bar{\boldsymbol{w}}$ is the

centroid of the point set W. Let $\bar{\boldsymbol{u}}$ be that of the point set U. Let $\boldsymbol{w}'_i = \boldsymbol{w}_i - \bar{\boldsymbol{w}}$ and $\boldsymbol{u}'_i = \boldsymbol{u}_i - \bar{\boldsymbol{u}}$. The minimization can be decomposed as follows:

$$\langle \hat{s}, \hat{R} \rangle = \operatorname*{argmin}_{s,R} \sum_{i=k}^{l} E\left[\left\| R(\boldsymbol{w}'_i - \boldsymbol{\xi}_i) - \frac{\boldsymbol{u}'_i}{s} \right\|^2 \right], \tag{2.29}$$

$$\hat{\boldsymbol{b}} = \bar{\boldsymbol{u}} - \hat{s}\hat{R}\bar{\boldsymbol{w}}, \tag{2.30}$$

where $E[x]$ is the average value of x. The details of decomposing (2.28) into (2.29) and (2.30) will be provided in Appendix B. Equation (2.29) shows that this problem is equivalent to find the relative scale $1/s$ applied onto the model curve $\mathcal{M}$. Expand the summation in (2.29) as

$$\sum_{i=k}^{l} \|\boldsymbol{w}'_i\|^2 - \frac{2}{s} \sum_{i=k}^{l} R\boldsymbol{w}'_i \cdot \boldsymbol{u}'_i + \frac{1}{s^2} \sum_{i=k}^{l} \|\boldsymbol{u}'_i\|^2 + 3(l-k+1)\sigma^2. \tag{2.31}$$

In the above, the noise $\boldsymbol{\xi}_i$, $k \le i \le l$, has been eliminated except its variance σ^2. Since the variance is a constant here, the scale $\hat{s}$ in (2.29) becomes independent of the noise $\boldsymbol{\xi}_i$, for all $1 \le i \le n$. This prevents the case that the data set W is scaled down to a point on the model segment $\mathcal{M}$ while repeatedly polishing T as described in Section 2.2.3. The prevention is especially effective when the scale s is not bounded by $s_{\min}$ or statistically undeterminable. The minimization in (2.29) is thus decomposed:

$$\hat{s} = \frac{\sum_{i=k}^{l} \|\boldsymbol{u}'_i\|^2}{\sum_{i=k}^{l} \hat{R}\boldsymbol{w}'_i \cdot \boldsymbol{u}'_i}, \tag{2.32}$$

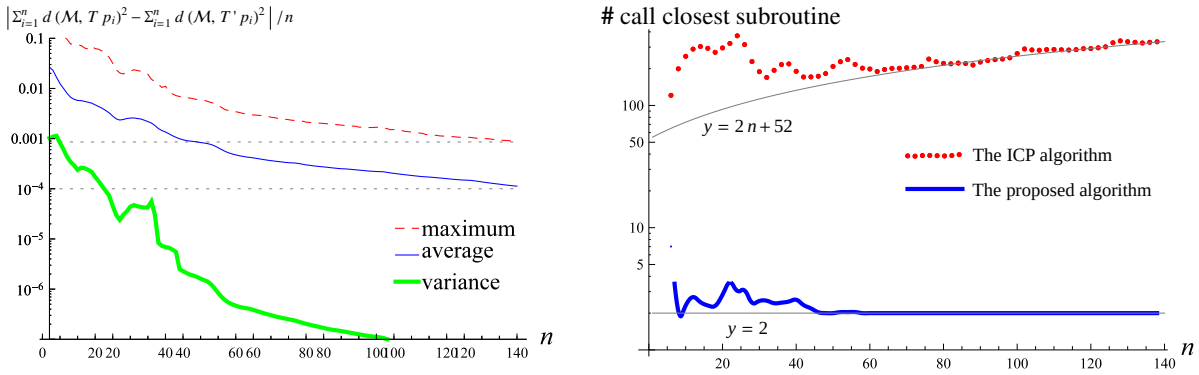
$$\hat{R} = \operatorname*{argmax}_{R} \sum_{i=k}^{l} R\boldsymbol{w}'_i \cdot \boldsymbol{u}'_i. \tag{2.33}$$

Getting back to the initial optimization problem (2.27), because $\Pr(s|s_{\text{pri}})$ is a filtering function, the optimal transformation $\langle s^*, R^*, \boldsymbol{b}^* \rangle$ can be determined in multiple steps as given in (2.16).

CHAPTER 3. EXPERIMENTS

The matching algorithm has been tested over synthetic data and range data. Synthetic data are sampled from splines as model curves under Gaussian noise. Range data are acquired using a desktop 3D scanner manufactured by NextEngine, Inc. The data points are generated by following a line of curvature[9] on a 3D mesh surface output from the scanner. The model curve is obtained by fitting a B-spline over these data points. The noise distribution for the data points is then estimated by comparing the model curve with the data points. The 7 number of B-splines and the 67 sequences of data points are tested as the synthetic and range data for experiments, respectively.

3.1 Comparison with the ICP Algorithm



(a) (b)

Figure 3.1: Comparisons of (a) accuracies and (b) performances of the ICP algorithm and our algorithm. (a) plots red (dashed), blue (solid), and green (thick) lines for the maximum, average, and variance of matching error differences between ICP and proposed algorithms, respectively. (b) plots the count of finding the closest point in each extending step by the ICP algorithm (dashed red) and by the proposed algorithm (solid blue), with their asymptotes $y = 2n + 52$ and $y = 2$.

Figure 3.1 compares the accuracies and performances of our algorithm and the ICP algorithm. Figure 3.1(a) plots the maximum (in a dashed red line), average (in a solid blue line),

[9]The tangent of the curve is always in the principal direction

and variance (in a thick green line) of the matching error differences between two algorithms as the number n of matching data points increases. The difference of the matching error is defined as $\frac{1}{n}\left|\sum_{i=1}^{n} d(\mathcal{M}, T\boldsymbol{p}_i)^2 - \sum_{i=1}^{n} d(\mathcal{M}, T'\boldsymbol{p}_i)^2\right|$, where T and T' are transformations returned by our algorithm and by the ICP algorithm, respectively.

Figure 3.1(b) plots the count of subroutine calls in each algorithm while extending a matching data segment $\mathcal{D}_{1..n-1}$ to $\mathcal{D}_{1..n}$. The subroutine finds the closest point to a transformed data point on the model curve; it is an elementary operation involved in both algorithms. The dashed (red) and solid (blue) lines represent the increases in the counts of subroutine calls by the ICP algorithm and by the proposed algorithm, respectively. The two thin (gray) solid lines show manually selected asymptotes of the two count increases: $y = 2n + 52$ and $y = 2$.

The two graphs imply that the accuracy of the online algorithm approximates that of the ICP algorithm, but the online algorithm ($O(n)$) is more efficient than the ICP algorithm ($O(n^2)$) when gradually extending the longest matching segment from an initially given match of two curves.

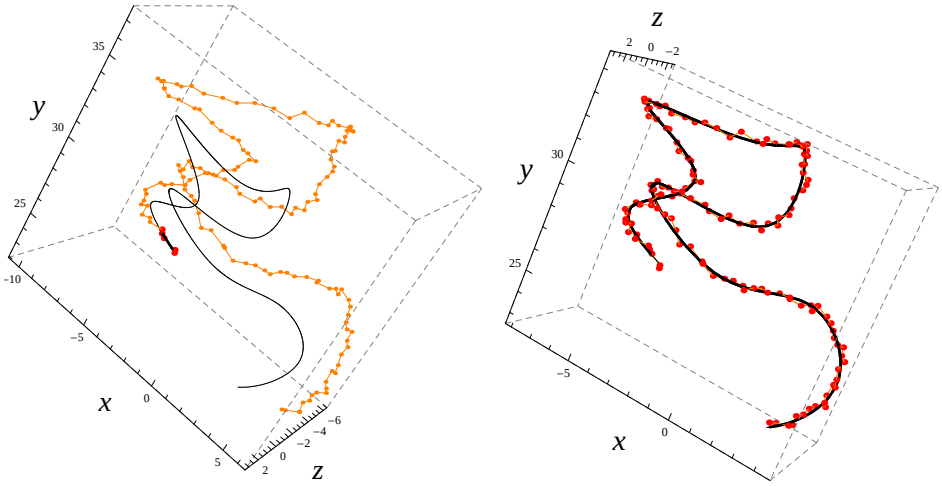


Figure 3.2: Initial configuration of one test case of the comparison to the ICP algorithm in (a), and its matching result in (b).

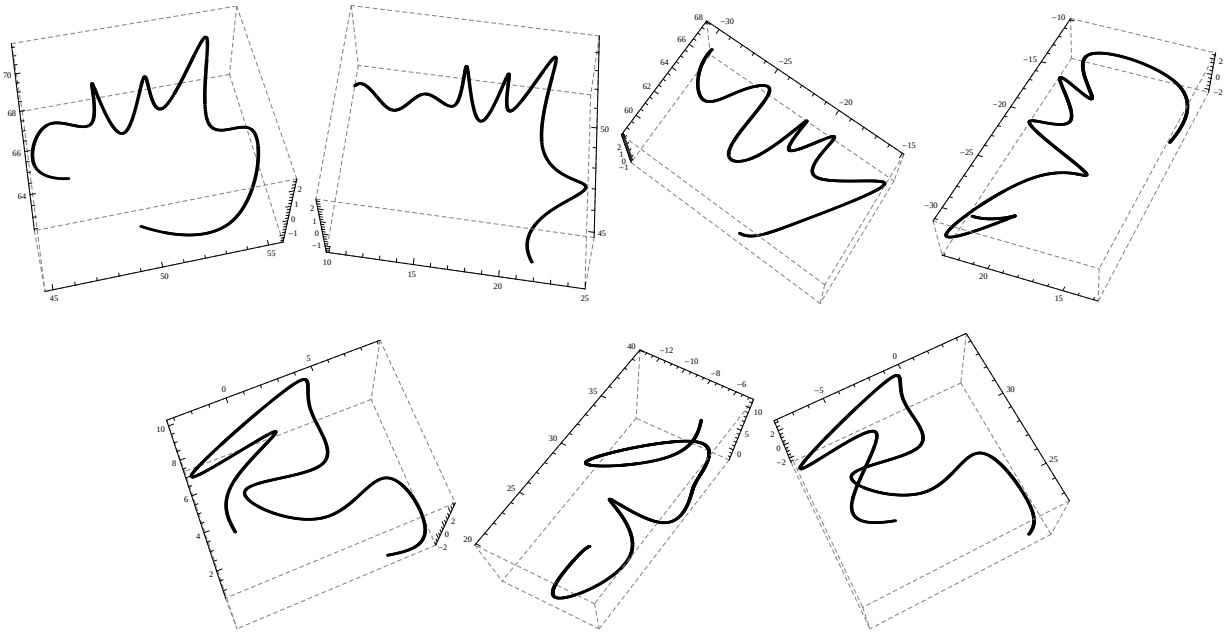
For the experiment, data points are sampled from a B-spline (model curve) under Gaussian noise $N(0, 0.05)$. Since the data curve fully matches the model curve, the initial matching data and model segments are set up as beginning parts of them with slightly different scale ratios, as shown in Figure 3.2(a). In the figure, red points and a bold black line highlight the initial matching data and model segments, respectively. Extension of the matching segment continues to the ends of both curves as shown in Figure 3.2(b), with collecting the accuracy and performance data to make the comparison with our algorithm and the ICP algorithm.

This experiment was performed with 7 different B-splines and 400 randomly generated data curves per B-spline. Figure 3.3 shows 7 B-splines in (a) and one of 400 data curves sets in (b) which are randomly generated from the B-splines. Figure 3.1 is the accumulated result of the performances and accuracies which are obtained from $7 * 400 = 2800$ experiments.

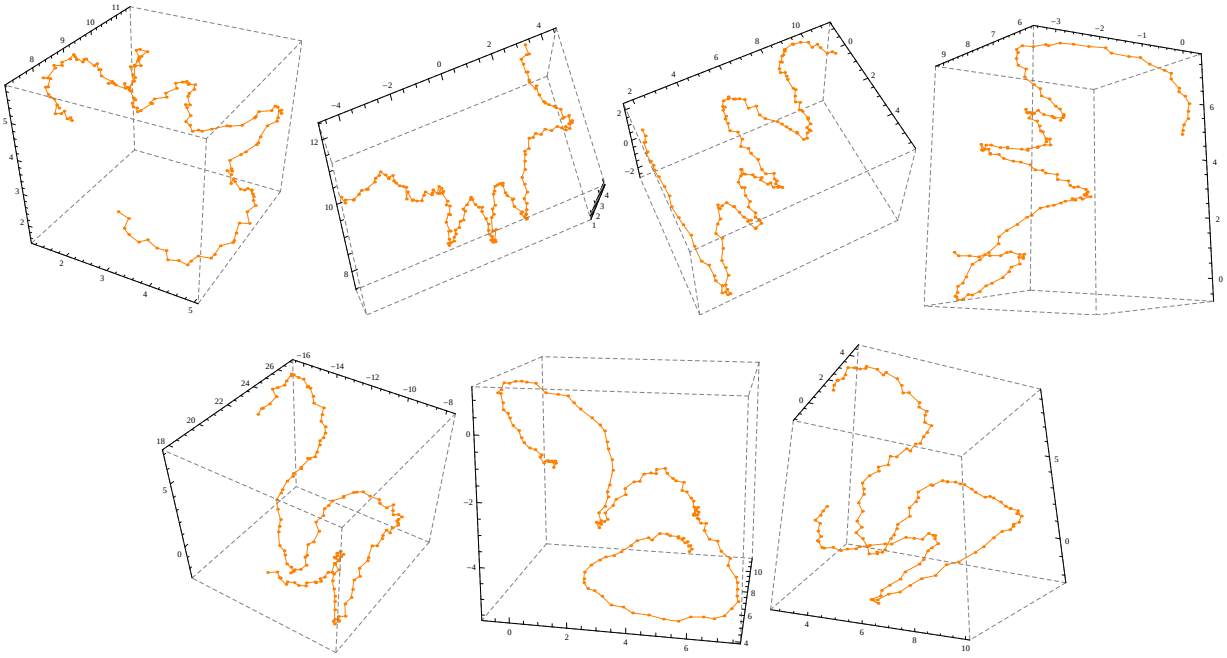
3.2 Initial Matches of Two Arbitrary Curves

Figure 3.4 and 3.5 show the result of finding the longest common segments of two unknown curves. As described in Section 2.1, a list of initial matches is selected using vertices and total curvatures of model and data curves, and the longest common segment is selected among their longest extensions. In Figure 3.4(a), a model curve (gray) $\mathcal{M}$ and a noisy data curve (orange) $\mathcal{D}$ are presented. The model curve is a B-spline with 15 knots. The data curve consists of 134 points sampled from another B-spline, which partially overlaps with the model curve (sharing the same subsequence of knots) in the middle before being scaled by a factor of $0.6 \approx 1/1.6667$. The 10 (and 8) initial matching segments of model (and data) curve are, respectively, in black (and red) in (a). Totally, $\binom{10}{8}$ combinations of initial matching segments are tested, and the configuration that draws the final result (b) is marked by dashed (blue) circle in (a). The found longest overlapping model and data sections are colored in (b). In this experiment, the allowed scale range is set to be $[s_{\min}, s_{\max}] = [0, \infty]$, assuming that the scale range is unknown. The result tells that the algorithm successfully approximates the scale factor 1.6667 as 1.6998. The longest matching data segment in (b) has 42 points while the count of calling the subroutine, finding the closest point, for this test is 81 out of the total count of 3164.

Figure 3.5 displays a case that both model and data curves are obtained from range data. The model curve is a B-spline generated from 425 data points without changing scale, and data curve has 519 points. The number of initial matching segments of the model and data curves are 31 and 16, respectively. The noise variance and allowed scale range are set as $\sigma^2 = 0.0189$ and $[s_{\min}, s_{\max}] = [0, \infty]$, respectively. The initial matching segment of the data curve in (a) is roughly overlapped to that of the model curve to present the initial transformation that draws the final result in (b). The final result shows that the proposed algorithm finds the longest common segment even though it is complex. The estimated output scale is $s = 0.9986$. After $\binom{31}{16}$ combination of tests, the total count of calling the subroutine is 15974. The longest common data segment in Figure 3.5(b) has 307 points while the count of calling the subroutine is 817.



(a)



(b)

Figure 3.3: Data set tested for the comparison with the ICP algorithm. (a) lists 7 model curves which are B-splines, and (b) shows some examples of noisy data points sampled from the model curves.

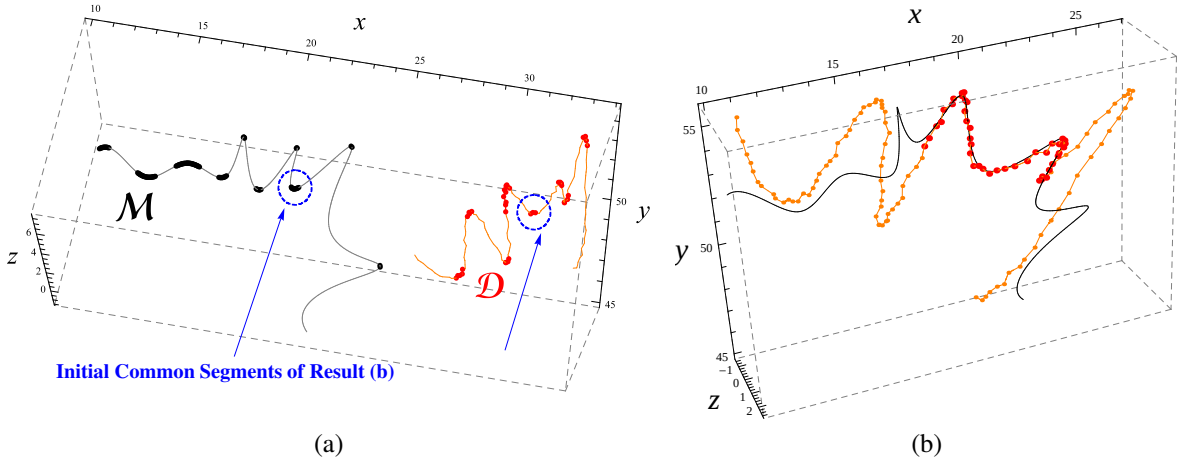


Figure 3.4: Partial match of a data curve $\mathcal{D}$ and a model curve $\mathcal{M}$, all shown in (a) with their initial matching segments. In the found match (b), the data curve is scaled 1.6998 times.

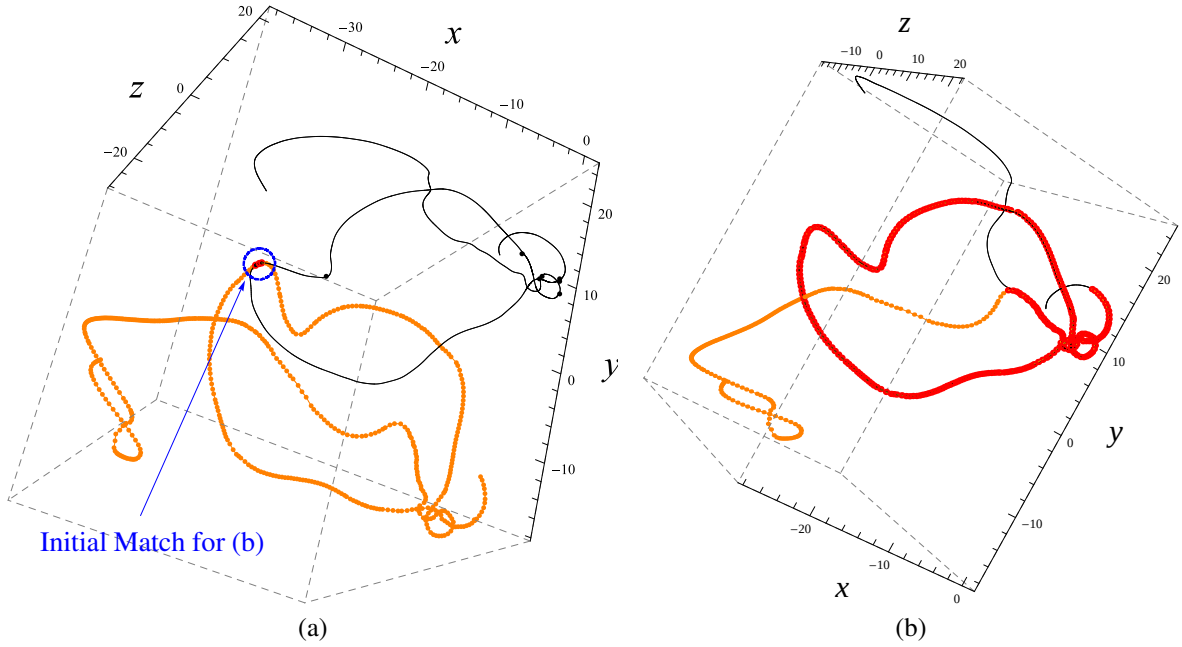
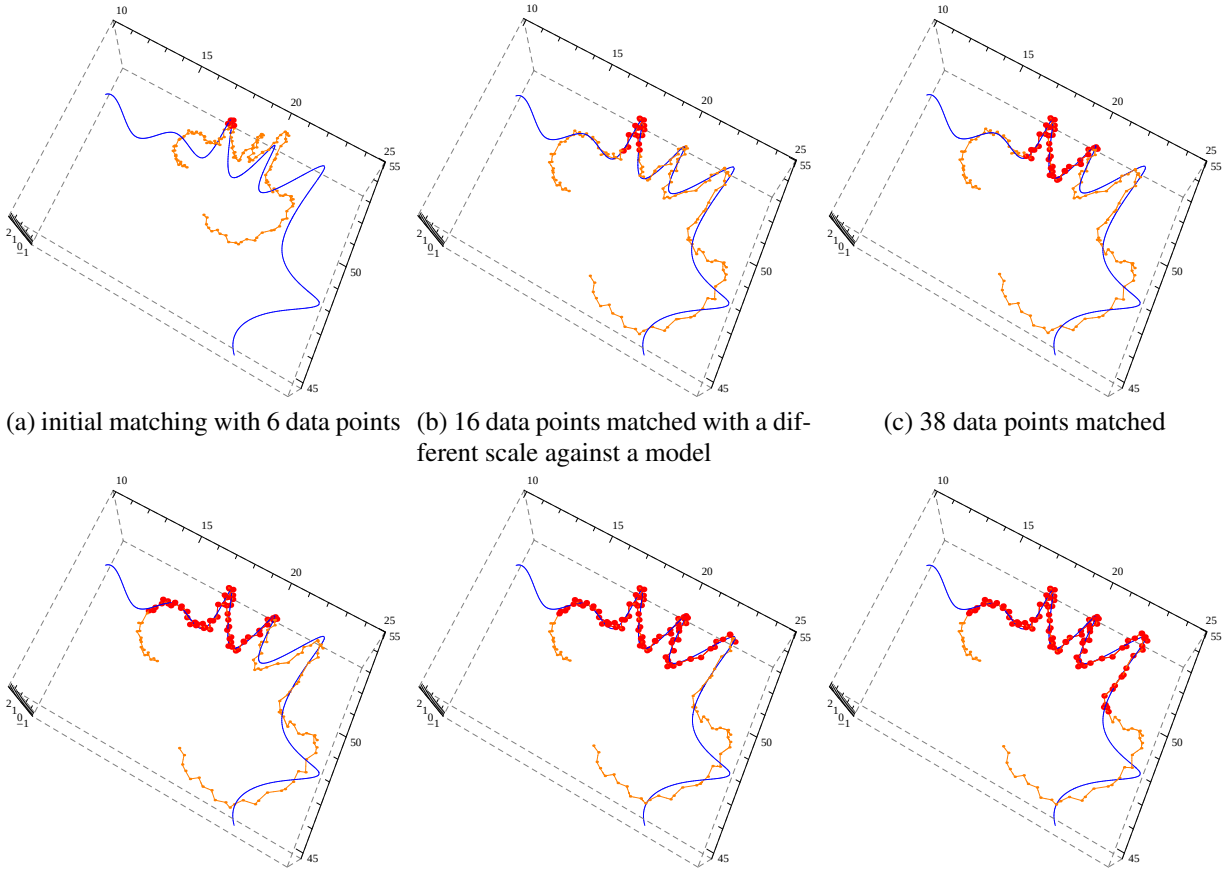


Figure 3.5: Experiment with range data. The initial pose to extend the initial matching segment is presented in (a), and its final matching result is displayed in (b).



(a) initial matching with 6 data points

(b) 16 data points matched with a different scale against a model

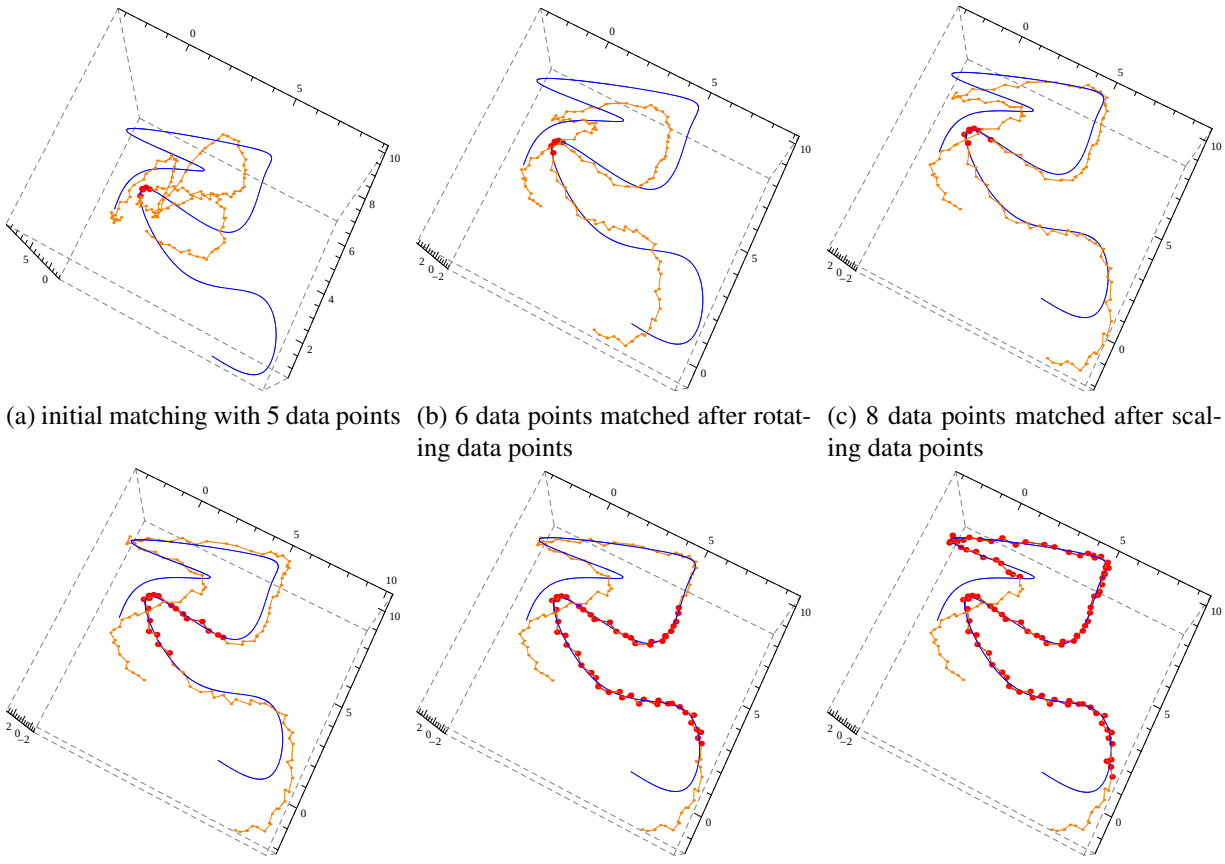
(c) 38 data points matched

(d) 60 data points matched

(e) 85 data points matched

(f) 100 data points in its final match

Figure 3.6: Progressive scaling and matching of a data segment against a model.



(a) initial matching with 5 data points

(b) 6 data points matched after rotating data points

(c) 8 data points matched after scaling data points

(d) 20 data points matched

(e) 53 data points matched

(f) 97 data points in its final match

Figure 3.7: Progressive scaling, rotating, and matching of a data segment against a model

3.3 Progressive Matching

Figure 3.6 and 3.7 show the progress of our algorithm while extending the matching segment. In those figures, the model curve and data curve are displayed as blue and orange lines, and the matching data segment is highlighted as red large points at the data points. In each test, the model curve is selected among 7 B-splines in Figure 3.3(a). The data points are sampled under Gaussian noise $N(0, 0.02)$ from another B-spline, which partially overlaps with the model curve (sharing the same subsequence of knots) in the middle before being scaled by a factor of $0.833 \approx 1/1.2$ and being rotated by axis $(1, 1, 1)$ and angle 0.1 radian. In each test, the allowed scale range is set as $[s_{\min}, s_{\max}] = [0, \infty]$.

In Figure 3.6(a), the initial data curve is shown to be incorrectly scaled. However, Figure 3.6(b)-(f) show that, as iteratively extending the matching segment of the two curves, our algorithm adjusts the scale factor to superpose the matching data segment onto the model curve, without updating the line approximations $\mathcal{L}_{k'}, ..., \mathcal{L}_{l'}$ of the model curve $\mathcal{M}$ for the data point $\boldsymbol{p}_{k'}, ..., \boldsymbol{p}_{l'}$ in the 4th step of the procedure in Section 2.2. Finally, our algorithm approximates the scale factor 0.833 as 0.843, and the rotation axis $(0.577, 0.577, 0.577)$ and angle 6.183 radian as $(0.690, 0.497, 0.526)$ and 6.155 radian, respectively. In the test, the subroutine, finding the closest point, is called 194 times.

Similarly, in Figure 3.7(a), the initial data curve is shown to be incorrectly scaled to and rotated. Figure 3.7(b)-(f) show that our algorithm adjusts the scale and rotation factors to superpose the matching data segment onto the model curve, without updating the line approximations $\mathcal{L}_{k'}, ..., \mathcal{L}_{l'}$ in the 4th step of the procedure in Section 2.2. Our algorithm approximates the scale factor 0.833 as 0.843, and the rotation axis $(0.577, 0.577, 0.577)$ and angle 6.183 radian as $(0.603, 0.517, 0.608)$ and 6.182 radian, respectively. In this test, the subroutine, finding the closest point, is called 189 times.

3.4 Table of Test Results with Range Data

Our algorithm is tested with 67 model curves and 67 data curves, which are obtained from range data. Totally, $\binom{67}{66}$ different combinations of model and data curve pairs are tested except the case that both model and the data curves are obtained from the same range data, to avoid full matches but collect the partial matches. Among the partial matches from $\binom{67}{66}$ combinations of test cases, 15 selections are displayed in Figure 3.8 and 3.9.

For the test, data points are generated by following a line of curvature on a 3D mesh surface from a random point on the surface. The model curve is obtained by fitting a B-spline over the data points. The noise variance for the data points is estimated by comparing the model curve

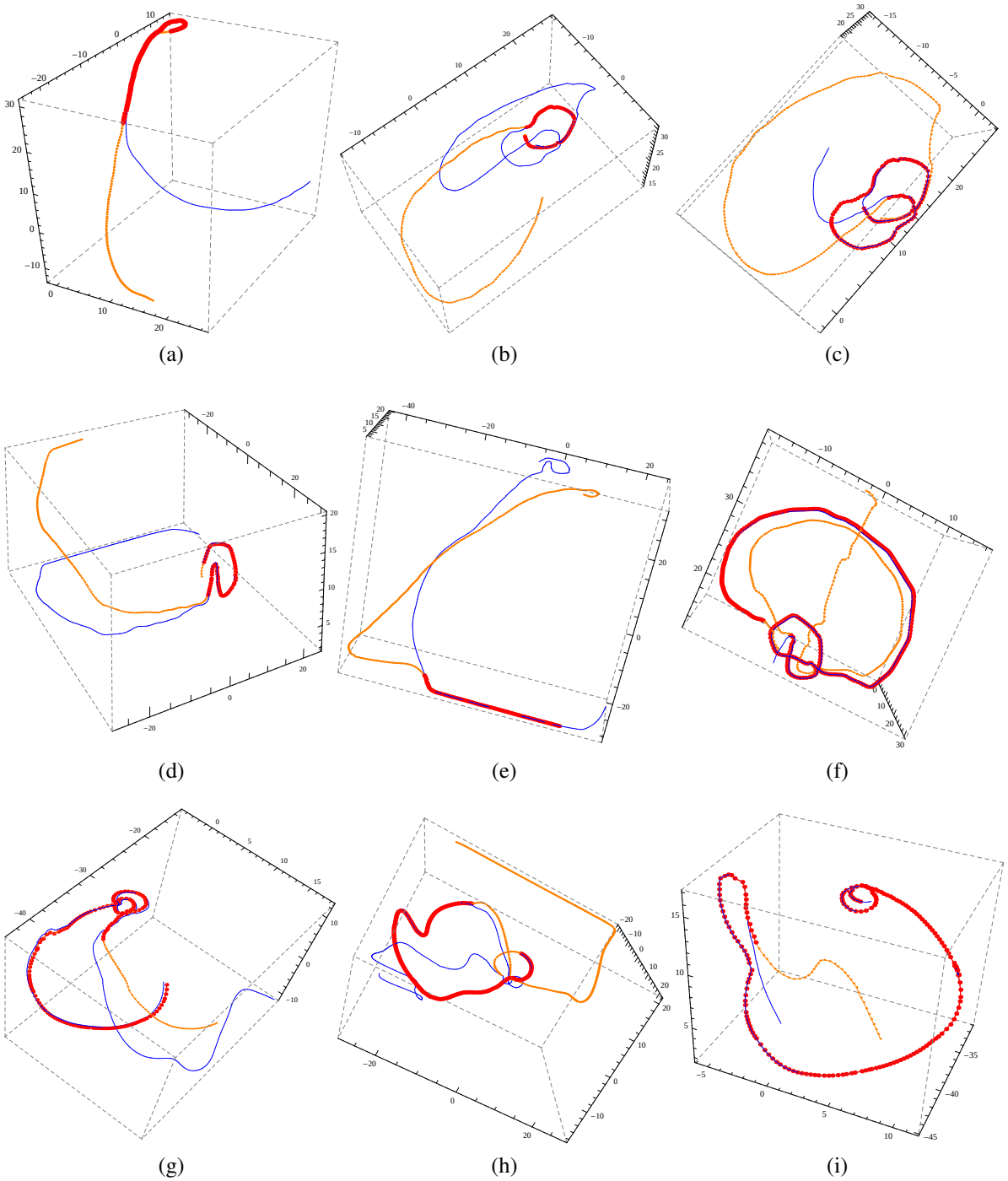
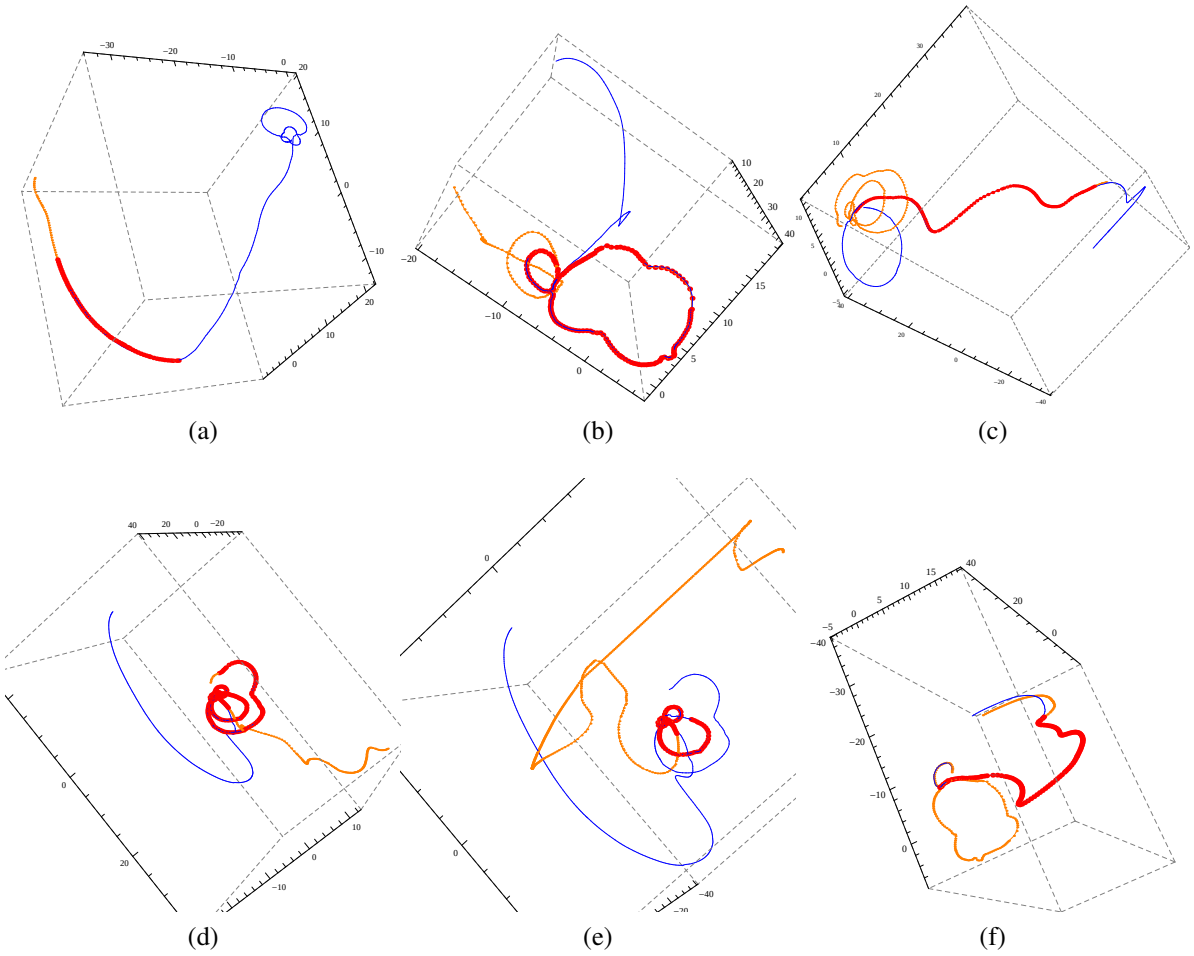


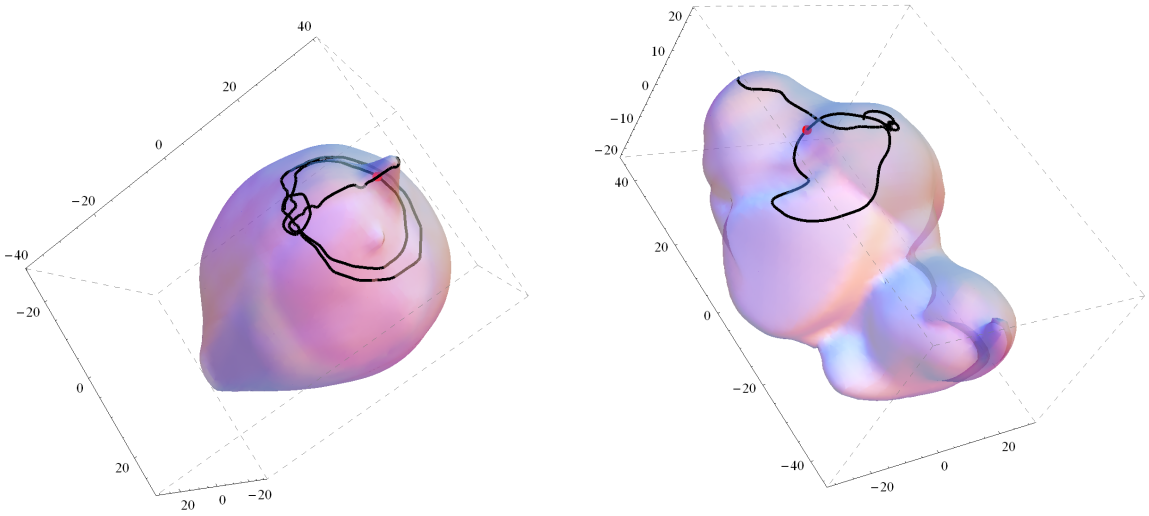
Figure 3.8: Pairs of matching range data segments (part 1)



(a) (b) (c)

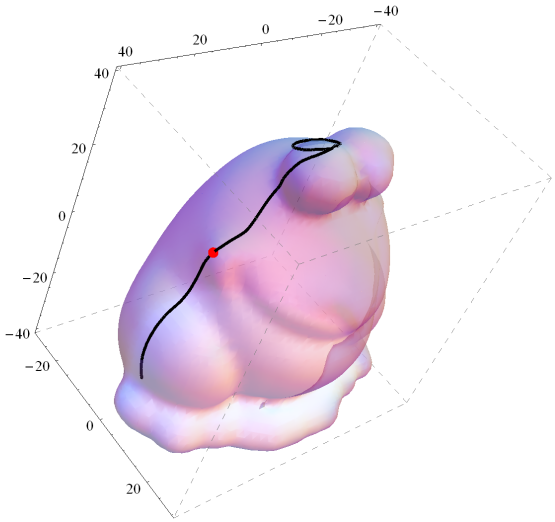
(d) (e) (f)

Figure 3.9: Pairs of matching range data segments (part 2)



(a) bird

(b) dog



(c) frog

Figure 3.10: Three data (black solid) curves superposed onto 3D mesh surfaces of (a) bird, (b) dog, and (c) frog. In each data curve, data points are generated by following a line of curvature on its corresponding mesh surface initially from a random (red) point on the surface. The 67 data curves are generated from the three mesh surfaces displayed in (a), (b), and (c) to obtain partial matches in Figure 3.8 and 3.9.

with these data points. In Figure 3.10, three mesh surfaces are displayed with some random points (red large points) and their corresponding data curves (black solid line). This procedure is repeated to obtain 67 model and data curves with the three mesh surfaces.

CHAPTER 4. CONCLUSION AND FUTURE WORK

Finding the longest matching segment of two curves is a problem with significant applications in computer vision and robotics. In this thesis, we present an algorithm for matching space curve data. The method finds the longest matching segment of two curves by gradually extending an initial matching portion in linear time. Here one (model) curve is any of type, and the other (data) curve is a polyline that joins a sequence of data points. We show that the statistics of the distance between a data point (on the data curve) and the model curve can be approximated to that of the distance between the point and a tangent line on the model curve. To achieve the online algorithm, we reformulated the matching error between the model and data segments as a recurrence. In a similar way, we determines the optimal transformation to superpose the model and data segments using recurrences and constant repetition in average. In the method, extending the matching segment is continued as long as its matching error is below some statistically defined tolerance.

The algorithm is compared with the well-known iterative closest point (ICP) algorithm. The experimental result shows that the accuracy of our algorithm is comparable to that of the ICP algorithm, but our algorithm is less expensive, in the problem of iteratively extending the matching segment of two space curves.

We also present a naive method that finds candidates of the initial matching segment of two space curves without any prior knowledge of them, to find the longest matching segment automatically. In the method, the initial matching segment of a curve is set up using vertices and total curvatures of the curve. The experimental result shows that our algorithm finds the longest matching segment of two curves of complex shapes and it adjusts well to noise distribution. Additionally, it shows that our algorithm works well even though the prior knowledge of scaling boundaries is unknown.

The proposed algorithm is particularly appealing when the character of the data is known in advance, such as the noise distribution of data, the prior knowledge of scale boundaries, and the salient region to set up the initial matching segment. Many optical and sensing devices, like 3-D scanners, have specified accuracies. This method is applicable partially in geometry-centered manipulations, such as robotic surgery and automated assembling of 3D objects, and

the control of capture device like a data glove.

One potential extension of this thesis is an algorithm that matches two surfaces. This is because equations for the surface matching problem can be formulated in the similar way to that for the proposed algorithm: the squared distance $d(\mathcal{L}_i, T\boldsymbol{p}_i)^2 = (\boldsymbol{q}_i - T\boldsymbol{p}_i)^t(I - \boldsymbol{a}_i\boldsymbol{a}_i^t)(\boldsymbol{q}_i - T\boldsymbol{p}_i)$ between a point $\boldsymbol{p}$ and a tangent line $\mathcal{L}_i : \boldsymbol{a}_i t + \boldsymbol{q}_i$ of a space curve can match the squared distance $d(\mathcal{H}_i, T\boldsymbol{p}_i)^2 = (\boldsymbol{q}_i - T\boldsymbol{p}_i)^t(\boldsymbol{n}_i^t\boldsymbol{n}_i)(\boldsymbol{q}_i - T\boldsymbol{p}_i)$ between the point $\boldsymbol{p}$ and a tangent plane $\mathcal{H}_i : \boldsymbol{n}_i \times t + \boldsymbol{q}_i$ of a surface. In the case, it would be a challenge to find a way to extend the boundary of the matching surface.

APPENDIX A. EXTENDED MATCHING SEGMENT

Equation (2.22) in Section 2.4 estimates the increase of the model segment length ζ under scale s. Here we provide how the equation is derived.

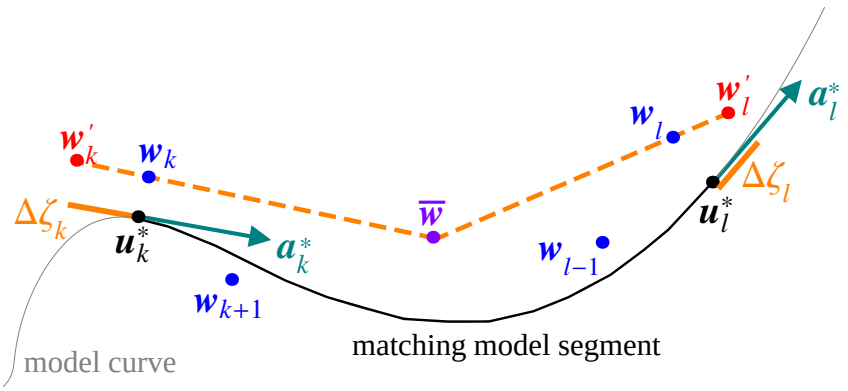


Figure A.1: The increase $\Delta\zeta_k$ (and $\Delta\zeta_l$) of the model segment length at its starting point $\boldsymbol{u}^*_k$ (and ending point $\boldsymbol{u}^*_l$), which corresponds to the starting point $\boldsymbol{w}_k$ (and $\boldsymbol{w}_l$) of the transformed data segment $\mathcal{D}_{k..l}$. In the figure, $\boldsymbol{a}^*_k$ (and $\boldsymbol{a}^*_l$) is the unit tangent to the model curve at $\boldsymbol{u}^*_k$ (and $\boldsymbol{u}^*_l$), $\bar{\boldsymbol{w}}$ is the centroid of the points set $\{\boldsymbol{w}_k, ..., \boldsymbol{w}_l\}$, and $\boldsymbol{w}'_k$ (and $\boldsymbol{w}'_l$) is the new position of $\boldsymbol{w}_k$ (and $\boldsymbol{w}_l$) scaled by a factor s from the centroid $\bar{\boldsymbol{w}}$.

Recall that $\boldsymbol{w}_k, ..., \boldsymbol{w}_l$ are the transformed data points $T\boldsymbol{p}_k, ..., T\boldsymbol{p}_l$ of the data segment $\mathcal{D}_{k..l} = (\boldsymbol{p}_k, \boldsymbol{p}_{k+1}, ..., \boldsymbol{p}_l)$ which is superposed onto the model curve under the transformation T, $\boldsymbol{u}^*_i$ is the point on the model curve $\mathcal{M}$ closest to $\boldsymbol{w}_i$, and $\boldsymbol{a}^*_i$ is the unit tangent vector to $\mathcal{M}$ at $\boldsymbol{u}^*_i$, where $i \in \{k, l\}$. In this configuration, the length ζ of a matching model segment is the arc length between $\boldsymbol{u}^*_k$ and $\boldsymbol{u}^*_l$ in the model curve. Figure A.1 illustrates the configuration and the approximated increase $\Delta\zeta = \Delta\zeta_k + \Delta\zeta_l$ of the matching model segment length when the transformed data segment $(\boldsymbol{w}_k, ..., \boldsymbol{w}_l)$ is scaled. In the figure, $\bar{\boldsymbol{w}}$ is the centroid of points $\{\boldsymbol{w}_k, ..., \boldsymbol{w}_l\}$, $\boldsymbol{w}'_k$ (and $\boldsymbol{w}'_l$) is the new position of $\boldsymbol{w}_k$ (and $\boldsymbol{w}_l$) scaled by the factor of s from the centroid $\bar{\boldsymbol{w}}$, and $\Delta\zeta_k$ (and $\Delta\zeta_l$) is the approximated increase of ζ at the point $\boldsymbol{u}^*_k$ (and $\boldsymbol{u}^*_l$) by the scale factor s. Here the approximation $\Delta\zeta_l$ is determined by the projection of $(\boldsymbol{w}'_l - \boldsymbol{w}_l)$ onto $\boldsymbol{a}^*_l$ as follows:

$$\begin{aligned} \Delta\zeta_l &= ((\boldsymbol{w}'_l - \boldsymbol{w}_l) \cdot \boldsymbol{a}^*_l) \\ &= (s-1)\,((\boldsymbol{w}_l - \bar{\boldsymbol{w}}) \cdot \boldsymbol{a}^*_l) \end{aligned} \tag{A.1}$$

In a similar way, $\Delta\zeta_k$ is determined. In sum, the approximated increase $\Delta\zeta$ of the matching model segment length is defined as follows:

$$\begin{aligned} \Delta\zeta &= \Delta\zeta_k + \Delta\zeta_l \\ &= (1-s)\Big(((\bar{\boldsymbol{w}} - \boldsymbol{w}_k)\cdot \boldsymbol{a}_k^*) - ((\bar{\boldsymbol{w}} - \boldsymbol{w}_l)\cdot \boldsymbol{a}_l^*)\Big). \end{aligned} \tag{A.2}$$

APPENDIX B. OPTIMAL POSE AND SCALE

Equation (2.28) in Section 2.4 is decomposed to (2.29) and (2.30), which determines the rotation $\hat{R}$, scale $\hat{s}$, and translation $\hat{\boldsymbol{b}}$ that minimizes the pointwise superposition error of two sets $W = \{\boldsymbol{w}_k, ..., \boldsymbol{w}_l\}$ and $U = \{\boldsymbol{u}_k, ..., \boldsymbol{u}_l\}$. Here we provide how the equation is decomposed. First, recall the equations:

$$\langle \hat{s}, \hat{R}, \hat{\boldsymbol{b}} \rangle = \underset{s,R,\boldsymbol{b}}{\operatorname{argmin}} \sum_{i=k}^{l} \left\| R(\boldsymbol{w}_i - \boldsymbol{\xi}_i) + \frac{\boldsymbol{b}}{s} - \frac{\boldsymbol{u}_i}{s} \right\|^2; \tag{B.1}$$

$$\langle \hat{s}, \hat{R} \rangle = \underset{s,R}{\operatorname{argmin}} \sum_{i=k}^{l} E\left[\left\| R(\boldsymbol{w}'_i - \boldsymbol{\xi}_i) - \frac{\boldsymbol{u}'_i}{s} \right\|^2 \right]; \tag{B.2}$$

$$\hat{\boldsymbol{b}} = \bar{\boldsymbol{u}} - \hat{s}\hat{R}\bar{\boldsymbol{w}}, \tag{B.3}$$

where $\bar{\boldsymbol{w}}$ and $\bar{\boldsymbol{u}}$ are the centroids of W and U, respectively, $\boldsymbol{\xi}_i$ is noise, $\boldsymbol{w}'_i = \boldsymbol{w}_i - \bar{\boldsymbol{w}}$, and $\boldsymbol{u}'_i = \boldsymbol{u}_i - \bar{\boldsymbol{u}}$, $k \le i \le l$. Introduce a new variable $\boldsymbol{h} = \boldsymbol{b}/s$, and redefine the term $\langle \hat{s}, \hat{R}, \hat{\boldsymbol{h}} \rangle$ that minimizes the summation in (B.1):

$$\langle \hat{s}, \hat{R}, \hat{\boldsymbol{h}} \rangle = \underset{s,R,h}{\operatorname{argmin}} \sum_{i=k}^{l} E\left[\left\| R(\boldsymbol{w}_i - \boldsymbol{\xi}_i) + \boldsymbol{h} - \frac{\boldsymbol{u}_i}{s} \right\|^2 \right]. \tag{B.4}$$

The summation in (B.4) is quadratic in $\boldsymbol{h}$. If we know the values $\hat{s}$ and $\hat{R}$, then $\hat{\boldsymbol{h}}$ is determined using the partial derivative of the summation with respect to $\boldsymbol{h}$:

$$\begin{aligned} \hat{\boldsymbol{h}} &= \text{root } \boldsymbol{h} \text{ of } \left[\frac{\partial}{\partial \boldsymbol{h}} \sum_{i=k}^{l} E\left[\left\| \hat{R}(\boldsymbol{w}_i - \boldsymbol{\xi}_i) + \boldsymbol{h} - \frac{\boldsymbol{u}_i}{\hat{s}} \right\|^2 \right] \right] \\ &= \frac{\bar{\boldsymbol{u}}}{\hat{s}} - \hat{R}\bar{\boldsymbol{w}}. \end{aligned}$$

In the above equation, the noise term $\boldsymbol{\xi}_i$ has been eliminated because $E[\boldsymbol{\xi}_i^t \cdot R^t R \cdot \boldsymbol{\xi}_i] = 3/\sigma^2$ and $E[\boldsymbol{\xi}_i \cdot \boldsymbol{c}] = 0$ for any rotation matrix R and constant vector $\boldsymbol{c}$.

Getting back to the original problem, we expand the summation in (B.4) as

$$\sum_{i=k}^{l} E\left[\left\| R(\boldsymbol{w}'_i - \boldsymbol{\xi}_i) - \frac{\boldsymbol{u}'_i}{s} \right\|^2 + \left\| R\bar{\boldsymbol{w}} + \boldsymbol{h} - \frac{\bar{\boldsymbol{u}}}{s} \right\|^2 + 2\left(R(\boldsymbol{w}'_i - \boldsymbol{\xi}_i) - \frac{\boldsymbol{u}'_i}{s} \right)^t \left(R\bar{\boldsymbol{w}} + \boldsymbol{h} - \frac{\bar{\boldsymbol{u}}}{s} \right) \right].$$

The second and third summands of the summation become zero with $\hat{s}$, $\hat{R}$, and $\hat{\boldsymbol{h}}$ because $\boldsymbol{w}_k, ..., \boldsymbol{w}_l$ will be rotated and scaled at their centroid $\bar{\boldsymbol{w}}$. Therefore, the original problem (B.4) is decomposed as follows:

$$\begin{aligned} \langle \hat{s}, \hat{R} \rangle &= \underset{s,R}{\operatorname{argmin}} \sum_{i=k}^{l} \left\| R(\boldsymbol{w}'_i - \boldsymbol{\xi}_i) - \frac{\boldsymbol{u}'_i}{s} \right\|^2, \\ \hat{\boldsymbol{b}} &= \hat{s}\,\hat{\boldsymbol{h}} = \bar{\boldsymbol{u}} - \hat{s}\hat{R}\bar{\boldsymbol{w}}. \end{aligned}$$

APPENDIX C. ITERATIVE CLOSEST POINT ALGORITHM

In Section 3.1, the proposed algorithm was compared with with the well-known iterative closest point (ICP) algorithm [3]. Here we provide the detail of the ICP algorithm.

The ICP algorithm finds the optimal rigid body transformation T (rotation and translation) superposing a shape $P = \{\boldsymbol{p}_1, \boldsymbol{p}_2, ..., \boldsymbol{p}_n\}$ composed of the set of points onto another shape X which can be a combination of line segments, surface patches and points. It optimizes the transformation T from its initial one T_0 by iteratively minimizing the sum of squared distances from points $\boldsymbol{p}_1, \boldsymbol{p}_2, ..., \boldsymbol{p}_n$ to the model X, as the following procedure:

1. Initiate T as T_0.
2. Construct the set of points $\{\boldsymbol{x}_i \mid \boldsymbol{x}_i \in X \text{ closest to } T\boldsymbol{p}_i, 1 \leq i \leq n\}$.
3. Estimate a transformation T' minimizing the error $\sum_{i=1}^{n} \left\|\boldsymbol{x}_i - T'\boldsymbol{p}_i\right\|^2$, and update the transformation T using T',
4. Repeat from the second step until the error converges.

In the third step, the rotation R^* and translation $\boldsymbol{t}^*$ of the transformation T' are determined by the steps of computation, which is the subset of (2.16), as follows:

$$\begin{aligned} R^* &= \underset{R}{\operatorname{argmax}} \sum_{i=1}^{n} R(T\boldsymbol{p}_i - T\bar{\boldsymbol{p}}) \cdot (\boldsymbol{x}_i - \bar{\boldsymbol{x}}); \\ \boldsymbol{t}^* &= \bar{\boldsymbol{x}} - R^* T\bar{\boldsymbol{p}}, \end{aligned} \tag{C.1}$$

where $\bar{\boldsymbol{p}}$ and $\bar{\boldsymbol{x}}$ are the centroids of point sets $\{\boldsymbol{p}_1, ..., \boldsymbol{p}_n\}$ and $\{\boldsymbol{x}_1, ..., \boldsymbol{x}_n\}$, respectively. For the proper comparison of the accuracy and performance differences between the proposed and ICP algorithms in Section 3.1, we used (2.16) instead of (C.1) in step three in order to find the optimal rotation R^*, translation $\boldsymbol{t}^*$, and scale s^* of the transformation T'.

BIBLIOGRAPHY

[1] Gady Agam and Suneel Suresh. Warping-based offline signature recognition. *IEEE Transactions on Information Forensics and Security*, 2(3-1):430–437, 2007.

[2] Y.S. Avrithis, Y. Xirouhakis, and S.D. Kollias. Affine-invariant curve normalization for object shape representation, classification, and retrieval. 13(2 2001):80–94, 2001.

[3] P.J. Besl and N.D. McKay. A method for registration of 3-D shapes. *IEEE Trans. Pattern Anal. and Mach. Intell.*, 14(2):239–256, February 1992.

[4] Tolga Can and Yuan-Fang Wang. Ctss: A robust and efficient method for protein structure alignment based on local geometrical and biological features. In *Proceedings of the IEEE Computer Society Conference on Bioinformatics*, CSB '03, pages 169–, Washington, DC, USA, 2003. IEEE Computer Society.

[5] Yang Chen and Gérard Medioni. Object modelling by registration of multiple range images. *Image Vision Computing*, 10:145–155, April 1992.

[6] Fernand S. Cohen, Zhaohui Huang, and Zhengwei Yang. Invariant matching and identification of curves using b-splines curve representation. *IEEE Transactions on Image Processing*, 4:1–10, 1995.

[7] Isaac Cohen and Isabelle Herlin. Curves matching using geodesic paths. In *In IEEE Proceedings of Computer Vision and Pattern Recognition*, pages 741–746, 1998.

[8] Isaac Cohen and Isabelle Herlin. Tracking meteorological structures through curves matching using geodesic paths. In *In Proceedings of ICCV*, pages 396–401, 1998.

[9] O D Faugeras and M Hebert. The representation, recognition, and locating of 3-D objects. *Int. J. Rob. Res.*, 5(3):27–52, 1986.

[10] J. Feldmar and N. Ayache. Rigid, affine and locally affine registration of free-from surfaces. *IJCV*, 18(18):99–119, 1994.

[11] Armin Gruen and Devrim Akca. Least squares 3d surface and curve matching. *ISPRS Journal of Photogrammetry and Remote Sensing*, 59:151–174, 2005.

[12] Douglas R. Heisterkamp and Prabir Bhattacharya. Matching of 3-D curves. In *ICRA96*, volume 4, pages 3490–3495, Apr 1996.

[13] Berthold K. P. Horn, Hugh M. Hilden, and Shahriar Negahdaripourt. Closed-form solution of absolute orientation using unit quaternions. *Journal of the Optical Society of America A*, 4:629–642, 1987.

[14] Zhaohui Huang and Fernand S. Cohen. Affine-invariant b-spline moments for curve matching. *IEEE Transactions on Image Processing*, 5:1473–1480, 1996.

[15] I. T. Jolliffe. *Principal Component Analysis*. Springer, 2002.

[16] B. Kamgar-Parsi and B. Kamgar-Parsi. Matching 3-D arcs. *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, page 28, 1997.

[17] Eyal Kishon, Trevor Hastie, and Haim J. Wolfson. 3-D curve matching using splines. In *ECCV*, pages 589–591, 1990.

[18] Weixin Kong and Benjamin. B. Kimia. On solving 2d and 3d puzzles using curve matching. *cvpr*, 2:583, 2001.

[19] B. Krebs, B. Korn, and F. Wahl. 3d b-spline curve matching for model based object recognition. In *ICIP97*, pages II: 716–719, 1997.

[20] D. J. Kriegman and Jean Ponce. On recognizing and positioning curved 3-d objects from image contours. *IEEE Trans. Pattern Anal. Mach. Intell.*, 12:1127–1137, December 1990.

[21] Stan Z. Li. Invariant representation, matching and pose estimation of 3d space curves under similarity transformations. *Pattern Recognition*, 30:447–458, 1997.

[22] X. Liu and Y. Jia. Character stroke extraction based on b-spline curve matching by constrained alternating optimization. In *ICDAR '07: Proceedings of the Ninth International Conference on Document Analysis and Recognition (ICDAR 2007) Vol 1*, pages 13–17, Washington, DC, USA, 2007. IEEE Computer Society.

[23] Ezio Malis, Graziano Chesi, and Roberto Cipolla. 2-1/2d visual servoing with respect to planar contours having complex and unknown shapes. *International Journal of Robotic Research*, 22:841–854, 2003.

[24] Jonah C. McBride and Benjamin B. Kimia. Archaeological fragment reconstruction using curve-matching. *Computer Vision and Pattern Recognition Workshop*, 1:3, 2003.

[25] Raffaele De Amicis Minh-Son Dao. A fusion method of geometric and topological features for boundary-based shape matching and retrieval. pages 217–220, Oct 2006.

[26] Raffaele De Amicis Minh-Son Dao. A new method for boundary-based shape matching and retrieval. In *ICIP06*, pages 1485–1488, Oct 2006.

[27] Farzin Mokhtarian and Alan Mackworth. Scale-based description and recognition of planar curves and two-dimensional shapes. *IEEE Trans. Pattern Anal. Mach. Intell.*, 8(1):34–43, 1986.

[28] Sang mook Lee, A. Lynn Abbott, Neil A. Clark, and Philip A. Araman. Spline curve matching with sparse knot sets: applications to deformable shape detection and recognition. In *29th Annual Conference of the IEEE Industrial Electronics Society*, pages 1808–1813, 2003.

[29] H. Murase and S.K. Nayar. Parametric eigenspace representation for visual learning and recognition. In *Proceedings of The International Society for Optical Engineering (SPIE)*, volume 2031, pages 378–391, 1993.

[30] T. Pajdla and L. Van Gool. Matching of 3-D curves using semi-differential invariants. *iccv*, 00:390, 1995.

[31] Euripides G. M. Petrakis, Aristeidis Diplaros, and Evangelos Milios. Matching and retrieval of distorted and occluded shapes using dynamic programming. *IEEE Trans. Pattern Anal. and Mach. Intell.*, 24(11):1501–1516, 2002.

[32] Arie Pikaz and Its'hak Dinstein. Matching of partially occluded planar curves. *Pattern Recognition*, 28(2):199–209, 1995.

[33] Jacob T. Schwartz and Micha Sharir. Identification of partially obscured objects in two and three dimensions by matching noisy characteristic. *Int. J. Rob. Res.*, 6(2):29–44, 1987.

[34] Thomas B. Sebastian, Philip N. Klein, and Benjamin B. Kimia. On aligning curves. *IEEE Trans. Pattern Anal. Mach. Intell.*, 25(1):116–125, 2003.

[35] Ying Shan and Zhengyou Zhang. Corner guided curve matching and its application to scene reconstruction. In *CVPR*, pages 1796–1803, 2000.

[36] Richard Szeliski. Matching 3-d anatomical surfaces with non-rigid deformations using octree-splines. *International Journal of Computer Vision*, 18:171–186, 1996.

[37] Yue Wang and Eam Khwang Teoh. 2d affine-invariant contour matching using b-spline model. *IEEE Trans. Pattern Anal. and Mach. Intell.*, 29(10):1853–1858, 2007.

[38] H. J. Wolfson. On curve matching. *IEEE Trans. Pattern Anal. Mach. Intell.*, 12(5):483–489, 1990.

[39] Minghui Xia and Bede Liu. Image registration by 'super-curves'. *IEEE Trans. Image Processing*, 13(5):720–732, 2004.

[40] Dong Xu and Wenli Xu. Description and recognition of object contours using arc length and tangent orientation. *Pattern Recogn. Lett.*, 26(7):855–864, 2005.

[41] Shihui Ying, Jigen Peng, Shaoyi Du, and Hong Qiao. A scale stretch method based on ICP for 3D data registration. *Automation Science and Engineering*, 6(3):559–565, July 2009.

[42] Liangjia Zhu, Zongtan Zhou, and Dewen Hu. Globally consistent reconstruction of ripped-up documents. *IEEE Trans. Pattern Anal. and Mach. Intell.*, 30(1):1–13, 2008.