University of Alberta

Geometric Approach to Multi-scale 3D Gesture Comparison.

by

Víctor Manuel Ochoa-Mayorga

A thesis submitted to the Faculty of Graduate Studies and Research
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

Department of Computing Science

Fall 2010
Edmonton, Alberta

Examining Committee

Pierre Boulanger, Ph. D., P. Eng.,
Professor.
Department of Computing Science,
University of Alberta.

Roy Eagleson, Ph. D., P. Eng.,
Associate Professor.
Department of Electrical & Computer Engineering,
University of Western Ontario.

Brian Maraj, Ph. D., Professor.
Department of Physical Education & Recreation,
University of Alberta.

Dale Schuurmans, Ph. D., Professor.
Department of Computing Science,
University of Alberta.

Herb Yang, Ph. D., Professor.
Department of Computing Science,
University of Alberta.

In loving memory to my father, Maximino Ochoa Rivas.
To my mother, Alicia Mayorga Varela.
To my siblings, Sergio and Karla.
To my nephews.

Abstract

The present dissertation develops an invariant framework for $3D$ gesture comparison studies. $3D$ gesture comparison without Lagrangian models is challenging not only because of the lack of prediction provided by physics, but also because of a dual geometry representation, spatial dimensionality and non-linearity associated to $3D$-kinematics.

In $3D$ spaces, it is difficult to compare curves without an alignment operator since it is likely that discrete curves are not synchronized and do not share a common points in space. One has to assume that each and every single trajectory in the space is unique. The common answer is to assert the similitude between two or more trajectories as estimating an average distance error from the aligned curves, provided that the alignment operator is found.

In order to avoid the alignment problem, the method uses differential geometry for position and orientation curves. Differential geometry not only reduces the spatial dimensionality but also achieves view invariance. However, the nonlinear signatures may be unbounded or singular. Yet, it is shown that pattern recognition between intrinsic signatures using correlations is robust for position and orientation alike.

A new mapping for orientation sequences is introduced in order to treat quaternion and Euclidean intrinsic signatures alike. The new mapping projects a $4D$-hyper-sphere for orientations onto a $3D$-Euclidean volume. The projection uses the quaternion invariant distance to map rotation sequences into $3D$-Euclidean curves. However, quaternion spaces are sectional discrete spaces. The significance is that continuous rotation functions can be only approximated for small angles. Rotation sequences with large angle variations can only be interpolated in discrete sections.

The current dissertation introduces two multi-scale approaches that improve numerical stability and bound the signal energy content of the intrinsic signatures. The first is a multilevel least squares curve fitting method similar to Haar wavelet. The second is a geodesic distance anisotropic kernel filter.

The methodology testing is carried out on $3D$-gestures for obstetrics training. The study quantitatively assess the process of skill acquisition and transfer of manipulating obstetric forceps gestures. The results show that the multiscale correlations with intrinsic signatures track and evaluate gesture differences between experts and trainees.

Acknowledgements

My heartfelt admiration and gratitude to Dr. Pierre Boulanger, to whom now I can call my friend. His commitment to science and humanitarianism has given me the inspiration to be true to myself and to never give up. He made his support available in a number of ways that taught me how to be a human being first and then, a scientist.

I am thankfully indebted to Dr. Mihn-Tu Pham from INSA Lyon. His smart and yet, troublesome questions took me out of my self-complacency. My gratitude to Dr. Richard Moreau, "Monsieur Richard" then, for his friendship.

This little self-achievement would not have come to fruition without the love and the support from Idanis Diaz during these last years.

My earnest gratitude to Lynn Schlecker, a true friend in this once, foreign land. She gave me a chance to find myself in a moment in which I was lost.

To my surrogate family Connie Molina-Albiter, Dr. Horacio Gómez-Acevedo and his wife Sheila, and Martha Benítez-Angulo.

My gratitude to the Computing Science Department Staff at the University of Alberta, but specially to Edith Drummond for her support. Now, she can put my troublesome student ashes in a shelf urn. Thank you Edith.

To my friends at the Advanced Man-Machine Interface Laboratory (AMMI Lab), specially Matthew Hamilton and Maryia Kazakevich.

And lastly, my gratitude to Mexico's National Council for Science and Technology (CONACYT *Consejo Nacional de Ciencia y Tecnología*) for their support during the first years of my program.

Contents

List of Figures

Chapter 1

Literature Review

The present dissertation is concerned with the quantitative comparison of $3D$, non-periodic sequences of rigid motion in free- and articulated-bodies. Very often, the term "gesture" appears in the literature to describe sequences of rigid motion, specially in the case of anthropomorphic or mechanisms with multiple degrees-of-freedom. However, "gesture" acquires different meanings depending on the research area [24]. With the lack of a better term to describe different aspects of motion sequences, Bobick [24] broadened out the meaning of "gesture" to include manipulative aspects instead of only considering communicative ones. The main consideration is that often, artificial systems lack information about the context. The context is absolutely necessary to infer the meaning of a motion sequence in an information exchange. And although sequences of motion for tool manipulation do not have a communicative context, they certainly have a well defined purpose. Under such considerations and hereafter, one embraces the term "gesture" to consider sequences of motion with manipulative purposes.

In this chapter, one reviews the current state-of-the-art and categorizes various systems for non-periodic motion sequence analysis and recognition for purpose of sequence comparison. The topic of recognition and quantification of articulated motion is broad. Several areas such as biomedical engineering, computer vision, robotics, and data analysis made important contributions; some with very compelling results. The purpose is to focus on what are the contributing factors for an accurate and efficient $3D$ motion sequence comparison. We review various aspects such as: purpose, input type, feature space, internal model type and structure, spatial dimensionality, and technical implementation. Different categorizations have different merits, however the scope and main interest lie around two important aspects: non-stationarity and system complexity for discrimination.

The non-stationarity of a motion recognition system refers to the ability for the system to recognize non-periodic sequences. An underlying aspect of non-stationarity is the time variance in the sequences. Similar sequences may follow the same spatial trajectory but completely different timing. Should they be classified within the same or a different class base on their dynamics? Is the trajectory more important? What is the noise in the trajectory? All these aspects are not easily discernible in non-periodic sequences.

The system complexity for discrimination is an aspect related to the internal model used in the recognition system, namely, the classifier. The degree-of-certainty to identify a motion sequence is usually a function of the classifier complexity. The recognition efficiency and speed of a classifier depends on several factors such as feature space dimensionality, linearity, and representability. Although the complexity of a classifier is proportional to the algorithm complexity (big $\boldsymbol{O}$ notation), while the efficiency in discrimination is not. The efficiency in discrimination is rather related to the linearity of the feature space. If the feature space is linear and separable, then the discrimination should be also high. However if the feature space is nonlinear or not separable, then the discrimination efficiency using linear classifiers is low, even with complex classifiers.

A lot of research in $3D$ motion sequence recognition for anthropomorphic mechanisms has been done in the area of artificial visual systems and computer science, yet the results are limited due to biological and technological reasons. Thus we review the limits of artificial visual systems for motion sequence classification from a technological perspective. One reviews the capabilities of different systems, their advantages and disadvantages, and specific requirements necessary to achieve articulated motion recognition.

Modern visual systems used in the capture and the analysis of articulated motion have foundations in perceptual-neuroscience studies trying to understand how the human brain perceives and identifies articulated motion, especially biological motion. These studies were conducted using unreferenced moving light displays [130], [131], [115]. Their conclusions initially supported the hypothesis that articulated motion could be inferred from just basic features without identifying the underlying structure of the motion mechanism. This is the basic idea behind a lot of work in human tracking and the analysis of articulated biological motion using simple feature spaces such as optical flow, color, and texture cliques. Such feature spaces are used in conjunction with affine linear models [189], [8], [158], [159] to track images. And later, with the use of temporal templates [25], [2], [26], [65] as a way to decode complex articulated motion in a fast and reliable way.

Other studies [179], [134], aimed at testing the moving light display hypothesis. They led to the discovery of specialized sub-adjacent brain structures that play an important part in the human visual system. The importance of such structures lies in their exclusivity in revealing the underlying body structure of articulated motion from motion cues. The discovery of such specialized structures presented a new basis for developing artificial recognition systems. This new insight state that the capture of articulated motion can be significantly improved by integrating internal models representing the geometrical information into the system. Usually, internal models are abstract representations of the physical structure of an articulated body.

1.1 Design Philosophies

Integrating geometric structures within the system somewhat contradicts Johanson's observations [179], [93], [94] and yet, it complements the notion that it is possible to detect complex articulated motion without first discovering the underlying body structure. This perspective has generated two opposing standings for articulate motion tracking within the computer vision community. The first considers that a system should encompass a bottom up construction. This perspective states that a model can be constructed directly from adequate feature spaces by using clustering and generative models that could eventually reach a higher abstractions of motion model representation. This is better known as bottom-up view. The second, and opposing view, proposes to consider first higher abstractions of complex motion and then generate appropriate features. This standing is better known as the top-down view.

A bottom-up model starts from observing basic features in early stages of data capture, often fulfilling some statistical measures of similitude between elements. This design philosophy assumes that more complex relational constructions will develop from the features by using generative models. These generative models would initially find random relationships between elements and sets of chosen statistical models. The statistical models would provide parametric groups that can be coordinated into organized relationships later. And by using an iterative approach, such relationships will grow into orderly sets of parameters identifying the phenomenon.

Conversely, a top-down model would initially start from a very generic abstraction of the problem. This initial model provides a coarse simplification for later refined versions that may include other factors. A designer can take appropriate decisions about features and implementation issues from this early simplified problem.

A common trait in bottom-up and top-down methodologies for motion recognition is the need to impose structural constraints to the motion data. These constraints can be used to explain or to smooth the data. Geometric models are often used to impose such structural constraints to the data. The major difference between the two approaches is the background motivation for including a geometric model into a recognition system.

1.1.1 Bottom-up Models

Bottom-up models usually start-off using an available feature space. It is assumed that the chosen feature space reflects key characteristics from the phenomenon, although this may not be guaranteed. An example of such proposition is the generation of texture filters using Markov random fields [181], [199]. A generative model is a statistical model based on indirect observations of the phenomenon. Generally, it is expressed as a set of conditional probabilities. The fundamental statement of a generative model is that one can solve the problem of finding the general relationships within feature spaces by observing only their conditional probabilities. But it is not really clear whether the *a posteriori* conditional probability of such filters is higher than the *a priori* conditional probabilities. Nor their repeatability, the feature space for textures is plagued with local minima that produces redundancies in the resulting filters.

On the second stage, the challenge is to find plausible relationships between feature elements. These relationships are established using a generative model based on a statistical measure of similarity among feature states [118], [199]. Other methods may use vectorial arrays or polynomial coefficients as a descriptors. The correspondence between features and higher relational abstractions is generally proposed as a combinatorial optimization problem. An example of such approach is the definition of “textons” or texture atoms [118], [181], [128], [199]. An analogous approach is to define “motons” or motion atoms [200] for motion dynamics as a directed graph problem. Similar approaches define “motons” with dynamic state-space models for two-dimensional motions [160] while the relationships are proposed using a Markovian graph model.

The key assumption in a bottom-up model is that the temporal and spatial relationships will naturally arise from solving the combinatorial problem. The primary issue with the bottom–up methods is that without first establishing the underlying motion structure, the relationships between different regions of the feature space may remain hidden or even lost. This is due to local minima with in the feature space, a common problem when identifying motion using optical flow [128]. In a multivariate space, a global minimization solution is less probable. In order to avoid local minima, often it is necessary to try different combinations of relation candidates to find feasible relationships. The combinatorial problem usually grows exponentially with the number of candidate relationships and frequently falls into local minima.

Bottom-up approaches also tend to have very hard implementation issues that often require some form of bootstrapping [117],[119],[63]. Bootstrapping in bottom–up models usually consists on imposing constraints that limit the number of possible combinations between features. Geometric constraints imposed on the body structure often limits the number of elements in the candidate set. Often the problem and interaction complexities are reduced for the feature space by using a geometric constraints. Consequently, this also simplifies the combinatorial task in finding open sets of relationship candidates. Moreover, the geometric constraints do not even have to be complex since a simple bi-dimensional kinematic model can be used to establish relationships between moving regions [4].

1.1.2 Top-down Models

Top-down models often arise from different research areas. Studies in neuropsychology [93], [134], [92] and motion perception [94], [166] in the human brain usually yield newer relational models. The main concern is the construction of simpler mathematical and computational models that can explain the observations.

In a top-down approach to motion recognition, the first abstraction deals with how biological visual systems discover the underlying motion structure. Since finding how such a neurological system works may prove to be difficult, the first step is to propose a simplified modeling structure that establishes general relationships between moving regions instead. Usually, the geometrical and mechanical nature of articulated motion simplifies the model since the structure can be either a geometrical model, a mechanical model, or a combination of both. The first internal model is often inaccurate and error prone. In order to tune–up the initial model, the trend [118], [1], [41], [42] is to increase the model complexity by adding parameters to reduce the initial error. The process is applied iteratively until an acceptable error is reached.

Other top-down approach prefer to use dynamic physical models. They depend on a state-space in order to relate the variations in a feature space with simple dynamic constraints. The objective is to relate different regions of the feature space with simple stationary dynamics. As an example, if a spring model is added [117], depending on the initial conditions it will produce a linear approximation to a stationary response. Regions within an optical field representing parts of an articulated body can be related using such a mechanical approximation. This also provides a physical meaning to the interactions between regions and limits the range and velocity of particular regions in a stationary manner [117]. This approach requires parameter tuning in order to synchronize the tracked motion. Unfortunately, non–stationary conditions included in transient responses are not considered as part of the model since the input is actually unknown. This limits their usefulness in gesture comparison.

Advanced visual motion recognition systems use a battery of hardware solutions to overcome or go around such problems as occlusion and region segmentation. [85], [83], [38], [39] use both kinematic and dynamic constraints. Sophisticated systems use combinations of basic and abstract feature spaces [39]. A basic feature space is comprised of low level processing features such as luminance, color, and edge information while an abstract feature space encircles basic features that satisfies certain constraints within a structure. Some examples of abstract feature spaces are the point distribution model [111], [20], contour models [21], [22], and gradient flow vectors [196], [195].

Advanced visual motion recognition systems use a battery of redundant information to tackle problems such as occlusion and region identification. Complex visual recognition systems employ multiple cameras to tackle occlusion while color coded body suits [84, 85] tackle the region segmentation problem. These systems commonly use internal models to represent the body geometry with simple three-dimensional geometrical objects such as cylinders and spheres [82]. These systems define $3D$ motion as a set of discrete body poses or configurations [4], [53], [65], [35]. These pose sets are better known as body postures. The body poses are a code book from which a discrete sets of fixed states comprises a gesture trajectory while the interpolated poses intermediate configurations. The identification problem is then reduced to a combinatorial optimization problem. The gesture recognition is achieved by solving the fitting minimization problem between the observed posture sequence and the code book sequence [101], [124].

In the case of stationary motion sequences, such as *gait motion*, [117], [106], [143], [148], [129], the problem is simplified even further since the motion sequence is periodic. The spatio-temporal trajectories are sets of homogeneous transformation matrices within a bounded range. Usually, the segmentation process is done by manually selecting specific known postures. Then, the identification problem is expressed as a dynamic-programming minimization problem. The fitness function minimizes a probabilistic form of a distance between a specific state in the posture state sequence and the current estimated posture. The estimation of the posture is done by determining the approximate correspondence between tracked points and the system's internal model [100], [57], [113], [178].

Top-down models promote the use of kinematic and state-space model constraints by design. These constraints are intrinsic to the model and they are not used to bootstrap the model as in the case of bottom-up models. Many top-down models can be found in the robotics literature [68], [72], [114] where the dynamics of articulated mechanisms is known. Advanced frameworks such as the Muybridge project [41] use a combination of multilevel probabilistic constraints for feature clustering and deterministic models to improve the posture estimation problem [38], [41]. Such combination is usually applied in target optical capture systems that do not require retro-reflective or active targets, a.k.a. target-less or free-target optical capture systems. The requirements of a target-less system increase the complexity of the body's region segmentation problem for any articulated body. Some assumptions about the number of segments and range of motion for each segment are necessary in order to identify specific body parts [63]. By definition, these models require a geometrical description of the articulated body. Such description specificity specializes the model exclusively to certain types of articulated bodies. The main issue in any target-less system is solving the correspondence problem between the geometrical description and the visualized areas. Different solutions to the correspondence problem are proposed using methods such as: Markov random fields analyzing clusters [41], [35], [144], [174], active contours [182], [127], [22], [186], principal component analysis [19], [17], and binary template matching [40], [15].

1.2 Motion Decoding and Encoding Trade-offs

Let's start with an example designed to track and analyze American–sign language from video cameras. It is assumed that such a system can measure the six parameters of the scalar fields $\{\boldsymbol{r}, \boldsymbol{\theta}\}$ for a $3D$ trajectory for every tracked target or region. This ability provides a rich output with a complete set of six-dimensional trajectories for the upper-limps. An important issue is to identify each segment a full conversation event. Several questions arise from this conjecture: how to best represent such a trajectory? What is the best model to represent motion: a state-space model or a probabilistic approach? And what are the "critical" points that segment the conversation process with a chosen model?

In this example, the system has a very rich and yet, an incomplete input about the phenomenon. The system has access to a complete trajectory description but it lacks a description of the exerted forces on the mechanisms, plus it lacks a detailed mass description for the body. What type of model can one use to obtain a reliable segmentation. From the classification standpoint, how can one create a prediction for the next stage in a conversation? Additionally, other constraints are necessary for an on-line implementation such as tractability and polynomial time complexity.

1.3 Abstraction Levels

As a solution to our hypothetical system, our recognition algorithm requires a cognitive structure with three main elements:

1. a context for the gesture,
2. a grammar or dictionary, and
3. a segmentation process.

These abstraction levels are part of an artificial cognitive structure that has been accepted as necessary by the computer vision [24] and neuroscience communities [23].

According to this structure, the context is the highest level of abstraction. The context are external significations that are conditioned by the environment. As such, the context is generally out of the scope of interpretation of many recognition systems. “Context” is the matching problem between specific state of “intention,” “purpose,” or “emotion,” to particular sequences of $3D$ trajectories. The mechanisms on how humans interpret context and their associated cognitive structures are unknown [23], [156]. Artificial recognition systems define such labels implicitly by design [24]. Current recognition systems cannot interpret nor define the context by themselves, such definition is embedded onto the system by the designer.

The intermediate level of the abstraction is the grammatical structure. It is mainly concern with the ordering aspects or the configuration of a sequence of motion states while the uniqueness of representation of the motion states falls into the domain of the segmentation model.

Motion grammar can be seen as organizing model parameters into state clusters that can be later used to identify motion trajectories. There are some parallels with this approach to the context of automatic signature recognition systems. The key idea of “atoms” of motion is particularly interesting for the fields of robotics and motor-skill transfer. Robotic mechanisms could greatly benefit by constraining the inverse kinematic solutions of redundant manipulators and solving piecewise linear dynamical models [171], [188].

Motor-skill transfer is a new field interested in the accurate evaluation and repeatability of gesture and motor execution by humans. The order in which these atomic states of motion are executed is seen as a configuration problem for the motion sequence. A set of motion states is a set containing all possible values available for a motion sequence. Such a set is said to be an alphabet of motion [24], [110].

Finally, the base element of the abstraction structure is the segmentation process. The segmentation process is concerned at finding the uniqueness for each possible state in a set of motion states. This is the key element of articulated motion analysis, whether if it is possible for the motion to be decomposed into 'segments," or "atomic" elements [163], [194]. A segmentation process for articulated motion analysis requires that one must be able to analyze temporal sequences with distinct and definitive parameters. The purpose of the segmentation process is to define the states of motion relating a structure motion model to data.

While the grammar structure is concerned with the combinatorial aspects, the segmentation focuses on modeling aspects such as finding distinctive parameter cliques in the model representation. Using a music analogy, the segmentation process is similar to determine the fundamental tone for each one of the music keys. A segmentation process in music would be equivalent to find a set of parameters for the fundamental tone from a bank of discriminating filters. Then, the filter bank response would be used as a unitary element for which a musical composition can be analyzed using those discrete states. Other forms of similar segmentation processes can be done with autoregressive models for stochastic signals such as electro-encephalography [152], [153].

Analogously, motion segmentation is seen as an organizing model in which the parameters of state-clusters uniquely identify motion trajectories. Motion segmentation processes with dynamical models benefit from physical motion properties. Here, the goal is to locate points in the sequence that are considered as "remarkable" or "critical." The definition of "remarkable" or "critical" point is somewhat arbitrary and highly dependent on the particulars of the model used to describe motion. The critical points may be defined as the points in which there is a significant event that occurs, such as the variation of third statistical moments or jerk, a sudden change in direction, a stop, or an acceleration [55], [68].

The inherent properties of a particular segmentation technique generally depends on solving a minimization function for the trajectory model or a transformation space such as scale space or differential geometry invariants (curvature). Moreover, there are difficulties and trade-offs when expressing what "remarkable" or "critical" constitutes between a model and the physical world. Certain aspects of what could be of interest for interpreting a phenomenon may be completely masked by the choice of a particular feature space. Thus, the desired segmentation pattern is unattainable for a particular set of features.

In purely geometric models, the idea of achieving motion segmentation with time series analysis is very tempting. It is always possible to define state-space descriptions using low-order autoregressive models [46], [45]. In this thesis, we will show later that it is not always possible to analyze articulate human motion with such signal processing tools due to fundamental constraints in the Fourier domain.

1.4 Previous System Classifications

A classification based on feature complexity suggests that visual systems for articulated motion capture can be divided into system for rigid and non-rigid body motion [1]. Dedicated systems for rigid motion will always assume that the features will provide some form of invariance such as: scaling, translation, rotation, and occlusion. Non-rigid body dedicated systems, on the other hand, make no assumption about such invariance to deformation, but they state the requirement that the deformation in the feature space is always tractable.

A great deal of computer vision systems for automatic motion recognition, tackle the problem of human motion recognition following an architecture with the following stages: preprocessing, parameter clustering, code-book creation, sequence classification, and finally, prediction. A general processing scheme is shown at Figure (1.1). Scene understanding refers to the problem of segmenting the corresponding regions of the tracked objects from the rest of the image. Different relationships between the distinct segments in the image are often inferred from optical flow. The resulting information allows to create a motion model that includes a hierarchy of regions with common features such as velocities, textures, or intensity values. The clustering of the regions is often done using statistical techniques such as Markov random fields [66]. While with occlusion the problem is predicting where the motion will emerge from a temporal singularity, *i.e.*, the impossibility to follow the motion due to the disappearance of part of the object from the scene.

A proposed taxonomy divides the tracking and motion capture systems in terms of the technical structure [136]. It classifies systems based on technology and required assumptions such as appearance, model initialization, tracking methodology, pose estimation, and recognition methods. The classification is composed of two major groups: low level motion recognition and higher level recognition systems.

In low-level recognition systems, there is no internal model and therefore such systems lack any prediction capabilities. Low-level recognition systems

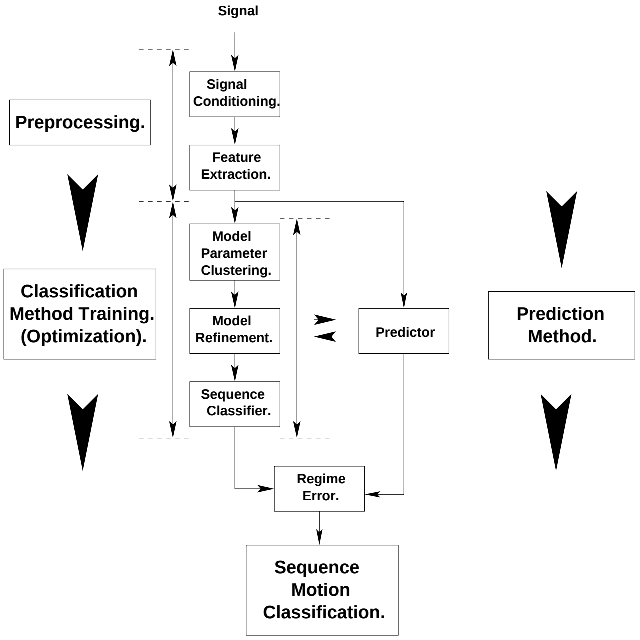


Figure 1.1: Conceptualized General Block Diagram for Visual Systems for Articulated Motion Analysis.

follow the assumption that motion can be recognized without establishing a prior relationship between co-jointed parts. Great deal of applications within this category involve automatic surveillance systems and systems to identify human motion from basic features.

Higher level recognition systems on the other hand do possess an internal kinematic model as part of the prediction phase. Such systems make assumptions about the pose, pre-modeled kinematics, and some prediction scheme. In most of the visual system specific for human motion, the internal model is reduced onto a two dimensional kinematic space or a form of representing changes in the feature space. Appearance models, for example, use linear models to effectively predict the position of a group of projected pixels with some basic assumptions on velocity. Deformable models use simple affine models to capture the sequence of changes that an active contour encounters as the image evolves. The randomness introduced by the digitalization of the visual system is often dealt using a smoothing filter such as the Kalman filter, or dealt with model identification with discrete sequence models such as hidden Markov models.

1.5 Internal Models

Our aim is to construct a categorization based on the internal model assumptions and its classification of computational requirements. It is also necessary to review the associated recognition techniques and classify them into comprehensive groups. Whether we design a bottom-up or top-down system, one important issue to discuss is the choice of the internal model governing the system for articulated motion analysis. This choice essentially depends on the articulated body characteristics and the amount of information provided by the sensors. The internal model usually falls within one of following categories: descriptive and predictive models.

Descriptive Models are not concerned with the causes of motion and they describe motion in terms of geometric affine transformations. Descriptive models are used in the classification of motion using conditional probabilities. The models can be group into two main subcategories: geometric and statistically-based.

- geometric, or kinematic, models are sequences of affine transformations that describe the spatio-temporal trajectories of an object;

- statistically-based models describe the motion in terms of a state-space, but the main difference with predictive models is that the state-space remains unobservable and one cannot guarantee that the model can predict beyond a specific scope of training.

Descriptive models do not, in general, predict the motion as predictive models do. Prediction in descriptive models is imposed with polynomial parameterizations of motion. A trajectory description is accomplished by keeping a discrete set of sample trajectories. The sample length may be different for each trajectory and the classification is done with different techniques. A popular approach to compare multi-dimensional trajectories is to use dynamic time warping [13] [37].

Predictive Models are physically-based linear state-space models. State-space models solve differential equation for the discrete value in the state-vector [78]. They are predictive models with high level of detail for variables representing a system given specific inputs and conditions. These models require elaborate descriptions of a body's physical properties and the external forces exerted onto it. Lagrangian state-space models [78] are particularly useful when complete force dynamics are available. Used in conjunction with kinematic models describing the geometry body, it is possible to attain complete spatio-temporal trajectory descriptions. Also, because the dynamics are encoded into the Lagrangian state-space, it is also possible to obtain non-stationary predictions. The main hinder with state-space models is the elaborate description of the body and their computational complexity.

The reason for including statistically based models in this classification is that although they may include a state-space, either continuous as a Kalman filter or discrete as a hidden Markov model, the prediction stage is limited to only the next step (Markovian condition). Statistically based models are conceptually similar to state-space models [86], [88], [87], [168], although an explicit description of the dynamics of the body is not readily available, the stages of a motion are implicitly described in the form of a directed graph. Graph models may also provide a prediction stage but such prediction is usually limited. Also, the scope of such prediction is limited to one step at a time as in Markovian processes.

A primary example of predictive models is motion dynamics in robotics. Motion dynamics in manipulators require models which take into account both position and orientation of each of the segments comprising the mechanism. Additionally, robotic mechanisms require to handle the static and dynamic forces exerted onto them, thus such models also use additional dynamic Lagrangian mechanics. In such cases, the required characterization of both the object and applied forces is very detailed. In the case of organic motion, such descriptions are not available without having a complex experimental set-up. Thus, a generalized description of motion for organic objects is usually based on kinematics rather than Lagrangian mechanics.

1.5.1 Template and Parametric Models

Motion tracking and gesture recognition applications for human motion that do not use state-space models are based on statistical properties of recorded template signatures and parametric spaces [66], [64], [65]. In general, these applications rely on complete signatures from the activities. The tracked trajectories are not decomposed any further. These type of approaches can be classified as motion template recognition methods.

The popularity of motion template approaches is due to quasi-real time recognition for small sets of gestures [26]. The reason behind this quasi-real time recognition is that the template evaluation is time linear $\boldsymbol{O}(n)$. The working principle of temporal templates relies on the capture of a complete spatio-temporal trajectory. Each spatio-temporal trajectory contains a complete sequence of motions that total the gesture as a whole. In the case of artificial visual systems, the captured sequence is treated as a complete set of frames of pre-segmented binary images. The binary images are added together time-wise forming a temporal template. Temporal templates achieve invariance to translation, rotation, and scale by using statistical moments [25], [66]. Template classification is achieved with statistical distances that are measured between the captured sequence and the stored template. This is a scalar value from the total area of the temporal template. Nonetheless, such template methods are not projective invariant since the method is designed to work in two-dimensional image space. Therefore, any change in camera position requires new template sets for distinct camera viewing angles. A major differentiation of the template approaches from dynamical approaches is that template methods do not segment motions into a sequence of possible configurations or principal modes.

More recent application of temporal templates is proposed for a hand gesture recognition system [96]. As any template, it does not use any form of a state-space, instead it uses a geometrical abstraction of the hand to create an abstract template. The orientation and position of the hand is extracted from disparity information using a stereo camera system. For each gesture available to the system, an individual template is created by manipulating the geometrical model, thus creating a sequence from individual templates. In order to compare sequences, the system uses disparity information extracted from a selected regions of interest. A form of projective invariance is accomplished by un-warping the images. The statistical properties of selected features such as color and region area determine a distance measure to the available templates. The approach is similar to the temporal template techniques [66], [64], [35].

Another form of template technique is to use parametric modeling. In parametric modeling, gestures are modeled as parameters or coefficients from possible affine transformations [149]. The approach is similar to kinematic transformations using interpolating schemes. A complete gesture is described in terms of coefficient states that form a configuration of movements.

1.5.2 State-Space

The term state-space is frequently used with two related yet distinctive connotations. In control engineering, a state-space representation is a mathematical model of a physical system described as a set of inputs, outputs, and state variables; all related by first-order differential equations. Whereas in computer science, a state-space is a description of a configuration of state machines. Such description is better posed by graph theory as a set of nodes (states), vertices (state transitions), initial states (inputs), and goal states (outputs). A major distinction between the two connotations is that whereas the former definition refers to very specific physical laws with transient and stationary conditions from a system's transfer function, the latter can only be a continuous or discrete set of possible combinations from observed variables.

State-spaces admit matrix algebra descriptions. Such systems can be described in one of two major canonical forms [105]: Controller and observer. The observer canonical form is of special importance since it guarantees that the internal state-space is observable. This is of vital importance to the estimation of the internal state of the process.

A state-space of a system is usually represented by the general equation

$$\begin{aligned} \dot{\boldsymbol{x}}[k+1] &= \boldsymbol{A}\boldsymbol{x}[k] + \boldsymbol{B}\boldsymbol{u}[k], \\ \boldsymbol{y}[k+1] &= \boldsymbol{C}\boldsymbol{x}[k+1] + \boldsymbol{D}\boldsymbol{u}[k]. \end{aligned} \tag{1.5.1}$$

where $\boldsymbol{A}$, $\boldsymbol{B}$, $\boldsymbol{C}$, and $\boldsymbol{D}$ are matrices and $\dot{\boldsymbol{x}}$, $\boldsymbol{x}$, $\boldsymbol{y}$, and $\boldsymbol{u}$ are vectors. The index k indicates an arbitrary time step. For time-dependent systems, the matrices $\boldsymbol{A}$, $\boldsymbol{B}$, $\boldsymbol{C}$, and $\boldsymbol{D}$ will also contain an index k. The matrix $\boldsymbol{A}$ usually contains the system's poles, while the matrix $\boldsymbol{B}$ contains the system's zeros. Since in any system the poles determine the stability, $\boldsymbol{A}$ is also known as the characteristic matrix of the state-space system. Whether the system is stable or not, depends on the position of the poles in $\boldsymbol{A}$ which is dependent on the eigenvalues of $\boldsymbol{A}$. If all the eigenvalues of $\boldsymbol{A}$ are all real, then $\boldsymbol{A}$ is linearly independent and $\boldsymbol{x}$ is guaranteed to have a stable solution. If all the eigenvalues of $\boldsymbol{A}$ are complex conjugates, then $\boldsymbol{x}$ has an oscillatory solution, also known as critically stable. An finally, if the eigenvalues of $\boldsymbol{A}$ are a mixture of real and complex values, then $\boldsymbol{x}$ may be nonlinear and unstable.

Hereafter, one will denominate any system which include a linear formulation of motion in control engineering state-space as explicit state-space system. One will make this specific distinction for systems whose descriptions can also be formulated as state-spaces, but where the challenge of a technique is to find a best formulation for the state-space. Such state-spaces are called implicit state-spaces. Explicit state-space formulation is commonly used in the predictive stage of a tracking system. It is applied to low level and high level features in visual systems for motion tracking. And the most common form of estimation is through a variant of a Kalman filter [105]. Implicit state-space formulation is commonly found in systems in which the states are hidden and an explicit formulation of the dynamics is not readily available.

1.5.2.1 Explicit State-Space Models

The category of explicit state-space models describes all those modeling techniques for motion that decompose a gesture using a state-space description. In particular, this category has numerous applications that make use of Gaussian modeling techniques such as linear models or Kalman filters, several variations such as hidden Markov Models (HMMs), particle filters, and Gaussian mixture models.

An application using an explicit state-space formulation does not necessary express the state-space in terms of the object physical properties. Instead, such a formulation can be expressed using properties from indirect sources such as feature observations. Visual tracking systems for articulated motion often use a state-space formulation that is specific for the application or feature space used. This feature space is usually a bi-dimensional state-space (y, v) formulation. In cases for which the tracking system deals with a $3D$ problem, the state-space might achieve projective invariance using multiple views.

From the system identification perspective, segmentation of motion sequences is accomplished using explicit state-spaces modeling physical parameters such as velocity and acceleration. The observe parameters are measured from particular features with the tracking system and a configuration space is selected. The configuration space, ordered state-space sequences is expressed as a regime problem. A "regime" refers to the principal mode of operation of a system. The principal mode of operation is identified by observing the eigenvalues and eigenvectors of the control matrix $\boldsymbol{A}$. As a regime changes so does the eigenvalues and consequently, the operation mode.

Regimes are identified as basic modes of operation. This has led to identify these basic modes as "atoms" in dynamical systems. Dynamic models [68] aim to discover the principal modes of operation and the configuration problem, *i.e.* the switching sequence at which the state-space model changes for a given system. An analogous idea has appeared for articulated motion analysis. The terms change and several names had been given: "movelet" codewords [76], behavior models [80], [81], and corpus of three-dimensional motions [91].

In order to observe a regime it is necessary to continuously solve the eigenvalue for the matrix $\boldsymbol{A}$, which may be a hinder. Another possible way to quantize the principal modes of operation, without solving the eigenvalue problem for the system, is to quantize velocity profiles as part of a HMM [108]. In this way, gestures can also be viewed as likelihood state machine. The issue with this type of representation lies in the hidden state-machine of an HMM. However, the classification efficacy in a HMM improves as the hidden state-machine approximates a deterministic one [56].

One way to train a curve-based HMM for gesture recognition can be obtained by fitting discrete samples of a principal curve [29]. The state representation is usually obtained by manual means and the minimization algorithm is complex and not be always stable. The classification of a gesture with principal curves is performed using an error function for the configuration sequence.

It only works for three-dimensional positioning leaving the orientation to be solved by other means.

We will now review basic fundamentals of some important explicit state-spaces for visual tracking. Such state-spaces do not solve a pose configuration space for an internal kinematic model, instead they use the evolution through the state-space for different samples to describe a configuration based on principal component analysis. However, the final configuration space does not provide an exact solution to the gesture problem, although it recognizes sequences of motion by finding the sequence of eigenvectors that fit the available data.

1.5.2.1.1 Active Contour and Active Appearance State-Space

The basic description of a deformable model and an active shape model are the same. The spatial part of an active contour defines a shape using a closed spline.

On the other hand, the temporal dynamics of an active contour are stated in terms of a state-space model that consists of two parts. The first part takes into account the modifications into the shape by modifying the location of the control and the node points in the spline. Certain applications [22], [82], [111], [18], [16] define spline families that use interpolating control points in order to simplify the estimation of the final shapes. The control point array for the spline is known as Point Distribution Model (PDM). The second part consists at tracking the shape in time in the image.

The main distinction between active shape and active appearance models is how the shape is estimated in both. In an active shape model, the shape is estimated only once at the beginning of the image sequence since the shape won't change through out the image sequence. The problem to be solved is to identify newer positions as the sequence evolves. On the other hand, in an active appearance model, the shape of the tracked object changes at every instant and the estimation has to be done at every frame. The models has to take into consideration all the possible changes in shape of the object through out the image sequence.

The main issue in active appearance models is how to estimate the corresponding affine transformation sequence for the control points in the active contour. This is in fact a kinematic problem. There are several solutions to this problem that do not solve the pose estimation configuration [112], [20].

When the motion is considered non-rigid then the dynamics of motion can be described as a sequence of transformations that are applied to the active shape when the articulated body is being tracked. The dynamics described by the active shape models and particle filters rely on projective transforms. The affine transformations required to modify the non-rigid motion by comparing the deformation sequence; and the estimation of the parametric angles of the superimposed stick model could proved costly.

Additionally, the Kalman filter estimating the kinematic parameters may not be able to overcome the problem of local minima. The matrix inversion required by the Riccati's equation is computationally expensive for multidimensional spaces [71]. A cheaper solution is to use a weighted approach using particle filters [69], [70]. Nonetheless, on the current hardware, the solution is not fast enough to be easily implemented in real-time. These constraints have a lot to do with the feature extraction process and; how costly the contour determination is. It is an expensive method since the computation complexity is $\boldsymbol{O}(n^3)$ for a three degrees-of-freedom joint.

1.5.2.2 Implicit State-Space Models

Implicit state-space systems are black-box formulations of state-spaces. They are usually based on Bayesian formulations with Gaussian assumptions. A popular formulation of implicit state-space models used for articulated motion tracking are the Hidden Markov models (HMM) [161], [88], [87]. To some extent, HMMs can be considered as a discrete Bayesian formulation of a Kalman filter [168], [89]. The latter is due to the homologous formulation between a Kalman filter and a hidden Markov model [167] but, in general terms, HMMs are graph models [169], [165].

Implicit state-space models are commonly used on those systems for which the internal variables can be inferred only from indirect observations. A great deal of tracking systems that use low-level features such as optical flow and pixel neighborhoods use HMMs to solve the configuration problem between internal states [187], [193], [6] and state segmentation [147], [36]. A major disadvantage of the implicit state-space models is the requirement to initially guess the number of states in the model [132]. There is no solution for the minimal number of internal states for a implicit state-space model.

Implicit state-space models are frequently found in applications where human gesture is being segmented into "atoms" of motion. Several approaches in gesture segmentation have been tried with mixed results. A popular method is to use linear models in the form of HMMs to create spatial invariants which can be matched in the temporal sequence [27]. The applications use a series of poses or orientations determined from features such as color or shapes [51].

In a commonly used formulation for gesture recognition, the gesture itself is viewed as a position tracking problem with quantized probabilistic states [22], [44], [43], [27], [192], [51]. The applications use a series of fixed poses or orientations for which a specific gesture is traced. A downside of considering a gesture as a state machine is the lack of model flexibility. Depending on the complexity of the gesture, the proposed state machine can contain a large number of states making model training difficult. The intuition is then to apply a divide and conquer scheme. In the divide and conquer scheme with HMMs, a complex model is transformed into a series of simpler, more manageable models [44], [43]. Divide and conquer schemes constitute a higher abstraction level for classes of simpler models that are fast to train and compute, but the interactions between the models may also turn to be complex. Different types of HMMs for multiple observations, multiple states, and parametric models are described in the literature [142], [148].

In general, the interaction between lower order models and multiple observations is not intuitive enough to have a simple state description. An important caveat in HMMs is that the number of internal states has to be determined empirically [133]. A gesture class cannot only contain a large number of models, but also the models may have different number of internal states. Under such circumstances, it is not possible to find a unique solution to gesture interpretation. And finally, HMMs do not allow a natural inclusion scheme for increasing the number of model classes that can constitute a family of gestures [27], [192], [191].

A better strategy to solve the problem of larger gesture database is by using hybrid models. Hybrid models include a form of candidate pre-selection phase [150], [151], [154]. The pre-selection phase smooths the candidate models with a Kalman filter. The Kalman filter smooths the autoregressive model parameters [150] for a clustering phase which provides the priors for the HMMs. The purpose of the Kalman filtering phase is to smooth the likelihood function of the hidden states using the signal covariance. Thus, the unsupervised method provides a better estimate of the number of internal states for the HMM.

However, applying such approach to three-dimensional gesture recognition may prove difficult since the trajectories in the three-dimensional Euclidean space are hardly stochastic. The traces not only are composed of low Fourier frequencies but they are also deterministic in nature. Therefore, autoregressive models may not be adequate or discriminative enough to differentiate between close, but distinctive, clusters.

An abiding issue in unsupervised methods is to determine the optimal number states that best describes a parametric space. The idea is that a simple model provides a coarse approximation to any phenomenon and that the error can be reduced by incrementing the number states in the model. However, an indiscriminate increase in the state number will yield a problem commonly known as the “dimensionality curse.” Models must rely on finding an optimum number of parameters in order to best describe the phenomenon.

How to achieve this goal is an enduring question. A way to measure the effect of the parameters in the model representation is to use a metric known as information criterion. There are several information criteria available, most of them are based on the information entropy principle or Shannon entropy [172]

One useful criterion is the Minimum Description Length Criterion (MDLC). MDLC is a modified version of Akaike's information criterion (AIC) [123], [3]. Its purpose is to automatically find a plausible cluster configuration that best describe a parametric data set with an optimal number of parameters [34], *i.e.*, it aims to automatically determine the number K for a given data set. The MDLC aims to find a minimum number of states while preserving most of the information conveyed by the feature space without falling into the "dimensionality curse." The method itself is a blind recursive recalculation of different clustering configurations for a particular data set. When the criterion reaches a minimum scalar value inside a series of possible configurations, the configuration with the minimum value is chosen as prior candidate to the hidden state machine in the HMM.

1.5.3 Motion Description Using Geometric Models

Geometric models, a.k.a. kinematic models, are descriptive models. Geometric models are used to describe complex articulated motion when state-dynamics are not available. Motion is described as a series of affine transformations that are relative to a reference frame $\mathcal{F}$. The position and orientation of the frame $\mathcal{F}$ are completely arbitrary. One can define a fixed global reference frame $\mathcal{F}_o$ such that any motion is relative to this reference frame.

When motion is described using kinematic models, it is necessary to achieve some form of invariance in order to guarantee recognition. There are two common ways to achieve invariance in the motion recognition problem: Fourier invariance and state–parametric invariance.

1.5.3.1 Fourier Invariance

Stationarity or periodicity is the most important factor to achieve invariance in the Fourier domain. Periodicity implies that a sequence of states repeats itself after a certain time interval.

Some kinematic problems are considered stationary, *e.g.* gait, cyclist's leg-pedal complex, arm swimming stroke, or certain repetitive gestures with communicative or manipulative purposes. However, not all motion sequences can be considered stationary from a Fourier perspective. Thus, a natural classification for motion sequences based on the Fourier domain are stationary and non-stationary motion sequences.

In general, manipulative gestures fall in the non-stationary motion sequence category. In these situations, starting and finishing points have variable timings and the motion dynamics have also execution time differences for a same gesture [28], [116], [180], [104], [120].

1.5.3.2 Parametric State-vector Invariance

Invariance in the state-parametric space is achieved through the geometric descriptive parameters of a body. This means that a complex motion, described in state-vector sequences will remain the same independently of an arbitrary external reference frame $\mathcal{F}$. Articulated bodies often use a modified state-space representation for individual link-joint pairs. The state vector pair $\boldsymbol{x}$ is not composed of six elements, but only four elements:

$$\begin{aligned} \boldsymbol{x}_k^{\mathsf{T}} = [\boldsymbol{d}_k,\, \boldsymbol{\chi}_k]^{\mathsf{T}} = [\mathrm{z}_k,\, \mathrm{x}_k,\, \chi_k,\, \alpha_k]^{\mathsf{T}} \\ \forall \quad k = \{1, \ldots, m\} \quad m \in \mathbb{N}. \end{aligned} \tag{1.5.2}$$

where the parameters $\boldsymbol{d}_k = [\mathrm{z}_k,\, \mathrm{x}_k]^{\mathsf{T}}$ are Euclidean distances measured along the link and the joint k respectively; $\boldsymbol{\chi}_k = [\chi_k,\, \alpha_k]^{\mathsf{T}}$ are the rotation angles for the joint and the angular difference between links measured about the normal vector to two consecutive direction links $\hat{z}_k$ and $\hat{z}_{k-1}$, .i.e, about the direction $\hat{z}_k \times \hat{z}_k$. One refers the $\boldsymbol{x}_k^{\mathsf{T}}$ state vector as the Denavit-Hartenberg vector for the k link-joint pair.

The reduction in the number of descriptive parameters is possible due to the Denavit-Hartenberg (DH) notation [60] uses for the kinematic link (KL) a constrained description. The insight of the DH notation is that it describes any KL intrinsically. Thus, the DH notation is independent from an external reference frame $\mathcal{F}$. It also simplifies the KL complex pose sequences by solving one single variable at the time.

1.5.4 Geometric Models in Neuroscience

A major field of study in Neurophysiology is the study of reaching gesture tasks for neuro-control of the upper limb [184], [185], [9], [145], [77]. The first step in these studies is the characterization of a “geometric stage” of the 3D–Euclidean space [184]. The geometric stage refers to the problem of first finding geometric solutions for a reaching gesture for the upper limb. The methodology has two important assumptions. First, the initial state is a resting configuration where the kinematic energy is kept at minimum and the potential energy does not change. Second, the reaching gesture passes from the initial resting state to the reaching posture with minimum kinematic energy change.

Usually, the solution involves solving all parameters for the end effector in the configuration space, and in general, the solution is not unique. The goal of the geometric stage is to find out geometric constraints that reduce the space to the most likely configurations. These studies produce high–likelihood parameter configurations from data of exemplary sets of reaching gestures. Although such solutions may not reproduce the internal processes occurring in the human neuro-control system, they provide important clues about the hidden dynamics of the upper limb.

1.5.5 Prediction With Geometric Models

A downside of geometric models is that they cannot deal with motion prediction and tracking uncertainty without *a priori* assumption. Since external forces to the body are unknown, the prediction of the dynamic behavior can be only assumed. In the absence of a differential equation, one can only assume a model, obtain statistics for average estimates for the trajectory. A popular choice is to assume a function such as the “constant acceleration” model [77],[91]. Another possibility is to assume multiple linear dynamic models and then choose a winner. In this situation Kalman filters are useful to consider the prediction error for different models. The main issue is that such models assume a free body where no possibility for modifying the inertia by altering the momentum by itself. An improvement is to consider a Gaussian function with limited support for the ‘third order derivative and then integrate [90],[145]. Although motion can be described by geometric models, the assumptions imposed on the model severely constraint higher hierarchical stages of motion prediction, smoothing and recognition.

If no physical constraints or motion model given, motion becomes a random variable. Often, it is assumed the most simple random motion: Brownian motion. However, under random motion assumptions it is never guaranteed that two points in a three-dimensional Euclidean space would ever share the same value, as it happens in a two-dimensional Euclidean space. In a two–dimensional Euclidean space if a particle sweeps a bounded section with a random Brownian motion, it is guaranteed that the same particle will pass at the exact position twice in undefined amount of time. The implication is that it is not advisable to use generative models in a three-dimensional Euclidean space in order to predict the next point in the curve.

Since a gesture does not behave randomly, other simpler constraints are required to delimit the body’s motion and eliminate the Brownian motion. One can assume Markovian constraints [51], [164], [73] for which the particle cannot randomly change directions, but to follow previous directions with a small random coefficient that depends on the previous state. Such constraints can be expressed in terms of conditional probabilities or smoothing conditions for the trajectories. Those constraints are not often easily identifiable for particle motion, but are somewhat empirical approximations to assuming quasi-Newtonian conditions.

Fortunately, there are some other intriguing experimental cues indicating that the speed control of reaching gestures is path independent [9], [145]. A seemingly evident assumption has been disproved, *e.g.*, a reaching gesture, the final arm posture depends on the initial shoulder and elbow angle configuration [176]. Instead, an energy function that minimizes the work amount exerted when passing from a point A to a point B in reaching gestures has been adopted as for subsequent studies [145], [184], [185].

The work minimizing function is proposed as quasi-dynamic constraint. By considering the motion direction of the upper limb, a reaching gesture can have different weights for arm elevation and arm lateral pronation. The weights are empirical approximations to physics constraints that minimizes a for minimal energy cost function [184].

Other possible constraint to predict motion using geometric models emerges from an invariance condition. A speed invariance has been reported [9], [184], [185] for reaching gestures of the upper limb. In motion planning, if the distance from the starting pose to the final pose configuration is known and if the maximum reaching speed is available, then the invariant functional can be applied to a motor control approximation. Other weighted constraints, such as comfort [184] or muscle metabolic intake consumption limits [5].

Although all of the above constraints allows us to restrict solutions for under-determined reaching gestures, there should be clear that the main problem in reaching gestures is the definition of the configuration space for the gestures. The initial and the final poses are known. However, in gesture recognition there is not such a clear cut definition. Usually, motion recognition and motion planning have broader expectations that involve difficult recognition problems. These problems may include:

1. *Description.* A geometric trajectory can be easily described in two-dimensional Euclidean space. It is even guaranteed that for a given surface in a two-dimensional Euclidean space, if one follow a Brownian motion, one will pass on the same point at least twice in an undetermined amount of time. Such a condition does not exist for a three-dimensional Euclidean space.

2. *Temporal dynamics.* Should one consider temporal dynamics to determine whether a spatio-temporal trajectory belongs into a given family

of trajectories is being intensively discussed. But it actually depends on how the spatio-temporal trajectories are being described. State-space descriptions are time-dependant and therefore they should include temporal dynamics in the model. Implicit state-space models such as hidden Markov models also include temporal dynamics but they try to find a form of time-invariance to the sequence, *i.e.* the observation vectors can change their length but the statistical properties of the space configuration remains. In other words, the state and state transition probabilities do not change unless the model is also modified.

3. *Constraints.* An empirical constraint is necessary to limit an energy cost function when a minimization is required. Such constraint may not be realistic and requires parameter tuning. Also, it demands the use of a training set of data. The descriptive data set used for training can be map into a different space, in the hope of enhancing key features, but it is not guaranteed. It is difficult to provide a single description for all possible cases and all possible configurations, so constraints tend to be application specific.

4. *Noise sensibility.* Although the noise can effectively be reduced in a two-dimensional Euclidean space with uncertainty distances to the main trajectory. Other considerations have to be taken into account when dealing with noise in three-dimensional Euclidean space. For example, when using principal curves method [103] for comparison, one also requires integrating an area along the trajectory to be used as uncertainty surface. Ideally, such a surface should be described as a function of the normal to the tangent point on the main trace describing a family of traces.

In general, $3D$ spatio-temporal curves are viewed as the composition of two separate and geometrically different parametric spaces: position and orientation. Aside the intrinsic space geometry, the major issue with any $3D$ sequence recognition is that most recognition algorithms for sequence recognition are frequently limited to only one or two variables, but not three. Such situation has created the need for other types of description that are able to reduce the dimensionality. One of such descriptions is differential geometry with osculating reference frames. Differential geometry defines two main orthogonal, osculating axes to represent a $3D$-curve with two parameters: curvature and torsion.

The definition of curvature and torsion in differential geometry can be formulated using the Frenet-Serret (FS) formulas [99]. The notation uses uppercase letters to represent vectors. This is against the notation in this document.

However, since the notation is traditional when dealing with FS frames, this convention will be preserved.

A FS frame defines n orientation vectors, one for each dimension, see Appendix A.1 . In $3D$, a FS frame defines three normalized vectors: $\widehat{\mathsf{T}}$, $\widehat{\mathsf{N}}$ and $\widehat{\mathsf{B}}$. The first, the normalized tangent vector $\widehat{\mathsf{T}}$ is defined as:

$$\widehat{\mathsf{T}} \equiv \widehat{\dot{\boldsymbol{r}}}, \quad \widehat{\dot{\boldsymbol{r}}} = \frac{\dot{\boldsymbol{r}}}{\|\dot{\boldsymbol{r}}\|} \tag{1.5.3}$$

where $\widehat{\dot{\boldsymbol{r}}}$ is the normalized velocity vector at the point on the curve $\boldsymbol{r}(t)$. $\widehat{\mathsf{N}}$ is the normalized normal vector and orthogonal to $\widehat{\mathsf{N}}$, is defined as:

$$\widehat{\mathsf{N}} = \dot{\widehat{\mathsf{T}}} = \widehat{\ddot{\boldsymbol{r}}}, \quad \widehat{\mathsf{N}} = \frac{\ddot{\boldsymbol{r}}}{\|\ddot{\boldsymbol{r}}\|} \tag{1.5.4}$$

where $\ddot{\boldsymbol{r}} = \nabla \cdot \dot{\boldsymbol{r}} = \nabla \cdot (\nabla \boldsymbol{r}) = \nabla^2 \boldsymbol{r}$ is the Laplacian of $\boldsymbol{r}(t)$. And the last normalized vector $\widehat{\mathsf{B}}$ is orthonormal to both $\widehat{\mathsf{T}}$ and $\widehat{\mathsf{N}}$ by construction:

$$\widehat{\mathsf{B}} = \widehat{\mathsf{T}} \times \widehat{\mathsf{N}} = \frac{\dot{\boldsymbol{r}} \times (\ddot{\boldsymbol{r}} \times \dot{\boldsymbol{r}})}{\dot{\boldsymbol{r}} \cdot (\ddot{\boldsymbol{r}} \times \dot{\boldsymbol{r}})} = \frac{\dot{\boldsymbol{r}} \times (\ddot{\boldsymbol{r}} \times \dot{\boldsymbol{r}})}{\|\dot{\boldsymbol{r}}\| \, \|\ddot{\boldsymbol{r}} \times \dot{\boldsymbol{r}}\|}. \tag{1.5.5}$$

For a $3D$-curve $\boldsymbol{r}(t) = \{x(t), y(t), z(t)\}$ the arc-length s is defined as:

$$s^2 = \int_0^t \big\|\nabla \boldsymbol{r}(t)\big\| dt \tag{1.5.6}$$

where $\dot{\boldsymbol{r}} = \nabla \boldsymbol{r}(t)$ is the velocity vector. By normalizing Equations (1.5.3), (1.5.4) and (1.5.5) against the curve's arc-length s, Equation (1.5.6), a FS frame is defined as:

$$\begin{bmatrix} \widehat{\mathsf{T}}' \\ \widehat{\mathsf{N}}' \\ \widehat{\mathsf{B}}' \end{bmatrix} = \begin{bmatrix} 0 & \kappa & 0 \\ -\kappa & 0 & \tau \\ 0 & -\tau & 0 \end{bmatrix} \begin{bmatrix} \widehat{\mathsf{T}} \\ \widehat{\mathsf{N}} \\ \widehat{\mathsf{B}} \end{bmatrix} \tag{1.5.7}$$

where κ is the curvature and τ is the torsion. The curvature and torsion can be computed directly from the parametric curve description as:

$$\kappa(s) = \frac{\|x'(s)y''(s) - x''(s)y'(s)\|}{\left(x'(s)^2 + y'(s)^2\right)^{\frac{3}{2}}} \tag{1.5.8}$$

and

$$\begin{aligned} \tau(s) = \quad & \frac{z'''(s)(x'(s)y''(s) - y'(s)x''(s)) + z''(s)(x'''(s)y'(s) - x'(s)y'''(s))}{(x'(s)^2 + y'(s)^2 + z'(s)^2)(x''(s)^2 + y''(s)^2 + z''(s)^2)} \\ & + \frac{z'(s)(x''(s)y'''(s) - x'''(s)y''(s))}{(x'(s)^2 + y'(s)^2 + z'(s)^2)(x''(s)^2 + y''(s)^2 + z''(s)^2)}. \end{aligned} \tag{1.5.9}$$

The most important advantage of using curvature and torsion is that any curve in a $3D$-Euclidean space is defined uniquely. Another advantage is the reduction of dimensionality in one dimension. This allows to apply classification methods for sequences on the re-parameterized tuple sets $\{\boldsymbol{\kappa}, \boldsymbol{\tau}\}$.

By contrast, the issues of using the $(\kappa - \tau)$ parametrization for curves are not less important. First, both the curvature κ and the torsion τ, as expressed by Equations (1.5.8) and (1.5.9) respectively, have numerical issues. The curvature κ is very sensitive to noise while the torsion τ is numerically unstable due to its dependency on the third derivative of $\boldsymbol{r}(s)$. And second, both signal processing and classification methods for sequences usually assume that sequences are continuous with no vanishing points. And yet, it is almost ensured that the curvature κ and the torsion τ will contain vanishing points.

Numerical stability is the most important issue. Classification results depend on the accuracy of the $(\kappa - \tau)$ representation. It has been reported that classification results for HMMs with recognition rates around 95% using differential geometry re-parametrization is possible if the training and classification algorithm use polynomial interpolation filters to estimate higher-order derivatives [52].

In general, conventional approaches use the $(\kappa - \tau)$ re-parametrization by first fitting the curve $\boldsymbol{r}(s)$ with a $3D$-spline [162], [49], [129], [141] and then use the fitted curve to approximate curvature and torsion from the spline. However, $3D$ spline fitting for curves is most difficult. First, spline fitting depends on a minimization criteria that takes into account the characteristics of a spline family [157]. Second, there is no proven method that solves the minimization for fitting a spline onto a $3D$ curve [95].

Cubic splines have some advantageous characteristics since their control points are the same as their knot nodes. This particularity simplifies the minimization strategy when fitting data sets to splines. It is guaranteed that it will produce a smooth curve with minimal curvature. And finally, cubic splines are less computationally expensive than other spline interpolation algorithms. However, cubic splines tend loop and wiggle due to a sensitivity to variations in the second order initial conditions. The reason is that the control points and the spline knots are the same.

Other spline families may have better smoothing characteristics. B-splines are one of such families. However, the control nodes in the B-splines are not part of the curve. The control points in the B-spline family lie at convex positions outside the fitting curve. This particular characteristic posses a difficult minimization problem. The construction algorithm of a B-spline is based on the DeCasteljau algorithm. Although, the DeCasteljau algorithm is well behaved in two dimensional fitting problems, it does not extend well for the three-dimensional problem [61]. The difficulty is that the control points of a $3D$ B-spline may lie on an unrestricted number of possible locations, all local minima, but the global minimum is not guaranteed.

Another way to smooth a trajectory data set is by using principal curve methods [29]. Principal curves are trajectory reconstructions based on statistical estimates of the curvature from numerous complete trajectory samples [103]. The main difference between principal curves and splines is that a principal curve does not use a single linear polynomial. Instead, the shape of principal curve is reconstructed recursively using principal component analysis (PCA) and least squares methods to fit polynomials of increasing order.

All of the above methods apply to the analysis of curves in Euclidean spaces. However, $3D$ orientation and rotation take place in a non-Euclidean space. Rotation sequences also describe trajectories that one should be able to match. Yet, due to the characteristics of how orientations and rotations are represented, trajectory matching is not an easy feat. One should make a distinction between orientation and rotation. Orientation is the direction of a rigid body relative to a reference frame. Rotations are mapping operators that work upon the direction of a body.

Initially, a discrete orientation is represented as a state-space vector

$$\boldsymbol{\theta} = \{\theta, \phi, \psi\}\,, \tag{1.5.10}$$

containing a scalar field with the Euler angle magnitudes for corresponding rotation axes. Consequently, rotation sequences are represented as a state-vector sequences for every discrete step i of the rotation sequence, *i.e.* a rotation sequence is represented as a set of vectors $\theta = \{\boldsymbol{\theta}[1],\, \boldsymbol{\theta}[2],\, \cdots,\, \boldsymbol{\theta}[i],\, \cdots,\, \boldsymbol{\theta}[M]\}$. At every step i of the sequence, each specific orientation vector is decoded with a rotation operator. The operator can be in the form of a matrix $\boldsymbol{R}$ or other mathematical object that handles orientations.

An n-dimensional rotation operator can be obtained as the product of individual rotation operators in lesser dimensions. However, rotation operators are non-commutative. Their sequential order is not interchangeable but it is reversible. Consequently, the order in which the product is obtained is important. All rotation operators use an specific order to obtain the general operator. This specific order is known as “convention.” Some common notations are: the aeronautical convention (YPR for Yawl,Pitch and Roll), the ZYZ convention, and the XYZ convention [60]. In this sense, representing the orientation with a state-vector is advantageous since it is independent of the rotation convention.

A rotation sequence represents an orientation trajectory. Orientation trajectories, unlike Euclidean trajectories, are impossible to describe as continuous functions. A naive approach is to consider the state-vector set $\boldsymbol{\theta}$ as linearly independent functions for each Euler angle and use linear relationships for interpolation. Such a description fails when the variations in $\boldsymbol{\theta}$ are not small, *i.e.*

$$\boldsymbol{\theta} + \delta\boldsymbol{\theta} \not\approx \boldsymbol{\theta}. \tag{1.5.11}$$

The reason is that the rotation operator is nonlinear for variations of $\boldsymbol{\theta}$ outside a small neighborhood [173]. It is common that the rotation operators obtained under such premises contain unforseen singularities. This problem is commonly known as the "gimbal lock." The reason behind is that many of the properties of Euclidean spaces are not available for rotation operators [125].

Establishing interpolation functions for orientation trajectories is another important issue. Approximations to the orientation trajectory using linear interpolation with rotation matrices usually yield jittering trajectories. The jitter is because of the first-order approximation with the Jacobian matrix [14], [185]. Additional constraints are necessary to smooth the inverse of Jacobian matrix [183], [50]. Euler angles and rotation matrices are ill suited for such a task because of the afore mentioned reasons. Other descriptions such as quaternions allow interpolation functions for ranges outside a small variation of $\boldsymbol{\theta}$ [173]. However, such functions consider geodesic paths [62]. The consequence is that a rotation sequence is set of geodesic paths between fixed orientations.

The most important limitation in comparing orientation sequences refers to how to measure differences between two different orientation state-vectors. Orientation matrices do define a distance between two orientation state-vectors. Another important is that each orientation trajectory has to be preserved in precisely n number of states comprising the whole rotation sequence.

An advantage of the orientation state-vector is that it is minimal and complete representation. An orientation state-vector is minimal because there is no further reduction in the number of parameters. It is also complete because it fully represents the rotation operation. Rotations are also represented with other operations that require a greater number of parameters such as vector-angle operations, homogeneous matrices, and unitary quaternions.

Rotation representation with either vector-angle rotation or unitary quaternions use four parameters. A rotation with vector-angle notation uses a three-dimensional vector and an scalar. The unitary vector describes a directed rotation axis while the scalar indicates the rotation magnitude. Quaternions have a similar geometric interpretation.

However, one can reduce the number of parameters as a rotation vector with only three parameters. The reduction involves a re-mapping from Cartesian to Polar coordinates. Instead of representing a direction as a unitary three-dimensional vector, the axis direction can be viewed as a point on the surface of a unitary sphere. The sphere manifold requires only two parameters to indicate a direction. The direction is given by a set of coordinates which direct an outward unitary-vector normal to the sphere surface at that specific set of angular coordinates ϕ and ψ. Thus, the scalar set $\boldsymbol{\theta}$ describes a vector-angle operation, a representation that it is minimal.

Other forms for representing orientation paths are based on using a subset of a Grassmann algebra, *i.e.* quaternions [74], [10], [11]. Quaternions are a non-Abelian group, non-commutative algebra; meaning that the multiplication order cannot be freely exchanged and the quaternion inverse has a negative sign. An orientation and a rotation operation are generally represented with a quaternion subset: unit quaternions. The unit quaternion subset is defined by all the quaternion numbers with norms equal to one. Unfortunately, the unitary quaternion subset is not a close set for addition and substraction. Algebraic operations such as addition and substraction will remain in the quaternion domain, but their solutions will not have unit norm. Only the multiplication and the division are closed operations for the unit quaternion subset. Additionally, the rotation representation with quaternions is not minimal, both an orientation and a rotation are represented with four parameters in quaternions.

Computer graphics is one of the first applications representing rotations with unit quaternions. Quaternions have an interpolation framework for rotations that is commonly used in computer animation. However, rotation interpolation does not use quaternions directly. Rotation interpolation involves an indirect approach in which two quaternion numbers mark the initial and the final orientations. Then, the procedure divides the geodesic trajectory between the two specific quaternions. Each point in the geodesic is estimated using a linear weighted function and the corresponding orientation positions are returned as Euler angles. The algorithm was first introduced by Shoemake [173] and it has been used ever since. Other and more complex algorithms construct splines using geodesics between specific orientation points, but all use Shoemake's algorithm as a foundation [62]. Quaternions are non-convex and non-linear spaces. This is the limiting factor that makes their use in applications other than computer graphics difficult.

In this thesis, we will demonstrate that it is in did possible to use quaternion spaces for motion sequence recognition. We will also develop some arguments over the properties of such spaces, some simplifications, and how they can be actually used to simplify recognition tasks, even if these spaces are non-linear and difficult to conceptualize.

1.6 Summary

Several points can be drawn from the previous review of the literature.

1. The effectiveness of any recognition system is greatly enhanced when the feature space truly represents the phenomena.

2. Any state-space representation has to include the geometric interpretation of the articulated motion. Many constraints need to be imposed to restrict conditions such as scene or sequence length, producing specific systems for very specific conditions. There are no general solutions.

3. Template approaches are computationally inexpensive but they do not provide invariance to other factors, nor they can be used to segment the elements of the gesture.

State-space models need a great amount of description about the body structure and the external forces acting on it. For the case of human articulated motion recognition systems such descriptions are not usually available. In the case of robotics, it is possible to provide such descriptions, but they also require dynamic models for motion planing which is not easy to provide. Motion planning requires multidimensional minimization that often is ill–constrained yielding local minima solutions. Trajectory planning in robotics is a fertile ground for geometrical models of motion where the state-space models of dynamics are complementary to each other.

Explicit and implicit space models can be seen as the two sides of the same coin. Both tackling different perspectives and complementary aspects of the problem. Explicit state-spaces have powerful differential descriptions of a system. Such descriptions have to deal with some degree of randomness and uncertainty due to system conditions and measurements. The best estimates for such systems are provided with adaptive filtering techniques such as the Kalman filter. Such filtering techniques require knowledge about the signal uncertainty and therefore some assumptions are made about the noise. Implicit state models are proxy models that do not have an explicit description of the system and can be viewed as black boxes. The premise is to extract information from indirect observations from inputs and outputs. Usually such observations are expressed in form of discrete sets of conditional probabilities.

What it is expected from a proxy state-space system is that a Shannon maximum entropy input will yield a minimum uncertainty output. This is the rationale when a system is analyzed with implicit state–space methods, one expects from the system to have some deterministic features that will be reflected on the system's output. This principle is applied to signal processing, signal compression, and system identification and it is also valid to implicit state-space techniques such as hidden Markov models.

Some issues when using state-space models for gesture recognition are:

1. Determining a similitude metric. In other words, how one can determine that a gesture belongs to a specific family of gestures from two different spatial geometric spaces: position and orientation.

2. Explicit state-space models: when the sequence of consecutive observations remain in the same modal forms (eigen-values).

3. Implicit state models: when the sequence between comparative model and the observed sequence remain within a probability threshold.

4. Geometric models: when the observed and pattern trajectories remain within a threshold distance and their curvatures preserve the same sequences.

Current homogeneous representation of geometric models, a.k.a. kinematic models, have shortcomings when representing rotations. Such deficiencies are similar to state model representations in the sense that it is impossible to obtain a short representation of the $3D$ trajectories of a rigid body.

In order to compare kinematic trajectories from a body, one has to compare segments of the complete trajectory in the form of sequences of state-space vectors. The memory and computational requirements to store individual trajectories are great. In addition, there is no mathematical formulation for comparing directly sequences of state-vectors outside an Euclidean space. Orientations and rotation operators possesses spherical geometry and linear approximations on the sphere are ineffectual outside of a small neighborhood. This implies that for fast changes in orientation the sampling rate for the trajectory should be high. How high the sampling rate should be is speculative at the moment. One must know the characteristics of the orientation space.

An important difficulty with rotation matrices is that they do not have a geometric interpretation. The rotation matrices conform a special group, namely special orientation group $SO(3)$ [60], [125] in lie algebras. Often, in the analysis of spatio-temporal trajectories, the first intuition is to extract the Euler state-vector from the orientation matrices. Most of all state information on the position and the orientation is provided in a state-vector form. The position state is an Euclidean space, a space with metric. It is possible to use interpolation to infer intermediate states. On the other hand, Euler spaces do not have intrinsic metrics, and thus linear interpolation do not converge to an accurate value in between Euler space-states. One has to use a different metric, other than Euclidean, to correctly interpolate points between two distinctive orientations.

Rotors or quaternions are one specific Grassmann algebra that provides a space with better comparative features for orientations, yet the spaces remain mostly unexploited in pattern recognition. The reason is that quaternions are far less intuitive than rotation matrices. A well known interpolation algorithm that uses quaternions is the Spherical Linear Interpolation (SLERP) algorithm [62].

More complex trajectories in the orientation space use other interpolation algorithms, but they use SLERPs as primitives. A SLERP defines a geodesic arc on the surface of a sphere such that it is analogous to straight line in a Euclidean space. Rotation sequences can be segmentally described and classified with sequences of geodesic arcs.

Chapter 2

Mathematical Foundation for 3D Gesture Comparison

2.1 Introduction

In the previous chapter, we discussed different approaches to the motion sequence recognition problem based on dynamical and geometric models found in the literature. The solutions are divided into two main groups: continuous and discrete.

Continuous methods solve Lagrangian models under structural constraints imposed by a geometric model. Lagrangian models are predictive and require detailed physical parameters while geometric models are descriptive and their parameter sets are rather small. Usually, a Kalman filter is used to smooth the force estimates from the Lagrangian models. Continuous models define structural and physics–based constraints for path trajectories in $3D$ space.

Discrete methods are proxy models to physical models. Unlike Lagrangian methods, proxy models avoid large parameter sets using distribution likelihoods instead. A motion recognition problem is viewed as a discrete optimization problem for a set of state sequences. The motion is recognized when a state sequence has an overall minimum distance to a pattern sequence. The main concern is to define a distance metric between sequences.

From the signal processing standpoint, $3D$-gesture comparison can be seen as solving a template matching problem where motion sequences are identified using template correlations. Fundamentally, a correlation is a linear measure of similitude. In signal processing, correlations identify specific patterns with a known spectral characteristic inside of a sequence of unknown patterns.

Spectral characteristics often depend on how the sequence and the metric are defined. The ability to match any pattern within a sequence or a structure is highly dependent on the invariance of the spectral description. In the Fourier domain, the spectral characteristic of a signal is the signal's power or energy content per cycle unit $\left[\frac{2\pi}{\text{sec}}\right]^2$. The estimate from the signal's power distribution in the Fourier domain is known as the signal's spectrum. However, in a geometric domain, the shape of a manifold is determined by its spectral characteristic which is generally defined by solving the eigenvalue problem:

$$\boldsymbol{A}\boldsymbol{x} = \lambda \boldsymbol{x} \quad \forall\, \boldsymbol{x} \neq \mathbf{0}. \tag{2.1.1}$$

Most geometric spaces define intrinsic measures of distance from their spectral characteristic. By definition, Euclidean spaces define their intrinsic distance as the inner product $\langle\ ,\ \rangle$ of the vector difference $\boldsymbol{d}$ between two points $\boldsymbol{x}[j]$ and $\boldsymbol{x}[k]$ such that $\boldsymbol{d} = \boldsymbol{x}[k] - \boldsymbol{x}[j]\ \forall\ \{j, k \in \mathbb{Z}\}$ and under the Euclidean inner product:

$$d^2 \equiv \langle \boldsymbol{d}, \boldsymbol{d}\, \rangle = \boldsymbol{d}^{\mathsf{T}} \boldsymbol{d} = \|\boldsymbol{d}\|^2. \tag{2.1.2}$$

The distance d is invariant and extensible to points in Euclidean spaces of higher dimensions. Other geometric spaces with geometries different to Euclidean also define spectral distances accordingly.

Gesture comparison, as defined in Chapter 1, is fundamentally a problem of similitude metrics in spaces with two different geometries. Gesture comparison is not only concerned with identifying a particular sequence of states, but also with measuring the individual similitude between states in a sequence. In human gesture, the comparison problem is more than simply identifying specific temporal sequences of motion. The comparison has to take into account not only specific trajectories in a three-dimensional space, but their temporal characteristics as well. Gesture comparison cannot be solved by only considering the state distribution in the feature space. Gestures are not just collections of random points in space, they are deterministic, they have constraints and they have limiting conditions due to structural relationships.

Another important aspect of gesture comparison are time dependencies. Motion sequences may have the same spatial trajectories, but they may be executed with different timing at different stages. A form of invariance involving time is the stationarity or periodicity of a signal. There are motion sequences that are also periodic and thus stationary. Often motion sequences are compared based on their stationarity. A prime example of stationary motion are gaits. Gaits and other complex kinematic motion can be analyzed using time or its reciprocal in the Fourier domain. However, aperiodic motion sequences such as gestures are not stationary. This implies that for a real-life gesture recognition system, spatial invariance is essential but not sufficient; time invariance is also required.

One of the main contributions of this thesis are the use of invariant representations and distance metrics for both position and orientation to perform temporal gesture comparison in $3D$ space. Gesture comparison cannot rely only on solving the identification problem for a sequence, but it also has to find spatial invariance and space metrics. This is key since the comparison process has to be independent of the point of view of an arbitrary reference frame $\mathcal{F}$.

This chapter is divided into five main sections. Section 2.2 introduces the concept of similitude metrics under geometric constraints for kinematics. Section 2.3 describes the mathematical foundations for trajectory analysis of position alone. Section 2.4 introduces fundamental concepts for orientation trajectory analysis based on three different representations: rotation matrix, vector-angle rotation, and quaternions. Section 2.5 introduces different quaternion projections onto $3D$-Euclidean spaces. We introduce a projection space that allow us to use Euclidean differential geometry directly to quaternions, an important contribution of this thesis. We also analyze different metrics for orientation transformation and dimensional reduction. We will then summarize the advantages and disadvantages of the proposed method and the thesis contributions.

2.2 Similitude Metrics and Geometric Invariance

Invariance is the key feature for any similitude metric. Invariance is the property of a vector $\boldsymbol{x}$ to remain unaltered by a n-dimensional affine mapping $\boldsymbol{f}$. The specific mapping may include one of the following operations: translation, rotation, and scaling. The invariance is usually characterized by the mapping $\boldsymbol{f}$ operating over a n-dimensional vector $\boldsymbol{x}$ such that:

$$\|\boldsymbol{f} \circ \boldsymbol{x}\| \mapsto \|\boldsymbol{x}\|, \tag{2.2.1}$$

where $\circ$ defines a general operation of $\boldsymbol{f}$ over $\boldsymbol{x}$. The solution to Equation (2.2.1) is equivalent to finding a solution of the Eigenvalue problem:

$$\boldsymbol{A}\boldsymbol{x} = \lambda \boldsymbol{x} \quad \forall \ \boldsymbol{x} \neq \boldsymbol{0} \text{ and } \lambda = 1. \tag{2.2.2}$$

In both Equation (2.2.1) and Equation (2.2.2), the inherent metric is one. The functional form of $\boldsymbol{f}$ requires invariance in scaling, thus its metric is fixed to one. In the Eigenvalue problem, if a solution for $\boldsymbol{A}^{-1}$ exists then, the orthogonal vector corresponding to the smallest eigenvalue defines the principal

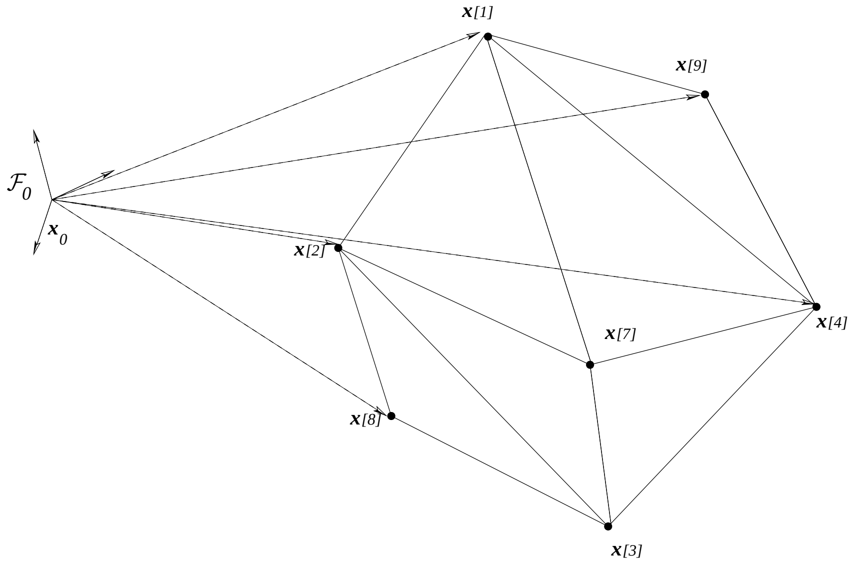


Figure 2.1: A set of points $\mathsf{x} = \{\boldsymbol{x}[1], \boldsymbol{x}[2], \ldots, \boldsymbol{x}[M]\}$ describes a $3D$ structure referenced to an external reference frame $\mathcal{F}_0$. The object defined by the set x is not invariant to translation or rotation from the external reference frame $\mathcal{F}_0$.

direction that does not modify the vector $\boldsymbol{x}$, while the smallest eigenvalue defines its metric.

The invariance property is not always guaranteed. Invariance is lost either because an increase in the number of dimensions or because the intrinsic space geometry diverges from an Euclidean representation. Here, the concept of mathematical manifolds emerges to reconstitute an Euclidean space at infinitesimal scales and thus to preserve some invariance at neighborhood level.

In the case of disjoint spaces, such as Euclidean spaces, most similitude metrics are based on the Euclidean distance, see Equation (2.1.2). The Euclidean distance d^2 is an invariant metric of similitude between points $\boldsymbol{x}$ in a n–Euclidean space since if $\boldsymbol{x}[i+1] \equiv \boldsymbol{x}[i]$ then $d^2 = 0$ and $d^2 > 0$ otherwise.

Now, suppose that a sequence or a structure is defined by a set x containing M points $\boldsymbol{x}$ in an n-dimensional Euclidean space, *i.e.*:

$$\mathsf{x} = \{\boldsymbol{x}[1], \boldsymbol{x}[2], \cdots, \boldsymbol{x}[M]\}^{\mathsf{T}} \tag{2.2.3}$$

referred by an external reference frame $\mathcal{F}_0$, see Figure (2.1). It is assumed that the data set x follows the rules of linear algebra and tensor notation.

From the reference frame $\mathcal{F}_0$ perspective, the object represented by x varies according to changes in position and orientation. Affine invariance for set x is achieved by referring it to a local reference frame $\mathcal{F}_\ell$ and then referring $\mathcal{F}_\ell$ to the general reference frame $\mathcal{F}_0$, see Figure (2.2).

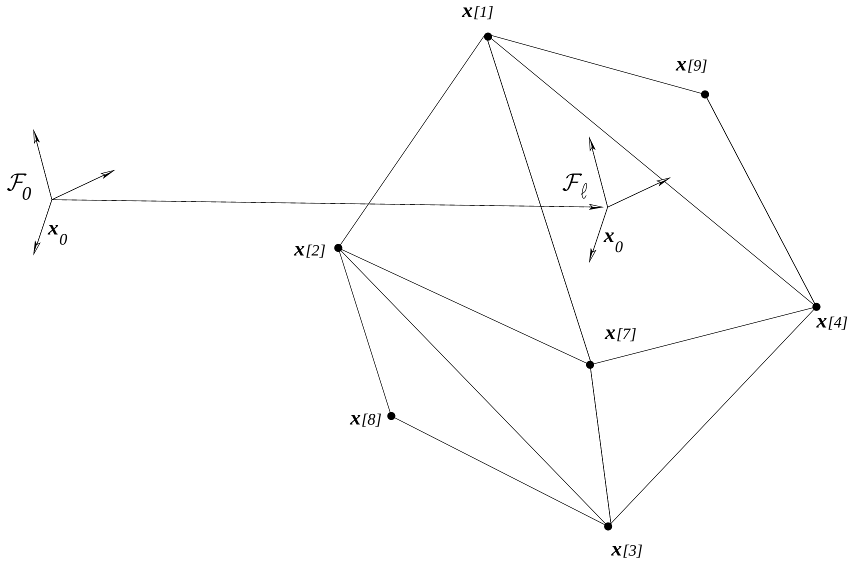


Figure 2.2: The set of points $\mathbf{x}$ is now translation invariant since the structure is referenced by a local reference frame $\mathcal{F}_\ell$. All distances to $\mathcal{F}_\ell$ are constant.

Essentially, positioning and orienting a local reference frame $\mathcal{F}_\ell$ follows one of two basic strategies. The first one is based on statistical moments and it involves placing the frame $\mathcal{F}_\ell$ outside of the manifold, yet within an optimum distance from all the elements in the set $\mathbf{x}$. The second one, places the local reference frame $\mathcal{F}_0$ on the manifold at one element in the set $\mathbf{x}$ such that the position $\boldsymbol{r}_\ell$ of $\mathcal{F}_\ell$ is $\boldsymbol{r}_\ell = \boldsymbol{x}[\ell]$.

In the statistical method, the position $\boldsymbol{r}_\ell$ for the local reference frame $\mathcal{F}_0$ depends on the mean from the data set $\bar{\mathbf{x}}$ such that the frame position $\boldsymbol{r}_\ell$ is:

$$\boldsymbol{r}_\ell = \bar{\mathbf{x}} = \mathscr{E}\left[\mathbf{x}\right]. \tag{2.2.4}$$

The frame orientation, denoted as $\mathbf{U}_\ell$, is also determined using second order statistical moments. The statistical variance of the data set $\mathbf{x}$, denoted as $\boldsymbol{\Sigma}(\mathbf{x})$ is the second statistical moment:

$$\boldsymbol{\Sigma}(\mathbf{x}) = \mathscr{E}\left[(\mathbf{x} - \bar{\mathbf{x}})(\mathbf{x} - \bar{\mathbf{x}})^{\mathsf{T}}\right]. \tag{2.2.5}$$

And the solution of the eigenvalue problem

$$\det(\boldsymbol{\Sigma}(\mathbf{x}) - \lambda \boldsymbol{I}) = 0 \tag{2.2.6}$$

provides the eigenvectors $\boldsymbol{u}_i(\mathbf{x})$ for the local frame orientation $\mathbf{U}_\ell$.

The second method is based on differential geometry of curves and surfaces in space. The position $\boldsymbol{r}_\ell$ for the frame $\mathcal{F}_\ell$ is, more o less, unconstrained.

However, without distinction of a curve or a surface, the location of initial local frame $\mathcal{F}_\ell[0]$ is always within the manifold, *i.e.*

$$\boldsymbol{r}_\ell = \boldsymbol{x}[\ell]. \tag{2.2.7}$$

with the only constraint that the chosen point $\boldsymbol{x}[\ell]$ is not an inflection point, *i.e.*

$$\lim_{\mathbf{x}\to\mathbf{x}_\ell} f(\mathbf{x}) = L \quad \text{usually,} \quad \left\| \frac{d^{(n)}\,\mathbf{x}}{d\,s^n} \right\|_{\boldsymbol{x}[\ell]} < \infty. \tag{2.2.8}$$

where s is the cumulative arc-length for the manifold.

Now, let us consider an homogeneous superset $\mathbf{X}$ composed from individual sets:

$$\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_j, \cdots, \mathbf{x}_k, \cdots, \mathbf{x}_N\} \quad N \in \mathbb{N}. \tag{2.2.9}$$

describing N geometric objects either curves or surfaces. There are no constraints imposed on the individual element sets $\mathbf{x}_j$ as for the M_j number of points within in each set. The location and orientation of the individual local reference frames $\mathcal{F}_\ell[j]$ is also dependent on the individual data set $\mathbf{x}_j$. The problem is how to assert the similitude of the sequences or the structures represented by the individual sets $\mathbf{x}_j$. It is not possible to use a simple similitude metric such as the Euclidean norm from Equation (2.1.2). The Euclidean norm is not a similitude metric for structures or sequences in $\mathbf{X}$. One must take other factors in consideration.

The first factor is the individual number of samples M_j in each subset $\mathbf{x}_j$. The general case assumes that

$$M_k \neq M_j \quad \{j \neq k\} > 0 \ \in \ \mathbb{Z}. \tag{2.2.10}$$

One procedure to compare two different sets $\mathbf{x}_j$ and $\mathbf{x}_k$ with different sample number M_j and M_k is to equate the sample number to the largest of the two. Usually, the procedure involves a combination of interpolation functions and re-sampling operations onto the set with the smallest number of samples.

The second factor is how to measure the overall similitude between the different objects $\mathbf{x}_j$ and $\mathbf{x}_k$. It is possible to compare sets $\mathbf{x}_j$ and $\mathbf{x}_k$ with different sample numbers by either using dynamic programming techniques that rely on the definition of a distance or using differential geometry that uses a re-parametrization. However some minimum conditions are expected. In $1D$ objects, the basic assumption is that $\mathbf{x}_j$ and $\mathbf{x}_k$ are sampled at the same frequency rate. For $2D$ manifolds, the resolution, sampling rate over manifold arc-length, may be different due to projection conditions but the basic assumption is that both manifolds are analytic and aligned in the same direction or, at least, they are invariant to $2D$ affine mappings. Finally, for $3D$

manifolds, besides the analytic condition, the minimum expectation is that a $3D$ alignment function exists.

Similitude metrics based on distances are subject to minimization constraints. Since the invariance for geometric objects is based on local reference frames $\mathcal{F}_\ell$, the correlation is highly dependent on the alignment between reference frames. The usual constraint is expressed as:

$$d = \min \sum_{i=1}^{Q-1} \left\| \left(\mathbf{x}_j[i] - \bar{\mathbf{x}}_j\right) - \widehat{\mathbf{R}}_j^k \left(\mathbf{x}_k[Q-i] - \bar{\mathbf{x}}_k\right) \right\|^{\frac{1}{2}} \tag{2.2.11}$$

for the discrete case and

$$d = \min \int_{-\infty}^{\infty} \left\| \left(\mathbf{x}_j(t) - \bar{\mathbf{x}}_j\right) - \widehat{\mathbf{R}}_j^k \left(\mathbf{x}_k(t+\tau) - \bar{\mathbf{x}}_k\right) \right\|^{\frac{1}{2}} d\tau. \tag{2.2.12}$$

for the continuous case. $\widehat{\mathbf{R}}_j^k$ is a n-dimensional tensor represented in matrix form and it is an estimated alignment operator.

Finding the estimated alignment operator $\widehat{\mathbf{R}}_j^k$ is known as "the registration problem." The geometrical estimation of $\widehat{\mathbf{R}}_j^k$ is based on the well known Iterative Closest Point (ICP) algorithm [198]. The ICP algorithm maximizes the trace of the covariance, $\mathrm{Tr}\left(\boldsymbol{\Sigma}(\mathbf{x})\right)$, for all the paired points as means to determine the optimal alignment operator $\widehat{\mathbf{R}}_j^k$. The main issue with ICP is that its solutions require a good initial estimate of the initial pose profiles. If the initial pose is under- or over-estimated, the minimization function may fall into a local minimum. This is mainly because ICP assumes that the global minimum is close to the initial estimate and thus there is only a small orientation difference between the individual sets $\mathbf{x}_j$ and $\mathbf{x}_k$.

In signal processing, one can compare two $1D$ functions $g(t)$ and $h(t)$ by simply using cross correlation:

$$C(\tau) = (g * h)(\tau) = \int_{-\infty}^{\infty} g^*(\tau) h(t+\tau)\, d\tau \tag{2.2.13}$$

where g^* indicates a complex conjugate of g, $C(\tau)$ is a scalar number, $-1 \leq C(\tau) \leq 1$, that quantifies the similarity between the two sequences $f(t)$ and $g(t)$. Or equivalently in the Fourier domain:

$$C(f) = G^*(f) H(f). \tag{2.2.14}$$

The latter form is popular because it is computationally inexpensive when using the Fast Fourier Transform (FFT), only $\boldsymbol{O}(N \log_2 N)$. The cross correlation coefficient $C(\tau)$ is invariant for one-dimensional functions or continuous sequences with respect to time. However, it is not possible to apply Equation (2.2.13) in higher dimensions.

The reason lies in the geometrical nature of the $3D$ data sets contained in $\mathbf{X}$. Equation (2.2.13) lacks invariance. The cross-correlation has to be modified in order to take into account the alignment problem between distinct subsets $\mathbf{x}_j$ and $\mathbf{x}_k$. Assuming that the sampling rate is the same for both, one defines $Q = M_j + M_k$ to also include the individual lengths for the sets $\mathbf{x}_j$.

The modified n-dimensional correlation is:

$$C[i] = \sum_{i=1}^{Q-1} \left(\mathbf{x}_j[i] - \bar{\mathbf{x}}_j\right)^{\mathsf{T}} \, \widehat{\mathbf{R}}_j^k \, \left(\mathbf{x}_k[Q-i] - \bar{\mathbf{x}}_k\right). \tag{2.2.15}$$

for the discrete case and

$$C(\tau) = \int_{-\infty}^{\infty} \left(\mathbf{x}_j(t) - \bar{\mathbf{x}}_j\right)^{\mathsf{T}} \, \widehat{\mathbf{R}}_j^k \, \left(\mathbf{x}_k(t+\tau) - \bar{\mathbf{x}}_k\right) \, d\tau \tag{2.2.16}$$

for the continuous case. The n-dimensional cross-correlation coefficients $C[i]$ and $C(\tau)$ are one-dimensional vectors of length $Q-1$ and they are subject to the alignment constraints in Equations (2.2.11) and (2.2.12) for they respective cases.

The last similitude metric is based on differential geometry. A signature is needed in order to compare two distinct curves or surfaces. The difference between two or more intrinsic signatures from different curves or surfaces establishes a similitude metric that is invariant to affine mappings. In differential geometry, curves and surfaces are manifolds $\mathbf{x}_\ell$ that can be described with sets of local reference frames $\{\mathcal{F}_\ell[i]\}$. The location and the attitude of each frame $\mathcal{F}_\ell[i]$ are function of each manifold derivatives $\partial^n \boldsymbol{x}[i]$. Every frame's location is function of the zeroth derivative, which is $\boldsymbol{x}[i]$. Each frame's attitude, which is a tensor $\mathbf{U}_\ell[i]$, is determined by the directions of subsequent n-order derivatives $\partial^n \boldsymbol{x}[i]$. The geometrical reasoning is that the n-order derivative set $\partial^n \mathbf{x}$ conforms an orthogonal directional set for each point $\boldsymbol{x}[i]$ on the manifold $\mathbf{x}$.

The intrinsic signatures relate the set of tensors $\{\mathbf{U}_\ell[i]\}$ for each point $\boldsymbol{x}[i]$ on the manifold $\mathbf{x}$.

The position of the initial reference frame is always on the manifold, *i.e.* $\boldsymbol{r}_\ell \in \mathbf{x}$. In this case, the distance metric is defined by a local approximation to the Euclidean distance at infinitesimal intervals. The distance between two points in the manifold is the accumulation of all consecutive points in between.

2.3 Geometric Models for Motion

In the current section, we tackle the problem of kinematic representation. Representing motion with homogeneous transformations presents the problem of unrestricted minimization in a $6D$-dimensional space. Generally, inverse solutions for articulated bodies with more than six degrees-of-freedom are ill-conditioned and not unique.

Geometric models, or kinematic models, are descriptive models. They do not provide information about system dynamics. A point state-vector $\boldsymbol{p}$ is described by a $6D$ set $\{x, y, z, \phi, \psi, \varphi\}$. The state-vector $\boldsymbol{p}$, namely location, is subdivided into two state-vectors: position $\boldsymbol{r} = [x, y, z]^\mathsf{T}$ and orientation $\boldsymbol{\theta} = [\theta, \phi, \psi]^\mathsf{T}$. The evolution of location, denoted as $\boldsymbol{p}(t)$ and $\boldsymbol{p}[i]$, uses these two distinctive spaces:

$$\boldsymbol{p}(t) = \begin{bmatrix} \boldsymbol{r}(t) \\ \boldsymbol{\theta}(t) \end{bmatrix} \quad \text{and} \quad \boldsymbol{p}[i] = \begin{bmatrix} \boldsymbol{r}[i] \\ \boldsymbol{\theta}[i] \end{bmatrix} \tag{2.3.1}$$

for the continuous and the discrete cases respectively. When comparing free rigid motion using Equation (2.3.1), one has to solve two identification problems in different geometrical spaces: Euclidean and hyper-spherical, respectively.

Two important concerns in a $3D$-trajectory comparison are space dimensionality and space metric. Both intertwine at different levels.

For example, Euclidean spaces can be both bounded or unbounded. Yet, they guarantee that a solution exists for an trajectory identification problem when the number of dimensions is limited to two; even for unbounded spaces. However, with three dimensions, a solution existence for the identification problem is not guaranteed, even when the space is bounded. On the other hand, orientation spaces are always bounded and unlimited. However, the number of space dimensions can be greater than three. Unlike Euclidean spaces that define a metric in terms of an internal product, orientation spaces are adimensional since they define an adimensional intrinsic metric measured in degrees or radians. The metric is defined in terms of state–vector projections, not internal products.

The dimensional complexity for comparing a curve is diminished by projecting the curve onto an intrinsic manifold. A high dimensional curve, such as $\boldsymbol{p}(t)$ when it is projected onto a manifold, preserves its characteristics with a reduced number of dimensions. However, not any projection nor any manifold has the ability to preserve the curve's characteristics. The projection is limited by inherent constraints of the curve. For example, the description of an articulated object can be simplified using fixed constraints given by the position and the orientation of each link and joint in the mechanism structure. Consequently, the mechanism's intrinsic structure provides constraints that reduce the dimensionality. The Denavit-Hartenberg notation [60] is an example of dimensionality reduction using structural constraints.

In differential geometry, the structure of curves is best described by frames and the relative geometrical relationships between them. The advantage is that manifolds are described in terms of apparent Euclidean distances between frames. However, curves and manifolds may have geometries other than Euclidean. The reduction in dimensionality appears because the frames lose at least one degree of freedom when establishing relationships to other frames.

2.3.1 Frames

A frame $\mathcal{F}$ is a vector field in a n-dimensional space. Sometimes, a frame is also referred as "tensor." A n-dimensional frame $\mathcal{F}$ defines a set of linearly independent but otherwise arbitrary vectors:

$$\mathbf{U}_{\mathcal{F}} = \left[\boldsymbol{e}_1, \ldots, \boldsymbol{e}_i, \ldots, \boldsymbol{e}_n\right] \tag{2.3.2}$$

known as basis vectors. Such basis vectors are usually defined in terms of the differential properties of the particular space.

The tensor $\mathbf{U}_{\mathcal{F}}$ is an arbitrarily oriented coordinate system. An n-dimensional point $\boldsymbol{x}_\ell$ becomes infinitesimally directed $\overrightarrow{\boldsymbol{x}}_\ell$ by means of a local coordinate system $\mathbf{U}_\ell$ acting upon the point, *i.e.*

$$\overrightarrow{\boldsymbol{x}}_\ell = \mathbf{U}_\ell \boldsymbol{x}_\ell. \tag{2.3.3}$$

The subindex $\ell \in \mathbb{Z}$ uniquely identifies each point in a space. Usually, the subindex $\ell = 0$ identifies a directed point as the global reference point for the space.

In a set of directed points

$$\overrightarrow{\mathbf{x}} \equiv \left\{\overrightarrow{\boldsymbol{x}}_0, \overrightarrow{\boldsymbol{x}}_1, \ldots, \overrightarrow{\boldsymbol{x}}_i, \ldots, \overrightarrow{\boldsymbol{x}}_M\right\} \quad \{i, M\} \in \mathbb{Z}^+, \tag{2.3.4}$$

passing between individual points $\overrightarrow{\boldsymbol{x}}_\ell$ is done by a mapping function $\mathbf{R}^i_\ell : \mathbb{R}^3 \mapsto \mathbb{R}^3$ such that:

$$\overrightarrow{\boldsymbol{x}}_i = \mathbf{R}^i_\ell \overrightarrow{\boldsymbol{x}}_\ell \tag{2.3.5}$$

or, equivalently,

$$\overrightarrow{\boldsymbol{x}}_i = \mathbf{R}^i_\ell \mathbf{U}_\ell \boldsymbol{x}_\ell. \tag{2.3.6}$$

The inverse mapping is simply

$$\overrightarrow{\boldsymbol{x}}_\ell = \mathbf{R}^\ell_i \overrightarrow{\boldsymbol{x}}_i \tag{2.3.7}$$

or

$$\overrightarrow{\boldsymbol{x}}_{\ell} = \mathbf{R}^{\ell}_{i}\mathbf{U}_{i}\boldsymbol{x}_{i}. \tag{2.3.8}$$

In the case of an sequential directed set $\overrightarrow{\mathbf{x}}$, each element $\overrightarrow{\boldsymbol{x}}[\ell]$ has a monotonically increasing order such that the index is $\ell = \{0 < 1 < \cdots < i < \cdots < M\} \in \mathbb{Z}^{+}$. The mappings from the local frame $\mathcal{F}_{\ell}$ to the general one $\mathcal{F}_{0}$ and viceversa are:

$$\left.\begin{aligned} \overrightarrow{\boldsymbol{x}}[0] &= \mathbf{R}^{\ell}_{0}\mathbf{U}_{\ell}\boldsymbol{x}[\ell] \\ \overrightarrow{\boldsymbol{x}}[\ell] &= \mathbf{R}^{0}_{\ell}\mathbf{U}_{0}\boldsymbol{x}[0] \end{aligned}\right\} \text{for } \ell = \{1, \cdots, M\}. \tag{2.3.9}$$

Under Equation (2.3.9), the sequential directed set $\overrightarrow{\mathbf{x}}$ is not invariant to rotations acting upon the global reference frame at $\overrightarrow{\boldsymbol{x}}[0]$, although it is invariant to translations upon $\overrightarrow{\boldsymbol{x}}[0]$.

2.3.2 Directed Curves

The sequential evolution of a directed point, as defined in Equation (2.3.3), describes a discrete directed curve in a space $\overrightarrow{\boldsymbol{p}}[i]$ with respect to a origin point $\overrightarrow{\boldsymbol{p}_{0}}$ that defines a global reference frame $\mathcal{F}_{0}$. Such a curve has mappings defined by Equation (2.3.9). Now, let us assume that the individual directional frames $\mathbf{U}_{i}$ have the same characteristics and that each basis vector $\boldsymbol{e}_{i}$ are at least orthogonal. Under such conditions, the directed curve $\overrightarrow{\boldsymbol{p}}[i]$ can be expressed in terms of a state vector containing individual positions $\boldsymbol{r}[i]$ and Euler state vectors $\boldsymbol{\theta}[i]$

$$\overrightarrow{\boldsymbol{p}}[i] \equiv \begin{bmatrix} \boldsymbol{r}[i] \\ \boldsymbol{\theta}[i] \end{bmatrix}. \tag{2.3.10}$$

that contains the necessary information to recover $\mathbf{U}[i]$ and $\overrightarrow{\boldsymbol{x}}[i]$ from the general reference point $\overrightarrow{\boldsymbol{p}}_{0}$.

Equation (2.3.10) defines a discrete directed curve in terms of two geometrically distinct subspaces. The analysis of gesture trajectories is usually limited to the analysis of the position part, $\boldsymbol{r}[i]$. The reason is the Euclidean metric. The directional part, $\boldsymbol{\theta}[i]$ is generally relegated due to the lack of a linear metric. In the following sections, methods to achieve invariance in both position and direction are introduced and reflected upon.

2.3.3 Gesture Comparison Using Position Alone

Gesture comparison using only position is deceptively simple. It is first assumed that for each directed point in the directed set $\overrightarrow{\mathbf{x}}$, the local frames have the same direction as the global reference $\overrightarrow{\boldsymbol{p}}_0$, *i.e.*

$$\mathbf{U}_i = \mathbf{U}_0 \;\; \forall \;\; i. \tag{2.3.11}$$

Thus, Equation (2.3.9) simplifies to

$$\left.\begin{aligned} \overrightarrow{\boldsymbol{x}}[0] &= \mathbf{R}^{i}{}_{0}\boldsymbol{I}\boldsymbol{x}[i] \\ \overrightarrow{\boldsymbol{x}}[i] &= \mathbf{R}^{0}_{i}\boldsymbol{I}\boldsymbol{x}[0] \end{aligned}\right\} \text{for } i = \{1, \cdots, M\}, \tag{2.3.12}$$

which is simply a change in direction from the global reference frame $\overrightarrow{\boldsymbol{p}}_0$.

As the position $\boldsymbol{x}$ evolves as function of time t, then the direction from the global reference point $\overrightarrow{\boldsymbol{p}}[0]$ to the point $\boldsymbol{x}[i]$ also changes:

$$\left.\begin{aligned} \Delta\overrightarrow{\boldsymbol{x}}[0] &= \mathbf{R}^{i}{}_{0}\Delta\boldsymbol{x}[i] + \Delta\mathbf{R}^{i}{}_{0}\boldsymbol{x}[i] \\ \Delta\overrightarrow{\boldsymbol{x}}[i] &= \mathbf{R}^{0}_{i}\Delta\boldsymbol{x}[0] + \Delta\mathbf{R}^{0}_{i}\boldsymbol{x}[0] \end{aligned}\right\} \text{for } i = \{1, \cdots, M\}. \tag{2.3.13}$$

As Equation (2.3.13) shows, there is a nonlinear term in the expression, $\Delta\mathbf{R}^{i}{}_{0}$ and its inverse. For simplicity, it is assumed that the change in direction $\Delta\overrightarrow{\boldsymbol{x}}[0]$ is mainly due to the change in position $\Delta\boldsymbol{x}[i]$ and $\Delta\mathbf{R}^{i}{}_{0}$ is negligible. Then Equation (2.3.13) is

$$\left.\begin{aligned} \Delta\overrightarrow{\boldsymbol{x}}[0] &= \mathbf{R}^{i}{}_{0}\Delta\boldsymbol{x}[i] \\ \Delta\overrightarrow{\boldsymbol{x}}[i] &= \mathbf{R}^{0}_{i}\Delta\boldsymbol{x}[0] \end{aligned}\right\} \text{for } i = \{1, \cdots, M\}. \tag{2.3.14}$$

2.3.3.1 Curve Invariance with Local Frames

The independent variable of time t plays an important role in many aspects of motion, but one has to consider if the variable t is really valuable. Dynamics are not part of the motion description using kinematic models. A motion is described only in terms of geometric transformations. Time is valuable when other physical variables such as mass, density, and momentum are part of the model, but is meaningless otherwise. For these reasons, it seems reasonable to leave the notion of time as a parameter to analyze gesture.

An invariant re-parameterizations is to redefine time parameter t in terms of the cumulative arc-length s instead. By replacing time t with the cumulated arc-length s, a curve is no longer influenced by an external variable but only by the intrinsic geometry of the curve. The position trajectory $\boldsymbol{r}(t)$ is now invariant under s and one can re-map $\boldsymbol{r}(t) \mapsto \boldsymbol{r}(s)$ directly. This effectively changes the position of a point in the trajectory determined by an external variable t to an intrinsic variable s which can be normalized between $[0, 1]$.

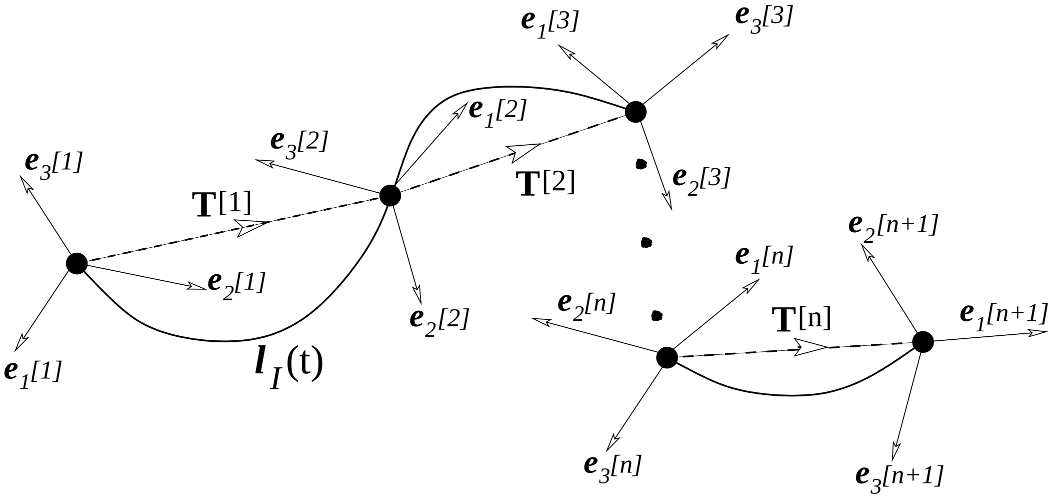


Figure 2.3: A set of discrete transformations describing a sequence of translations and rotations. The discrete transformation $\boldsymbol{T}[i]$ translates and re-orients the previous state vector $\boldsymbol{p}[i]$, effectively moving the origin for the next transformation $\boldsymbol{T}[i+1]$. Thus, the approximation $\boldsymbol{p}[i]$ is view–invariant as the continuous form $\boldsymbol{p}_I(t)$.

The arc-length s of a limited continuous trajectory between $[0,t]$ is defined in Equation (1.5.6). For the discrete case, the arc-length is approximated by a finite number N of discrete samples:

$$s^2 = \sum_{i=1}^{N-1} \langle \nabla \boldsymbol{r}[i], \nabla \boldsymbol{r}[i] \rangle \tag{2.3.15}$$

where $\langle \nabla \boldsymbol{r}[i], \nabla \boldsymbol{r}[i] \rangle$ is the Euclidean metric between two discrete points in space. The analytic properties of $\boldsymbol{r}(s)$ are equivalent to the analytic properties of $\boldsymbol{r}(t)$.

Gesture comparison is based mainly on comparing spatial trajectories. Thus, it is critical to find viewpoint invariant descriptions to trajectory motion, independent of time. With the proposed time re-parametrization, the problem complexity is reduced to spatial variables only, although time remains implicitly expressed as a scaling constant of the trajectory sequence.

2.3.4 Viewpoint Invariant Analysis of Position Using the Frenet-Serret Frame

View invariance refers to the problem of analyzing trajectory in space independently of the coordinate reference frame. An reference frame $\mathcal{F}$ determines the position of a point in space. Often, the reference frame location is independent of the point and consequently, extrinsic to the point. A consequence of this relativistic description is that there is no single description which is above any other. Thus, the point evolution, or trajectory $\boldsymbol{r}$, is not invariant to distinct external reference frames.

The only description of a particle's trajectory that is invariant for any location in space is one obtained by setting a reference frame $\mathcal{F}$ within the particle. Since the reference frame "travels" with the particle it is said that it is local to the particle or intrinsic. Any set of transformations changing the position and/or orientation of the particle becomes relative to this frame. Within any given sequence of transformations, the trajectory is fully determined by following the transformation sequence back and forth. Any comparison between such descriptions is invariant. The disadvantage of using intrinsic frames is that a particular trajectory $\boldsymbol{r}(s)$ has to be traveled recursively from the beginning to the end with the intrinsic frame.

Frenet-Serret frames map a global reference frame $\mathcal{F}_0$ into a local frame $\mathcal{F}_\ell$. The matrix representation of Equation (1.5.7) uses the curvature $\kappa(s)$ and torsion $\tau(s)$, and it shows how the dimensionality is reduced to only two parameters.

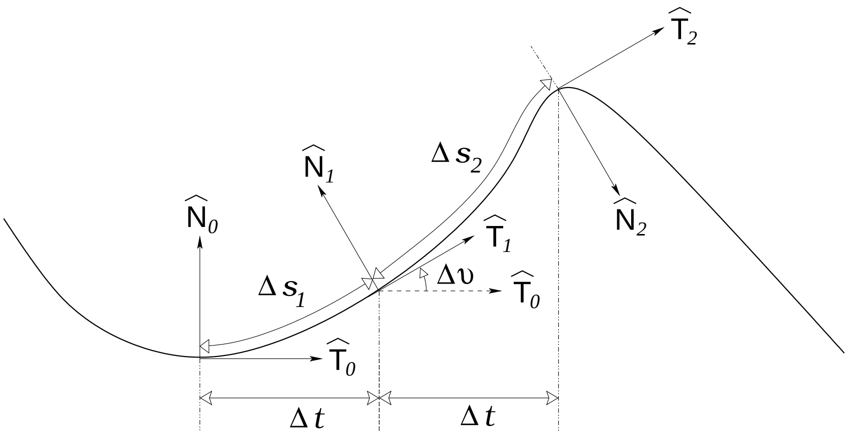


Figure 2.4: A Planar perspective of a curve for geometrical interpretation of the curvature κ.

As it is seen at Figure (2.4), the tangent vector $\widehat{\mathsf{T}}$ and the normal vector $\widehat{\mathsf{N}}$ are characteristics of a planar curve. As the curve is sampled from an extrinsic point of view, represented by Δt, the arc-length of the individual sections Δs_i may differ between them. A linear approximation to the curvature $\kappa[i]$ is defined by the rate:

$$\kappa[i] \approx \frac{\Delta\upsilon_i}{\Delta s_i} = \frac{1}{\Delta s_i} \arccos\left(\widehat{\mathsf{T}}_i \cdot \widehat{\mathsf{T}}_{i-1}\right). \tag{2.3.16}$$

The inverse of the curvature, the radius of curvature $r_\kappa[i]$ can be seen as the distance to the intersecting point between $\widehat{\mathsf{N}}_i$ and $\widehat{\mathsf{N}}_{i-1}$. When $\widehat{\mathsf{T}}_i$ and $\widehat{\mathsf{T}}_{i-1}$ are parallel, the intersection point occurs at the infinity and thus the curvature is zero, the opposite is also true. This situation causes an singularity in $\kappa[i]$.

Similarly to Equation (2.3.17), the approximation to the torsion is

$$\tau[i] \approx \frac{1}{\Delta s_i^2} \frac{\arcsin\left(\widehat{\mathsf{N}}_i - \widehat{\mathsf{N}}_i \cdot \widehat{\mathsf{N}}_{i-1}\right)}{\arccos\left(\widehat{\mathsf{T}}_i \cdot \widehat{\mathsf{T}}_{i-1}\right)}, \tag{2.3.17}$$

which is a rotation rate about $\widehat{\mathsf{T}}_i$.

The outer product $\hat{\mathsf{T}} \times \hat{\mathsf{N}}$ defines a bi-osculating plane on which the curve is moving, and the outer product $\hat{\mathsf{T}} \times \hat{\mathsf{B}}$ defines a rate at which the curve $\boldsymbol{r}(s)$ leaves the plane $\hat{\mathsf{T}} \times \hat{\mathsf{N}}$ as shown at Figure (2.5).

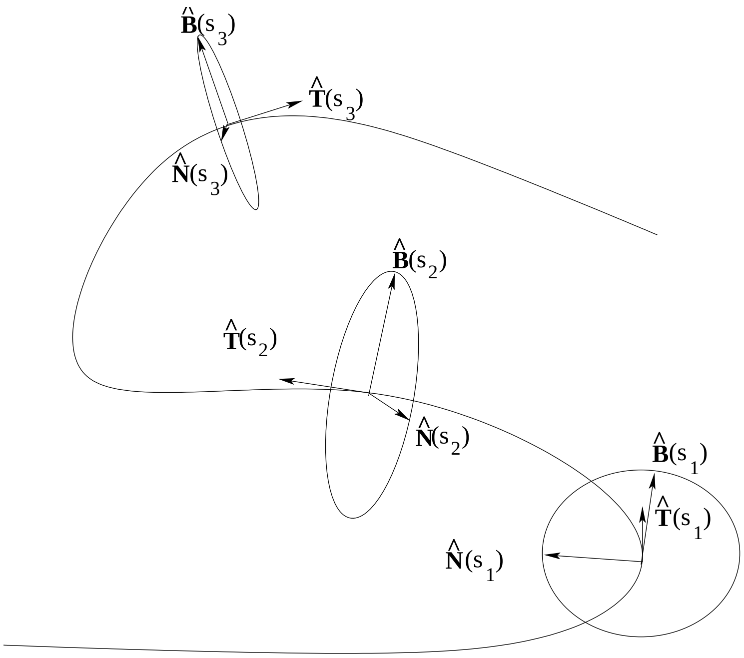


Figure 2.5: Frenet-Serret Frames are characterized by two parameters: curvature $\kappa(t)$ and torsion $\tau(t)$. The vectors $\hat{\mathsf{T}}, \hat{\mathsf{N}}$ and $\hat{\mathsf{B}}$ are orthonormal, *i.e.* $\|\hat{\mathsf{T}}\| = \|\hat{\mathsf{N}}\| = \|\hat{\mathsf{B}}\| = 1$ and $\hat{\mathsf{T}} \cdot \hat{\mathsf{N}} = 0$, $\hat{\mathsf{T}} \cdot \hat{\mathsf{B}} = 0$ and $\hat{\mathsf{N}} \cdot \hat{\mathsf{B}} = 0$

The name bi-osculating coordinate system is based on the fact that both parameters, curvature $\kappa(s)$ and torsion $\tau(s)$, are inverse values of radius distances, namely curvature radius r_κ and torsion radius r_τ. The curvature radius is circumscribed to the plane $\hat{\mathsf{T}} \times \hat{\mathsf{N}}$ while the torsion radius is circumscribed to the orthogonal plane $\hat{\mathsf{T}} \times \hat{\mathsf{B}}$.

2.3.5 Advantages and Disadvantages of Frenet-Serret Frames

View invariance and dimensionality reduction are the main advantages of using Frenet-Serret frames for trajectory descriptions. The reduction of the dimensionality is done by swapping extrinsic reference frames for intrinsic frames to the curve $\boldsymbol{r}(s)$. The descriptive space is a $(n-1)$-dimensional orthonormal space. Such reference frames are invariant to how the n-dimensional curve is embedded in the space.

However, although the curve description is invariant, the associated costs may offset the advantages. Intrinsic frames require smoothness from the trajectories. As expressed by the generalization of Frenet-Serret frames to higher dimensions in Appendix (A.1), the local frame is described in terms outer products of the (n)-order derivative. For a two–dimensional curve, the corresponding Frenet-Serret frame requires the second spatial derivative, for a three–dimensional curve, the associated Frenet-Serret frame requires the third spatial derivative and so on.

Although higher derivatives may not be an issue as a theoretical description, for practical applications such as, real-time filtering, and comparison, the numerical computation are challenging at best due to the effect of round off errors as noise in the approximation.

2.3.6 Numerical Computation of Curvature and Torsion

Curvature and torsion in $3D$ are approximated from the data set derivatives with respect to the arc-length, $\boldsymbol{r}', \boldsymbol{r}''$ and $\boldsymbol{r}'''$ using outer products:

$$\begin{aligned} \kappa &= |\boldsymbol{r}' \wedge \boldsymbol{r}''|, \\ \tau &= \frac{|\boldsymbol{r}' \wedge \boldsymbol{r}'' \wedge \boldsymbol{r}'''|}{|\boldsymbol{r}' \wedge \boldsymbol{r}''|}. \end{aligned} \tag{2.3.18}$$

Numerical approximations of the derivative vectors $\boldsymbol{r}', \boldsymbol{r}''$ and $\boldsymbol{r}'''$ can be done by either convolving nonlinear interpolation filters [170],[177] or using finite-difference methods [107]. In either case, the assumption is that the curve's sampling rate is approximately constant or varies smoothly:

$$\delta s \approx c. \tag{2.3.19}$$

Mapping a constant sampling interval from time to distance is more or less guaranteed in Euclidean spaces due to the orthogonality of the space.

Still, when the space geometry is such that the measure for distance is not linearly constant, the method for approximating the derivatives must be compensated or it will suffer from distortion [107], [155]. In following sections, we analyze the respective cases for convolution and finite-difference methods with non-equidistant samples.

2.3.6.1 Convolution Method for Spatial Derivative Approximations

2.3.6.1.1 The Signal

The key idea behind a convolution filter is that an observed n-dimensional signal $\boldsymbol{x}[t]$, $n \times 1$ can be approximated as a linear combination of locally smooth functions $\boldsymbol{B}\,\boldsymbol{y}[t]$ plus and error ε:

$$\boldsymbol{x}[t] = \boldsymbol{B}\boldsymbol{y}[t] + \varepsilon. \tag{2.3.20}$$

In general, Equation (2.3.20) is an over-determined system and thus its design matrix $\boldsymbol{B}$ is of size $m \times n$ and $\boldsymbol{y}[t]$ is a vector of size $m \times 1$. The true signal $\boldsymbol{y}[t]$ is approximately the expected value of the observed signal $\boldsymbol{x}[t]$, *i.e.*

$$\boldsymbol{y}[t] \approx \bar{\boldsymbol{x}}[t] \equiv \mathscr{E}\left[\boldsymbol{x}[t]\right] \tag{2.3.21}$$

The expected value $\bar{\boldsymbol{x}}[t]$ takes place within a window with support $2\mathsf{w}+1$ and $\mathsf{w} \in \mathbb{Z}$. The error ε is considered of Gaussian nature with zero mean and σ^2 variance, *i.e.* $\varepsilon = \mathcal{N}(0, \sigma^2)$.

The smoothing filter is obtained from solving the signal model in Equation (2.3.20):

$$\boldsymbol{y}[t] = \left(\boldsymbol{B}^\mathsf{T}\boldsymbol{B}\right)^{-1}\boldsymbol{B}^\mathsf{T}\,\boldsymbol{x}[t] + \zeta \tag{2.3.22}$$

for which it is expected that $\zeta << \varepsilon$.

2.3.6.1.2 The Noise

A tacit notion in convolution filtering is that the signal-to-noise ratio (SNR) must be greater than a constant k. The SNR is defined for a discrete signal $x[t]$:

$$\mathrm{SNR} = k\,\log_{10}\left(\frac{|x[t]|}{\varepsilon}\right) \tag{2.3.23}$$

Since practical convolution is calculated using a binary representation, the SNR includes the noise from the binary quantization $k = \mathit{20}\,\log_{10} 2 \times (\#\text{bits})$.

The error $\boldsymbol{\varepsilon}$ is also defined as an average:

$$\boldsymbol{\varepsilon}^2 = \frac{1}{T_0} \int_{-\frac{1}{2}T_0}^{\frac{1}{2}T_0} (x[t] - \bar{x}[t])^2 dt \quad \leq \sigma^2. \tag{2.3.24}$$

When SNR $\leq k$ then the filter is inadequate since the noise level borderlines a Wiener estimation problem.

2.3.6.1.3 Savitzky-Golay Filter

A Savitzky-Golay (SG) filter [170] is a moving averaging filter or low pass filter. The SG filter smooths $\boldsymbol{x}[t]$ while it preserves the signal's higher statistical moments. Higher statistical moments are preserved because the SG filter employs a local fit of a high order polynomial [170], [177]. A high order polynomial ensures that the smoothed output $\boldsymbol{y}[t]$ is $\mathcal{C}^{(n)}$ continuous for the local neighborhood. Also because the SG filter fits a local high order polynomial, it can be used to compute time derivatives.

In classic digital signal processing, it is better to express a digital filter in terms of the z-transform. The z-transform defines z is a complex number that lies in the unit circle of the complex plane:

$$z = \mathrm{e}^{\,\mathrm{j}\frac{\omega}{2\mathrm{w}+1}} \quad \forall \quad \|z\| = 1. \tag{2.3.25}$$

which is the discrete Fourier frequency domain. ω has a periodic range, $[0, 2\pi n)$ $\forall$ $n \in \mathbb{Z}$, that represents the discrete Fourier frequency domain which limits at the Nyquist sample rate f_N:

$$\omega = 2\pi n f_N. \tag{2.3.26}$$

Consequently, the z-transform is unique because integer powers of z are cyclic by definition:

$$\mathrm{e}^{\,\pm\mathrm{j}\frac{1}{2\mathrm{w}+1}\omega} = \mathrm{e}^{\,\pm\mathrm{j}\frac{2}{2\mathrm{w}+1}\omega} = \cdots = \mathrm{e}^{\,\pm\mathrm{j}\frac{(n-1)}{2\mathrm{w}+1}\omega} = \mathrm{e}^{\,\pm\mathrm{j}\frac{n}{2\mathrm{w}+1}\omega} \quad \forall \quad n \in \mathbb{Z} \tag{2.3.27}$$

The smoothed signal $\boldsymbol{y}[z]$ is expressed with the smoothing model in Equation (2.3.22) but dependent on the discrete z polynomial as:

$$\boldsymbol{y}[z] = \boldsymbol{W}\boldsymbol{x}[z] + \boldsymbol{\zeta} \tag{2.3.28}$$

where $\boldsymbol{W}$ is the smoothing weight matrix, $\boldsymbol{x}[z]$ is the polluted input expressed in terms of the z-transform.

Since one can consider the spatial components in Equation (2.3.22) as independent, one can express the SG filter for a single component $y[t]$ and then generalize it for the rest. The expected signal value $y[t_0]$ is computed from a linear combination of weights for window of size $(2\mathsf{w}+1)$ where $\mathsf{w} \in \mathbb{Z}$ such that

$$\boldsymbol{w} = \left\{ \frac{\delta[i]}{\sqrt{2\mathsf{w}+1}} \right\} \quad i = \{-\mathsf{w}, -\mathsf{w}+1, \ldots, 0, \ldots, \mathsf{w}-1, \mathsf{w}\} \tag{2.3.29}$$

where $\delta[j]$ is an impulse function, a.k.a. dirac function. The dirac function is simply defined as:

$$\delta[i] = \begin{cases} 0 & \text{at } i \neq j. \\ 1 & \text{otherwise.} \end{cases} \tag{2.3.30}$$

The sequential structure of the vector $\boldsymbol{w}$ in Equation (2.3.29) is expressed as a z-transform polynomial:

$$\begin{aligned} \boldsymbol{w}[z] = \frac{1}{\sqrt{2\mathsf{w}+1}} \big(& a_{-\mathsf{w}} z^{-\mathsf{w}} + a_{-(\mathsf{w}-1)} z^{-(\mathsf{w}-1)} + \cdots \\ & + a_0 z^0 + \cdots + a_{\mathsf{w}-1} z^{\mathsf{w}-1} + a_{\mathsf{w}} z^{\mathsf{w}} \big) . \end{aligned} \tag{2.3.31}$$

And the n-order derivative in Equation (2.3.31) is simply:

$$\begin{aligned} \boldsymbol{w}^{(n)}[z] = \frac{1}{\sqrt{2\mathsf{w}+1}} \Big(& \left(a_{-\mathsf{w}}(-1)^n \mathsf{w}^n\right) z^{-\mathsf{w}} + \left(a_{-(\mathsf{w}-1)}(-1)^n(\mathsf{w}-1)^n\right) z^{-(\mathsf{w}-1)} + \cdots \\ & + 0 + \cdots + \left(a_{\mathsf{w}-1}(\mathsf{w}-1)^n\right) z^{\mathsf{w}-1} + \left(a_{\mathsf{w}}\mathsf{w}^n\right) z^{\mathsf{w}} \Big) . \end{aligned} \tag{2.3.32}$$

A Jacobian matrix $\boldsymbol{J}$ is the $\boldsymbol{B}$ term in signal model, Equation (2.3.20). $\boldsymbol{J}$ is fully represented with a Vandermonde matrix:

$$\boldsymbol{J}[z] = \begin{bmatrix} (-1)^0\mathsf{w}^0, & \cdots & (-1)^{n-1}\mathsf{w}^{n-1}, & (-1)^n\mathsf{w}^n \\ (-1)^0(\mathsf{w}-1)^0, & \cdots & (-1)^{n-1}(\mathsf{w}-1)^{n-1}, & (-1)^n(\mathsf{w}-1)^n \\ \vdots & \cdots & \vdots & \vdots \\ 0^0, & \cdots & 0^{n-1}, & 0^n \\ \vdots & \cdots & \vdots & \vdots \\ (\mathsf{w}-1)^0, & \cdots & (\mathsf{w}-1)^{n-1}, & (\mathsf{w}-1)^n \\ \mathsf{w}^0, & \cdots & \mathsf{w}^{n-1} & \mathsf{w}^n \end{bmatrix} \tag{2.3.33}$$

of size $((n+1) \times (2\mathsf{w}+1))$. The maximum derivative order (n) is determined by the window size w which is approximately $(n) \approx \left\lfloor \frac{\mathsf{w}}{2} \right\rfloor$.

It follows from the smoothing model, Equation (2.3.22), that the polynomial fit matrix $\boldsymbol{W}$ is:

$$\boldsymbol{W} = \left(\left(\boldsymbol{J}^\mathsf{T}\boldsymbol{J}\right)^{-1} \boldsymbol{J} \right), \tag{2.3.34}$$

and thus the smoothing for the SG filter is:

$$\boldsymbol{y}[z] = \left(\left(\boldsymbol{J}^\mathsf{T}\boldsymbol{J}\right)^{-1} \boldsymbol{J} \right) \boldsymbol{x}[z] + \zeta. \tag{2.3.35}$$

2.3.6.1.4 Computing Derivatives With a SG Filter

The partial solution of the weighting matrix $\boldsymbol{W}$ for the SG filter, Equation (2.3.34), contains a complementary set of difference filters from the Jacobian matrix $\boldsymbol{J}$ in Equation (2.3.33), *i.e.*

$$\boldsymbol{G} = \boldsymbol{J}^{\mathsf{T}}\boldsymbol{J}. \tag{2.3.36}$$

$\boldsymbol{G}$ is a square matrix of size $(2\mathsf{w}+1) \times (2\mathsf{w}+1)$ since it arises from an overdetermined system. It is possible to obtain $\boldsymbol{G}^{-1}$ with several methods however, it is possible to obtain a geometric interpretation by using QR decomposition [177], [155]:

$$\boldsymbol{G} = \boldsymbol{Q}\boldsymbol{R} \tag{2.3.37}$$

where $\boldsymbol{Q}^{\mathsf{T}}\boldsymbol{Q} = \boldsymbol{I}$ and $\boldsymbol{R}$ is upper triangular, such that the inverse is

$$\boldsymbol{G}^{-1} = \boldsymbol{R}^{-1}\boldsymbol{Q}^{\mathsf{T}}. \tag{2.3.38}$$

A more detailed geometrical interpretation can be seen at Appendix (A.2).

The SG filter is not an optimal filter. Although, it is the filter with the best characteristics when it is necessary to preserve higher statistical moments in a smoothed signal $\boldsymbol{y}[z]$. The amount of distortion caused by the SG filter against the amount of improvement in the signal-to-noise ratio depends in two main factors: the polynomial degree and the window size $(2\mathsf{w}+1)$.

Three general recommendations are given in order to improve the SNR and to reduce distortion in the SG filter. The first is considered critical. It consists in adjusting the window size according to the data set $\mathbf{x}$. The second involves choosing the polynomial degree in order to consider higher or lower statistical moments during filtering. However, the polynomial order is restricted by the window size, a.k.a window support. The third and final recommendation involves the SG filter support itself. Both small and large window sizes for the SG filter are not recommended. Noise is pervasive in SG filters with small window sizes and low order polynomials. Small window size causes greater inaccuracies and lower SNR. However, large window sizes may also cause problems. The numerical stability of the SG filter is compromised for larger window sizes due to numerical overflow with high order polynomials. While increasing computation times are because the SNR improvement is asymptotic for larger window sizes [155].

The above recommendations highlight the dependencies of the SG filter to the data set $\mathbf{x}$. Different data sets such as $\mathbf{x}_j$ and $\mathbf{x}_k$ may require specific SG filters adapted for each data set. This is particularly important for multi-scale smoothing because it shows that the filters are to be designed accordingly to the data set at hand [47]. However, the optimal window size is commonly found by empirically testing the data sets $\mathbf{x}$. Some theoretical approximations to the optimal size have been done by comparing the filter with Lagrange polynomials [155].

As for the application of convolution filters to find high order derivatives for curvature and torsion analysis, one can conclude that although advantageous from the online implementation perspective, SG filters present serious limitations. According to Persson [155], third order derivative filter, as required by the Frenet-Serret frames, requires a polynomial degree of at least fourth order with a minimum window size of twenty five samples. However, the theoretical relationship is asymptotic, so SG filter with a fifth order polynomial with forty-nine samples is needed. For small sample data-sets this proves costly in terms of numerical stability and signal tail filtering.

2.3.6.2 Numerical Approximations for Computing Spatial Derivatives using Finite Differences

Finite-difference methods are used to compute time and spatial derivatives. These methods are favored because higher-order derivatives can be easily approximated by recursion. There are three main forms commonly used: forward, backward, and central differences. The functional forms in the one-dimensional case are:

$$\begin{array}{l}\text{Forward differences:}\\ \qquad \Delta\left[f(u[i])\right] = f(u[i+h]) - f(u[i])\\ \text{Backward differences:}\\ \qquad \nabla\left[f(u[i])\right] = f(u[i]) - f(u[i-h])\\ \text{Central differences:}\\ \qquad D\left[f(u[i])\right] = f\left(u\left[i+\tfrac{1}{2}h\right]\right) - f\left(u\left[i-\tfrac{1}{2}h\right]\right)\end{array} \tag{2.3.39}$$

The central differences method is the most unbiased method for approximating the derivative. Mistakenly, it is usual to associate the difference order to derivative order, *i.e.*

$$\ddot{f}(u[i]) = \frac{1}{h}D^{(2)}\left[f(u[i])\right] + \epsilon. \tag{2.3.40}$$

Although, the difference order is independent from the derivative order. Intuitively, any derivative order can be approximated with a single finite-difference of the same order or a linear combination of irreducible low-order differences, *i.e.*

$$D^{(n)}\left[f(u[i])\right] = \prod_{i=1}^{n-1}\left(f\left(u\left[i+\tfrac{1}{2}h\right]\right) - f\left(u\left[i-\tfrac{1}{2}h\right]\right)\right). \tag{2.3.41}$$

The choice of a particular difference order for a derivative approximation is based on two competing conditions: accuracy and computation time. Higher accuracy requires longer computational time while faster computational time requires lesser approximation accuracy.

However, the difference order is just not enough to guarantee higher accuracy for the computation of the derivative. The main factor is the interval h. Most finite difference methods assume that it is possible to approximate the derivative using a constant difference. However, in the case in which is not possible to establish a constant interleave for a mesh, the practical solution is to decrease the interleave space h, even with simple differential equations. A large mesh works against the computational speed, where computational complexities of $\boldsymbol{O}(n^3)$ are not unheard. This situation favors other methods such as finite-elements which reduce the mesh size, but include other factors such as the finite-element geometry.

Another way to increase the computational speed with finite-difference methods is to use non-equidistant meshes. Non-equidistant meshes compensate the finite-difference polynomial for the h parameter, Equation (2.3.39) [107]. The polynomial compensation assumes that higher derivatives can be obtained using recursion and consequently, only formulas for the first and second derivative are available. An example are the central difference expressions for the first derivative with second order accuracy with four point support [107]:

$$u_x(i) = \frac{2(au_{i-2} + bu_{i-1} + cu_i + du_{i+1})}{(\Delta x_i + \Delta x_{i+1})} \tag{2.3.42}$$

a, b, c and d are constants that depend on the ratio g_i between two grid values: Δx_i and Δx_{i+1}:

$$\mathrm{g}_i = \frac{\Delta x_{i+1}}{\Delta x_i} \tag{2.3.43}$$

the values for the constants are obtained by solving a linear system:

$$\begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ -\left(1 + \dfrac{1}{\mathrm{g}_{i-1}}\right) & -1 & 0 & \mathrm{g}_i \\ \left(1 + \dfrac{1}{\mathrm{g}_{i-1}}\right)^2 & 1 & 0 & \mathrm{g}_i^2 \\ 0 & 0 & 0 & 0 \end{bmatrix}^{-1} \begin{bmatrix} 0 \\ \frac{1}{2}\left(1 + \mathrm{g}_i\right) \\ 0 \\ 0 \end{bmatrix} \tag{2.3.44}$$

Approximations to second order derivatives with second order accuracy have more complex equations.

The issue at hand is how to accurately calculate adequate differentials for curvature κ and torsion τ. Combinations of first and second derivative approximations using finite differences use a recurrence such that

$$f'''(u[i]) = f'(f''(u[i]) + \epsilon_2) + \epsilon_1. \tag{2.3.45}$$

that yields a multiplicative error ϵ that depends on the approximation accuracy of previous steps. Given the sensitive nature of the torsion parameter τ, approximations using finite differences in spaces other than Euclidean is hard and difficult.

2.4 Analysis of Orientation Alone

Characterizing the evolution of the orientation $\boldsymbol{\theta}[i]$ is difficult. The reason is the lack of a intrinsic metric in terms of frames and geometry. Euclidean spaces provide simple intrinsic measures with the internal product of the difference between two points. Orientations on the other hand do not have a simple metric, it depends on the representation. For the above reasons, gesture is commonly analyzed using only position, leaving the orientation as a secondary parameter.

Traditionally, rotation sequences are represented as state-space vectors $\boldsymbol{\theta}$ containing Euler angles, namely the Euler state-space vector:

$$\boldsymbol{\theta} = [\phi, \psi, \varphi]^{\mathsf{T}}. \tag{2.4.1}$$

Unlike Euclidean spaces, there is no distance for such a vector. The Euler state-vector $\boldsymbol{\theta}$ does not represent a metric space as the position state-vector $\boldsymbol{r}$ does in an Euclidean space. Consequently, concepts such as curvature κ and torsion τ do not extend to the analysis of orientation trajectories.

In gesture comparison using orientation, the assumption is that the direction for each directed point in the directed set $\overrightarrow{\boldsymbol{\mathsf{x}}}$ has a different direction than the global reference $\overrightarrow{\boldsymbol{p}}_0$, *i.e.*

$$\mathbf{U}_i \neq \mathbf{U}_0 \ \ \forall \ \ i. \tag{2.4.2}$$

Consequently, Equation (2.3.9) remains as it is. For convenience it is reproduced here:

$$\left.\begin{aligned} \overrightarrow{\boldsymbol{x}}[0] &= \mathbf{R}^{i}_{0}\mathbf{U}_0\boldsymbol{x}[i] \\ \overrightarrow{\boldsymbol{x}}[i] &= \mathbf{R}^{0}_{i}\mathbf{U}_i\boldsymbol{x}[0] \end{aligned}\right\} \text{for } i = \{1, \cdots, M\}. \tag{2.4.3}$$

Now, as the position $\boldsymbol{x}$ evolves as function of time t, the direction from the global reference point $\overrightarrow{\boldsymbol{p}}[0]$ to the point $\boldsymbol{x}[i]$ has a specific sequence of transformations in the directional tensor as well:

$$\begin{aligned} \Delta\overrightarrow{\boldsymbol{x}}[0] &= \mathbf{R}^{i}_{0}\mathbf{U}_0\Delta\boldsymbol{x}[i] + \big(\mathbf{R}^{i}_{0}\Delta\mathbf{U}\boldsymbol{x}[i] + \Delta\mathbf{R}^{i}_{0}\mathbf{U}_0\boldsymbol{x}[i]\big) \\ \Delta\overrightarrow{\boldsymbol{x}}[i] &= \mathbf{R}^{0}_{i}\mathbf{U}_i\Delta\boldsymbol{x}[0] + \big(\mathbf{R}^{0}_{i}\Delta\mathbf{U}\boldsymbol{x}[0] + \Delta\mathbf{R}^{0}_{i}\mathbf{U}_i\boldsymbol{x}[0]\big). \end{aligned} \tag{2.4.4}$$

Although Equation (2.4.4) looks ominous, what is really involved in the term $\Delta\mathbf{U}$ is another transformation between the local tensor $\mathbf{U}_i$ and the global reference tensor $\mathbf{U}_0$. Assuming that the tensors in each directed point $\overrightarrow{\boldsymbol{x}}[i]$ are orthonormal and the only difference is a rotation operator, then one can assume that Equation (2.4.4) reverts to:

$$\left.\begin{aligned}\Delta\overrightarrow{\boldsymbol{x}}[0] &= \mathbf{R}^{i}_{0}\mathbf{U}_i\Delta\boldsymbol{x}[i] + \Delta\mathbf{R}^{i}_{0}\mathbf{U}_i\boldsymbol{x}[i]\\ \Delta\overrightarrow{\boldsymbol{x}}[i] &= \mathbf{R}^{0}_{i}\mathbf{U}_0\Delta\boldsymbol{x}[0] + \Delta\mathbf{R}^{0}_{i}\mathbf{U}_0\boldsymbol{x}[0]\end{aligned}\right\}\text{for } i = \{1,\cdots,M\}. \tag{2.4.5}$$

The simplification is now to assume that the change in direction $\Delta\overrightarrow{\boldsymbol{x}}[0]$ due to the change in orientation $\Delta\mathbf{R}^{i}_{0}$. Then Equation (2.4.5) is simply

$$\left.\begin{aligned}\Delta\overrightarrow{\boldsymbol{x}}[0] &= \Delta\mathbf{R}^{i}_{0}\mathbf{U}_i\boldsymbol{x}[i]\\ \Delta\overrightarrow{\boldsymbol{x}}[i] &= \Delta\mathbf{R}^{0}_{i}\mathbf{U}_0\boldsymbol{x}[0]\end{aligned}\right\}\text{for } i = \{1,\cdots,M\}. \tag{2.4.6}$$

In order to characterize rotation sequences with differential geometry, it becomes necessary to define an appropriate metric for the operators $\Delta\mathbf{R}^{i}_{0}$ and $\Delta\mathbf{R}^{i}_{0}$. This is a specially sensitive issue. In the following paragraphs one will discuss how different representation for orientations and rotations are inadequate to define a metric in the orientation space. Although it is possible to define a metric using projections for quaternions that map the trajectories onto a $3D$ volume.

2.4.1 Using Euler Angles for Analysis

There are two main methods in which Euler angles play a key role in determining the orientation of a rigid body in three-dimensional space: rotation matrices and vector axis-angle rotation.

2.4.1.1 Rotation Matrices

A $3D$-space defines three rotation operators $\mathbf{R}_{\boldsymbol{\mathsf{h}}}$, where $\boldsymbol{\mathsf{h}}$ is the set of vectors serving as rotation axes, *i.e.* $\boldsymbol{\mathsf{h}} = \{\vec{i},\vec{j},\vec{k}\}$. Each operator is defined in matrix form as:

$$\begin{gathered}\mathbf{R}_{\vec{k}}(\phi) = \begin{bmatrix}\cos\phi & -\sin\phi & 0\\ \sin\phi & \cos\phi & 0\\ 0 & 0 & 1\end{bmatrix},\quad \mathbf{R}_{\vec{j}}(\psi) = \begin{bmatrix}\cos\psi & 0 & \sin\psi\\ 0 & 1 & 0\\ -\sin\psi & 0 & \cos\psi\end{bmatrix}\\ \mathbf{R}_{\vec{i}}(\varphi) = \begin{bmatrix}1 & 0 & 0\\ 0 & \cos\varphi & -\sin\varphi\\ 0 & \sin\varphi & \cos\varphi\end{bmatrix}.\end{gathered} \tag{2.4.7}$$

The general rotation operator is defined as the product of these three individual operators:

$$\mathbf{R}(\boldsymbol{\theta}) = \mathbf{R}_{\,\mathsf{h}[3]}(\phi)\mathbf{R}_{\,\mathsf{h}[2]}(\psi)\mathbf{R}_{\,\mathsf{h}[1]}(\varphi). \tag{2.4.8}$$

$\boldsymbol{\theta}$ is the Euler state-space vector defined in Equation (2.4.1). The sub-index set h indicates the axis about the individual rotation matrix $\mathbf{R}_{\mathsf{h}}$ operates.

There are combinations with repetition for which the indexes in the set h can be ordered. Each individual combination constitutes what is known as "rotation convention." A rotation convention is the particular order in which the individual rotation operators have to be multiplied in order to obtain the general rotation matrix $\mathbf{R}_{\mathsf{h}}(\boldsymbol{\theta})$. Due to the properties of rotation and linear algebra, the different combinations are not interchangeable. Once a particular conventions has been chosen, further operations should remain using the same convention for consistency. The most notable conventions are the yawl-pitch-roll with axis order $\vec{k}$,$\vec{j}$,$\vec{i}$ for aeronautics and the ZYZ with axis order $\vec{k}$,$\vec{j}$,$\vec{k}$ in robotics.

Rotation operators are orthogonal operators, *i.e.*

$$\mathbf{R}_{\mathsf{h}}^{-1} = \mathbf{R}_{\mathsf{h}}^{\mathsf{T}}, \text{ thus } \mathbf{R}_{\mathsf{h}}^{\mathsf{T}}\mathbf{R}_{\mathsf{h}} = \mathbf{R}_{\mathsf{h}}\mathbf{R}_{\mathsf{h}}^{\mathsf{T}} = \boldsymbol{I}. \tag{2.4.9}$$

The set of rotation operators defined in Equation (2.4.7) form a group, known as rotation group, but they are also known as special orthogonal group in three dimensions or SO3.

The main characteristic of any rotation operator, whether it has a matrix representation or any other is its scalar value. In the case of rotation matrices or tensors, the scalar value is given by its determinant:

$$\det\left(\mathbf{R}_{\mathsf{h}}\right) = 1. \tag{2.4.10}$$

Other representations may define this scalar as a vector norm, but it will always be the unit. This is characteristic of any space or sub-space used to represent rotation operators.

In order to recover the Euler state-space vector $\boldsymbol{\theta}$ from a rotation operator $\mathbf{R}_{\mathsf{h}}$, it is necessary to convert the matrix into a vector-angle representation.

2.4.1.1.1 Rotation Sequences and Rotation Matrices

A rotation sequence is an ordered set of Euler state-space vectors as

$$\uptheta = \{\boldsymbol{\theta}[1], \boldsymbol{\theta}[2], \cdots, \boldsymbol{\theta}[i], \cdots, \boldsymbol{\theta}[M]\} \quad \{i, M\} \in \mathbb{N}. \tag{2.4.11}$$

The rotation sequence $\uptheta$ defines a general rotation matrix, product of the individual rotation matrices in the set, *i.e.*

$$\mathbf{R}_{\mathsf{h}}(\uptheta) \equiv \mathbf{R}_{\mathsf{h}}\big(\boldsymbol{\theta}[1]\big)\mathbf{R}_{\mathsf{h}}\big(\boldsymbol{\theta}[2]\big)\cdots\mathbf{R}_{\mathsf{h}}\big(\boldsymbol{\theta}[i]\big)\cdots\mathbf{R}_{\mathsf{h}}\big(\boldsymbol{\theta}[M]\big). \tag{2.4.12}$$

2.4.1.2 Vector Axis-Angle Rotation

A vector axis-angle rotation is a corollary of the Euler rotation theorem. The Euler theorem states that any rotation or rotation sequence can be reduced to a single operation representing a rotation about a directed axis. The vector axis-angle operator uses the Euler state-vector, Equation (2.4.1), to find a unitary direction vector $\widehat{\boldsymbol{e}}_\ell$ and a scalar ϑ that defines the magnitude of the rotation.

The vector axis-angle operation is defined for an arbitrary point $\boldsymbol{x}$ in the Euclidean space as:

$$\begin{aligned} \vartheta &= \arccos\left(\tfrac{1}{2}(\mathrm{Trace}(\mathbf{R}(\vartheta, \widehat{\boldsymbol{e}}_\ell)) - 1)\right) \\ \widehat{\boldsymbol{e}}_\ell &= \frac{1}{2\sin\vartheta}\left[(R_{3,2} - R_{2,3}), (R_{1,3} - R_{3,1}), (R_{2,1} - R_{1,2})\right]^\mathsf{T} \end{aligned} \tag{2.4.13}$$

where $R_{i,j}$ are the elements of a SO(3) matrix. The unitary direction vector $\widehat{\boldsymbol{e}}_\ell = [e_1, e_2, e_3]^\mathsf{T}$ and the arbitrary point $\boldsymbol{x}$ define together a rotation plane while the scalar ϑ provides the rotation magnitude.

In order to recover the Euler state vector from vector axis-angle operator it is necessary to first reconstruct the rotation matrix $\mathbf{R}(\boldsymbol{\theta})$:

$$\mathbf{R}(\boldsymbol{\theta}) = (1 - \cos\vartheta)(\boldsymbol{e}_\ell\, \boldsymbol{e}_\ell^\mathsf{T}) + \cos\vartheta\, \boldsymbol{I} - \sin\vartheta\, \boldsymbol{S} \tag{2.4.14}$$

where

$$\boldsymbol{S} = \begin{bmatrix} 0 & -e_3 & e_2 \\ e_3 & 0 & -e_1 \\ -e_2 & e_1 & 0 \end{bmatrix} \tag{2.4.15}$$

is the skew-symmetric matrix. The outer product $\boldsymbol{e}_\ell \boldsymbol{e}_\ell^\mathsf{T}$ constructs a plane normal to the rotation axis. The rotation magnitude is given by the term $(1 - \cos\vartheta)$. The rotation magnitude is one half of the orientation increase $\Delta\boldsymbol{\theta}$ at the general reference frame.

2.4.1.3 Advantages and Disadvantages of Each Representation

Rotation matrices are intuitive. They are popular because they describe complex kinematic chains with linear algebra. One can quickly construct a rotation operator from an Euler state-vector, Equation (2.4.1).

However, there are disadvantages in using rotation operators $\mathbf{R}_\mathsf{h}(\boldsymbol{\theta})$ in matrix form for $3D$-gesture comparison. First, the lack of a metric. It is not possible to distinguish a rotation sequence from another in the same form as an Euclidean trajectory. Second, it is not possible to predict when the matrix will loose rank. Rotation sequences may contain unpredictable vanishing points for at least one degree of freedom. And third, rotation matrices do not define interpolation functions between orientations. Although, some applications in computer graphics and in games specially, do define and use linear spherical interpolation for small variations in angle $\delta\boldsymbol{\theta}$ by trading speed for accuracy.

Vector axis-angle rotations seem to be a good solution for tracing the orientation trajectory, but requires greater computation resources. They require to constantly calculate the solution of eigenvalue problem for each Euler state-vector in the rotation sequence. Additionally, the solution of the eigenvalues for rotation matrices commonly yields complex values that are not easily interpreted. This factor combined with the duality in direction for a rotation cause ambiguities in the trajectory that are not easily explained or interpreted in the geometrical sense. Finally, since the vector axis-angle rotation is dependent on finding a rotation matrix, the operator also has the same caveats as any rotation matrix. It is no possible to establish a distance function that relates different vector-axes trajectories without imposing sets of heuristic rules.

2.4.2 Quaternions for Orientation Analysis

Quaternions represent orientations and rotations using two vectors and hence the name of bi-vector operators. Essentially, quaternions define an orientation as a linear combination of two irreducible vectors. “Irreducible” is a term related to the orthogonality property of a vector space. As demonstrated by Grassmann [74], any geometric operation that uses two n-dimensional vectors in its definition, implicitly defines a hyper-plane, (see Appendix (A.3)).

In order to show how magnitude–constrained quaternions represent individual orientations and rotation operations, one can use an analogy to point positions and point translations in the Euclidean space. A quaternion defines an orientation as a point in its $4D$ scalar field while a translation of that point in the quaternion scalar field is analogous to a rotation operator. A quaternion is, actually, a plane embedded in a $3D$ Euclidean space:

$$\begin{aligned} &\text{vector form:} && \boldsymbol{n}(\boldsymbol{r}-\boldsymbol{r}_0)=0 \\ &\text{explicit form:} && a\mathrm{x}+b\mathrm{y}+c\mathrm{z}+d=0 \end{aligned} \tag{2.4.16}$$

Equations (2.4.16) are the definition of a plane from its normal vector $\boldsymbol{n}$, the intersecting point between the normal and the plane $\boldsymbol{r}_0$ and the distance from $\boldsymbol{r}_0$ to any point on its surface $\boldsymbol{r}-\boldsymbol{r}_0$.

In a magnitude constrained quaternion the norm has a fixed magnitude. The implication is that the value of the norm is shared between the magnitude of the normal vector defining the plane and the plane area. As such, the normal vector stops being always unitary. At certain directional values, the norm of the normal may be zero or one and inversely, the scalar representing the area of the plane. One can think of a unit quaternion as a shrinking or expanding vector or viceversa, as a expanding or shrinking plane, in that order.

The quaternion rotation operator is defined in terms of the differences of such a variable vector-area planes. However, although the above geometrical representation is easy to understand, unit quaternions are multiplicative mathematical entities. It is possible to define differential forms based on linear operations such as addition and substraction, but they have no relation as to measure the distance between two different orientations. Distances with unit quaternions are defined as products, not as differences.

The shortest distance between two orientations in the unit quaternion space is not a straight line but a geodesic. Geodesics are fairly well known and documented [79],[102],[99]. Rotation sequences are represented as curves composed of segment geodesics in the unit quaternion space [62],[125]. Smooth and continuous orientation trajectories are difficult to interpolate. The reason lies on the nonlinear nature of the hyper-sphere. The amount of error introduced by the nonlinear distances often limit the quality of a smooth interpolated trajectory.

As in any other rotation operation, quaternions have a dual representation for orientations and rotation operators in the unit quaternion space. However, the dualities lie on antipodal point of the unit quaternion space. Additionally, the singularities are always predictable, unlike rotation matrices. The key issue is whether or not quaternions are better at representing rotation sequences.

2.4.2.1 Quaternions in Cartesian Space

A quaternions as Cartesian spaces are defined by a four parameter set

$$\boldsymbol{q} = w + x\,\widehat{i} + y\,\widehat{j} + z\,\widehat{k} \tag{2.4.17}$$

where the parameter set $\{w, x, y, z\}$ defines a $\mathbb{R}^4$ space. The set of orthogonal vectors $\{\,\widehat{i}\,, \widehat{j}\,, \widehat{k}\,\}$ defines an orthogonal orientation tensor matrix $\boldsymbol{\Xi}$ such that:

$$\boldsymbol{q} = \boldsymbol{\Xi}[x,\, y,\, z]^{\mathsf{T}} + w. \tag{2.4.18}$$

It is possible to redefine the expression in Equation (2.4.18) into linear algebra terms using homogeneous coordinates:

$$\boldsymbol{q} = \boldsymbol{Q}\boldsymbol{x} \quad \text{where} \quad \boldsymbol{Q} = \begin{bmatrix} \boldsymbol{\Xi} & \boldsymbol{0} \\ \boldsymbol{0}^{\mathsf{T}} & 1 \end{bmatrix} \quad \text{and} \quad \boldsymbol{x} = [x,\, y,\, z,\, w]^{\mathsf{T}} \tag{2.4.19}$$

However, Cartesian quaternions are of little interest for representing rotations. The parametrization of $\boldsymbol{x}$ is independent and thus the continuous functions governing each parameter are independent. Such functions cannot represent rotations. Cartesian coordinates for quaternions are analogous to represent general states in matrix representation.

As in any other rotation operator, quaternions must impose a constraint in their magnitude in order to represent a rotation. This is analogous to state the Special Orientation group SO(3), then $\{\boldsymbol{\Xi} \in \mathrm{SO}(3)$ and $|\boldsymbol{\Xi}|^2 = 1\}$.

2.4.2.2 Polar Representation of Quaternions

Hamilton [97] suggested to decompose a quaternion as a product of a magnitude and a phase components and he suggested two product operators:

$$\boldsymbol{q} = \mathsf{M}\left(\boldsymbol{q}\right) \mathbf{U}\left(\boldsymbol{q}\right) \tag{2.4.20}$$

where the operator $\mathsf{M}\left(\cdot\right)$ is the quaternion norm:

$$\mathsf{M}\left(\boldsymbol{q}\right) = \|\boldsymbol{q}\| = \langle \boldsymbol{q}, \boldsymbol{q} \rangle^{\frac{1}{2}} \tag{2.4.21}$$

and the operator $\mathbf{U}\left(\cdot\right)$ defines the quaternion phase such that:

$$\widehat{\boldsymbol{q}} \equiv \mathbf{U}\left(\boldsymbol{q}\right) = \frac{\boldsymbol{q}}{\mathsf{M}\left(\boldsymbol{q}\right)}, \quad \|\mathbf{U}\left(\boldsymbol{q}\right)\| = 1. \tag{2.4.22}$$

$\widehat{\boldsymbol{q}}$ describes a particular quaternion subset namely, unit quaternions. Unit quaternions $\widehat{\boldsymbol{q}}$ are characterized by:

$$\widehat{\boldsymbol{q}} \subset \mathbb{H} \quad | \quad \mathsf{M}\left(\boldsymbol{q}\right) = 1. \tag{2.4.23}$$

Three–dimensional rotations are completely described by unit quaternions.

The proposed analysis for quaternions is similar to the analysis of complex numbers in polar coordinates. However, in the analysis for quaternions the elements include scalar and vector parts, *i.e.*:

$$\widehat{\boldsymbol{q}} = (\alpha, \boldsymbol{\xi}) \tag{2.4.24}$$

where

$$\alpha = w, \quad \text{and} \quad \boldsymbol{\xi} = \boldsymbol{\Xi}[x, y, z]^{\mathsf{T}}. \tag{2.4.25}$$

where α is the scalar part and $\boldsymbol{\xi}$ is the vector part of the quaternion respectively. By definition, the magnitude of the unit quaternions is constrained to the unit:

$$\|\widehat{\boldsymbol{q}}\| = 1. \tag{2.4.26}$$

The implication is the interdependence in the magnitudes of their parameters, *i.e.*,

$$\begin{aligned} &\text{if} \quad \alpha^2 + \|\boldsymbol{\xi}\|^2 = 1 \\ &\text{then} \\ &\quad \|\boldsymbol{\xi}\| = (1-\alpha^2)^{\frac{1}{2}}. \end{aligned} \tag{2.4.27}$$

The quaternion polar representation arises naturally from the magnitude constraint. Lets assume that magnitude constrained quaternions are represented as:

$$\|\boldsymbol{q}\| = \rho, \quad \forall \quad \rho^2 = \alpha^2 + \|\boldsymbol{\xi}\|^2 \tag{2.4.28}$$

where $\rho \geq 0 \in \mathbb{R}$. Any magnitude constrained quaternion holds to the relationship:

$$\widehat{\boldsymbol{q}} = \frac{\alpha}{|\rho|} + \frac{(\rho^2-\alpha^2)^{\frac{1}{2}}}{|\rho|}\widehat{\boldsymbol{\xi}}. \tag{2.4.29}$$

It is easily shown that both elements, scalar and vectorial, have angular relationships:

$$\left.\begin{aligned} \cos\theta &= \frac{\alpha}{|\rho|} \\ \sin\theta &= \left(1 - \left(\frac{\alpha}{|\rho|}\right)^2\right)^{\frac{1}{2}} \end{aligned}\right\} \quad \forall \quad 0 \leq |\alpha| \leq |\rho|. \tag{2.4.30}$$

Then, for any magnitude–constrained quaternion $\boldsymbol{q}$, the normalized quaternion $\widehat{\boldsymbol{q}}$ in (2.4.29) is analogous to the Euler formula for complex numbers:

$$\widehat{\boldsymbol{q}} = \cos\theta + \widehat{\boldsymbol{\xi}}\sin\theta \equiv \exp(\widehat{\boldsymbol{\xi}}\theta) \tag{2.4.31}$$

where $\widehat{\boldsymbol{\xi}}$ is a normalized direction vector, *i.e.* $\|\widehat{\boldsymbol{\xi}}\| = 1$. From the matrix-vector product in Equation (2.4.18), the relationship between $\boldsymbol{\xi}$ and $\widehat{\boldsymbol{\xi}}$ is such that

$$\begin{aligned} &\boldsymbol{\xi} = \widehat{\boldsymbol{\xi}}\sin\theta \\ &\text{and} \\ &0 \leq \|\boldsymbol{\xi}\| \leq 1 \end{aligned} \tag{2.4.32}$$

The vector $\boldsymbol{\xi}$, sometimes called "versor" [175], is interpreted as a great–circle arc lying on the $4D$ hyper-sphere. The normalized vector $\widehat{\boldsymbol{\xi}}$ is called "right-versor." A "right-versor" is defined as a "pure" quaternion since the magnitude of the scalar part is zero, that is:

$$\widehat{\boldsymbol{\xi}} \equiv \left(0, \frac{1}{\sin\theta}\boldsymbol{\xi}\right) \tag{2.4.33}$$

A pure quaternion occurs for every instance in which $\sin\theta = 1$ or conversely, $\theta = \frac{(2n-1)}{2}\pi$. Thus, the polar representation of a general magnitude-constrained quaternion has the form:

$$\boldsymbol{q} = \|\boldsymbol{q}\| \exp(\widehat{\boldsymbol{\xi}}\,\theta). \tag{2.4.34}$$

As a corollary, a expression for the unit quaternion set is:

$$\widehat{\boldsymbol{q}} = \exp(\widehat{\boldsymbol{\xi}}\,\theta). \tag{2.4.35}$$

2.5 Curve Invariant Analysis in Quaternion Space

In the previous section, we reviewed different representations of $3D$-rotations: Homogeneous matrices, vector-axis angle rotations, and quaternions. Quaternions are probably the most consistent way for rotation representation in $3D$. The reason, as stated before, is their singularities are predictable and there are interpolation functions that use the arc-length of the spherical space. However, quaternion distances are often confused with angular values, but they are not, they are geodesic-arcs. During the following discussion one will review some concepts on how quaternions are used for interpolation and their importance to evaluate orientation trajectories.

Quaternions can be treated exchangeably as vectors or planes:

$$\begin{array}{llll}
\text{scalar set: } \{1\} & & & \\
\text{vector set: } \{\widehat{\boldsymbol{e}}_1, \widehat{\boldsymbol{e}}_2, \widehat{\boldsymbol{e}}_3\} & & & \\
\text{bivector set: } & \{\widehat{i} = \widehat{\boldsymbol{e}}_1\widehat{\boldsymbol{e}}_2, & \widehat{j} = \widehat{\boldsymbol{e}}_2\widehat{\boldsymbol{e}}_3, & \widehat{k} = -\widehat{\boldsymbol{e}}_3\widehat{\boldsymbol{e}}_1\} \\
 & \widehat{i}\,\widehat{j} = \widehat{k} & \widehat{j}\,\widehat{k} = \widehat{i} & \widehat{k}\,\widehat{i} = \widehat{j} \\
 & \widehat{j}\,\widehat{i} = -\widehat{k} & \widehat{k}\,\widehat{j} = -\widehat{i} & \widehat{i}\,\widehat{k} = -\widehat{j} \\
\text{pseudoscalar: } I = \widehat{\boldsymbol{e}}_1\widehat{\boldsymbol{e}}_2\widehat{\boldsymbol{e}}_3 \text{ and } I^2 = -1 & & & \\
 & \widehat{i}^{\,2} = \widehat{j}^{\,2} = \widehat{k}^{\,2} = \widehat{i}\,\widehat{j}\,\widehat{k} = -1 & & .
\end{array} \tag{2.5.1}$$

2.5.1 Three-Dimensional Projection Spaces

The intent of achieving a $3D$ projection for unit quaternions is to use the $3D$ Frenet-Serret framework in an orientation sequence. The unit quaternion subset is a spherical scalar field. Each point on the manifold is represented as:

$$\widehat{\boldsymbol{q}} = [w, x, y, z]^{\mathsf{T}} \quad \text{subject to } \|\widehat{\boldsymbol{q}}\| = 1 \tag{2.5.2}$$

with a common juncture point at

$$\widehat{\boldsymbol{q}}_0 = [0,0,0,0]^{\mathsf{T}} \tag{2.5.3}$$

which is a null space. The geometrical interpretation of the unit quaternion subset is of a spherical manifold embedded in a $4D$-space. Every quaternion has a negative conjugate at an antipodal position representing the same orientation rotation in the opposite direction:

$$\widehat{\boldsymbol{q}} = -\widehat{\boldsymbol{q}}^{*}. \tag{2.5.4}$$

However, this space representation is inadequate for comparison purposes.

Unit quaternions are not linearly independent scalar fields. The scalar part and the vectorial part of any magnitude-constrained quaternion are interdependent as shown by Equation (2.4.31). This parametric interdependence is hierarchical. The angular value θ defined in terms of the scalar component w:

$$\theta = \arctan\left(\frac{\sqrt{1-w^2}}{w}\right), \tag{2.5.5}$$

is used to represent the distance between two different orientations on the $4D$-spherical hyper-surface. The angular value θ is invariant for unit quaternions, but it cannot be applied as distance metric for an orientation sequence. The reason is that the manifold has constant curvature. Consequently, any mapping based on θ hides the true positions of an orientation sequence because it losses two degrees-of-freedom instead of one.

There are two commonly used projective spaces. The first uses a pole-projection, similar to a Mercator projection of a sphere. The second is a tangent space known as ternion space. It projects the unit quaternion onto a spherical volume. Both spaces are analytical with advantages and disadvantages of their own.

2.5.1.1 Natural Logarithm Quaternion Projection

The natural logarithm quaternion space emerges naturally from the quaternion polar representation:

$$\widehat{\boldsymbol{q}} = \exp(\theta\,\widehat{\boldsymbol{\xi}}) \tag{2.5.6}$$

then

$$\ln \widehat{\boldsymbol{q}} = \theta\,\widehat{\boldsymbol{\xi}} \tag{2.5.7}$$

Since unitary quaternions are relative to each other with no global reference, all projections use the initial orientation as reference for a whole sequence, *i.e.* $\widehat{\boldsymbol{q}}_0$ is the initial anchoring point in the space. The distance from

$\widehat{\boldsymbol{q}}_0$ to any other quaternion $\widehat{\boldsymbol{q}}_1$ is given by

$$d_g = \widehat{\boldsymbol{q}}_0^{-1}\widehat{\boldsymbol{q}}_1 \tag{2.5.8}$$

then the derivatives are ratios to the reference quaternion $\widehat{\boldsymbol{q}}_0$:

$$\begin{aligned} \frac{d(\ln\widehat{\boldsymbol{q}})}{\ln\widehat{\boldsymbol{q}}_0} &= \left(\frac{d\widehat{\boldsymbol{\xi}}}{\widehat{\boldsymbol{\xi}}_0} + \frac{d\theta}{\theta_0}\right), \\ \frac{d^2(\ln\widehat{\boldsymbol{q}}) - d(\ln\widehat{\boldsymbol{q}})^2}{\ln\widehat{\boldsymbol{q}}_0} &= \left(\frac{d^2\widehat{\boldsymbol{\xi}} - d\widehat{\boldsymbol{\xi}}^2}{\widehat{\boldsymbol{\xi}}_0^2} + \frac{d^2\theta - d\theta^2}{\theta_0^2}\right), \end{aligned} \tag{2.5.9}$$

See Figure (2.6) for a geometric interpretation.

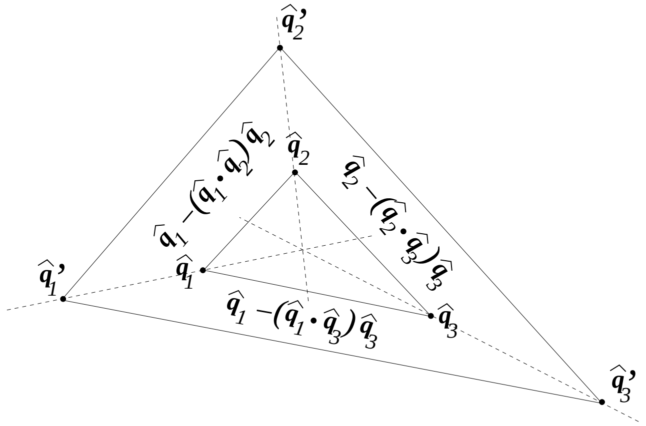


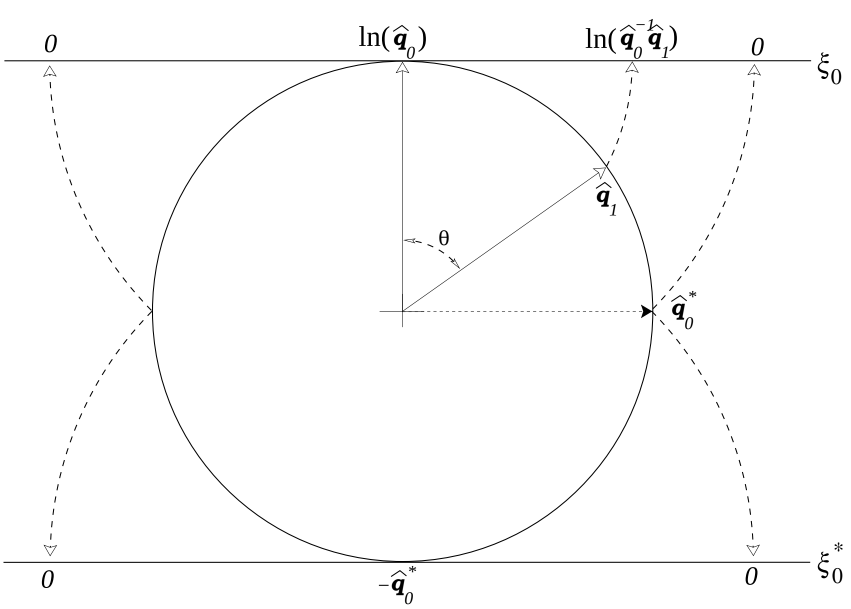
Figure 2.6: Geometric interpretation of differentials in the unit quaternion manifold. Proportional quaternions $\widehat{\boldsymbol{q}}'_1, \widehat{\boldsymbol{q}}'_2$ and $\widehat{\boldsymbol{q}}'_3$ have the same differential as $\widehat{\boldsymbol{q}}_1, \widehat{\boldsymbol{q}}_2$ and $\widehat{\boldsymbol{q}}_3$. The differential in the manifold can be interpreted as rotating the set of quaternions between $\widehat{\boldsymbol{q}}_1$ and $\widehat{\boldsymbol{q}}_2$ pivoting around $\widehat{\boldsymbol{q}}_2$ towards $\widehat{\boldsymbol{q}}_3$.

Higher order derivatives contain greater number of nonlinear elements. And yet, all derivatives are always relative to a point in the space, a global reference is undefined. The remaining higher order derivatives always consider the ratio $\dfrac{d\widehat{\boldsymbol{\xi}}}{\widehat{\boldsymbol{\xi}}}$, which is a vector projection.

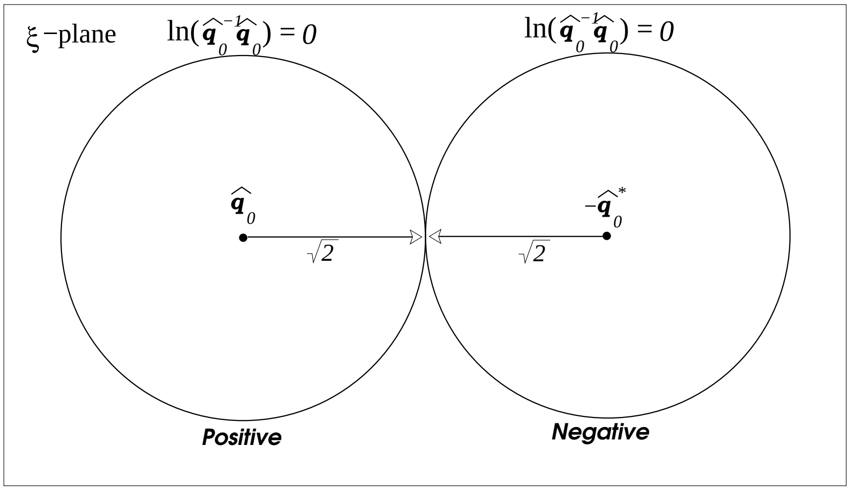
Every quaternion on the surface of $\widehat{\boldsymbol{q}}$ is re-projected on the $3D$-plane $\widehat{\boldsymbol{\xi}}\,\big|_{\widehat{\boldsymbol{q}}_0}$, hereafter one denominate this plane as $\widehat{\boldsymbol{\xi}}$-plane. The $\widehat{\boldsymbol{\xi}}$-plane is relative to the initial position of the sequence, $\widehat{\boldsymbol{q}}_0$. The mapping is similar to a geodesic polar projection, it also introduces distortions. The projection is illustrated at Figure (2.7a)

The antipodal quaternion $-\widehat{\boldsymbol{q}}_0^*$ is represented at Figure (2.7b) as the center point of a second circle. The second circle is a mirror for $\widehat{\boldsymbol{q}}_0$ since

$$\widehat{\boldsymbol{q}}_0 = -\widehat{\boldsymbol{q}}_0^* \tag{2.5.10}$$



(a) Natural logarithm mapping. The basic distance is the product $\widehat{\boldsymbol{q}}_0^{-1}\widehat{\boldsymbol{q}}$. The mapping is relative to the initial quaternion $\widehat{\boldsymbol{q}}_0$



(b) The natural logarithm domain has a radius $\sqrt{2}$. It is bounded for $\theta = (-\pi, \pi)$. For $\theta = \pm\pi$ the mapping degenerates to a scalar equal to zero. The two hemispheres are shown since $\widehat{\boldsymbol{q}}_0 = -\widehat{\boldsymbol{q}}_0^*$

Figure 2.7: Natural logarithm projection. The $\widehat{\boldsymbol{\xi}}_0$-plane is tangent to the reference quaternion $\widehat{\boldsymbol{q}}_0$. The distance from $\widehat{\boldsymbol{q}}_0$ to any other given quaternion $\widehat{\boldsymbol{q}}_1$ is $\ln\left(\widehat{\boldsymbol{q}}_0^{-1}\widehat{\boldsymbol{q}}_1\right)$. However, when $\widehat{\boldsymbol{q}}_1 = \widehat{\boldsymbol{q}}_0^*$ then the mapping degenerates to a scalar equals to zero.

are the same point. One concludes that both the initial $\widehat{\boldsymbol{q}}_0$ and the outer circle are the same points in the mapping. Thus, a magnitude radius $\|\theta\,\widehat{\boldsymbol{\xi}}\| = \sqrt{2}$ in the $\widehat{\boldsymbol{\xi}}_0$-plane represents a 4π rotation in Euler angles.

2.5.1.2 Ternion Projection

A ternions are another projection of quaternions. The natural logarithm rotates every point on the surface of the subset $\widehat{\boldsymbol{q}}$ onto the tangent plane to the reference quaternion $\widehat{\boldsymbol{q}}_0$. The ternion also uses the same $\boldsymbol{\xi}_0$-plane, however the mapping is the intersection point between the radial direction and the $\boldsymbol{\xi}_0$-plane, see Figure(2.8). The projection and its inverse are:

$$\begin{aligned} \boldsymbol{t} &= \mathscr{T}\left(\widehat{\boldsymbol{q}}\right) = \frac{1}{\cos\theta}\widehat{\boldsymbol{q}} - 1, \\ \widehat{\boldsymbol{q}} &= \mathscr{T}^{-1}(\boldsymbol{t}) = \cos\theta\,(\boldsymbol{t}+1). \end{aligned} \tag{2.5.11}$$

The resulting ternion $\boldsymbol{t}$ is not unitary, but its inverse yields a unitary quaternion $\widehat{\boldsymbol{q}}$. The quaternion distance $\widehat{\boldsymbol{q}}_0^{-1}\widehat{\boldsymbol{q}}_1$ maps onto the ternion space as a pole distance:

$$\boldsymbol{t}_0^{-1}\boldsymbol{t}_1 = \mathscr{T}\left(\widehat{\boldsymbol{q}}_0^{-1}\widehat{\boldsymbol{q}}_1\right) = \tan\theta\,\boldsymbol{\xi}. \tag{2.5.12}$$

where $\boldsymbol{\xi} = \boldsymbol{\xi}_1 - \boldsymbol{\xi}_0$ and $\theta = \arccos(\widehat{\boldsymbol{q}}_1 \cdot \widehat{\boldsymbol{q}}_0)$. Ternions are compact representations for $3D$-orientations, only three parameters are necessary.

The main issue of the ternion projection is that the distance function is nonlinear for $\theta > \dfrac{\pi}{4}$ and singular, $\tan\theta = \infty$, for values of $\theta \approx \pm\dfrac{\pi}{2}$. It is easy to see from Figure (2.8a) that the intersection has a distance:

$$\tan\theta = \sqrt{\sec^2\theta - 1}. \tag{2.5.13}$$

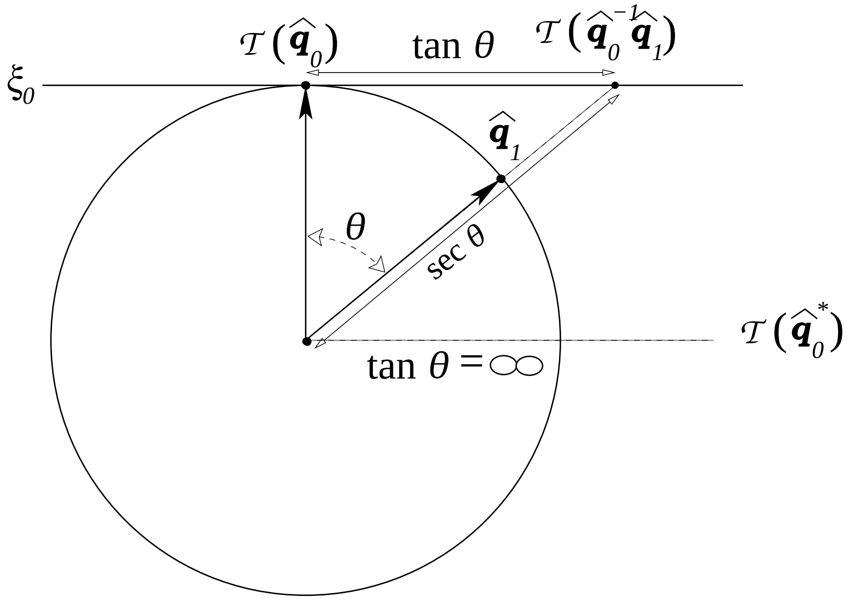
It is also easy to see that as θ approximates $\dfrac{\pi}{2}$, the intersection point in the $\boldsymbol{\xi}_0$-plane occurs at ∞. Consequently, the mapping is unbounded, see Figure (2.8b).

The ternion projection differential has the same form as expressed in the Equation set (2.5.9). However, the distance value for the scalar element has an unbounded domain from $(-\infty, \infty)$ unlike the natural logarithm.

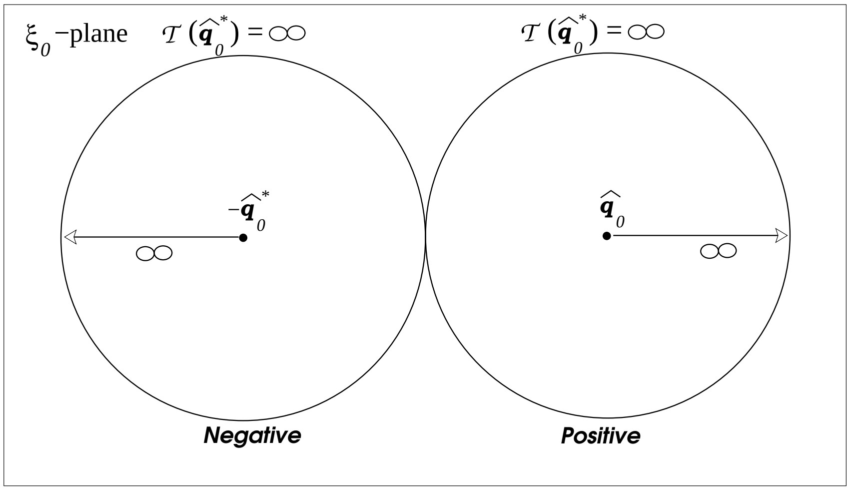
2.5.1.3 Geometric Projection

The quaternion geometric projection is based on the quaternion product. A quaternion $\widehat{\boldsymbol{q}}_1$ can be defined from an initial quaternion $\widehat{\boldsymbol{q}}_0$ with another quaternion $\widehat{\boldsymbol{p}}$ such that

$$\widehat{\boldsymbol{q}}_1 = \widehat{\boldsymbol{p}}^*\widehat{\boldsymbol{q}}_0\widehat{\boldsymbol{p}} = \widehat{\boldsymbol{p}}\widehat{\boldsymbol{q}}_0\widehat{\boldsymbol{p}}^*. \tag{2.5.14}$$



(a) Ternion Mapping



(b) Ternion Domain

Figure 2.8: The ternion mapping defines a pole distance. The pole distance is the tangent function of θ. In this mapping, the $\boldsymbol{\xi}_0$-plane contains all quaternions with the exception of $\widehat{\boldsymbol{q}}_0^*$ which is a singularity.

Since $\widehat{\boldsymbol{p}}$ is located in between $\widehat{\boldsymbol{q}}_0$ and $\widehat{\boldsymbol{q}}_1$, then it has a different angular value θ_p and a tangent plane $\widehat{\boldsymbol{\xi}}_p$:

$$\widehat{\boldsymbol{q}}_1 = (\cos\theta_p + \widehat{\boldsymbol{\xi}}_p \sin\theta_p)\widehat{\boldsymbol{q}}_0(\cos\theta_p - \widehat{\boldsymbol{\xi}}_p \sin\theta_p) \tag{2.5.15}$$

by expanding and canceling terms

$$\begin{aligned} \widehat{\boldsymbol{q}}_1 &= (1 - 2\sin^2\theta_p)\widehat{\boldsymbol{q}}_0 \\ \widehat{\boldsymbol{q}}_1 &= \cos(2\theta_p)\widehat{\boldsymbol{q}}_0 \end{aligned} \tag{2.5.16}$$

then

$$\begin{aligned} \widehat{\boldsymbol{q}}_1\widehat{\boldsymbol{q}}_0^* &= \cos(2\theta_p) \\ \theta_p &= \frac{1}{2}\arccos(\widehat{\boldsymbol{q}}_1\widehat{\boldsymbol{q}}_0^*) \end{aligned} \tag{2.5.17}$$

$$\theta_p = \frac{\theta}{2} \quad \Rightarrow \quad \widehat{\boldsymbol{p}} = \cos\frac{\theta}{2} + \widehat{\boldsymbol{\xi}}_p \sin\frac{\theta}{2}. \tag{2.5.18}$$

Because recursive quaternion rotations will halve the $\widehat{\boldsymbol{q}}_1\widehat{\boldsymbol{q}}_0^*$ geodesic, the recursive rotation is equivalent to propose a geometric series for the geodesic, thus the name of the projection.

It is notable that the direction of the plane $\widehat{\boldsymbol{\xi}}_p$ of the rotating quaternion $\widehat{\boldsymbol{p}}$ is not important, it could be any quaternion lying on a geodesic bisecting the geodesic defined by $\widehat{\boldsymbol{q}}_0$ and $\widehat{\boldsymbol{q}}_1$. For simplicity, the quaternion $\widehat{\boldsymbol{p}}$ is chosen such that it also lies on the geodesic joining $\widehat{\boldsymbol{q}}_0$ and $\widehat{\boldsymbol{q}}_1$. Two orientations can be represented using the initial orientation $\widehat{\boldsymbol{q}}_0$ and the rotating quaternion $\widehat{\boldsymbol{p}}$ using Equation (2.5.14). The geometric representation can be seen at Figure (2.9)

The projection is similar to the ternion projection, except that the geodesic distance from $\widehat{\boldsymbol{q}}_0^*$ to $\widehat{\boldsymbol{q}}_1$ is halved. Consequently, the projected distance from $\widehat{\boldsymbol{q}}_1$ onto the $\widehat{\boldsymbol{\xi}}_0$-plane is reduced by half as well. The new projection domain for a single hemisphere is not bounded $[0, \pm\infty)$. However, the singularity does not occur until the geodesic reaches a value of $-\widehat{\boldsymbol{q}}_0^*$.

2.5.1.4 Geodesic-arc Projection

The geodesic-arc projection is equivalent to re-parameterize the unit quaternion manifold by arc-length or geodesic-arc, see Appendix (A.3.1). This is a contribution in this dissertation and thus it will be reviewed with more detail than previous projective spaces. However, one can use properties of Grassmann algebras and quaternion functions to show intuitive aspects of the projection and avoid many details (see Appendix A.3.2).

The above quaternion mappings use an intrinsic function to the unit circle. All use basic circle relationships to map from n-space to $n-1$ manifold. Unit quaternions are an infinite quaternion subset in a $(n-1)$-manifold embedded

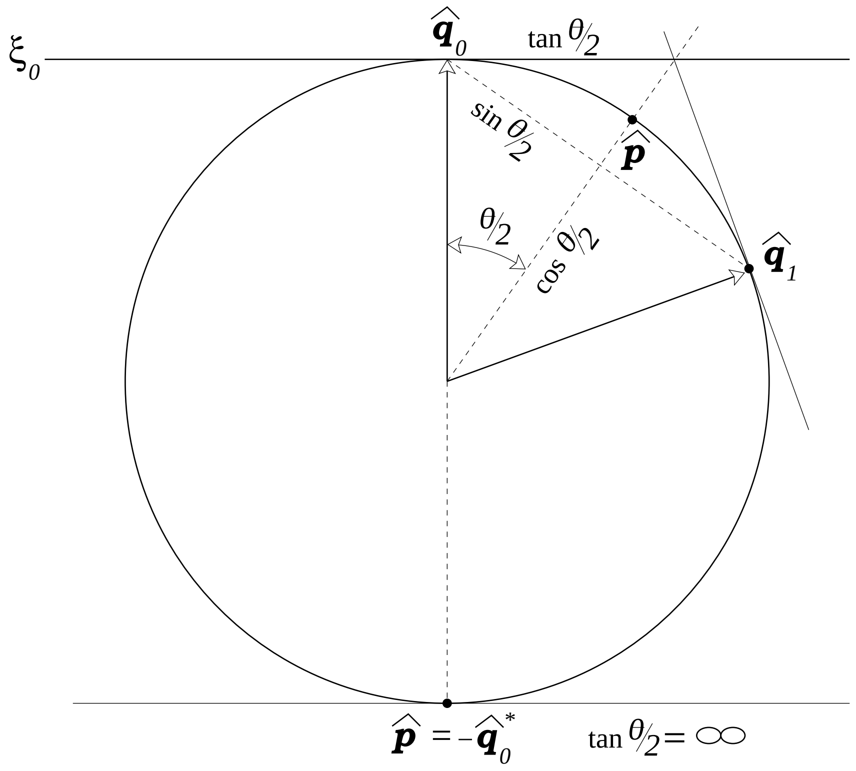


Figure 2.9: The geodesic $\widehat{\boldsymbol{q}}_0^{-1}\widehat{\boldsymbol{q}}_1$ is fully determined by the collinear quaternion $\widehat{\boldsymbol{p}}$. The projection halves its projected distance to $\tan\frac{\theta}{2}$ by using the quaternion rotation.

into a n-space. Any quaternion constrained by $\|\widehat{\boldsymbol{q}}\| = 1$ lie on the surface a sphere embedded in a $4D$-space. This sphere is actually a $3D$-manifold. The most notable properties for all quaternion projections are:

1. *Domain Range*: All projections have a domain range determined by the value of the intersection point to the $\widehat{\boldsymbol{\xi}}$-plane. The domain is unbounded if the intersection point is located at $\pm\infty$ and it is bounded if the intersection point has a limited value. A summary of the domain ranges for each projection can be found in Table (2.1)

2. *Projection singularities*: All quaternion projections have singularities in their domain. Singularities in the space are defined as the unit quaternion values $\widehat{\boldsymbol{q}}_i$ for which the limit:

$$\lim_{\boldsymbol{q}\to\widehat{\boldsymbol{q}}_i} \operatorname{proj} \widehat{\boldsymbol{q}}_i = \infty \tag{2.5.19}$$

   is true.

The specific values for which a singularity exist depends on the reference quaternion $\widehat{\boldsymbol{q}}_0$ and the geometric considerations for the projection. Such singularities are summarized in Table (2.1).

Table 2.1: Singularities for different unit quaternion projections

Projection	Singularities	Domain Range
Natural Logarithm	$\widehat{\boldsymbol{q}}_0^*$	$\ln\left(\widehat{\boldsymbol{q}}_0^{-1}\widehat{\boldsymbol{q}}_0^*\right)$
Ternion $\mathscr{T}(\widehat{\boldsymbol{q}})$	$\widehat{\boldsymbol{q}}_0^*$	$(-\infty,\infty)$
Geometric $\widehat{\boldsymbol{p}}^*\widehat{\boldsymbol{q}}\widehat{\boldsymbol{p}}$	$-\widehat{\boldsymbol{q}}_0^*$	$(-\infty,\infty)$

The cycloid projection is equivalent to state the angular value of θ between two quaternions: $\widehat{\boldsymbol{q}}_0$ and $\widehat{\boldsymbol{q}}_1$ in terms of the arc-length of the sphere. Because the modulus of the unit quaternion set is equal to one $\|\widehat{\boldsymbol{q}}\| = 1$, one may consider the arc-length relationship:

$$\beta\theta = \|\widehat{\boldsymbol{q}}\|\theta = |1|\theta \tag{2.5.20}$$

as an angular value, but it should not. The arc-length, $\beta\theta$, has the same dimensional units as β while the angular value θ is dimensionless. The geometric interpretation of β is a radius of the $4D$–hyper-sphere.

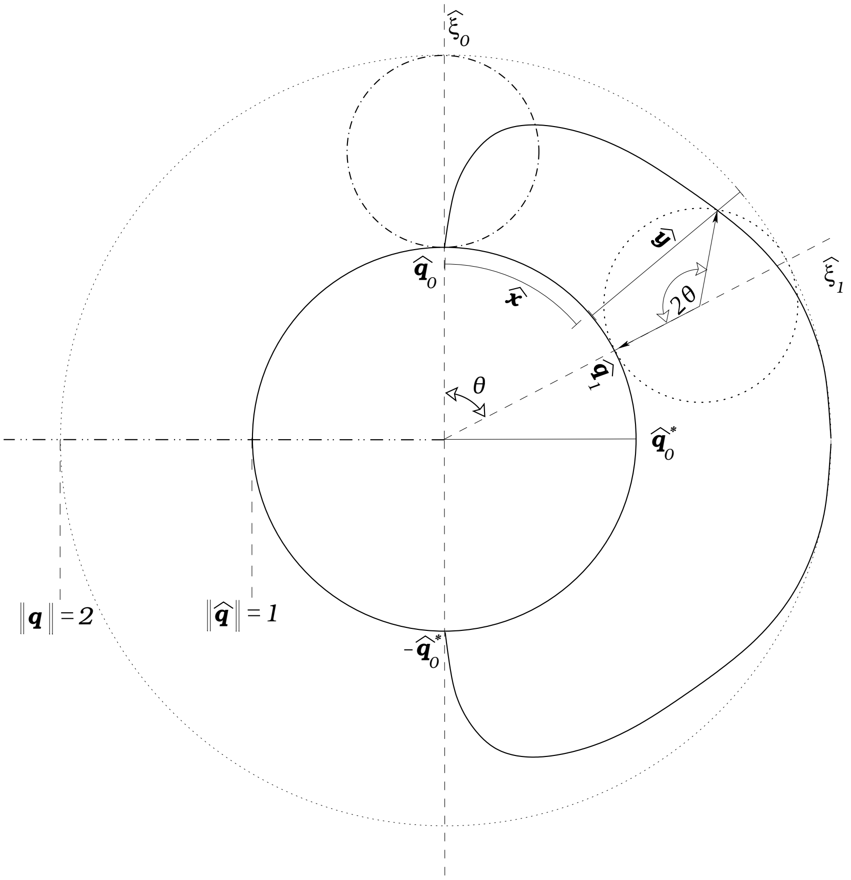
The geometry of the unit quaternion hyper-surface has constant curvature. The manifold distance can be easily measured. The geodesic distance on a sphere can be measured using another geodesic as a pattern. This is the key concept behind the cycloid mapping. One can transform a curved space into another space with an Euclidean metric. The geometric concept is to measure a fixed-position sphere with a planetary sphere moving along the periphery of the first sphere. This is illustrated at Figure (2.10a)

Because a sphere is being used to measure the distance, the distance of the geodesic-arc is implicitly expressed as a ratio of π. The mapping considers two quaternions $\widehat{\boldsymbol{x}}$ and $\widehat{\boldsymbol{y}}$ to indicate directions. We consider every quaternion on $\widehat{\boldsymbol{q}}_0^{-1}\widehat{\boldsymbol{q}}_i$ as a linear combination of these two directional quaternions $\widehat{\boldsymbol{x}}$ and $\widehat{\boldsymbol{y}}_i$

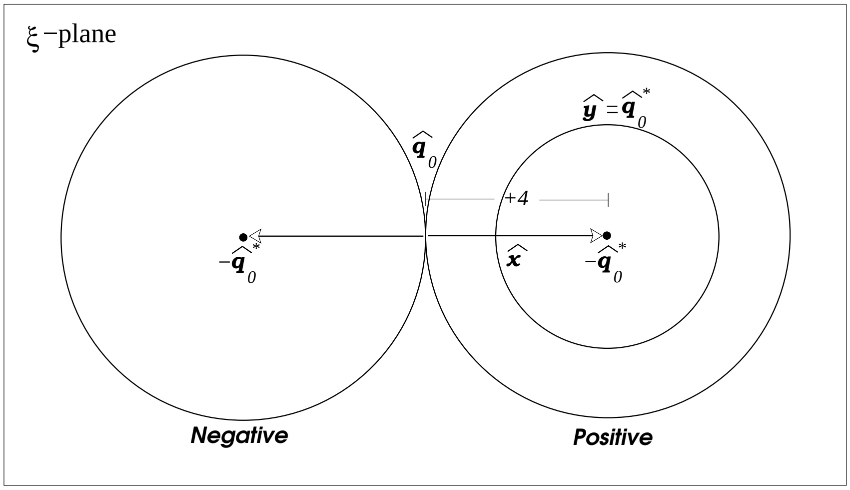
$$\widehat{\boldsymbol{q}}_i = \left(\frac{\|\widehat{\boldsymbol{x}}\|}{\|\widehat{\boldsymbol{x}}\| + \|\widehat{\boldsymbol{y}}_i\|}\widehat{\boldsymbol{x}} + \frac{\|\widehat{\boldsymbol{y}}_i\|}{\|\widehat{\boldsymbol{x}}\| + \|\widehat{\boldsymbol{y}}_i\|}\widehat{\boldsymbol{y}}_i\right)\widehat{\boldsymbol{q}}_0 \tag{2.5.21}$$

where $\widehat{\boldsymbol{x}}$ is a unit quaternion representing an Euclidean axis for a quaternion space. $\widehat{\boldsymbol{y}}$ contains the direction $-\widehat{\boldsymbol{\xi}}_i$ which is the radial direction from the center of the hyper-sphere to the individual quaternion $\widehat{\boldsymbol{q}}_i$ that lies on the geodesic $\widehat{\boldsymbol{q}}_0^{-1}\widehat{\boldsymbol{q}}_i$, see Figure (2.10a).

Explicitly, $\widehat{\boldsymbol{x}}$ always remains in contact with the unit quaternion manifold at only one point. The planetary hyper-sphere is related to each quaternion



(a) Cycloid Transform



(b) Cycloid Domain

Figure 2.10: The cycloid rises from measuring the arc-length of the sphere $|1|\theta$ with another sphere of proportional radius $a\theta$, $(a > 0 \,\forall\, a \in \mathbb{R})$. A ratio $a = \dfrac{1}{2}$ is illustrated.

conjugate $-\widehat{\boldsymbol{q}}_i^*$ since it has opposite direction to its correspondent. One can redefine $\widehat{\boldsymbol{x}} = -\widehat{\boldsymbol{x}}_i \ \forall \ i$ as the opposite direction as the geodesic $\widehat{\boldsymbol{q}}_i^{-1}\widehat{\boldsymbol{q}}_{i+1}$, thus:

$$\widehat{\boldsymbol{q}}_0^{-1}\widehat{\boldsymbol{q}}_i = \left(\frac{\|\widehat{\boldsymbol{x}}\|}{\|\widehat{\boldsymbol{x}}\| + \|\widehat{\boldsymbol{y}}_i\|}\widehat{\boldsymbol{x}} + \frac{\|\widehat{\boldsymbol{y}}_i\|}{\|\widehat{\boldsymbol{x}}\| + \|\widehat{\boldsymbol{y}}_i\|}\widehat{\boldsymbol{y}}_i \right). \tag{2.5.22}$$

The result is that every quaternion lying on geodesic-arc $\widehat{\boldsymbol{q}}_0^{-1}\widehat{\boldsymbol{q}}_i$ is defined in terms of two Euclidean quaternions:

$$x = \frac{\|\widehat{\boldsymbol{x}}\|}{\|\widehat{\boldsymbol{x}}\| + \|\widehat{\boldsymbol{y}}_i\|}, \quad y = \frac{\|\widehat{\boldsymbol{y}}_i\|}{\|\widehat{\boldsymbol{x}}\| + \|\widehat{\boldsymbol{y}}_i\|}. \tag{2.5.23}$$

Since, the reference point $\widehat{\boldsymbol{q}}_0$ is fixed, a point moving on the sphere $\widehat{\boldsymbol{q}}_0'$, the trajectory of $\widehat{\boldsymbol{q}}_0'$ is described by a cycloid:

$$\begin{aligned} x &= a\left(\frac{\theta}{a} - \sin\frac{\theta}{a}\right) \\ y &= a\left(1 - \cos\frac{\theta}{a}\right) \\ a &= \frac{r}{r'} \end{aligned} \tag{2.5.24}$$

where $a \in \mathbb{R}$ is a ratio between radii for the quaternion sphere $r = \|\widehat{\boldsymbol{q}}\|$ and the moving sphere r'.

The distance between two quaternions $\widehat{\boldsymbol{q}}_i$ and $\widehat{\boldsymbol{q}}_{i+1}$ lying on the same geodesic-arc $\widehat{\boldsymbol{q}}_0^{-1}\widehat{\boldsymbol{q}}_{i+1}$ is defined using the arc-length of a curve:

$$d(\widehat{\boldsymbol{q}}_i^{-1}\widehat{\boldsymbol{q}}_{i+1}) = \int_{\theta_i}^{\theta_{i+1}} \left(\left(\frac{dx}{d\theta}\right)^2 + \left(\frac{dy}{d\theta}\right)^2 \right)^{\frac{1}{2}} d\theta \quad \forall \quad i \in \mathbb{Z}. \tag{2.5.25}$$

The solution for the integral in Equation(2.5.25) when using the set of Equations (2.5.24) is:

$$\begin{aligned} d(\widehat{\boldsymbol{q}}_i^{-1}\widehat{\boldsymbol{q}}_{i+1}) &= -4a\cos\theta\Big|_{\theta_i}^{\theta_{i+1}} \\ &= 4a(\cos\theta_i - \cos\theta_{i+1}) \\ &= 4a\sin(\frac{\theta_{i+1}+\theta_i}{2})\sin(\frac{\theta_{i+1}-\theta_i}{2}) \\ &= 4a(\sin^2\theta_i - \sin^2\theta_{i+1}) \\ &= 4a(\cos^2\theta_{i+1} - \cos^2\theta_i). \end{aligned} \tag{2.5.26}$$

Since $\widehat{\boldsymbol{q}}_0 = -\widehat{\boldsymbol{q}}_0^{-1}$, the domain for the cycloid is bounded $[0, \pi]$, the transformation domain is also bounded to:

$$d(-\widehat{\boldsymbol{q}}_0^{-1}\widehat{\boldsymbol{q}}_0) = 8a. \tag{2.5.27}$$

Now one can see that the values for the quaternion axes $\widehat{\boldsymbol{x}}$ and $\widehat{\boldsymbol{y}}$ in the cycloid are:

$$\begin{gathered}\widehat{\boldsymbol{x}} = \frac{1}{2}(\widehat{\boldsymbol{q}}_i\widehat{\boldsymbol{q}}_0^{-1} + \widehat{\boldsymbol{q}}_0\widehat{\boldsymbol{q}}_i^{-1}), \quad \widehat{\boldsymbol{y}}_i = \frac{1}{2}(\widehat{\boldsymbol{q}}_0\widehat{\boldsymbol{q}}_i^{-1} - \widehat{\boldsymbol{q}}_i\widehat{\boldsymbol{q}}_0^{-1}) \\ \text{or} \\ \widehat{\boldsymbol{x}} = \widehat{\boldsymbol{q}}_0^{-1} \cdot \widehat{\boldsymbol{q}}_i \qquad\qquad \widehat{\boldsymbol{y}}_i = \widehat{\boldsymbol{q}}_0^{-1} \wedge \widehat{\boldsymbol{q}}_i.\end{gathered} \tag{2.5.28}$$

By choosing the value $a = \frac{1}{2}$, the transformation is equivalent to represent the quaternion values on the geodesic-arc $\widehat{\boldsymbol{q}}_0^{-1}\widehat{\boldsymbol{q}}_i$ by their half-quaternion rotations. However, both contributions from the internal product component and the outer product component are separated as function of the angle difference $\Delta\theta$.

In order to maintain the consistency between the outer and inner products, at least one of the vectors $\widehat{\boldsymbol{e}}_1$, $\widehat{\boldsymbol{e}}_2$ or $\widehat{\boldsymbol{e}}_3$ has to maintain the direction of the radial vector $\widehat{\boldsymbol{\xi}}_i$, that is, the internal product $\widehat{\boldsymbol{q}}_0^{-1} \cdot \widehat{\boldsymbol{q}}_i = \widehat{\boldsymbol{e}}_3 I$ while the outer product is left to track the trajectory followed by the other two bi-vectors, leaving only two degrees of freedom to $\widehat{\boldsymbol{q}}_0^{-1} \wedge \widehat{\boldsymbol{q}}_i$, *e.g.* $\widehat{\boldsymbol{e}}_2\,\widehat{\boldsymbol{e}}_3$ and $-\widehat{\boldsymbol{e}}_3\,\widehat{\boldsymbol{e}}_1$.

Consequently, the mapping looses a parameter in order to keep the relationship. The number of parameters is reduced from 4 to 3. The mapping is a compact representation like ternions or a tangential projection. The new projective rotation space is a spherical volume in $3D$. Its radius, ρ, is a nonlinear distance representing the cycloid's arc-length. The angular parameters ϕ and ψ are functions for the direction vector $\widehat{\boldsymbol{\xi}}_i$, as shown at Figure (2.11).

In the cycloid projection, all individual orientations are mapped as points inside the $3D$-sphere. Their radii is proportional to the cycloid arc-length. Their directions towards the unit sphere corresponds to their individual $\widehat{\boldsymbol{\xi}}_i$ vectors. For a proportionality ratio between the sphere radii $a = \dfrac{1}{2}$, the projection is:

$$\begin{aligned} &\text{Radius:} && \rho = 4\cos\theta_i \\ &\text{Angular Position:} && \phi = \arctan\left(\frac{y}{x}\right) \\ & && \psi = \arctan\left(\frac{z}{\sqrt{x^2+y^2}}\right) \end{aligned} \tag{2.5.29}$$

The radius ρ is independent from the angular coordinates (ϕ, ψ) that are a function of $\widehat{\boldsymbol{\xi}}_i$. The mapping yields trajectory segments in the Euclidean space. The segments correspond to the spherical geodesic-arc difference between different orientations.

Antipodal quaternions are the same orientation with opposite direction. They are mapped on another sphere with negative directions. The corresponding sign for the hemisphere is determined by the initial quaternion $\widehat{\boldsymbol{q}}_0$.

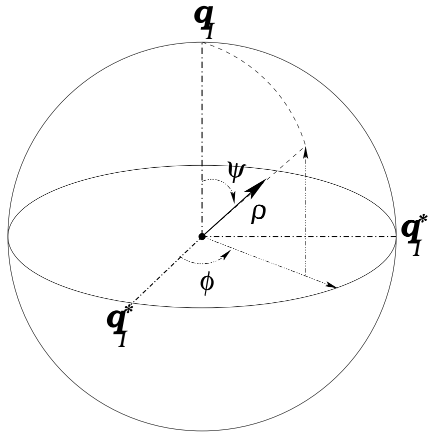


Figure 2.11: The sphere radius ρ is the arc-length of the cycloid function relating the angular value θ in the quaternion with the magnitude. The angular values of ϕ and ψ are the local coordinates for the spherical projection in $2D$. The quaternion conjugate $\boldsymbol{q}_I$ is the equator geodesic while the reference quaternion is $\boldsymbol{q}_I$.

One tends to use the positive sign. The mapping is used to express either orientations or rotation operations.

An important precaution is that the mapping is not Euclidean. The distance between two different orientation vectors is:

$$d(\widehat{\boldsymbol{q}}_i^{-1}, \widehat{\boldsymbol{q}}_{i+1}) = 2(\sin^2\theta_i - \sin^2\theta_{i+1}) \quad \forall \ i = \{1, \ldots, N\}, \ N \in \mathbb{Z} \tag{2.5.30}$$

and the total arc-length for the trajectory in a rotation sequence is the cumulative of the distances:

$$d(\widehat{\boldsymbol{q}}_0^{-1}, \widehat{\boldsymbol{q}}_N) = \sum_{i=1}^{N} 2(\sin^2\theta_i - \sin^2\theta_{i+1}) \ N \in \mathbb{Z} \tag{2.5.31}$$

However, an Euclidean approximation to the distance can be done when the difference magnitude in angular values for $\Delta\theta$ are small, when $\Delta\theta_{i+1,i} \approx \theta$. An empirical rule is $\Delta\theta < 10$ deg. Under such a constraint, the approximation to the arc-length for the sequence is:

$$d(\widehat{\boldsymbol{q}}_0^{-1}, \widehat{\boldsymbol{q}}_N) = \sum_{i=1}^{N} 2\left(\Big||\theta_{i+1}| - |\theta_i|\Big|\right) \ N \in \mathbb{Z} \tag{2.5.32}$$

Several advantages of using the cycloid to project unit quaternions over the Euclidean quaternion space:

1. A positive and accumulable distance function;
2. Bounded projection;
3. Compact representation a vector field in $3D$-Euclidean space.

2.5.2 κ and τ for Orientation in Projective Space

All previously discussed methods had been directed towards one point. To establish a comparison method for orientations in $3D$. The idea is to use the invariant FS framework for orientations as it is used for Euclidean trajectories in $3D$. The final objective is to simplify pattern recognition to simpler correlations without the need to use an alignment operator, see Equation (2.2.15). The necessary conditions to achieve such a goal are:

1. Establish a metric.

2. Guarantee view invariance, or solve the alignment problem.

Quaternion projections onto $3D$-Euclidean space are nonlinear and often unbounded. This is problematic since although one may describe an orientation trajectory in terms of the quaternion space, there is no way to guarantee that the correlation is bounded in terms of energy. This is the case for quaternion projections using polar projections such as ternions and geometric progression.

When projecting quaternions using the logarithm function, the projection is bounded but distorted because of the rotation from the surface to the tangential plane $\widehat{\boldsymbol{\xi}}_0$. The arc-length of a sequence cannot be measured nor compared using the natural logarithm function. And finally, it is highly dependent of the initial orientation point since the projection takes place in the tangent plane.

In this sense, the geodesic-arc projection offers better properties in order to separate the distance and direction from the angular component in the quaternion. The position of an specific orientation with respect to a reference point determines the quaternion magnitude and vice-versa. This is the major advantage of using the geodesic-arc projection.

However, the geodesic-arc projection does not allow us to use Frenet-Serret frames directly. The vectorial components in the projection are just approximations to the geodesics between two different orientations. There is no continuous function relating more than two quaternion points in the projective space. To use Frenet-Serret frames in the projection space is a mistake since there are straight lines and the derivative approximations introduce too much noise in the $3D$–projection for a numerical approximation to the curvature.

Nevertheless, it is still possible to measure a curvature from the spherical manifold obtained from the geodesic-arc projection. Using the transformation set in Equation (2.5.29), the mapping of $\widehat{\boldsymbol{\xi}}_i$ onto the sphere is expressed for the angular variables as:

$$\begin{aligned}
\phi_i &= \arctan\left(\frac{y_i}{x_i}\right) \\
\psi_i &= \arctan\left(\frac{z_i}{\sqrt{x_i^2 + y_i^2}}\right) \\
\widehat{\boldsymbol{\xi}}_i &= \widehat{\boldsymbol{q}}_{i-1}^{-1} \wedge \widehat{\boldsymbol{q}}_i, \quad \widehat{\boldsymbol{\xi}}_i = x_i\,\hat{\mathsf{i}} + y_i\,\hat{\mathsf{j}} + z_i\,\hat{\mathsf{k}}.
\end{aligned} \tag{2.5.33}$$

Then the curvature of the trajectory on the sphere $\boldsymbol{r}(\phi,\, \psi,\, t)$ is also parameterized as $\kappa(u,\, v,\, s)$ where s is the arc-length of a curve on the surface $S(u, v)$.

2.6 Proposed Solution: Combined Spaces

The proposed system analyzes position and orientation trajectories in two separate geometric spaces. The position is analyzed using general Frenet-Serret frames for two main reasons: space reduction and view-point invariance. The analysis of the orientation follows similar suit. Figure (2.12) illustrates a block diagram of the system.

The proposal is novel in the sense that it is a geometric approach that tackles at the same time the invariance problem for both position and orientation using their intrinsic spaces. The system uses time-tested techniques of signal processing adapted to the geometric problem using invariant templates and compact description spaces.

Previous sections have shown that invariance in the Euclidean space results from embedding local reference frames into the position curve. Although Frenet-Serret frames are the most common, other techniques are possible. One of this frames uses parallel transport. Parallel transport [99] avoids the problem of indeterminacy of the curvature κ by proposing a second curvature κ_2. In quaternion spaces, the definition of the Frenet-Serret frames is done by embedding the quaternion into the $3D$ frame [99].

2.7 Summary

Gesture comparison involves the classification of $3D$ spatio-temporal trajectories. The general solution often involves to establish a metric. However, even when the metric is available, the number of spatial dimensions complicates the problem. It is always desirable a low number of spatial dimensions.

Often, classification methods for state sequences rely on model properties to parameterize a sequence. An important issue, for both signal processing and data modeling, is that the data has to be representative independently

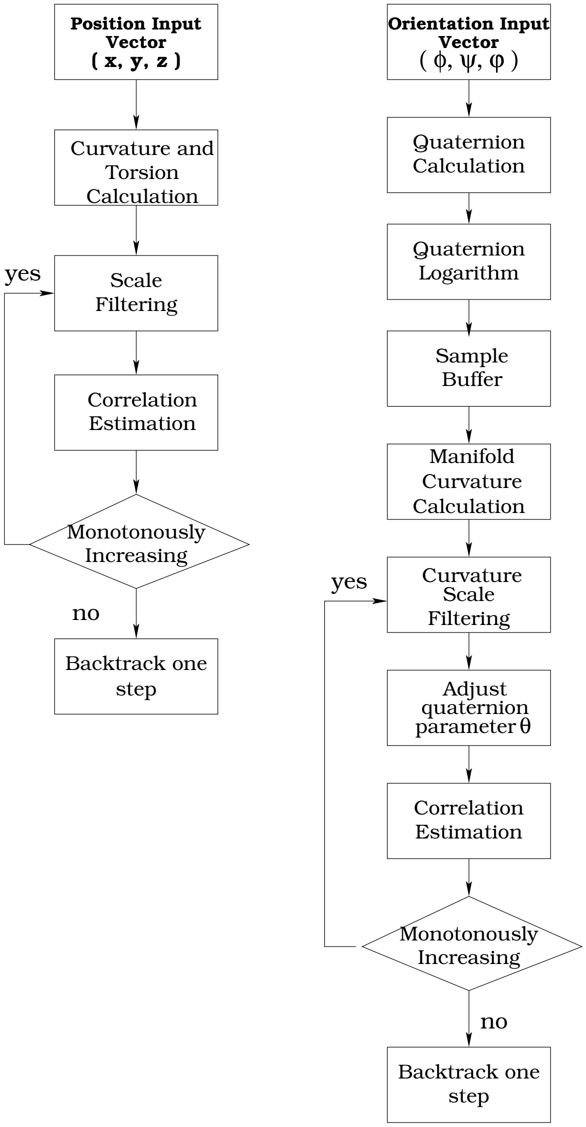


Figure 2.12: Block diagram for analysis of position and orientation. Both position and orientation follow similar paths. Orientation trajectories use the curvature of a projective space in order to obtain the curvature.

of point of view or time scalability. While time-invariance is usually achieved by means of the Fourier or Laplace domains, geometric invariance is more difficult to achieve. Geometric comparison depends on an alignment factor between reference frames. Time variations reflected by the Fourier domain depend on the stationary condition. Time-invariance is not guaranteed when a function is aperiodic. Since gestures are not necessarily periodic, comparison cannot depend on Fourier transformations to achieve time-invariance.

On the other hand, state models or differential equations in the Laplace domain, can achieve invariance by using the physical laws of motion in differential form. However, laws of motion, namely Lagrangian models, require detailed descriptions of the object and its properties, that in the case of gestures, are seldom available.

The geometric treatment of curves in space deals preferentially with Euclidean spaces. Local frame descriptions such as Frenet-Serret frames can even be extended to higher dimensions by assuming the extensibility of the analytic properties of the Euclidean space. They also assume that the curve is at least $\mathcal{C}^{(n)}$ continuous for the n-dimension.

A $3D$ curve $\boldsymbol{p}(t)$ is a description for a rigid body that uses two distinct geometric spaces: an Euclidean manifold and an 4-spherical manifold. The Euclidean manifold describes the position $\boldsymbol{r}(t)$. The $4D$-spherical manifold describes the orientation.

Differential frames provide an invariant description that, additionally, reduces the dimensionality, but it describes the trajectory in terms of two nonlinear parameters that should be treated carefully. Local frames are not the first choice to describe a curve. The reason behind is that the local frames assume that it is possible to always obtain the n-order derivative. However, when dealing with discrete functions, such analytic conditions cannot be ensured. Also, the signal-to-noise (SNR) ratio decreases exponentially when approximating derivative orders higher than the second order. Numerical approximations to the derivative with either convolution methods or finite differences introduce both noise and distortion.

Then why it is important. The reason is that although the curvature whether it is imprecise or not, contains enough information on the intrinsic space so it is still viable to retain invariance even if higher derivatives are not tenable. The difficulty lies in how to smooth the functions in the extrinsic space, in the n-manifold, without distorting the available information in the $(n-1)$-manifold or intrinsic space.

Orientation spaces do not have the same properties as Euclidean spaces. Rotations described by rotation matrices and vector axis-angle rotations do not have a metric space nor a descriptive trajectory. Quaternions are the only mathematical entity capable of giving a description of the shape of the orientation trajectory of a local frame. The definition of a distance metric in quaternions is only determined between two consecutive quaternions and the functions are delimited to operations on planes.

In this chapter, we have shown that although an orientation trajectory has special challenges. And yet, it is still possible to use the same descriptions of curvature for an Euclidean trajectory. In the case of spherical spaces, one modifies the sphere-space by re-projection of the sphere onto a more suitable space. Different mappings provide different characteristics, they may be unbounded or not. We developed a projection, namely geodesic-arc projection. The projection uses the intrinsic properties of circles. It is an arc-length projection and thus nonlinear. However it provides a spherical distance that which is closely related to navigation distances on $2D$-spheres. That hint us that the projection is correct.

An important contribution in the present discussion is a proposal for which position and orientation can be treated online with the same local differential framework. Invariance is key and it is achieved by switching to differential local frames. Here, curvature measures for the n-dimension are numerically stable, but not bounded. Our interest in such problem arises from the fact that correlation between sequences is a time-tested method in pattern recognition but it requires that the compared sequences are bounded. Correlations take different forms for different mathematical and statistical tools.

The method has advantages over others. First, the number of spatial dimensions is reduced with local differential frames for both, position and orientation spaces. This reduces the complexity of the recognition problem. Second, the method is scalable to kinematic chains. The recognition and comparison problem for a single point in space is scalable to kinematic mappings for both position and orientation. Third, one can use simpler online signal processing techniques with geometric descriptions. And last, view invariance for both position and rotation using the same framework.

Nevertheless, the method also has disadvantages. It is necessary to implement an off-line methodology for the acquisition of trajectory templates, for both position and orientation parameters. View and orientation invariance are achieved at the cost of non-linear metric spaces with high sensitivity to noise. Small variations have great repercussions in the comparison using curvature and torsion. To deal with non-linear metrics, it is also necessary to implement an adaptive filtering procedure for both template acquisition and on-line comparison. The purpose is to reduce the variability in the comparison.

In the next chapter, we will introduce a novel scale-space analysis that will allow us to compare two trajectories independently of the space geometry.

Chapter 3

Multi–scale Curvature Analysis

3.1 Extrinsic and Intrinsic Reference Frames

$3D$-geometric data is difficult to model. Solids and curves are frequently represented as sets of discrete points $\mathbf{f}$ in Euclidean spaces. These objects have an extrinsic description $\mathbf{f}$ when they are referred by a global reference frame denoted as $\mathcal{F}$. The $\mathbf{f}$ object does not have an invariant representation to affine transformations under $\mathcal{F}$. However, a unique representation independent of point of view, denoted by $\mathbf{g}$, can be achieved by moving the reference frame $\mathcal{F}$ to $\mathcal{F}_I$ within the object's manifold. Moving the external reference frame $\mathcal{F}$ inside or onto the object's manifold $\mathbf{g}$ creates an intrinsic description that is critical for robust pattern recognition. The situation is expressed mathematically as:

$$\begin{array}{lll} & \boldsymbol{G} : \mathbb{R}^n \mapsto \mathbb{R}^n & \\ \text{such that} & & \\ & \mathcal{F}_I = \boldsymbol{G}(\mathcal{F}) & \mathbf{f} \mapsto \mathbf{g} \\ & \mathcal{F} = \boldsymbol{G}^{-1}(\mathcal{F}_I) & \mathbf{g} \mapsto \mathbf{f} \end{array} \tag{3.1.1}$$

The mapping $\boldsymbol{G}$ localizes the global frame $\mathcal{F}$ for an object $\mathbf{f}$ either inside or onto the object's $(n-1)$–manifold $\mathbf{g}$ such that the geometric transformation $\mathbf{f} \mapsto \mathbf{g}$ is invariant to affine transformations of $\mathbf{g}$.

However, invariance has a cost. Moving the external reference frame $\boldsymbol{G}(\mathcal{F})$ also eliminates other information than the manifold's local distance $d(\mathbf{g})$. The manifold's local distance $d(\mathbf{g})$ is also blind to the global geometry since it is measured over the manifold. Another downside is that the manifold's local distance $d(\mathbf{g})$ may be nonlinear with respect to the extrinsic reference frame $\mathcal{F}$. At sufficiently small variations δd, the local distance d can be measured on the surface of $\mathbf{g}$ as an Euclidean distance and thus, $\mathbf{f} \mapsto \mathbf{g} \in \mathbb{R}^2$ defines an $(n-1)$-manifold of an Euclidean space.

Often it is assumed that when $3D$-curves are digitalized, the samples are located at evenly distributed distance intervals. In the case of smooth curves and surfaces, it is reasonable for the objects to be sampled at constant time intervals. And although constant time may translate into constant distance in Euclidean spaces for analytic $3D$-curves, often the assumption does not hold true for spaces with other geometries. Mapping time constancy into distance often justifies an approximation for the Nyquist theorem onto $3D$-geometric objects. However, it is inappropriate to hold that the same assumption will be true for geometries other than Euclidean. Constant time-sampling does not translate into constant distances in n-dimensional rotation geometries such as quaternions. Thus, intrinsic measures such as curvature have to be reinterpreted based on the characteristic inter–point distances for surfaces and curves in other geometries.

3.2 Curvature Scale-space Analysis

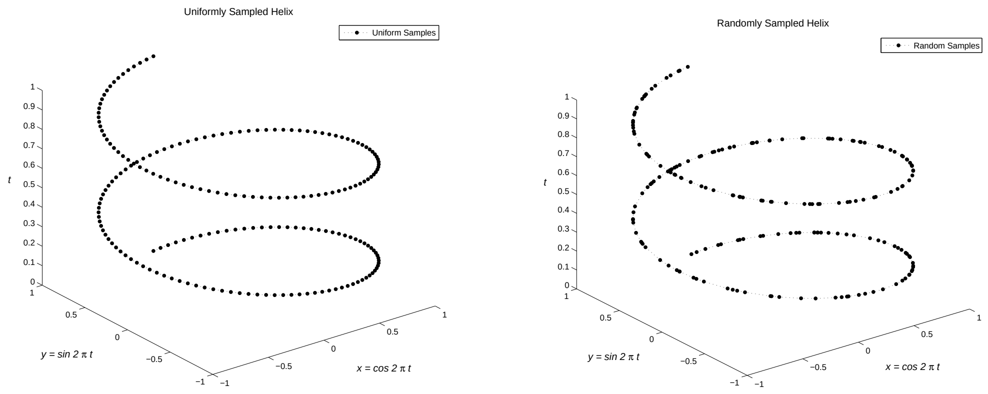
Curvature is a metric of bending. Specifically, it is a rate that measures how fast the tangent vector $\widehat{\mathsf{T}}$ of a curve changes in direction as the curve evolves. If the tangent vector $\widehat{\mathsf{T}}[0]$ at point $\boldsymbol{r}_0$ has the same direction as the tangent $\widehat{\mathsf{T}}[i]$ at point $\boldsymbol{r}_i$, the curvature κ curvature between $\boldsymbol{r}_0$ and $\boldsymbol{r}_i$ is zero because the two tangent vectors are collinear, i.e, the intersection points meet at $\pm\infty$. However, if the two tangents have different directions, then the curvature $\kappa \neq 0$. Since the rate at which the angular value of the intersection between the normals change is relative to the distance s measured on the curve $\boldsymbol{r}(s)$, the general definition of the two-dimensional curvature κ is

$$\kappa \propto \left\| \frac{d^2\, \boldsymbol{r}(s)}{d\, s^2} \right\| . \tag{3.2.1}$$

The curvature is always related to the second derivative of the curve $\boldsymbol{r}(s)$.

Because of how the curvature is defined, curvature correlations are seldom used because they are not bounded. Curvature is nonlinear and not bounded. This is a most important issue when comparing curves with curvature signatures. Curvature signatures for a curve $\boldsymbol{r}[i]$ can be easily singular at any point. Such singularities often occur either when a segment of the curve approaches a straight line or the curve has an inflexion point. In both cases, the curvature is singular either because there is no change in direction or because the change in direction occurs at the same point, so $ds = 0$. The usefulness of the curvature as intrinsic signature is severely compromised.

An implied factor with the nonlinearity of curvature signatures is noise. The curvature, as metric, is very sensitivity to noise. One can show the nonlinear nature of the sensitivity of the curvature to noise with a simple example by using an analytical curve such as an helix, at Figures (3.3a) and (3.3b). The helix curve has been uniformly and randomly sampled along the arc-length. The arc-length is the same for both helixes but the samples are placed at different positions along the curve. No other variations were used.



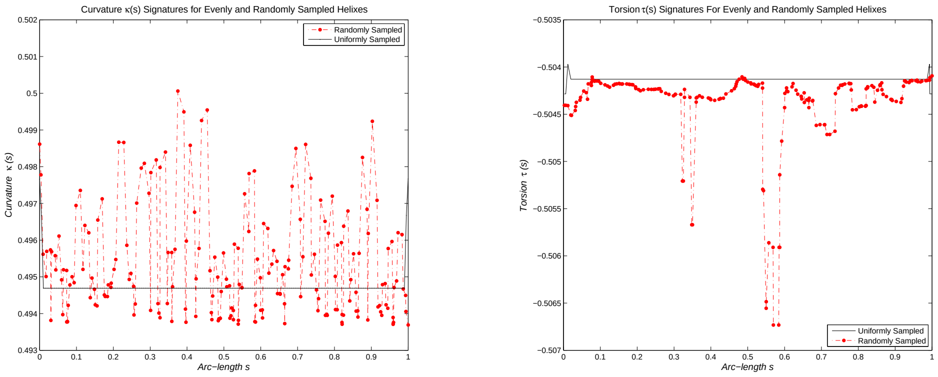
(a) Uniformly sampled helix **(b)** Randomly sampled helix

Figure 3.1: Two helixes with different sampling along the arc-length. The difference is the interleave from sample to sample: Uniform for Figure (3.3a) and random for Figure (3.3b)

Sensitivity to noise from the curvature and torsion signatures is an important concern. Even when the two curves are close to each other since their metrics are close, the variation in their curvature signatures is noticeably different. The key factor is to determine how much noise is allowed in the signature and still be able to achieve a correct identification (See Figure (3.2)).

The nonlinearity of both curvature and torsion metrics is evident as illustrated at Figures (3.2a) and (3.2b). The non-uniform sampling introduces noise to both curvature $\kappa(s)$ and torsion $\tau(s)$, although the latter is more severe. The explanation is that the numerical approximation for the derivative is not enough to cancel the odd terms of the Taylor series. Some improvement is accomplished by compensating non-uniform interleave in the h parameter for the Taylor series, but even these linear approximations cannot maintain the analytic properties of the signal.

The curvature-torsion mappings at Figure (3.1) illustrate the dilemma. The mapping is unique for the curve $\boldsymbol{r}(s)$, but it is numerically unstable. The stability of finite difference methods depend on canceling the odd terms in the Taylor series a function f. Canceling the odd terms depends on the uniformity of the intervals h.



(a) Curvature Signatures $\kappa(s)$ for Uniformly and Randomly Sampled Helixes, at Figures (3.3a) and (3.3b) respectively.

(b) Torsion for Uniform and Randomly Sampled Helixes, at Figures (3.3a) and (3.3b) respectively.

Figure 3.2: Curvature and Torsion signatures for an uniformly and randomly sampled helix.

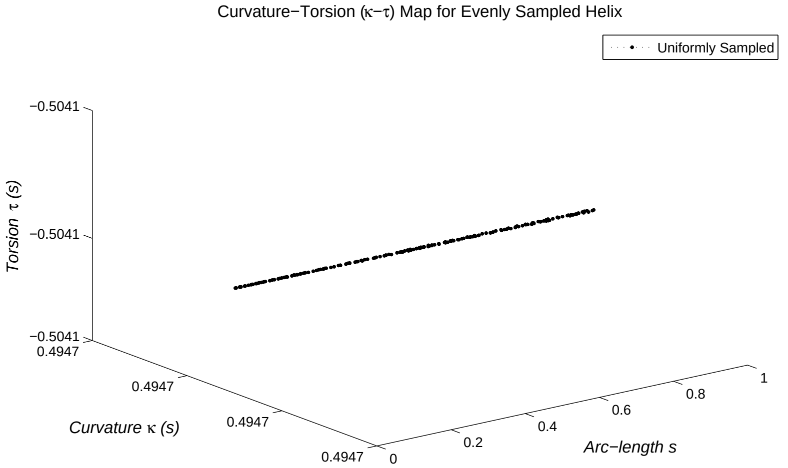
Numerical instability arises when the higher odd terms have residuals. Such a situation yields divergence for the derivative f'. Even when some compensation for the polynomial of h is possible, the finite difference method is sensitive to rapid changes in the distance h and performs poorly [107]. As shown at Figure (3.2a), a second order polynomial of h can maintain a degree of stability for the curvature calculation.

On the other hand and as shown in Section (2.3.6.1), convolution methods to obtain time–series derivatives, such as the Savitzky-Golay filter, also introduce a distortion due to aliasing when the interleave distance is uneven.

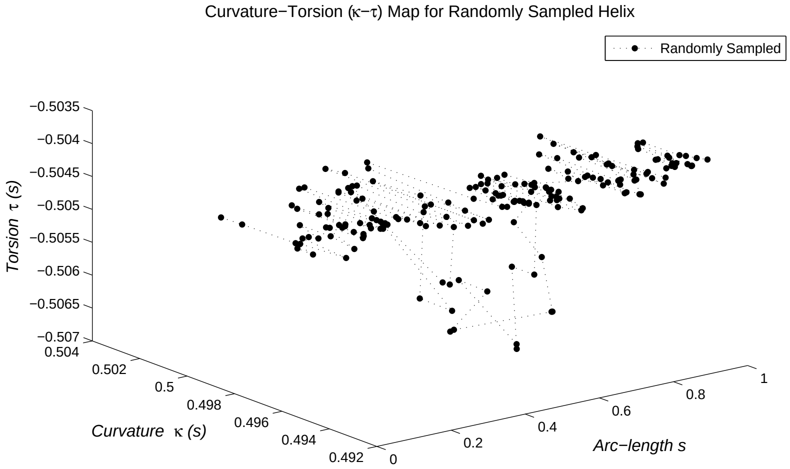
3.3 Correlations with Curvature Signatures

The matrix form for $3D$-Frenet-Serret, Equation (1.5.7), describes a curve $\boldsymbol{r}(s)$ which is the solution of a system of ordinary differential equations. As the curve evolves, both curvature κ and torsion τ also change along the curve's arc-length s.

In this sense, κ and τ are to local frames what the internal dot product $\boldsymbol{x}^{\top}\boldsymbol{x}$ is to Euclidean spaces, distances. However Frenet-Serret frames introduce a skew–symmetry in the distance matrix $\boldsymbol{K}$, characteristic of external products. It also shows that the curvature-torsion space is singular and critically stable. This is observed since its determinant is singular and consequently, all the



(a) Curvature-Torsion $(\kappa - \tau, s)$ Signature for an Uniformly Sampled Helix, at Figure (3.3a)



(b) Curvature-Torsion $(\kappa - \tau, s)$ Signature for a Randomly Sampled Helix at Figure (3.3b)

Figure 3.3: Curvature-Torsion $(\kappa - \tau, s)$ maps for two helixes with different sampling strategies. The randomly sampled data introduces arc-length differences in the derivatives. Consequently the noise level is higher.

eigenvalues of curvature matrix $\boldsymbol{K}$ are complex conjugates:

$$|\boldsymbol{K}| = \begin{vmatrix} 0 & \kappa & 0 \\ -\kappa & 0 & \tau \\ 0 & -\tau & 0 \end{vmatrix} = 0. \tag{3.3.1}$$

Complex conjugate eigenvalues indicate that the distances κ and τ are unique and irreplaceable for every point in the sequence. The reason is that the eigenvalues generates orbits in different parametric spaces of κ and τ. Indicating also that the curvature and torsion are orthogonal and independent from each other.

Unfortunately, all the above properties also indicate that convolving a filter to smooth the $\kappa - \tau$ signature is useless. Convolving a filter with the $\kappa - \tau$ signatures yield a projection onto a null space. The conclusion is that if one were to modify the signatures $\kappa(s)$ and $\tau(s)$, the operation on the intrinsic space would change the original curve $\boldsymbol{r}(s)$ into a completely different curve $\widetilde{\boldsymbol{r}}(s)$ in the extrinsic space. Proving that both parameters, curvature κ and torsion τ, are invariant and unique.

How is it then possible to propose a multi-scale curvature analysis? An example is the Curvature Scale-Space (CSS) in $2D$-image spaces [137]. The CSS analysis in $2D$ operates on the curvature sub-determinant:

$$|\boldsymbol{K}_{3,3}| = \begin{vmatrix} 0 & \kappa \\ -\kappa & 0 \end{vmatrix}. \tag{3.3.2}$$

CSS operates with smoothing filter on the extrinsic space before obtaining the curvature signature (3.3.2) with a scaling function matrix $\boldsymbol{\digamma}(\sigma \boldsymbol{I})$ such that:

$$\kappa(\sigma \boldsymbol{I}, \boldsymbol{x}) = f(\boldsymbol{\digamma}(\sigma \boldsymbol{I})\, \boldsymbol{x}). \tag{3.3.3}$$

The curvature κ depends on a scaling factor $\sigma \boldsymbol{I}$ and the $2D$ position vector $\boldsymbol{x}$ of an image feature such as an edge. Usually for simplicity, the diffusion operator $\digamma$ is a $2D$ Gaussian filter. Depending on how the factors in vector $\sigma \boldsymbol{I}$ are introduced in the operator $\digamma$, the scaling effects of the filter are either iso- or anisotropic. However, such approach is not adequate for the purposes here discussed. The smoothing operator $\boldsymbol{\digamma}(\sigma \boldsymbol{I})$ over-smooths relevant points of the curvature signature $\kappa(s)$.

In theory, it is possible to use a direct approach to compare two distinctive curves in a $3D$-Euclidean space by using a two-dimensional correlation that involves both signatures $\kappa(s)$ and $\tau(s)$. However, in practice, both signatures are highly nonlinear and not bounded. The $\kappa(s)$ and $\tau(s)$ are plagued with inflexion points for which

$$|\kappa(s)| = \infty \quad \text{or} \quad |\tau(s)| = \infty. \tag{3.3.4}$$

Such a situation implies that correlations using such signatures to be unbounded as well.

Correlations do not work in signals with unbounded energy. The condition is clearly expressed by the Parseval's theorem for a correlation between two signals $\mathsf{G}(f)$ and $\mathsf{H}(f)$ in the Fourier domain:

$$\left\| \mathsf{G}(f)\overline{\mathsf{H}(f)} \right\|^2 < \infty. \tag{3.3.5}$$

In other words, the total energy content in the correlation must be finite. However, this condition does not hold true in direct correlations with the curvature and the torsion signatures.

However, it is still possible to use the intrinsic signatures in correlation operations. The fundamental problem is how to bound the energy content for $\kappa(s)$ and $\tau(s)$. The following sections elaborate on two different multi-scale methods that keep intrinsic signatures bounded and thus, their correlation is also bounded.

A secondary issue with correlations is the length of the signals $\boldsymbol{g}[t]$ and $\boldsymbol{h}[t]$. The correlation operation expects that both signals have the same number of samples. But, usually, it is rare for $\boldsymbol{g}[t]$ and $\boldsymbol{h}[t]$ to have the same number samples. A common solution is to use re-sampling methods in order to equate the number of samples. However, re-sampling is essentially s nonlinear operator that tends to modify the signature. Due to the sensitive nature of the $\kappa(s)$ and $\tau(s)$ signatures, this situation is not desirable.

In order to solve these problems, we introduce two general multi-scale methods for $3D$-geometric signal analysis using the curvature space as invariant support. The purpose of our correlation methods is twofold. First, to deal with curvature correlations in a bounded way, and second, to compare gestures of different lengths while avoiding to use re-sampling methods. A valid concern is that our algorithm approximates the manifold's curvature signature using finite differences. Finite difference methods assume that the intrinsic distance between samples is equidistant with constant sampling rate. However, geodesic distances other than Euclidean do not hold this assumption true.

The hypothesis of an analytical signal introduces a non-linearity for the curvature signature in geometries different than Euclidean. A constant sampling rate does not guarantee equidistant samples in geometric problems. Consequently, the curvature and torsion signatures may not be preserved in a different geometry.

The first method is a non-linear least squares fitting algorithm that uses a parametric shape of known curvature. The parametric function fits recursively into the data by a process of fit and segmentation. The result is a multilevel approximation going from the general curvature value to the smaller neighborhood values. The output will include a residual from the fitting procedure.

The process is similar to approximating one-dimensional signals with limited support linear polynomials at distinctive scalar values: wavelets. The main difference is that the shape provides a predictable curvature ratio for the locally fitted neighborhood. One can use this shape fitting to approximate the local curvature values of the geometric signal.

The second method is a technique based on geometric signal processing kernel known as local-curvature filter. It is an anisotropic kernel filter because it uses geodesic distances to produce a multi-scale approximations to smooth the curve. A great advantages of the anisotropic kernel filter is that it does not need an accurate approximations of the curvature κ. A coarse approximation suffices to produce a smoothing kernel. The multi-scale is provided by the neighborhood size k. The neighborhood size k relates the filter to the geodesic distance and thus, it bounds the energy content of the neighborhood. The result is an energy bounded description of the curvature signature in the manifold. Our two general methods do multi-scale smoothing using the current available curvature information.

We achieve two important goals: first, to create an invariant and bounded metric of the manifold and second, to avoid using interpolation methods that would modify the manifold curvature signature. The shape fitting algorithm reduces the extrinsic curve into simpler local approximations to the curvature values while its residuals indicate us the error magnitude in terms of local energy. This technique is a general non-linear least squares fitting algorithm and gives similar results to signal processing of one–dimensional signals using wavelets. The intrinsic filter provides a non-linear smoothing function that approximates the curve in the extrinsic space in the least square sense. The advantage of the method is that it provides a simpler yet, powerful tool to analyze the trajectory in both coarse and detail levels. The method is inspired from techniques used in three-dimensional laser image ranging [32], [31], [30], [33]. Similar filters using the diffusion equation have appeared in different disciplines such as computer vision [137] and computational flow methods [75].

3.4 Curvature-based Correlation Methods

In this section two general methods for scale-space analysis in geometric data sets using curvature are introduced. The methods use linear correlations with nonlinear metrics. We apply both methods to the problem of $3D$-gesture comparison. The idea is to bound the signature energy by applying a linear combination of intrinsic distances. In a way, the methods re-define the energy content of a geometric path over the normalized arc-length $\left[\frac{1}{s}\right]$, this is equivalent to estimate the signal's power for a $3D$ geometric curve or surface. The concept is inspired from three-dimensional range image processing [32], [30],[31] where similar concepts apply and it is fundamental for our correlation methods.

The first method is a divide and conquer strategy by fitting a function with least-squares methods. The divide and conquer strategy divides the data set into coarse and detail levels. The coarse levels are composed of the parametric values for the fitting function while the detail levels are the corresponding residuals of the fitting function. This is a multi-scale method that approximates the curvature values from the parameters of the fitting, without calculating the derivatives.

The second method is a convolution filter called anisotropic kernel filtering. The main strength of the method is that it does not require for the derivative approximations to be accurate. The method accepts derivative approximations from low order difference methods or convolution filters such as the Savitzky-Golay filter. Although, it does require that the approximations are second order continuous, *i.e.*, $\mathcal{C}^2$ exists. The reason is that the method applies a nonlinear kernel to the metric, so a certain level of inaccuracy is tolerated. The weakness of the method is its computational complexity which is approximately $\boldsymbol{O}(n^2)$.

3.4.1 Curve Fitting for Curvature Analysis

This section introduces a curve fitting method that approximates the curvature with multi-scale resolution without computing local derivatives. Computing the curvature using curve fitting is hardly a novel technique, but it is advantageous for two reasons. First, the curve fitting produces a natural smoothing for the $3D$-data set depending on the scale. Second, the derivative order for the $3D$-data set completely depends on the model and not on the local differences since they tend to be numerically unstable.

The key idea is to fit families of parametric shapes with known curvature into the unknown data set. The shapes are fitted onto the unknown points by using an objective function. The fitting error, which it is also proportional to the curvature error, is evaluated by the objective function in the least square sense.

Fitting a curve or a surface to the cloud of $3D$ points introduces a relationship that abates the dimensionality and the complexity of the recognition problem. The initial hypothesis is that the set of points, $\mathbf{x}$, belongs to suspected surface in $3D$. The scalar field in the set $\mathbf{x}$ is represented by an embedded manifold with lower dimensionality such that there is a mapping

$\boldsymbol{f}(\mathbf{x}) : \mathbb{R}^n \mapsto \mathbb{R}^{n-1}$ plus an error $\boldsymbol{\varepsilon} \in \mathbb{R}^{n-1}$. In the particular case of $3D$, the mapping has a form:

$$\mathbf{x} = \{x, y, z\} \mapsto \boldsymbol{f}(u, v) + \boldsymbol{\varepsilon} \tag{3.4.1}$$

The known shape is also described as a three-dimensional multivariate function $\boldsymbol{g}(u, v : \boldsymbol{\beta})$ where $\boldsymbol{\beta}$ is a n-vector of adjustable parameters that controls the shape. The algorithm minimizes a metric error e between the unknown surface and the parametric surface by using the least-squares objective function:

$$e = \min_{\boldsymbol{\beta}} \sum^{N} \left\| \boldsymbol{f}(u, v) - \boldsymbol{g}(u, v; \boldsymbol{\beta}) \right\|^2 \tag{3.4.2}$$

where N is the number of parameters in vector $\boldsymbol{\beta}$.

Some curves and surfaces in $3D$-Euclidean spaces may admit other simplifications due to their geometry. Thus one can solve Equation (3.4.2) with linear programming, although it is not rare to find nonlinear minimization problems for complex shapes.

Curves in space are sequences that follow a temporal order and not only spatial locations. This characteristic is critical because it allows us to define the space's principal directions in terms of the sequence ordering. This approach is similar to the principal component analysis in the sense that one has to determine first the sequence's principal direction. Finding a principal direction for the sequence is a concern in the method. However, if it is not possible to find a principal direction for the data sequence, then a possible solution is to subdivide the ordering sequence into shorter sequences until a projection is found.

Determining an optimal length for the subsequences is outside of the scope of the current work. The reason is that determining the optimal length is analogous to finding statistical invariants for a time-series or conversely. In other words, it becomes necessary to solve a segmentation problem that fulfils a stationarity condition. The current assumption is that the sequences do have a principal direction and linear programming does yield results showing the direction of the curve in space.

Reducing the dimensionality of the data set $\mathbf{x}$ allows us to simplify pattern shapes to general quadratics such as the ones shown in Table 3.1. The curvature values for each shape are shown in the third column for the different curve patterns. An ellipse has a nice expression for its curvature, however it requires at least four non-concyclic points and the solution may not be unique. Both the parabola and hyperbolas are unbounded and asymptotic curvatures. They may present other difficulties for curvature analysis.

A circle has constant curvature. Any type of curve analysis fitting circles is a type of analysis with constant curvature. The circle is the simplest shape

Table 3.1: Quadratic Functions For Shape Fitting

General Function $A(x^2) + B(xy) + C(y^2) + Dx + Ey + F = 0$		
Shape	Conditions	Curvature
Circle	$B = 0,\ D = 0,\ E = 0$ Solve A, C and F for radius from the point means $\bar{\boldsymbol{X}}$ and the estimated center $\hat{\boldsymbol{c}}$, then $r = \lVert \bar{\boldsymbol{X}} - \hat{\boldsymbol{c}} \rVert$.	$\kappa = \frac{1}{r}$
Ellipse	$\begin{vmatrix} A & B \\ B & C \end{vmatrix} > 0$ Solve A, B, C, D, E and F for the estimated parametric foci angle t, the major axis a and minor axis b.	$\kappa = \frac{ab}{(b^2 \cos^2 t + a^2 \sin^2 t)^{\frac{3}{2}}}$
Parabola	$B^2 = 4AC \quad A \neq 0,\ C \neq 0$ Solve A, B, C, D, E and F for the distance from point mean $\bar{\boldsymbol{X}}$ to the estimated focal point $\hat{\boldsymbol{f}}$.	$\kappa = \frac{1}{2\lVert \bar{\boldsymbol{X}} - \hat{\boldsymbol{f}} \rVert}$
Hyperbola	$\begin{vmatrix} A & B \\ B & C \end{vmatrix} < 0$ Solve A, B, C, D, E and F for the estimated parametric foci angle t, the major axis a and minor axis b.	$\kappa = -\frac{ab}{(b^2 \cosh^2 t + a^2 \sinh^2 t)^{\frac{3}{2}}}$

available and it provides a nice geometric interpretation for the curvature κ. Constant curvature analysis has been used for automatic image query in database search problems [137], but it has not been applied to the problem of gesture comparison.

3.4.1.1 Why Curvature Only

Aside the numerical instabilities of the third derivative, there is a practical reason that justifies the use of only the curvature parameter. Smoothing a $3D$ curve based only on the curvature parameter is equivalent to project the $3D$ data set onto a thin plate. This only justifiable if the error introduced into the trajectory is small. The torsion parameter indicates the rate at which a $3D$ curve leaves the thin plate plane. However, in a noisy $3D$ data set, it does not indicate how far a peak point would be M samples ahead. It becomes then necessary to find the maxima at which such peaks occur from the projection plane. If the ratio between the total arc-length for the curve under consideration and the distance at which the peaks occurs is not significant, then the thin projection is justifiable.

In the case of our working data set, the above justification is valid. The ratio I between curve arc-length s vs. maximum peak distance in between Z zero crossings between curve and projection is small. The ratio is defined as

$$I = \frac{\sum_{i=1}^{Z-1} \max_i \left|\boldsymbol{x} - \boldsymbol{P}\boldsymbol{x})\right|}{s}. \tag{3.4.3}$$

Such situation is illustrated at Figures (3.4a) and (3.4b). Other ratios consider the total distance in between zero crossings:

$$I_i = \frac{(\boldsymbol{x} - \boldsymbol{P}\boldsymbol{x})^{\mathsf{T}}(\boldsymbol{x} - \boldsymbol{P}\boldsymbol{x})}{\Delta s_i} \quad \text{for} \;\; \Delta s_i = s_i - s_{i-1}. \tag{3.4.4}$$

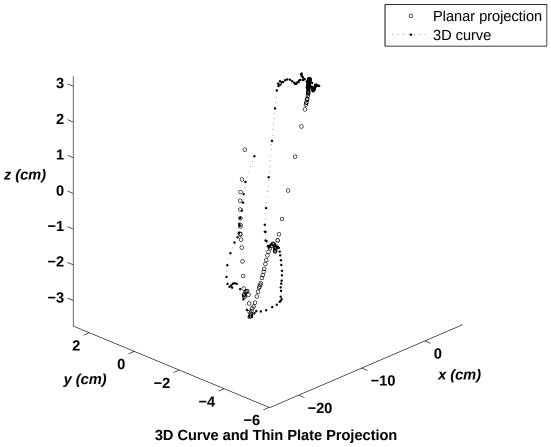
for which s_i and s_{i-1} two separate zero crossings occur.

3.4.1.2 Basic Algorithm for Constant Curvature Analysis

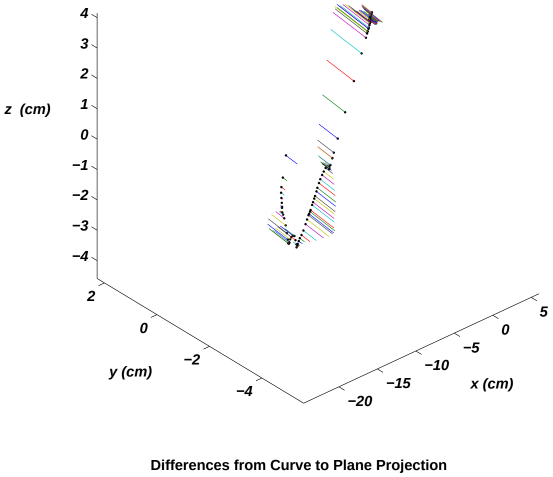
The shape fitting algorithm does not handle curvature values directly. However, it is numerically more stable and makes more physical sense to use the radius of curvature instead. Approximating curves and surfaces with circles and spheres converges rapidly at least at the proximity of the point of interest and it quickly provides a local measure of the curvature. The rapid convergence of the algorithm is because circle fitting admits a reduction in the complexity that other shapes do not [58].

The initialization of the algorithm consists in finding a common projection plane to the majority of points in the sequence. The plane is a least-square fit to all the points in the sequence. The next step is to construct a projection of all point in the sequence onto the common plane. For small to medium size sequences the method works well because curves composed of a limited number of elements can be approximated by planes.

Next, the algorithm uses a divide and conquer strategy in conjunction with a tree data structure for constant curvature analysis. The tree structure constructs different levels for which the curve $\boldsymbol{r}(s)$ fits a circle for different arc-lengths for a constant $\alpha \in \mathbb{R}$ such that $\alpha s,\ 0 < \alpha \leq 1$. The base node uses $\alpha = 1$ to include the full length of the curve $\boldsymbol{r}(s)$ and fit the known curvature. Subsequent nodes use proportional arc-lengths to fit subsequent shapes. Unlike wavelet analysis, curvature analysis does not guarantee equally split steps as in a geometric series. The complexity of $\boldsymbol{O}(n \log n)$ is due to the tree data structure.



(a) A $3D$ Forceps trajectory and its planar projection



(b) Differences between a $3D$ Forceps trajectory and its planar projection

Figure 3.4: The ratio between maxima peaks between a $3D$ curve and its planar projection vs. the arc-length is small. The reason is that the maximum number of crossings between projection and curve is small or because the distance differences at peaks are also small.

At each level, there are n_s possible nodes. The number of nodes varies according to the shape fitting. In order to determine the number n_s, the algorithm determines the high curvature values peaks encountered at each scale. It does not guarantee the same number of segments in a particular level. However, the method guarantees that the curvature values are quantized, and thus bounded. The residuals of the fitting function at each specific level constitute the total fitting error at each level node. The decomposition provides quantized coarse values of curvature for each approximation. The details or residuals are kept on the tree leaves.

The lower bound of the algorithm is reached at $N/3$ samples. A minimum of three samples are required to fit a circle. This lower bound is rarely needed because either the fitting error is too small at higher nodes or because the noise introduced by not considering lower levels is negligible and considered as an acceptable noise level for the signal-to noise ratio.

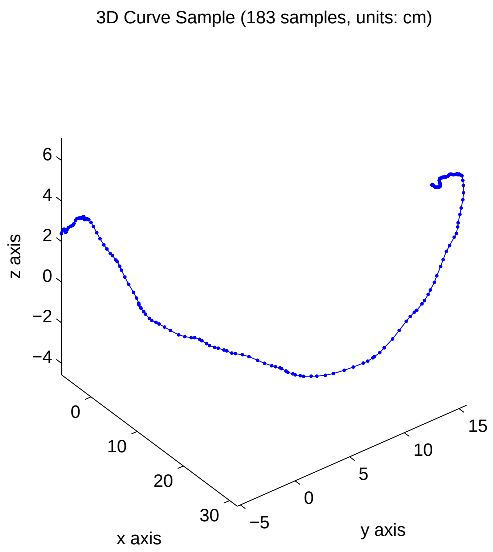
Other basic shapes can be used to fit the basic shape of the curve. Another choice would be the ellipse or a parabola for small sections. However, choosing an ellipse must take into account that the fitting may not be unique. The fitting algorithm to fit conics other than circles is still somewhat unreliable, and in practical applications, uses more than four samples. Open end conics such as the parabola would additionally require to find the dominant directions for the parabola. Continuity conditions may be also problematic at the extremity of the curve.

3.4.1.3 Constant Curvature Fitting Results

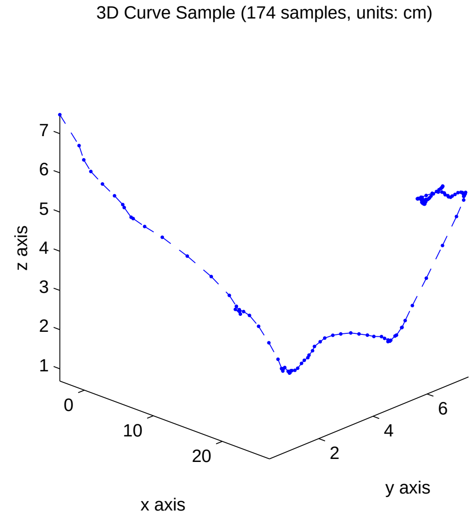
Lets consider a $3D$ curve as the one presented at Figure (3.5). The curve presented at Figure (3.5) is part of a gesture data set for Forceps handling gestures that will be explain later. Only the position is presented in this section.

Both trajectories at Figure (3.5) are for position. The curves are composed of discrete samples in a $3D$-Euclidean space. The trajectory at Figure (3.5a) has of a total of 183 samples while the trajectory at Figure (3.5b) has a total of 174 samples. The sampling rate was of 40 samples/s. The curves can be considered mostly planar and thus the planar projection is easy to do without previous segmentation.

Figure (3.6a) shows the planar projection for the curve at Figure (3.5) and the initial step in fitting circles in order to approximate the curvature at the projection plane. The curvature error for each point is inversely proportional to the closeness to the intersection points. This is clearly shown at Figures (3.6c) and (3.6d). The curvature spikes at Figures (3.6c) and (3.6d) represent the points for which the circle approximation intersects the thin–plate projection



(a) Expert Curve



(b) Novice Curve

Figure 3.5: A $3D$ position trajectory for a forceps blade. Two different trajectories are presented. Figure (3.5a) is a trajectory from an experienced user while Figure (3.5b) is a trajectory from an inexperienced user. Both have distinct number of samples. Units (cm.)

causing a high local curvature with respect to the circle approximation. Such intersections are the key segmenting points for the curve.

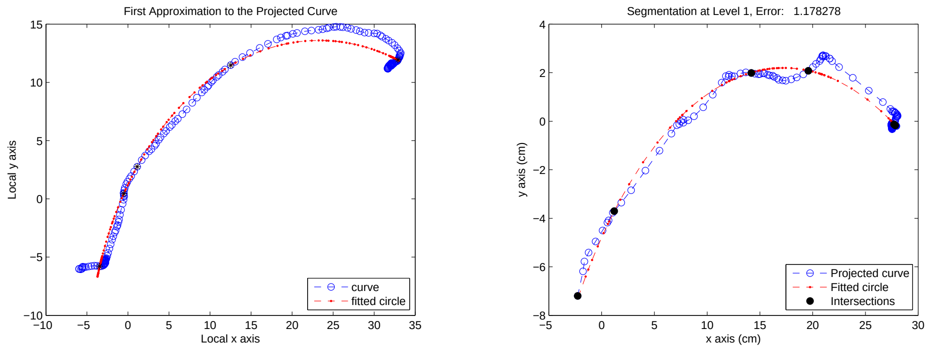
The curve is naturally segmented from the intersection points from the curve and the fitted shape. At each level, the local curvature minima points allows us to establish boundaries for which a new circle is fitted. The points are grouped by considering the local minima as boundaries for which in between sequential points belong into a group. This situation is clearly illustrated at Figures (3.7) and (3.8). The number of segments n_s in both figures is similar yet not equal. It is not expected that the number of sub-segments should be the same, only that the number of groups is similar.

The natural segmentation process at different iteration or decomposition level is illustrated at Figures (3.7a), (3.7c), (3.8a), and (3.8c). Different levels of analysis, from the initial fitted circle to all subsequent branch fittings up to the fifth iteration, are shown for an experienced gesture and quantized with the local reference curvature. Figures (3.7b), (3.7d), (3.8b), and (3.8d) correspond to the same curvature analysis for an inexperienced student after receiving instruction and being allowed to practice the same gesture.

The rapid convergence of the algorithm is illustrated at Figures (3.9) and (3.10). It is noticeable that only after two iterations, curvature analysis at level 3, the gesture has been segmented into several clusters containing points with approximately the same curvature. The reconstruction, using only curvature radii approximations for the curve at Figures (3.9c) and (3.9d), is piecewise continuous within each cluster but discontinuous at the segmenting points.

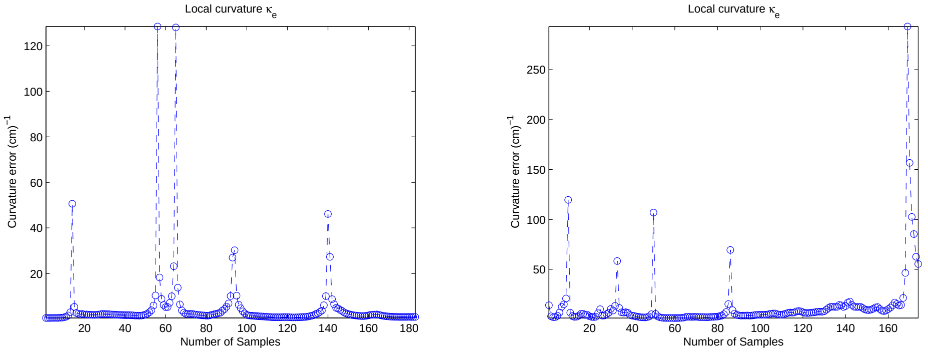
Further analysis using circle fitting increases the number of clusters segmenting the gesture curve. However, when the number of clusters increases, the number of samples per cluster decreases very rapidly. The significance is that even one can reduce the error to the curvature value per each sample, the generalization of the curvature to a group of samples is lost. The curvature quantization per sample is further specialized, but the curvature generalization is lost at the same time. This is illustrated at Figures (3.8) and (3.10) where at further iterations the fitting error is reduced, the number of clusters is increased and the number of samples in each cluster is small.

The resulting estimated curvature signatures, illustrated at Figures (3.11) and (3.12), are similar to a Haar wavelet analysis of a $1D$–signal [59]. Figure (3.11) illustrates the estimated curvature signature for the experienced gesture at different levels of analysis while Figure (3.12) illustrates the same estimate for an inexperienced user after explanation and practice. For clarity, we present five signatures, one per iteration, where Figure (3.11a) and (3.12a) are the initial circle fitting and Figures (3.11e) and (3.12e) are the outputs for last iteration of the algorithm. A negative sign in the value of the estimate for



(a) Plane projection for experienced user trajectory at Figure (3.5a) and fitted circle at the first level of approximation.

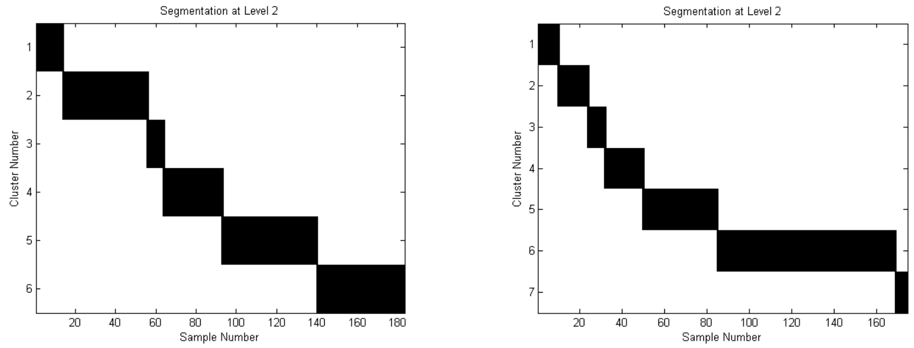
(b) Plane projection for inexperienced user trajectory at Figure (3.5b) and fitted circle at the first level of approximation.



(c) Local curvature for experienced user at level 1.

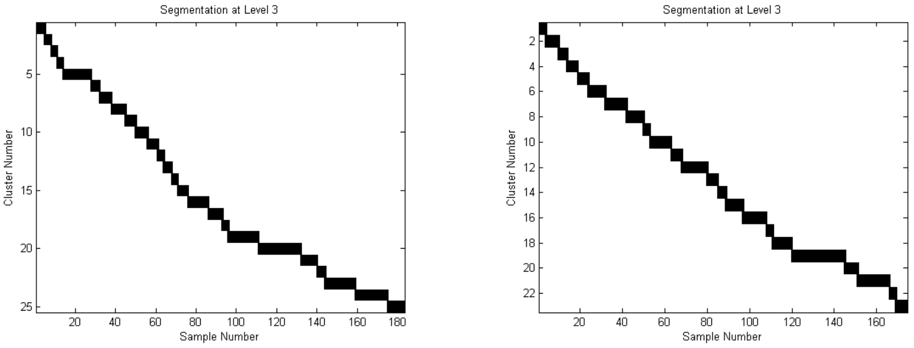
(d) Local curvature for inexperienced user at level 1.

Figure 3.6: Thin plate projection for the curve at Figure (3.5). The randomly sampled data introduces arc-length differences in the derivatives. The local curvature is measured from the fitted circle to the corresponding point in the curve. First approximation level.



(a) Experienced user at level 2 clustering. Total clusters: 6

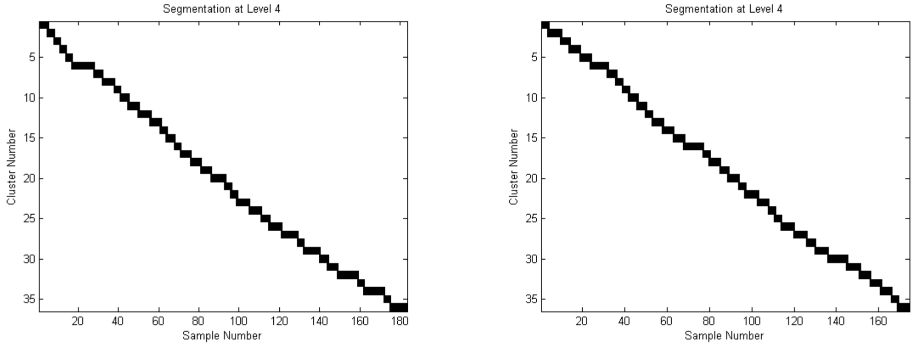
(b) Inexperienced user at level 2 clustering. Total clusters: 7



(c) Experienced user at level 3 clustering. Total clusters: 25

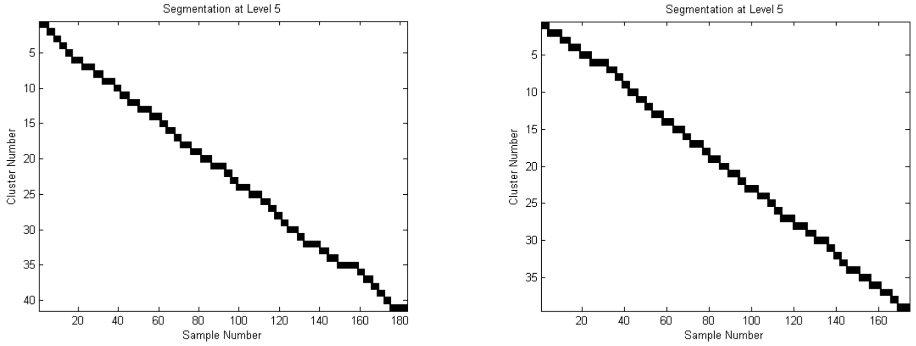
(d) Inexperienced user at level 3 clustering. Total clusters: 23

Figure 3.7: There is a rapid cluster progression as the algorithm progresses to smaller scales. At coarser levels, levels 2 and 3, the clusters contain large number of points, but the curve fitting returns large errors. The number of discrete states is low.



(a) Experienced user at level 4 clustering. Total clusters: 36

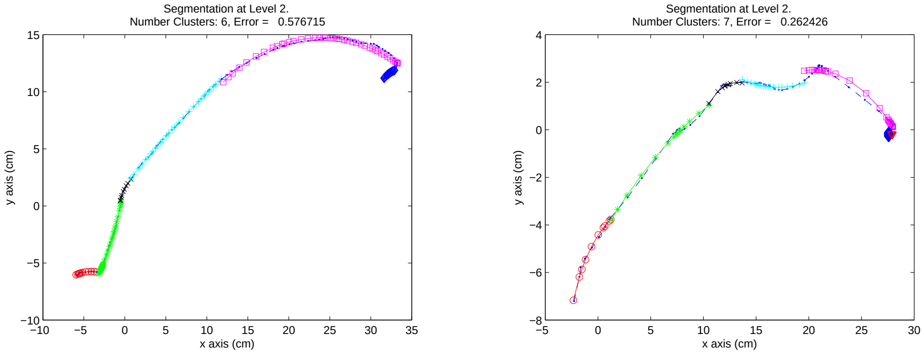
(b) Inexperienced user at level 4 clustering. Total clusters: 36



(c) Experienced user at level 5 clustering. Total clusters: 41

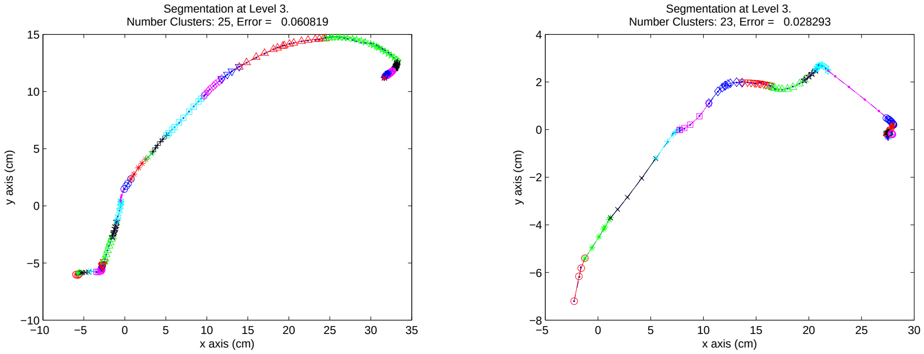
(d) Inexperienced user at level 5 clustering. Total clusters: 39

Figure 3.8: Rapid progression in the number of clusters. The clusters contain smaller and smaller sets of points as the algorithm progresses into finer scales. After reaching level 5, the clusters barely contain enough points to fit a curve, the finest level has been reach. The number of discrete states is high, probably inadequate for a HMM.



(a) Experienced user at level 2 segmentation.

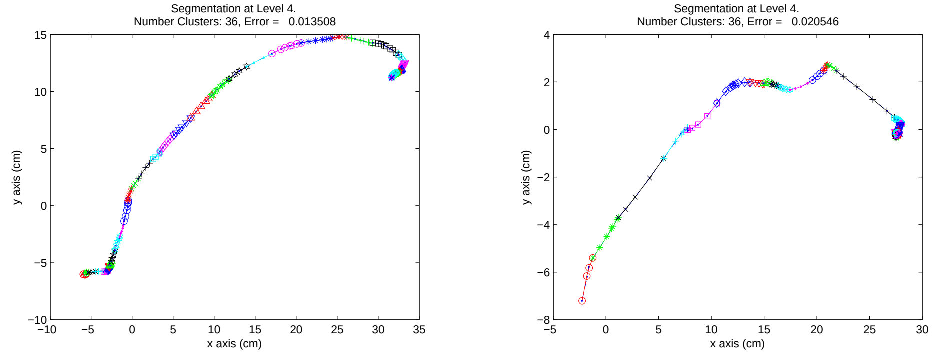
(b) Inexperienced user at level 2 segmentation.



(c) Experienced user at level 3 segmentation.

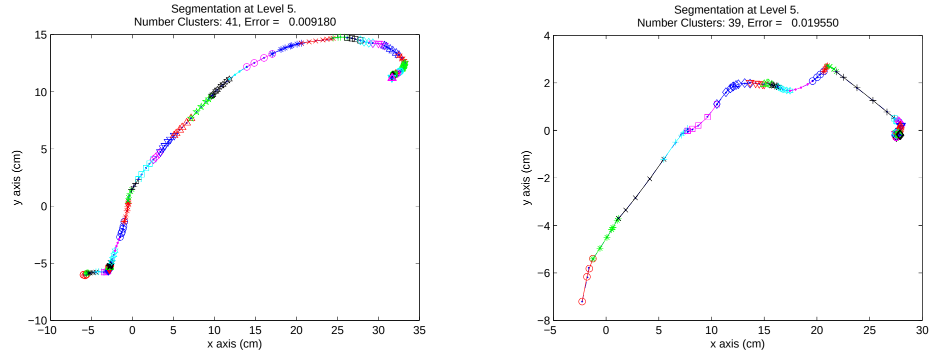
(d) Inexperienced user at level 3 segmentation.

Figure 3.9: At coarser levels, the approximation error for the circular basis function is relatively high, but they offer a better generalization with fewer discrete states. At level 3, a balance is reached between the approximation error of the basis function and the number of discrete estates that is more manageable than at finer scales.



(a) Experienced user at level 4 segmentation.

(b) Inexperienced user at level 4 segmentation.



(c) Experienced user at level 5 segmentation.

(d) Inexperienced user at level 5 segmentation.

Figure 3.10: At finer levels, the cluster numbers increase. However, the approximation error reaches a plateau. The reason is that the circular basis functions have reach its limits as shown at Figure (3.10d).

a single sample indicates its relative position with respect to the previous iteration. It simply indicates that the curvature radius estimate has a lower value from the intersection point perspective. The spikes at the curvature signature estimates are present because these are the boundary locations indicating high local curvature values at different iterations.

3.4.1.4 Advantages and Disadvantages

The main advantage of the algorithm is that the curvature estimate has a lower and an upper bound. The estimates are also a function of the scale since they are inversely proportional to the radii at different decomposition levels. The curvature scale analysis is analogous to Haar wavelet multi-scale analysis for curvature. Each decomposition level is an invariant signature for curve on the plane at a concordant neighborhood size w projection.

The algorithm output can be used to apply direct correlations between two different signatures. Each level in the decomposition is invariant in the arc-length sense. A correlation between the same levels yield a similitude measure providing a distance metric between the curves.

An important disadvantage of the method is the data structure in the comparison. The tree structure allows us to improve the multi-scale decomposition of the curvature signature. However, it does not guarantee that the same tree shape will remain for similar curves. Slightly different trees may represent the same curve with slight different branch shape and numbers. In this sense, the tree structure is unreliable. Consequently, it is preferable to maintain a low level decomposition with shallow levels for recognition purposes while the tree uncertainty is treated as noise at higher levels.

3.4.2 Anisotropic Kernel Filtering

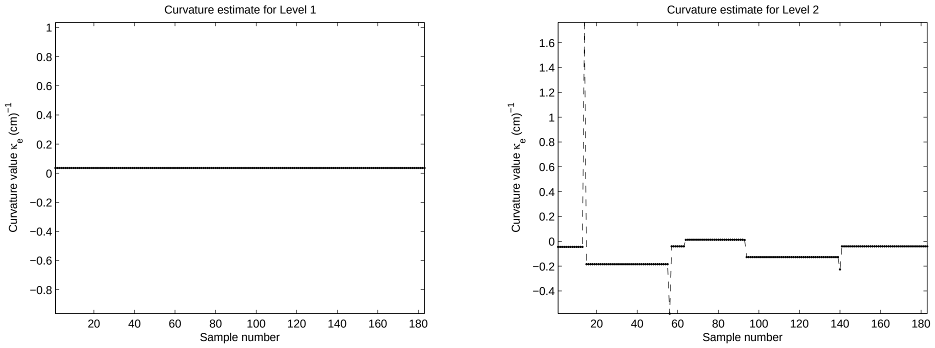
We start by defining two general representations $\boldsymbol{X}$ and $\boldsymbol{X}_M$ for two discrete n– and $(n-1)$–dimensional spaces where $\boldsymbol{X}_M \subset \boldsymbol{X}$. Both are vector representations of an extrinsic and an intrinsic spaces respectively.

The discrete extrinsic space is a set of points $\boldsymbol{x}$ of n-dimensions:

$$\boldsymbol{X}^{\mathsf{T}} = \Big\{\boldsymbol{x}[1], \boldsymbol{x}[2], \ldots, \boldsymbol{x}[i], \ldots, \boldsymbol{x}[m]\Big\} \quad \forall \quad i = \{1, 2, \ldots, m\} \quad m < \infty. \tag{3.4.5}$$

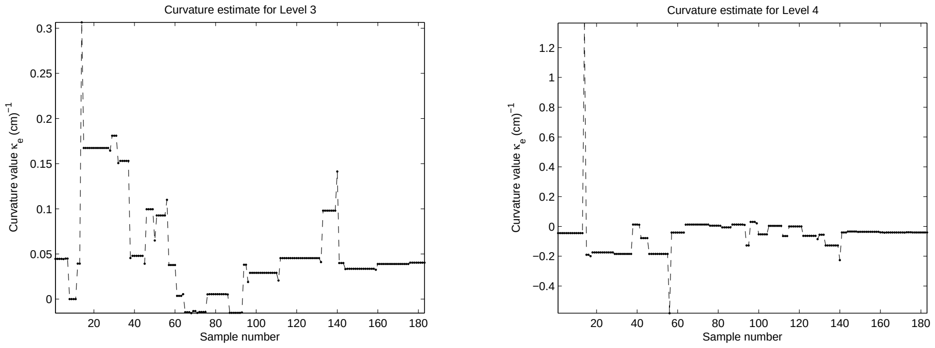
Each point in the space, $\boldsymbol{x}$, is represented as an n–tuple. In the other hand, an intrinsic space or manifold, is a discrete set of points $\boldsymbol{x}$ lying in the $(n-1)$–dimensional manifold M:

$$\boldsymbol{X}_M^{\mathsf{T}} = \Big\{\boldsymbol{x}_M[1], \boldsymbol{x}_M[2], \ldots, \boldsymbol{x}_M[i], \ldots, \boldsymbol{x}_M[m]\Big\} \in M \quad \forall \quad m < \infty \tag{3.4.6}$$



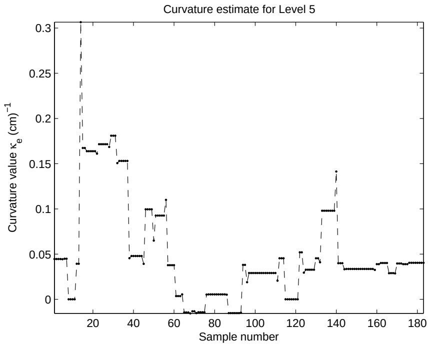
(a) Curvature estimate for level 1.

(b) Curvature estimate for level 2.



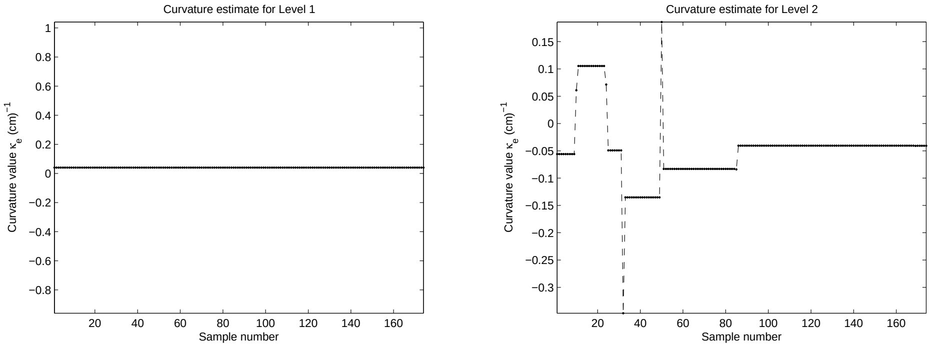
(c) Curvature estimate for level 3.

(d) Curvature estimate for level 4.



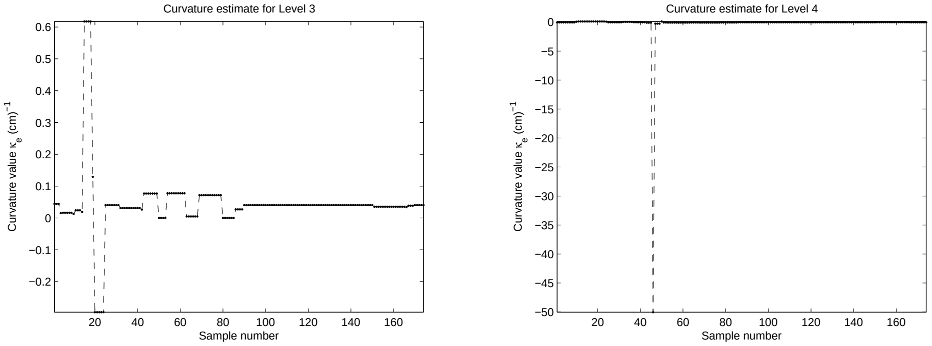
(e) Curvature estimate for level 5.

Figure 3.11: Discrete curvature estimates from fitting circles. The position of curvature values at finer levels depend on the coarser levels. However, a summation in the log-space produces a curvature profile containing coarse and fine scales. It presents a similar behavior as a Haar wavelet.



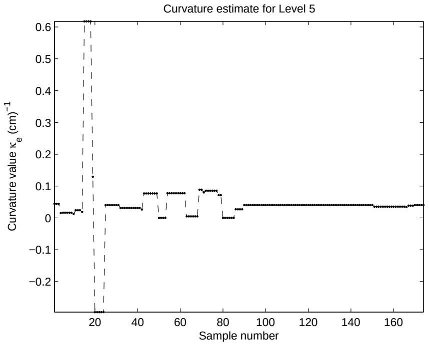
(a) Curvature estimate for inexperienced user at level 1.

(b) Curvature estimate for inexperienced user at level 2.



(c) Curvature estimate for inexperienced user at level 3.

(d) Curvature estimate for inexperienced user at level 4.



(e) Curvature estimate for inexperienced user at level 5.

Figure 3.12: Cumulative curvature profile from an inexperienced user. The estimates from the fitting algorithm rapidly and correctly localizes high curvature values at finer scales. Spikes in the finer scales indicate a sudden changes in direction, a situation that is not desirable.

where each point $\boldsymbol{x}_M$ is an $(n-1)$–tuple. The index i uniquely defines each point in both the space and the manifold. There are no repeated indexes and while the space may be infinite, the manifold is either finite or it has a closed form. One also assume that the manifold $\boldsymbol{X}_M$ is an invariant representation of an object in the $\boldsymbol{X}$ extrinsic space.

As in any other smoothing filter, the basic idea of the anisotropic kernel is to use a weighted average to smooth an object in $\boldsymbol{X}$. An important characteristic of the weighted average is that the kernel achieves signal smoothing using invariant distances. In the case of geometric signals, the expected invariance stands from the geometric standpoint. The smoothing process should not change an arrangement configuration or the expected output under a geometric transformation. In Euclidean spaces the natural relationships are distance between points and straight lines.

For geometries other than Euclidean where geodesic curves are analogs to straight lines, the geodesic distance provides unique scalar values to point clique. These unique scalars are averaged and then used to determine a smoother estimate for the expected central point in the neighborhood. In multidimensional geometric signal processing it is more adequate to refer to this type of filter as averaging or smoothing filter since one cannot define an intrinsic frequency domain as in Fourier analysis.

Cartesian spaces use the Euclidean distance as intrinsic metric. It is defined as an internal product. For a sequence of N discrete points $\boldsymbol{x}[i]$ in an Euclidean space, the geodesic metric is

$$d_g^2 = \sum_{i=1}^{N-1} (\boldsymbol{x}[i+1] - \boldsymbol{x}[i])^{\mathsf{T}} (\boldsymbol{x}[i+1] - \boldsymbol{x}[i]) \quad N \in \mathbb{Z} \tag{3.4.7}$$

also known as arc-length.

However, the definition of distance for other geometries Equation (3.4.7) may be inappropriate. The definition of a geodesic distance d_g is dependent of the manifold geometry and thus, anisotropic in nature.

3.4.2.1 Anisotropic Kernel and Gaussian Filters

The Gaussian filter is, in a sense, an isotropic or extrinsic filter. The weights do not depend on the manifold geometry as they are a function of an estimated covariance matrix $\boldsymbol{\Sigma}$ in the Euclidean space. The estimation of the covariance matrix is extrinsic to the points describing the manifold. Its reference frame is located outside the manifold.

The distance function for the Gaussian filter is the Mahalanobis distance. The matrix $\boldsymbol{D}$ contains the smoothed distance covariance from $\boldsymbol{x}[\lfloor \frac{\mathsf{w}}{2} \rfloor + 1]$ to each point $\boldsymbol{x}$ inside the window W:

$$\begin{aligned} &\boldsymbol{D}\left[\left\lfloor \tfrac{\mathsf{w}}{2} \right\rfloor + 1\right] = \boldsymbol{X}_{\mathsf{w}}^{\mathsf{T}}[i]\boldsymbol{\Sigma}^{-1}\boldsymbol{X}_{\mathsf{w}}[i] \\ &i = \{1, 2, \ldots, \mathsf{w}\}, \quad \mathsf{w} \in \mathbb{Z} < \infty. \end{aligned} \tag{3.4.8}$$

where $\boldsymbol{X}_{\mathsf{w}}$ is a difference distance matrix from points $\boldsymbol{x}$ starting at index i to the window center point $\boldsymbol{x}\left[\left\lfloor \frac{\mathsf{w}}{2} \right\rfloor + 1\right]$. The filter output $\boldsymbol{y}[i]$ is:

$$\boldsymbol{y}[i] = \frac{1}{((2\pi)^n\,|\boldsymbol{\Sigma}|)^{1/2}}\frac{1}{\mathsf{w}}\boldsymbol{O}_{\mathsf{w}}^{\mathsf{T}} \exp\left(-\frac{1}{2}\boldsymbol{X}_{\mathsf{w}}^{\mathsf{T}}[i]\boldsymbol{\Sigma}^{-1}\boldsymbol{X}_{\mathsf{w}}[i]\right)\boldsymbol{x}[i] \tag{3.4.9}$$

where

$$\boldsymbol{O}^{\mathsf{T}} = [\boldsymbol{o}_1, \cdots, \boldsymbol{o}_i, \cdots, \boldsymbol{o}_n], \qquad \boldsymbol{o}_i = \mathbf{1}, \ \ \|\boldsymbol{o}_i\| = \mathsf{w} \tag{3.4.10}$$

is a cumulation matrix of size $\mathsf{w} \times n$. The covariance matrix $\boldsymbol{\Sigma}$ is of size $n \times n$ with $n > 0 \in \mathbb{Z}$. It is a vectorial representation of the points inside a neighborhood window of size w. The shape of the window is considered at the moment of choosing the vectorization $\boldsymbol{X}_{\mathsf{w}}$ and it is defined at i as:

$$\boldsymbol{X}_{\mathsf{w}}^{\mathsf{T}}[i] = \begin{bmatrix} \boldsymbol{x}^{\mathsf{T}}[i] - \boldsymbol{x}^{\mathsf{T}}\left[i - \left\lfloor \frac{\mathsf{w}}{2} \right\rfloor\right] \\ \boldsymbol{x}^{\mathsf{T}}[i] - \boldsymbol{x}^{\mathsf{T}}\left[i - \left\lfloor \frac{\mathsf{w}}{2} \right\rfloor + 1\right] \\ \vdots \\ 0 \\ \vdots \\ \boldsymbol{x}^{\mathsf{T}}[i] - \boldsymbol{x}^{\mathsf{T}}\left[i + \left\lfloor \frac{\mathsf{w}}{2} \right\rfloor - 1\right] \\ \boldsymbol{x}^{\mathsf{T}}[i] - \boldsymbol{x}^{\mathsf{T}}\left[i + \left\lfloor \frac{\mathsf{w}}{2} \right\rfloor\right] \end{bmatrix}. \tag{3.4.11}$$

The matrix contains the difference vectors from points inside the neighborhood $\boldsymbol{x}[i],\ i = \left\{-\left\lceil \frac{\mathsf{W}}{2} \right\rceil, \ldots, -1, 0, 1, \ldots, \left\lfloor \frac{\mathsf{W}}{2} \right\rfloor\right\}$ to a central point, $\boldsymbol{x}\,[0]$.

The smoothing is provided by the covariance matrix $\boldsymbol{\Sigma}$. The main assumption is that a statistically significant number of points $\boldsymbol{x}$ in the space $\boldsymbol{X}$ are explained by the covariance matrix. The weight w at i is

$$w[i] = \frac{1}{((2\pi)^n\,|\boldsymbol{\Sigma}|)^{1/2}}\frac{\boldsymbol{o}_{\mathsf{w}}^{\mathsf{T}}}{\mathsf{w}} \exp\left(-\frac{1}{2}\boldsymbol{X}_{\mathsf{w}}^{\mathsf{T}}[i]\boldsymbol{\Sigma}^{-1}\boldsymbol{X}_{\mathsf{w}}[i]\right)\boldsymbol{o}_{\mathsf{w}} \tag{3.4.12}$$

In contrast to the Gaussian filter, an anisotropic kernel filter does not use the covariance matrix to smooth a value. The smoothing of the filter is solely determined by a scalar value acting upon the eigenvectors of covariance matrix of the anisotropic distance $d \mid \boldsymbol{x} : \mathbb{R}^n \mapsto \mathbb{R}$ in the corresponding manifold. As an example in Euclidean spaces, the distance function is the internal product:

$$d^2 \; |\boldsymbol{x} : \boldsymbol{x}^\mathsf{T}\boldsymbol{x}$$

$$w[i] = \frac{1}{\alpha(d(\boldsymbol{x}), \sigma)} \exp\left(-\frac{1}{2\sigma^2}\boldsymbol{x}_\mathsf{w}^\mathsf{T}[i]\boldsymbol{x}_\mathsf{w}[i]\right) \tag{3.4.13}$$

where $\alpha(d(\boldsymbol{x}), \sigma)$ is a normalization factor that depends on the values of the distance function $d(\boldsymbol{x})$ and the scaling parameter σ.

In order to extend the covariance of the anisotropic distance to other geometries as an internal product, one can apply geometric considerations about the manifold into Equation (3.4.13):

$$w[i] = \frac{1}{\alpha(d(\boldsymbol{x}), \sigma)} \exp\left(-\frac{1}{2\sigma^2}\boldsymbol{x}^\mathsf{T}[i]\boldsymbol{G}_\mathsf{w}^{-\mathsf{T}}[i]\boldsymbol{G}_\mathsf{w}[i]\boldsymbol{x}[i]\right) \tag{3.4.14}$$

where $\boldsymbol{G}_\mathsf{w}$ is a vectorial representation of a general windowed geodesic-distance difference matrix that includes information about the scalar curvature tensor for the manifold.

3.4.2.2 Curvature Scale-space and the Anisotropic Kernel Filter

We showed in Section 3.4.1 that a natural form of curvature scale-space emerges from recursively fitting a known-curvature parametric shape family along a three-dimensional curve. The analysis starts by initially fitting to all points in the trajectory in the least square sense and then locating the high curvature point values as natural segmentation points. Then recursively repeat the procedure until a desired error value is obtained or the number of points runs out. The scale-space thus produced has constant curvature values for different sections on the curve with different lengths. This curve analysis is only approximations to the real values of the curvature at each point in the curve. The advantage is that uses well known strategies but it misses the generality of an energy-based method.

In curvature scaling, the scale weight has to be modified without changing the curvature signature. In many ways the natural distance function for any space is the geodesic distance in Equation (3.4.14). The scalar curvature tensor is an invariant value closely related to the second derivative of the space. The significance is that the geodesic distance of any surface or any trajectory is a local geodesic function of a neighborhood. Closer points do affect the curvature value while farther points should not affect the neighborhood. The natural question is how do one choose the corresponding weighting values for the filter at different scale values.

The scalar curvature tensor has a projection matrix. If one analyze Equation (3.4.14), the matrix-vector product $\boldsymbol{x}^{\mathsf{T}}[i]\boldsymbol{G}_{\mathsf{w}}^{-\mathsf{T}}[i]\boldsymbol{G}_{\mathsf{w}}[i]\boldsymbol{x}[i]$ yields a squared scalar value. Using associative laws and dropping off the discrete index i, it easy to show that if:

$$\boldsymbol{a} = \boldsymbol{G}\boldsymbol{x}, \tag{3.4.15}$$

the scalar curvature tensor has the form of a distance function

$$d_g^2 = \boldsymbol{a}^{\mathsf{T}}\boldsymbol{a} = \boldsymbol{x}^{\mathsf{T}}\boldsymbol{G}^{-\mathsf{T}}\boldsymbol{G}\boldsymbol{x} \tag{3.4.16}$$

for each one of the points in the manifold. Each and every point on the curve or the surface is uniquely characterized by a geodesic distance denoted as d_g. If one places a reference frame onto the manifold, then Equation (3.4.16) must also include the Riemannian information $\boldsymbol{G}$ into the geodesic difference matrix as:

$$d_g^2 = \boldsymbol{a}^{\mathsf{T}}\boldsymbol{a} = \boldsymbol{x}^{\mathsf{T}}\boldsymbol{X}_{\mathsf{w}}^{\mathsf{T}}\boldsymbol{G}^{\mathsf{T}}\boldsymbol{G}\boldsymbol{X}_{\mathsf{w}}\boldsymbol{x} \tag{3.4.17}$$

where the geodesic difference matrix $\boldsymbol{X}_{\mathsf{w}}^{\mathsf{T}}$ may include either a small portion, or a complete manifold, by controlling the window size w. The specific geometry of the window is generalized by vectorizing the geodesic-distance difference matrix, so any kernel function $\boldsymbol{\Phi}(\mathsf{w})$ can be included.

By observing the covariance between the neighboring set of points in the manifold, one can gain an insight of the manifold's structure $\boldsymbol{X}$. The neighborhood is determined by a vector of size $\mathsf{w} \times 1$ with an odd number constraint $\mathsf{w} = 2n + 1$. The center point in the neighborhood $\boldsymbol{x}[i]$ defines our origin onto the manifold. It defines a window for a geodesic-distance difference vector $\boldsymbol{d}_{g,\mathsf{w}}$ such that:

$$\boldsymbol{d}_{g,\mathsf{w}}[i] = \begin{bmatrix} d_g\left[i\right] - d_g\left[i - \left\lfloor \frac{\mathsf{w}}{2} \right\rfloor\right] \\ d_g\left[i\right] - d_g\left[i - \left\lfloor \frac{\mathsf{w}}{2} \right\rfloor + 1\right] \\ \vdots \\ 0 \\ \vdots \\ d_g\left[i\right] - d_g\left[i + \left\lfloor \frac{\mathsf{w}}{2} \right\rfloor - 1\right] \\ d_g\left[i\right] - d_g\left[i + \left\lfloor \frac{\mathsf{w}}{2} \right\rfloor\right] \end{bmatrix} \tag{3.4.18}$$

and a geodesic-distance covariance matrix $\boldsymbol{D}_{g,\mathsf{w}}[i]$ from the manifold window projection matrix:

$$\boldsymbol{D}_{g,\mathsf{w}}[i] = \frac{\boldsymbol{d}_{g,\mathsf{w}}\boldsymbol{d}_{g,\mathsf{w}}^{\mathsf{T}}}{\sqrt{\boldsymbol{d}_{g,\mathsf{w}}^{\mathsf{T}}\boldsymbol{d}_{g,\mathsf{w}}}}. \tag{3.4.19}$$

The size of the geodesic-distance covariance matrix is $\mathsf{w} \times \mathsf{w}$. There is no assumption on the properties of $\boldsymbol{D}_{g,\mathsf{w}}$ except that is symmetric by construction. The geodesic-distance covariance matrix can be ill-conditioned depending on the structure of the manifold and the size of the neighborhood.

The geometric interpretation of the scaling the curvature appears when considering the extrinsic space as a $n+1$-dimensional space by generalizing the $(n-1)$–dimensional plane given by windowed-geodesic-distance covariance matrix as a projection:

$$\boldsymbol{P}_{\mathsf{w}} = \boldsymbol{D}_{g,\mathsf{w}} \tag{3.4.20}$$

where $\boldsymbol{P}_{\mathsf{w}}$ is the generalized projection matrix for the local neighborhood on a n-dimensional space. The key assumption is that the manifold lies in a $(n+1)$-dimensional space. By definition $\boldsymbol{P}_{\mathsf{w}}$ is singular and therefore is not invertible, but in exchange, it possesses some useful properties:

$$\begin{aligned} \boldsymbol{P}_{\mathsf{w}}^{\mathsf{T}} &= \boldsymbol{P}_{\mathsf{w}}, \\ \boldsymbol{P}_{\mathsf{w}}^{n} &= \boldsymbol{P}_{\mathsf{w}}. \end{aligned} \tag{3.4.21}$$

The smoothing process at different scales of the anisotropic kernel filter is explained by factoring the local projection matrix for a point in a n-dimensional manifold neighborhood. Using singular value decomposition to uncover the structure of the projection matrix $\boldsymbol{P}[i]$ for the local neighborhood:

$$\boldsymbol{P}_{\mathsf{w}}[i]\boldsymbol{V}_{\mathsf{w}}[i] = \boldsymbol{U}_{\mathsf{w}}[i]\boldsymbol{\Lambda}_{\mathsf{w}}[i] \tag{3.4.22}$$

$\boldsymbol{U}_{\mathsf{w}}[i]$ is the orthonormal extrinsic vector space, $\boldsymbol{\Lambda}_{\mathsf{w}}[i]$ is the singular value diagonal matrix and $\boldsymbol{V}_{\mathsf{w}}[i]$ is the orthonormal anisotropic vector space for the centered point $\boldsymbol{x}[i]$ in the manifold. The direction of the local extrinsic orthonormal vectors $\boldsymbol{U}_{\mathsf{w}}[i]$ depends on the choice of the adequate value for the corresponding $\boldsymbol{\Lambda}_{\mathsf{w}}^{-1}[i]$ steering the local anisotropic space $\boldsymbol{V}_{\mathsf{w}}[i]$, *i.e.*,

$$\boldsymbol{U}_{\mathsf{w}}[i] = \boldsymbol{P}_{\mathsf{w}}[i]\boldsymbol{V}_{\mathsf{w}}[i]\boldsymbol{\Lambda}_{\mathsf{w}}^{-1}[i]. \tag{3.4.23}$$

The effect of $\boldsymbol{\Lambda}^{-1}[i]$ is to maximize the effect of the anisotropic distances in the extrinsic space by choosing adequate scalar values. This is accomplished for the local neighborhood when the following condition applies:

$$\left(\boldsymbol{X}_{\mathsf{w}}^{\mathsf{T}}[i]\boldsymbol{U}_{g,\mathsf{w}}[i]\boldsymbol{X}_{\mathsf{w}}[i] - \boldsymbol{D}_{g,\mathsf{w}}[i]\boldsymbol{V}[i]\boldsymbol{\Lambda}^{-1}[i]\right)\ \boldsymbol{x}[i] = \boldsymbol{0}. \tag{3.4.24}$$

This set of normal equations degenerates into a scalar with no unique solution. However, Equation (3.4.23) shows that the family of solutions lie on the n-dimensional normal vector to the manifold plane given by the anisotropic distance covariance matrix $\boldsymbol{D}_{g,\mathsf{w}}$ at the centered point $\boldsymbol{x}[i]$. The projection matrix provides us with the general direction of the solution for the local neighborhood at the centered point.

The geodesic covariance matrix $\boldsymbol{D}_{g,\mathsf{w}}$ can be unbounded, meaning that the total energy of the anisotropic kernel filter may be undefined:

$$\sum_{i=1}^{\mathsf{w}} d_g^2[i] = \infty. \tag{3.4.25}$$

In order to avoid such a situation, a nonlinear kernel controls the geodesic distance values within the neighborhood. A reasonable choice for the kernel is the negative exponential function $\mathrm{e}^{-\left(\frac{x}{2}\right)^2}$. Then, our anisotropic kernel matrix is defined as:

$$\boldsymbol{W}_{g,\mathsf{w}}[i] = \boldsymbol{H}^{-1}(d_g, \mathsf{w})[i] \exp\left(-\frac{1}{2}\boldsymbol{D}_{g,\mathsf{w}}[i]\boldsymbol{V}^{\mathsf{T}}[i]\boldsymbol{V}[i]\boldsymbol{\Lambda}^{-2}[i]\right) \tag{3.4.26}$$

where $\boldsymbol{D}_{g,\mathsf{w}}$ is the anisotropic distance covariance matrix, $\boldsymbol{V}$ are the orthonormal anisotropic manifold direction vectors whose internal dot product $\boldsymbol{V}^{\mathsf{T}}[i]\boldsymbol{V}[i]$ is a diagonal matrix, and $\boldsymbol{H}(d_g, \mathsf{w})[i]$ (Uppercase Eta) is the matrix containing the singular values for the principal directions in the manifold. Further simplification on Equation (3.4.26) yields:

$$\boldsymbol{W}_{g,\mathsf{w}}[i] = \boldsymbol{H}^{-1}(d_g, \mathsf{w})[i] \exp\left(-\frac{1}{2}\boldsymbol{D}_{g,\mathsf{w}}[i]\boldsymbol{I}\boldsymbol{\Lambda}^{-2}[i]\right). \tag{3.4.27}$$

This shows that the singular value matrix $\boldsymbol{\Lambda}$ contains the scaling factors for the principal directions of the anisotropic covariance matrix $\boldsymbol{D}_{g,\mathsf{w}}$ for the neighborhood around the point $\boldsymbol{x}[i]$. In other words, one can manipulate the scaling factors of the principal directions that maximize the anisotropic smoothing for the data in $\boldsymbol{U}[i]$.

The normalization factor $\boldsymbol{H}(d_g, \mathsf{w})$ is a diagonal matrix containing the energy signatures for the directions of the local neighborhood. It is defined as

$$\boldsymbol{H}(d_g, \mathsf{w})[i] = \frac{1}{\mathsf{w}}\boldsymbol{O}_{\mathsf{w}}^{\mathsf{T}} \exp\left(-\frac{1}{2}\boldsymbol{D}_{g,\mathsf{w}}[i]\boldsymbol{V}[i]\boldsymbol{\Lambda}^{-1}[i]\right). \tag{3.4.28}$$

The diagonal vector $\boldsymbol{\eta}_{\mathsf{w}}$ in $\boldsymbol{H}(d_g, \mathsf{w})$ contains the total energy in the anisotropic manifold for a specific neighborhood size w, *i.e.*,

$$\boldsymbol{H}_{\mathsf{w}} = \begin{bmatrix} \eta_{\mathsf{w}1,1} & 0 & \cdots & 0 & 0 \\ 0 & \eta_{\mathsf{w}2,2} & \cdots & 0 & 0 \\ \vdots & & \ddots & 0 & 0 \\ 0 & 0 & \cdots & \eta_{\mathsf{w}m-1,m-1} & 0 \\ 0 & 0 & \cdots & 0 & \eta_{\mathsf{w}m,m} \end{bmatrix}. \tag{3.4.29}$$

The norm of $\|\boldsymbol{\eta}(g, \mathsf{w})\|^2$ is an important weighting signature. The reason is that once the normalization factors for all the manifold has been computed, the resulting vector diagonal $\boldsymbol{\eta}(d_g, \mathsf{w})$ preserves a complete description of the manifold that is also bounded and normalized at different window sizes.

One of our main contributions in this thesis is given by Equations (3.4.28) and (3.4.29). They are important and critical equations. They allow us to compare directly two curvature signatures, $\kappa'[l]$ and $\kappa''[l]$ in terms of curve power or energy per arc-length unit.

From a signal processing standpoint, the normalization factor $\boldsymbol{\eta}(\kappa, \sigma, \mathsf{w})$ defines the characteristic impulse response of a finite impulse response (FIR) filter, $\boldsymbol{\eta}_{\mathsf{w}}$, of size $m \times 1$ taps. $\boldsymbol{\eta}_{\mathsf{w}}$ is a vector containing the energy description of all the directions in the anisotropic manifold. Since the impulse response function characterizes the local geometry of the manifold at point $\boldsymbol{x}[i]$, the energy content is also invariant.

This alone provides the means to directly compare two distinctive curvature signatures. The filter characteristic summarizes all the necessary information to compare two different impulse responses by convolving the unknown impulse response into the known response filter. Now, one can quantify the signatures similitude in terms the power residuals based on a single anisotropic parameter: the curvature κ.

The general smoothing of the anisotropic kernel filter is defined for n-dimensions as:

$$\boldsymbol{y}[i] = \boldsymbol{H}^{-1}(d_g, \mathsf{w}) \left(\boldsymbol{X}_{\mathsf{w}}^{\mathsf{T}}[i] \boldsymbol{W}_{g,\mathsf{w}}[i] \boldsymbol{X}_{\mathsf{w}}[i] \right) \boldsymbol{x}[i] \tag{3.4.30}$$

and $\boldsymbol{W}_{g,\mathsf{w}}[i]$ as defined in (3.4.26) and of size $\mathsf{w} \times \mathsf{w}$. The matrix $\boldsymbol{X}_{\mathsf{w}}[i]$ is windowed the difference vectors for the local neighborhood in the extrinsic space, as defined in (3.4.11). The size of $\boldsymbol{X}_{\mathsf{w}}$ is $\mathsf{w} \times n$

3.4.2.3 Determining the Optimum Estimate $\widehat{\boldsymbol{\Lambda}}$

A direct solution to determining the values of $\boldsymbol{\Lambda}$ is to use singular value decomposition for each individual point in the manifold and solve the pseudo-inverse:

$$\boldsymbol{D}_{g,\mathsf{w}}^{-\dagger}[i] = \boldsymbol{V}[i] \boldsymbol{\Lambda}^{-1}[i] \boldsymbol{U}^{\mathsf{T}}[i]. \tag{3.4.31}$$

$\boldsymbol{D}_{g,\mathsf{w}}^{-\dagger}$ is the pseudo-inverse, usually obtain by means of the Singular Value Decomposition method. Its solution is computationally expensive, but it uniquely determines the adequate singular values for each direction in the anisotropic space.

Another, and less expensive, solution is to use a least squares approximation to a common value that is satisfactory for the majority of points in the anisotropic space. The values for $\mathbf{\Lambda}^{-1}$ are determined indirectly by observing the auto-correlation function between the curve $\boldsymbol{x}(l)$ in the anisotropic manifold and the resulting filtered signal $\boldsymbol{y}(l)$. The auto-correlation function will produce a maximum at the approximate value of the optimum estimate $\widehat{\mathbf{\Lambda}}$.

3.4.3 Multi-scale Curvature with the Anisotropic Kernel Filter

The behavior of the filter depends on selecting two main parameters: σ and k. The former defines the importance scaling for the anisotropic directions while the latter parameter defines the window support for the local neighborhood. Due to the similitude to the Gaussian filter, common sizes for the window are values like $k = 7, 9, 11, 13$. This window length ensures that, at least in theory, the support takes into consideration at least 99.9%, 99.994% or higher of the total energy content from the local neighborhood.

By using second order differences one can estimate the discrete derivatives $\dot{\boldsymbol{x}}$ and $\ddot{\boldsymbol{x}}$ in order to approximate the curvature value κ. However, the torsion parameter demands an estimate of the third derivative $\dddot{\boldsymbol{x}}$. Although our estimates for both the first and second derivatives of $\boldsymbol{x}$ are sufficiently stable, we found that the estimate for the third derivative is still numerically unstable and insufficient to obtain a reliable measurement estimate for the torsion, see Figure (3.2).

3.4.3.1 Anisotropic Kernel Smoothing

The algorithm is straightforward. Initially one relies on the auto-correlation coefficient ρ to provide a characterization for the filter's smoothing process. The pseudo-algorithm for the local-curvature anisotropic kernel filter is:

1. Provide initial estimates for filter tap number k and scaling factors $\widehat{\sigma}$ and $\delta\widehat{\sigma}$

2. Re-parameterize the curve $\boldsymbol{x}[t]$ as

$$\boldsymbol{x}[t] \mapsto \boldsymbol{x}[s] \quad \text{for } s = 1. \tag{3.4.32}$$

3. Estimate the initial curvature signature $\widehat{\boldsymbol{\kappa}}[0] \ \forall \ s$.

4. Calculate the curvature reciprocal vector

$$\widehat{\boldsymbol{r}}_{\kappa}[0] = \frac{1}{\widehat{\boldsymbol{\kappa}}[0]}. \tag{3.4.33}$$

5. Iterate with index i:

(a) Iterate for all s with index $[j]$:
   i. Calculate the filter window $\boldsymbol{X}_k[j] = \mathrm{kernel}(\widehat{\boldsymbol{r}}_k[i], \widehat{\sigma}[i], k)$
   ii. Use Equation (3.4.40) to calculate the anisotropic kernel filter $\boldsymbol{X}_k[j]$ and obtain the smooth vector $\widetilde{\boldsymbol{x}}$

$$\widetilde{\boldsymbol{x}}[i+1][j] = \boldsymbol{X}_k[j] * \boldsymbol{x}[i][j]. \tag{3.4.34}$$

(b) Determine auto-correlation coefficient $\rho[i]$ from the original $\boldsymbol{x}[i]$ and the smoothed cross-correlation coefficient $\widetilde{\rho}[i]$ from $\widetilde{\boldsymbol{x}}[i+1]$ and the original $\boldsymbol{x}[i]$. Take the first maximum.

(c) Termination condition:

$$\left|\rho[j] - \widetilde{\rho}[j]\right| < \text{threshold} \tag{3.4.35}$$

or

$$\max(\text{Iter}) < i. \tag{3.4.36}$$

(d) Calculate new curvature estimate $\widehat{\boldsymbol{\kappa}}[i+1]$

(e) Increase $\widehat{\sigma} = \widehat{\sigma} + \delta\widehat{\sigma}$.

(f) Increase index i

6. Return $\max \widehat{\sigma}(\boldsymbol{\rho})$

Only the initial curvature vector estimates $\widehat{\boldsymbol{\kappa}}$ and $\widehat{\boldsymbol{r}}_\kappa$ are important. In fact, both remain constant during the smoothing procedure since they are the best approximations to the manifold signature. The index i is an innovation index. At each step one use auto-correlation to determine an optimal value $\widehat{\sigma}[i]$ by locating the maximum correlation value from the previous value of $\sigma[i-1]$. Then, one proceed to recalculate the new curvature vector $\boldsymbol{\kappa}[i]$ for our gesture exemplar. At each step one consider the maximum auto-correlation coefficient and observe its evolution. An optimum value $\overline{\sigma}$ occurs when the coefficient reaches a maximum and then it starts decreasing. This is concordant with the rotation of the local values of curvature explained above. One uses $\widehat{\sigma}$ as the optimum value for the curvature signature in the cross-correlation with other exemplars.

3.4.3.2 Numerical Stability of the Anisotropic Kernel Filter

The anisotropic kernel filter, as stated in Equation (3.4.27), is numerically unstable when using curvature values κ for smoothing the extrinsic space. The reason is that the curvature covariance matrix

$$\boldsymbol{K}_{\mathsf{w}}[i] = \frac{\boldsymbol{\kappa}_{\mathsf{w}}[i]\boldsymbol{\kappa}_{\mathsf{w}}^{\mathsf{T}}[i]}{\boldsymbol{\kappa}_{\mathsf{w}}^{\mathsf{T}}[i]\boldsymbol{\kappa}_{\mathsf{w}}[i]} \tag{3.4.37}$$

is ill-conditioned for large curvature values and thereby, the numerical resolution is easily overflowed. However, one can eliminate this issue by using instead the curvature reciprocal or the curvature radius r_κ:

$$r_\kappa = \frac{1}{\kappa}. \tag{3.4.38}$$

The geodesic distance covariance matrix is now defined as

$$\boldsymbol{R}_{\mathsf{w}}[i] = \frac{\boldsymbol{r}_{\kappa,\mathsf{w}}[i]\boldsymbol{r}_{\kappa,\mathsf{w}}^{\mathsf{T}}[i]}{\boldsymbol{r}_{\kappa,\mathsf{w}}^{\mathsf{T}}[i]\boldsymbol{r}_{\kappa,\mathsf{w}}[i]}. \tag{3.4.39}$$

Such a substitution is viable because the curvature is an invariant scalar for the anisotropic space. When the curvature approximates a large numerical value, its reciprocal approximates zero. Thereby, one have only applied a conditioning factor and our new smoothing function is written as:

$$\boldsymbol{y}[i] = \boldsymbol{H}^{-1}(\boldsymbol{r}_\kappa, \hat{\sigma}, \mathsf{w}) \left(\boldsymbol{X}_{\mathsf{w}}^{\mathsf{T}}[i] \exp\left(-\frac{1}{2\hat{\sigma}^2} \boldsymbol{R}_{\kappa,\mathsf{w}}[i] \right) \boldsymbol{X}_{\mathsf{w}}[i] \right) \boldsymbol{x}[i] \tag{3.4.40}$$

with the new normalization matrix as

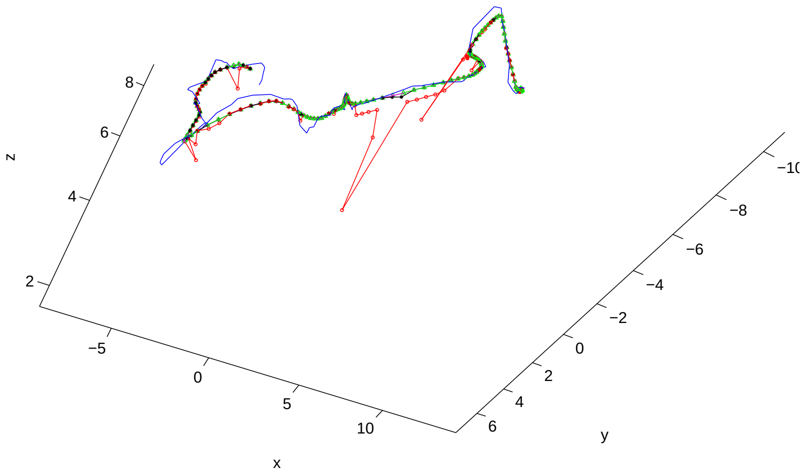
$$\boldsymbol{H}(\boldsymbol{r}_\kappa, \hat{\sigma}, \mathsf{w}) = \frac{1}{\mathsf{w}} \boldsymbol{O}_{\mathsf{w}}^{\mathsf{T}} \exp\left(-\frac{1}{2\hat{\sigma}^2} \boldsymbol{R}_{\kappa,\mathsf{w}}[i] \right) \boldsymbol{o}_{\mathsf{w}}. \tag{3.4.41}$$

and $\hat{\sigma}$ is our estimated value for the general steering factor $\boldsymbol{\Lambda}$.

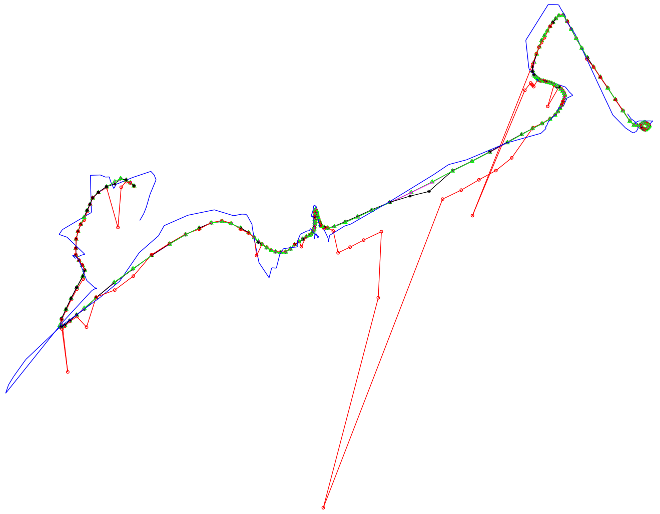
3.4.4 Parameter Estimation for the Anisotropic Kernel Filter

The problem of finding adequate values for the scalar value σ and window size k is illustrated at Figure (3.13). The residual error for the curvature approximation in the anisotropic kernel filter depends on the adequate parametrization. In the following, we determine the parametrization tuple (σ, k) for the anisotropic kernel filter on $3D$ Euclidean and quaternion curves. In order to determine a pair of general parameters, we used a $3D$ gesture data set containing six values for position and orientation.

We applied statistical data analysis on the curvature energy provided by Equation (3.4.41) and estimated the sum square error (SSE) of fitting residuals for different curves in a estimation data set. In order to create an estimation data set, several samples were randomly selected without replacement. A total of 24 samples containing position and orientation were used for the parameter estimation.



(a) 3D position curve. Viewing (Azimuth $= -32^o$, Elevation $= -28^o$)



(b) Close up panning for the same trajectory. Axes not shown.

Figure 3.13: A raw 3D position trajectory (continuous line) and the anisotropic smoothing for $\sigma = 0.01$ (red) , 0.1 (black), 1.0 (green) and 10.0 (magenta). The window size is $k = 11$. The nonlinear nature of the filter is clear at $\sigma = 0.01$ where several spikes appear at point where the curvature signature is close to indetermination.

In order to determine the most general interaction pair from different parametric values of (σ, k), we applied single factor ANOVA for a logarithmic sequence of values for $\sigma = \{0.01, 0.02, \cdots, 0.09, 0.1, 0.2, \cdots, 0.9, 1, 2, \cdots, 19\}$ and a linear sequence of odd window sizes $k = \{3, 5, 7, 9, 11, 13, 15\}$. We are interested in testing if different combinations of σ and k influence the sum of square error. Mathematically, the statistical hypotheses are stated as:

$$\begin{array}{ll} H_0: & \mu(\sigma_1) = \mu(\sigma_2) = \cdots = \mu(\sigma_n) \\ H_1: & \text{Not } \mu(\sigma_1) \end{array} \tag{3.4.42}$$

for σ and

$$\begin{array}{ll} H_0: & \mu(k_1) = \mu(k_2) = \cdots = \mu(k_n) \\ H_1: & \text{Not } \mu(k_1) \end{array} \tag{3.4.43}$$

for the window size. Due to the space geometries, the hypothesis are applied to each space, Euclidean and quaternion, independently.

3.4.4.1 Position Trajectories

The parameter interaction for σ and k for the position curves is shown at Figure (3.14). The behavior for σ and k is asymptotic with respect to the curvature SSE. It is apparent that the curvature SSE reaches a *plateau* for $\sigma > 0.2$ at a window size $k = 15$. However, the curvature SSE remains constant at different window sizes.

This behavior is corroborated by the box plots at Figure (3.15a) and Figure (3.15b). The p-value at the standard 0.05 significance level is $p_{0.05} = 0.716$ thus we cannot reject the null hypothesis H_0 for σ. This indicates that there are no significant statistical differences in the means for the SSE at different values of σ.

On the other hand, the probability value for the null hypothesis for the window size factor is $p_{0.01} = 4.52 \times 10^{-6}$, indicating a strong dependency of the SSE to the window size k and rejecting the null hypothesis. This is clearly illustrated at Figure (3.14) where the minimum SSE is presented at a window size $k = 5$. And although it is not shown at Figure (3.14), a window size $k = 3$ introduces a large SSE, measured at least of three orders of magnitude the error of a window size $k = 15$. This particular window size is not useful for the filter for this testing data set.

3.4.4.2 Orientation Trajectories

The parameter interaction for σ and k for the orientation curves is shown at Figure (3.16). The behavior for σ and k is also asymptotic with respect to the curvature SSE. It is also apparent that the curvature SSE reaches a *plateau* for

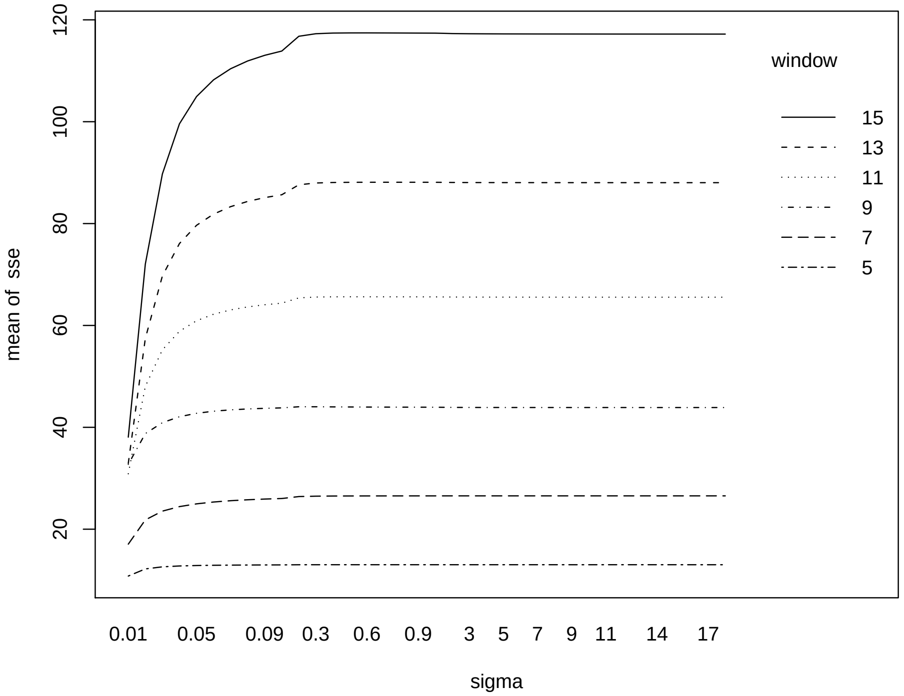
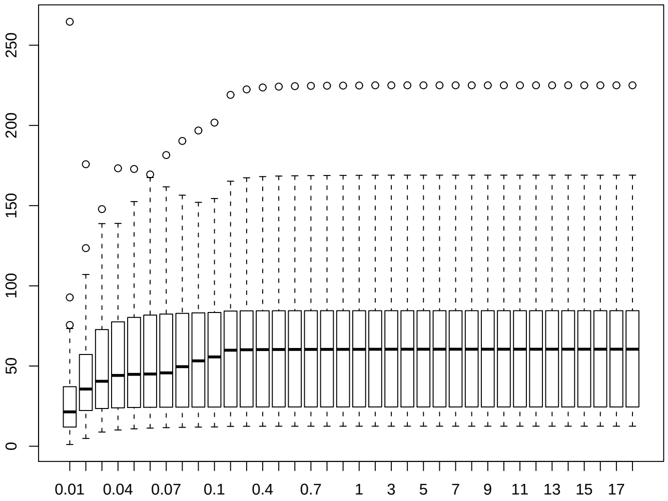


Figure 3.14: Parameter interaction in the intrinsic filter. On the abscissa, the value σ for different window sizes from 3 to 15 showing the minimum sum of the square error (SSE) for the curvature approximation. All the curves are asymptotic for $\sigma > 0.2$. The curve that minimizes the SSE is $k = 5$.

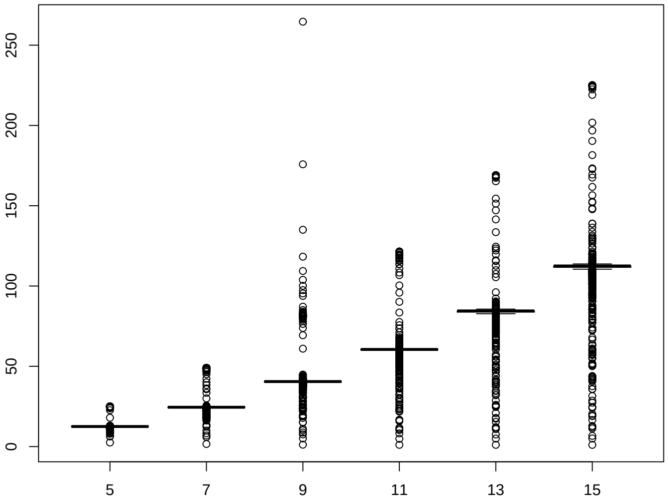
$\sigma > 0.03$ at a window size $k = 15$. The curvature SSE also remains constant at different window sizes.

Again, this is corroborated by the box plots at Figure (3.17a) and at Figure (3.17b). The p-value at the standard 0.05 significance level is $p_{0.05} = 0.914$ so we cannot reject the null hypothesis H_0 for σ, indicating that there are no significant statistical differences in the means for the SSE at different values of σ also.

The probability value for the null hypothesis for the window size factor is $p_{0.01} = 8.43 \times 10^{-6}$, so there is a strong dependency of the SSE to the window size k, rejecting the null hypothesis H_0 for the window size. This is illustrated at Figure (3.16) where the minimum SSE occurs at a window size $k = 3$. In this case, the very own nature of the quaternion space relates curvature directly to a window size of three points.



(a) σ means at different values



(b) k means at different window sizes

Figure 3.15: Box plots for different values of σ and window size k. The σ plot shows that there are no significant differences between data means for any $\sigma > 0.2$ with a p-value $p_{0.05} = 0.716$, accepting the null hypothesis.

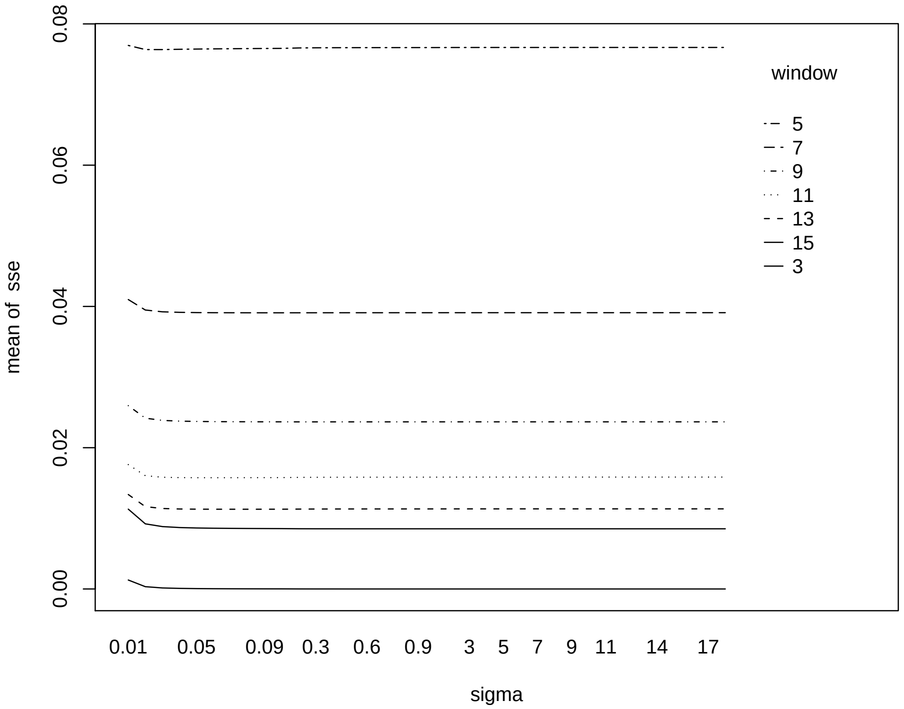


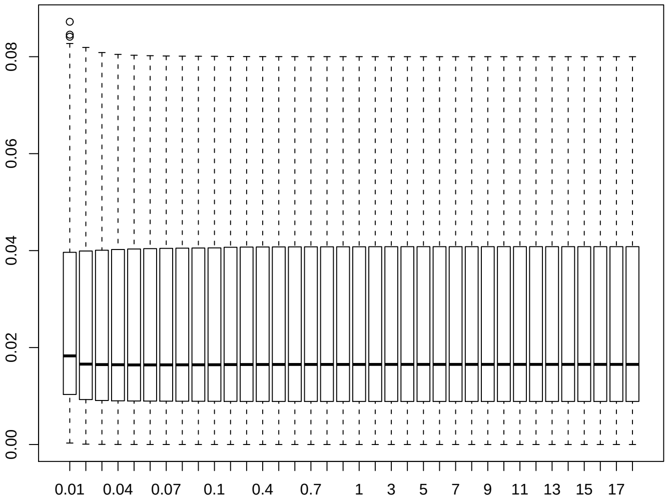
Figure 3.16: The minimum SSE error for the curvature residual for the orientation space is present for a window size $k = 3$ and any $\sigma > 0.09$.

3.4.4.3 Window Size vs. Scope

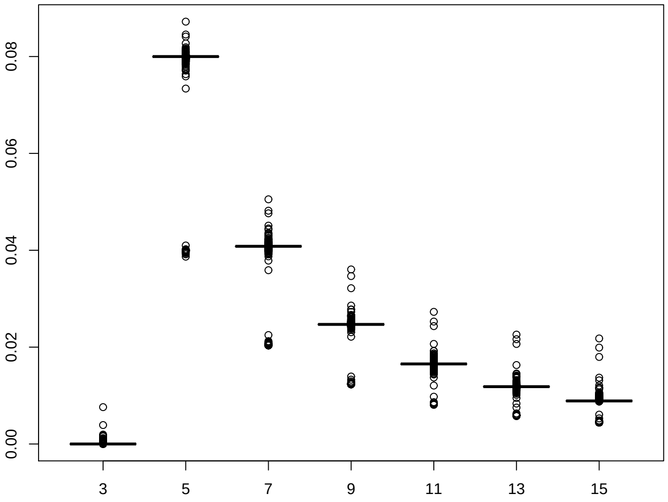
A particular point to be discussed is shown at Figures (3.15b) and (3.17b). The box plots for window size, for both position and orientation, indicate that there is no homogeneity in the variance for the curvature SSE. The reason behind this is anisotropic nature of the filter. The weights minimize, on different degrees, the curvature signature of the trajectory. By definition every curve has its own signature. Different window sizes k approach each signature differently and thus it is expected that the variance between signature to be non-homogeneous. The results for position and orientation for the data set are summarized in Table 3.2

Table 3.2: Values for window size k and scalar σ for $3D$ position and orientation trajectories.

	Position	Orientation
Window Size	$k = 5$	$k = 3$
σ	Any > 0.1	Any > 0.05



(a) σ means at different values



(b) k means at different window sizes

Figure 3.17: Box plots for different values of σ and window size k for the orientation trajectories. The σ plot shows that there are no significant differences between data means for any $\sigma > 0.03$ with a p-value $p_{0.05} = 0.914$, accepting the null hypothesis.

A characteristic of anisotropic smoothing is that the initial components for the trajectory may be smooth and analytic; however, that may not be the case once smoothed against the curvature signature. The filter minimizes the geodesic distance of the trajectory through out the use of different window sizes and reaches a limit for which the initial components may be distorted from their initial values. Nevertheless, the resulting curve in the space is smooth in curvature.

Also, since the filter does not considers the second curvature, the approximation to the initial curve lacks a degree of freedom that would it allow it to match the initial curve. In other words, the anisotropic kernel filter follows the curve in the curvature $\kappa(s)$ but not in torsion $\tau(s)$.

3.5 Summary

We introduced two new methods for geometric signal processing using the curvature κ as weighting parameter for gesture comparison. The first method, use of constant curvature as basis function to approximate general and local curvatures from a signal. The second method is based on a three-dimensional range geometric processing algorithm using anisotropic filters. Both methods share a common point, the use of eigenvalue decomposition in the form of singular value decomposition.

The first method using constant curvature decomposition is not a novel approach and it can be classified a top-down approach. The novelty lies in the application of the constant curvature approach to the problem of gesture comparison and the similarity to other scale-space approaches such as wavelets for signal processing. It relies heavily on a least squares curve fitting algorithm which can fail if a number of samples lie in a straight line. The strong point of the constant curvature fitting algorithm is that the number of iterations necessary to obtain a good estimate of the curvature is relatively low, usually two or three iterations for the exemplary gestures. It also provides a natural segmentation for the curve at point where the constant curvature and the local curvature of the signal cross providing a high curvature point in the residuals. Using such characteristics, gesture comparison can be done in a scale-space fashion with constant values of curvature and the residuals as analogous to the high frequency values.

A major advantage of the constant curvature decomposition over the anisotropic kernel approach lies in its direct application to rotation sequences with quaternions. Unitary quaternions are intrinsic spherical spaces with constant curvature geodesics. Complex orientation sequence trajectories sequences can be simplified using projections of the unitary quaternion space to the (ϕ, ψ)-

plane, as defined by the Equation set (2.5.29). Other considerations follow since the spherical projection is independent from the quaternion magnitude. The angular distance is kept separate. Without this independent value, the spherical projection is useless to characterize the rotation sequence.

The second method, anisotropic kernel filter, is a bottom-up approach in which the similarities between two arc-length vectors are measured in terms of a weighted average from the curvature distances of $\boldsymbol{x}(s)$. We analyzed the relationship between combinations of different values for the parameters (σ, k). For a particular data set, we concluded that the steering scalars σ have no major weight in our multi-scale curvature approach. The deciding factor is the window size k. Different geometries requires different window sizes.

Chapter 4

Study Case and Results

Gesture comparison main objective is to assess how the motor dynamics of a gesture are performed and passed on between individuals. Generally, the assessment of motor dynamics is done using indirect, subjective methods where apprentices are evaluated under the guidance of experts.

Medical training has unique requirements where individuals have to learn specific gestures and techniques to be applied in specific circumstances. Yet, the pending question is how to quantitatively assess how well a new practitioner acquires and continues to improve the various skills necessary to perform his work. As previously discussed, the main difficulty in assessing gesture motor skills lies in comparing three-dimensional gesture trajectories in space. The current approach relies on subjective assessments from experts that, frequently, are difficult to translate into quantitative measures of performance. Even when assessing skill differences between experts, it is difficult to quantitatively measure motor skill differences between them.

There has been an increased interest to quantitatively evaluate gesture in recent years. Due to both ethical and practical reasons, it has become desirable to use of simulators or emulators for training. The two main issues that simulators attempt to solve are: emulate real life-like situations and quantify an individual's skills solving specific situations.

In this chapter, we will evaluate how the proposed methodology can be applied to quantitatively assess the skill of obstetric surgeons for the use of forceps. We hypothesize that specific three-dimensional gestures and kinematic state sequences cannot only be recognized but also be compared to one another by using standard signal processing applied to invariant differential geometry representations of the three-dimensional gesture trajectories. Our primary hypothesis is tested in the gesture evaluation phase for a medical simulator, specifically a childbearing simulator.

BirthSIM is a robotic medical simulator system developed at the Ampère Laboratory at INSA of Lyon, France [138] [139] [140]. In our case study, the objective is to test our methodology for which one can compare the three-dimensional trajectories for the blade placement gesture of both right and left forceps blades placing onto the targeted area. An important constraint is time independence. Time independence refers to the fact that gesture analysis requires us to compare how similar the gesture is against a pattern. The hypothesis is that the execution of a specific gesture by an expert has a short execution time, yet, in the case of an apprentice, the gesture execution time is large due to ongoing learning.

4.1 Evaluating Obstetric Reaching and Grasping Gestures

One may think that the use of forceps is an outdated exercise. Mainly, due to the availability of new procedures for childbearing distress such as C-section or vacuum cup extraction. However, the technique is taught as a failsafe emergency procedure when the latter options are not possible.

In obstetrics, forceps are classified into three taxonomical groups: outlet, mid-cavity and rotational. Each type is appropriate for specific situations. Each one requires different levels of expertise. A general description of forceps structure is one of a pincer instrument composed of two independent flat-flanked, curved blades. The blade tips are shaped as a elongated spoons. The blade tip is known as the curve. The curve has a other specializations depending on the grasping region: cephalic or pelvic curve. The second structure in the forceps are the handles. The length and retraction action of the handle is designed with the type of intended reaching area in mind. Forceps with short handles are usually intended for use in the perineal and the vaginal region when the fetus is crowning at the vulva. Forceps with medium length handles, called also mid-cavity forceps, are intended to engage the fetus at the pelvic brim in the birth canal. Forceps with longer handles are intended to reach the fetus at the mid-cavity region. A third characteristic to be considered in forceps is related to the rotation action intended for the rotation and repositioning of the fetus. This type of handle provides a sliding action besides the grasping and clamping action.

Forceps blades are clamped and secured at one junction point in the handle section. The position and securing method of this clamping point varies from design to design, but the end purpose is to provide a mechanical pincer-like action so the targeted regions can be mechanically grasped and secured.

4.1.1 Forceps Application Procedure

The application of forceps has three major stages: placing, clamping, and rotation and traction. Placing the forceps involves various specific techniques that have to be passed on to new trainees. It is a critical phase in the use of obstetric forceps. In the case of outlet cephalic forceps, the obstetrician first finds the sagittal plane, by either visual or tactile inspection of the cranial sagittal fissure in the crowning fetus. This stage is important to locate the mento-vertical line. The mento-vertical line is a reference line located between mid-section of the temporal bone in the cranium, the mandibular condyle and the mandibular angle. The target placement area for the cephalic forceps is located along the mandibular angle and the temporal bone. Both blades are placed onto the opposite sides of the cranium, separately. The blades placement in the correct position is critical to avoid tissue damage to the fetus during the rotation and traction phase.

At the clamping stage, the blades are mechanically secured at the pincer's pivoting point. Depending on the forceps design, the pivoting point may have one or two degrees of freedom, allowing for a gentle rotation of the fetus' skull or just by securing the grasping points at the placement areas. At the grasping stage, just before clamping, the obstetrician must check that the pressure is correctly applied to the grasping areas, or it may cause damage to soft tissues or even may cause neurological damage.

Finally, the traction stage is performed once the forceps have been mechanically secured onto the correct regions. In order to expedite the fetus' expulsion, the obstetrician applies a gentle traction and/or rotation action during the contraction phase of the labor. Obstetricians have to learn how to synchronize the traction motion along the contraction phase to avoid unnecessary stress for both the mother and the fetus. If traction is applied during the relaxation phase, the action may cause tension on the soft tissue that may create further damage to the fetus.

4.1.2 BirthSiM Simulator

BirthSiM is an anatomically correct robotic simulation system. The mechanical core of the simulator is a pneumatic piston with two degrees of freedom, a linear displacement and a rotation axis around the piston's shaft. The piston is terminated with an anatomically correct dummy of a fetus cranium. The birth canal surrounding the piston shaft is also completed with realistic anatomical reference points that obstetricians use to orientate themselves when locating the forceps. The perineum region is also realistic, presenting both labia majora and minora, see Figure (4.1).

The expulsion motions of the birth canal are simulated by small displacements of the pneumatic piston. A previous calibration was performed to simulate contraction motions as realistic as possible. Different orientations of the piston's end effector simulate different delivery orientations for the fetus cranium. This allows obstetricians to practice the different referencing techniques, both visual and tactile, at distinct stages of the labor. The realism and correctness of the fetal cranium is indispensable to locate the correct grasping points along the temple-mandibular-angle line.

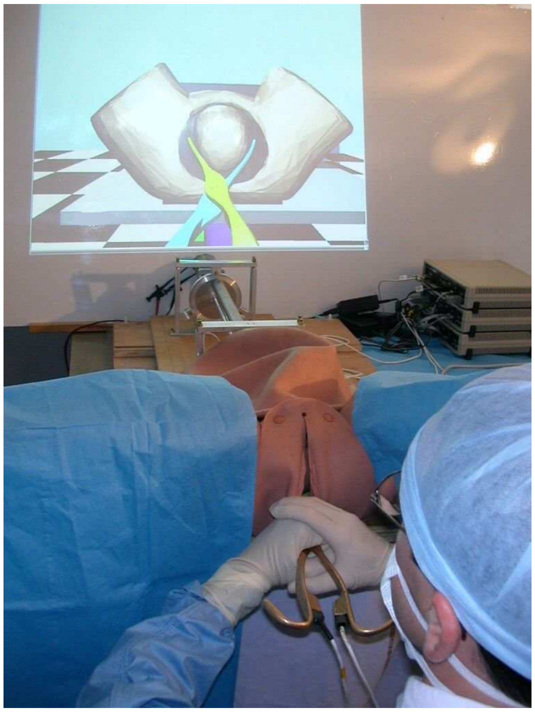
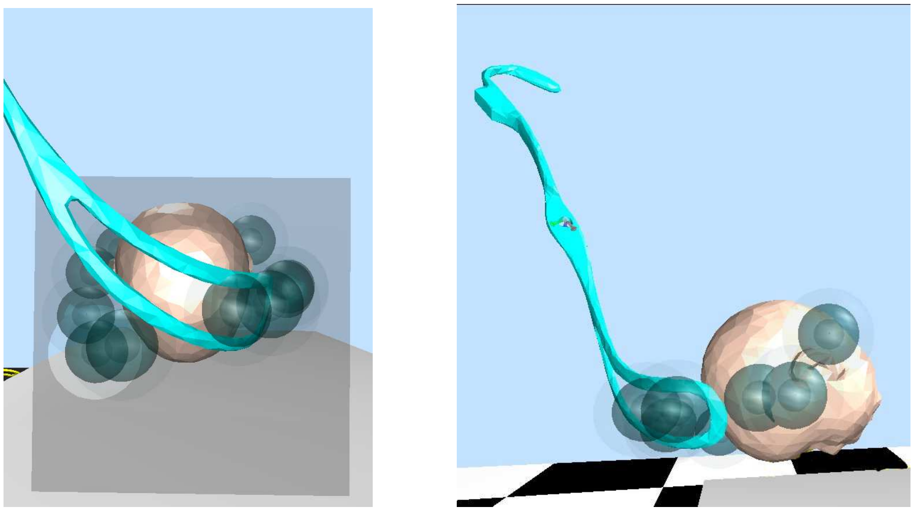


Figure 4.1: Robotic application BirthSIM

4.1.2.1 Interaction with BirthSIM

BirthSIM process data in real time. Both experts and apprentices can observe the position and orientation of each blade using a two-dimensional display of the region. The graphical interface uses a low-resolution polygonal approximation for the robotic elements in the emulator by using actual information from electromagnetic sensors tracking the forceps and pneumatic piston position and orientation.

A first approach to help students to follow a correct trajectory path is to present a series of goal positions to be reached. At each position, a spherical marker representing a point in space for which the blade has to go through, is placed along the chosen optimal trajectory to reach correctly the location at the fetus' cranium. The positioning of the markers was done by dividing an averaged expert trajectory in equal length segments. Different projective views are possible, as illustrated at Figures (4.2a) and (4.2b). The sphere sizes are reduced in order to increase the difficulty at different training stages. At this point, the optimal trajectory is just a visual aid for students and not a comparison between experts and juniors.



(a) Transverse plane view **(b)** Sagittal plane view

Figure 4.2: Visual aids for different perspectives for the neonatal pose presentation and cue markers for both blade trajectories

4.1.2.2 Digitalizing Gesture

The position and orientation data from the forceps reaching gesture's are captured using electromagnetic receivers mounted at the handles of each blade, see Figure (4.3). The sampling rate for the motion capture system is of 50 samples/s per channel. In order to reduce the electromagnetic field distortion, the simulator's forceps are not actual obstetric forceps but they are made of non-magnetic bronze. Bronze has lower conductivity coefficient than copper which reduces the reflection of the electromagnetic field. Bronze has also greater tensile strength making the blades more brittle, so they have to be handled carefully.

For simplicity, each forceps is considered as a free moving object in $3D$-space. The actual location of the end-effector of a forceps is a complex kinematic chain. However, the kinematic chain remains hidden to the sensor and was calibrated by scanning the forceps with a laser scanner. Other assumptions had to be taken into account as well. Ideally, the sensor positioning should be at the tip of the forceps blade, but that would affect the handling and placing of the blade's curve on the targeted area. Instead, the sensor is placed at the forceps' handle. This introduces a trade-off between accuracy and practicality. The trade-off is the assumption that there are no deformations nor vibration between the blade's curve and its handle. Although such an assumption may not be entirely true, it is necessary in order to estimate the rigid transformation between the tip and the sensor's location. The coordinate transformation from the tip of the blade to sensor's coordinates is shown at Figure (4.3).

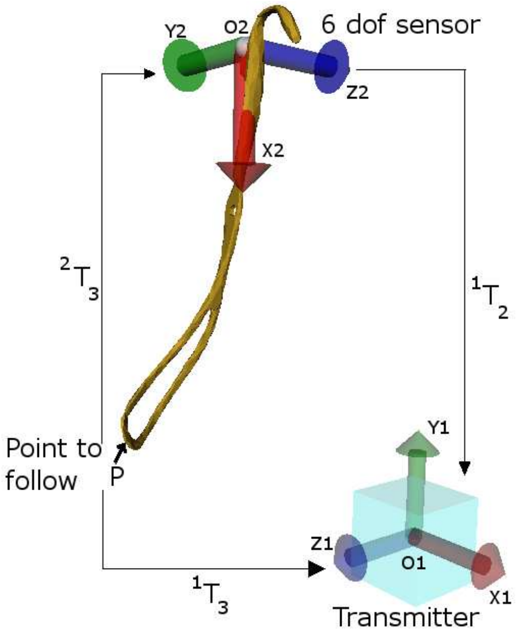


Figure 4.3: Reference frame transformation for both trajectory and orientation on a forceps blade.

4.1.2.2.1 Electromagnetic (EM) Tracking System

Electromagnetic (EM) sensors provide us with information for both position $\boldsymbol{r}$ and orientation $\boldsymbol{\theta}$. EM trackers measure the intensity of an electromagnetic field from a nearby source. The main advantage of EM trackers is their immunity to occlusion. However, their disadvantages are the need for a calibrated space and body restriction.

The EM tracker accuracy is highly dependent on the stability of the near–source EM field. An EM near–field is easily distorted by the presence of any ferromagnetic material [197], [109]. The reason is that metallic surfaces with magnetic properties deflect EM waves, effectively distorting the near–field. It is recommended to avoid the presence of any magnetic metal object during motion capture.

The distortion in the EM near–field source due to environmental elements such as equipment and materials lying near to the tracking space is called static field distortion. A certain amount of static field distortion for the EM trackers can be compensated for position. The compensation process requires an involved calibration procedure of a volume in tracking space [48], [67], [121], [122], [126]. However, the static field distortion for orientation has very little representation in the literature [135], [146].

A major disadvantage of earlier EM tracking systems is motion restrictiveness. Earlier versions of EM tracking sensors used heavy shielded cables thus restricting motion. In general, the tracking space is mapped to a global reference frame, usually located at the transmitter antenna. Newer version of the systems, such as the Mini-Bird from Ascension Technologies$^{\text{TM}}$, offer less motion restriction by using a wireless interface between sensors and portable data acquisition units. The sensor placement always considers the position and orientation correspondence to the tracked link.

4.1.2.2.2 Tracker Accuracy

EM trackers are sensitive to the presence of ferromagnetic materials in their surroundings. After a calibration procedure, the presence of such materials can be compensated and the static accuracy is improved at the calibrated points but diminished for volumes in between the calibration points. The calibration procedure was done for a mesh of 10 × 10 × 10cm. The average isotropic error at the center of the cube was 8 mm^3. The static isotropic error is then

$$\varepsilon = \frac{\text{max Volume Error}}{\text{Total Volume}} \times 100\% = \frac{8\ \text{mm}^3}{1000\ \text{mm}^3} \times 100\% = 0.8\%. \qquad (4.1.1)$$

A trajectory error is 0.8% of the arc-length in any direction.

A more accurate model is to consider an anisotropic error since the directional variations in the electromagnetic field will have different scalars for different directions. However, since the static error is lower than 1%, the experimental measurements are considered accurate with in a cube of 2 × 2 × 2 mm.

4.2 Analysis of $3D$-Trajectories

4.2.1 Analysis of the Three-dimensional Position

As previously mentioned, $3D$-trajectory analysis must have two important characteristics: viewpoint and time invariance. In the current dissertation the view invariance is tackled by switching global reference frames to local trajectory reference frames. Whereas the time invariance is solved by replacing the evolutionary time variable by a geometric equivalent: the trajectory arc-length.

As mentioned in the previous chapter, one can minimize the number of dimensions necessary to describe a trajectory in a three-dimensional Euclidean space using Frenet-Serret frames with two main variables, namely curvature κ and torsion τ. Numerically, both parameters, κ and τ are very sensitive to noise. Curvature and torsion estimates depend on numerical approximations of the spatial derivatives of the underlying trajectories.

Though in theory both curvature κ and torsion τ are essential to describe a three-dimensional trajectory. Our comparison procedure has to be limited to use the curvature κ alone. There are two justifications to do so based on a physical and a numerical constraints.

The physical, or intrinsic, constrains reduce the comparison dimensionality for only the curvature. In humans, the number degrees of freedom for the upper limbs is redundant, yet limited. This implies that any trajectory described by the conjunction of all possible rotational joints indeed does not present high values of torsion. The torsion of a curve is the rate at which the normal vector $\hat{\mathsf{N}}$ changes direction along the arc-length s. In other words, this means that one do not expect sudden changes by the rolling of the joints.

In a free-body, the torsion τ has to be considered in lesser degree of importance if the numerical approximations to the third order derivative are noisy. In practice, approximating the third derivative is tricky and it can only be done with acceptable numerical error for analytical functions. A possible solution is to use function fitting with thin plates for small trajectory sections in order to estimate the direction change. However, with this approach overfitting is always a latent problem. More often, approximations to the third derivative for discrete data contain large approximation errors, resulting in poor approximations of the torsion parameter τ (see Figure (3.2b)).

During the discriminating stage, a sampled trajectory curvature is calculated and then compared against a reference. To compute the corresponding curvature, the data are first expressed according to their cumulated arc-length s as described in Chapter 3. Three-dimensional gestures are first assessed according to their path length. Here, the cumulated arc-length (s) is used to guarantee time independence. The discrete evolution of the position $\boldsymbol{r}$ at time t is expressed as an array $\boldsymbol{r}(t) = [x(t),\ y(t),\ z(t)]^{\mathsf{T}}$. The cumulated arc-length $s(t)$ is calculated from an initial fixed time index $t = 0$. Then, the position data are expressed with respect to this parameter, *i.e.* $\boldsymbol{r}(s) = [x(s), y(s), z(s)]^{\mathsf{T}}$.

4.2.2 Analysis of the Sensor Orientation

In the case of forceps placing gesture, the evolution is traced by tracking the end-effector's position and orientation with respect time. Our end-effector is one forceps blade. The electromagnetic tracking sensor is physically placed at the handles. By assuming a rigid transformation, the physical coordinates of the sensor are translated and re-oriented at the tip of the forceps blade. We consider the blade as a free kinematic rigid body. In order to fully determine the corresponding analysis, both position and orientation trajectories are considered.

Metric for orientation trajectories are nonlinear. The definition of a distance is done in a four parametric space, the quaternion space. Quaternions have different projections that can be used to represent trajectories, see Section 2.5.1. Each one has advantages and disadvantages. However only two spaces are bounded, other projective spaces are unbounded. The meaning of bounded in this case refers to finite distances. Projective spaces such as ternions and polar projections are not bounded because at one point they project a point into infinity. In order to solve this issue, one can use a quaternion three-dimensional projection defined by another quaternion space (see Section (2.5.1.4)).

The projection is defined by a relative distance between two quaternions and a trajectory on the surface of a sphere. The $(\phi-\psi)$-manifold is the surface of a unit sphere. $\widehat{\boldsymbol{q}}_0$ is the initial orientation in the quaternion space.

$$\begin{aligned} &\text{Metric:} && \mathsf{\rho}[i] = d\left(\widehat{\boldsymbol{q}}\,[i], \widehat{\boldsymbol{q}}\,[i+1]\right) = \sin^2\theta[i] - \sin^2\theta[i+1] \\ &(\phi-\psi) && \phi = \arctan\left(\frac{z}{\sqrt{x^2+y^2}}\right) \\ & && \psi = \arctan\left(\frac{y}{x}\right) \end{aligned} \tag{4.2.1}$$

Unlike other projections, the angular parameters ϕ and ψ are completely independent from the value of $\mathsf{\rho}_i$.

The distance $\Delta\mathsf{\rho}_i$ admits an Euclidean simplification if $\Delta\theta[i] = \theta[i+1]-\theta[i]$ is less than 10 degrees:

$$\mathsf{\rho}[i] \approx \theta[i+1] - \theta[i] \quad \forall \quad \theta < 10 \ \text{deg.} \tag{4.2.2}$$

The arc-length s for a N point sequence in the quaternion space is then defined as:

$$s = \sum_{i=1}^{N-1} \mathsf{\rho}[i] \left(\Delta^2\phi[i] + \Delta^2\psi[i]\right)^{\frac{1}{2}} \tag{4.2.3}$$

which is nonlinear.

In order to compare two orientation trajectories one can use $2D$ curvature analysis since N tends to be large and $\mathscr{E}[\Delta r_i] = k, \quad K = Nk = \text{constant}$. These two conditions simplify the curvature analysis for orientations in the plane $(\phi - \psi)$:

$$s \approx K \sum_{i=1}^{N} \left(\Delta^2\phi[i] + \Delta^2\psi[i]\right)^{\frac{1}{2}}. \tag{4.2.4}$$

The approximation is only valid for a large number of points N and small increments between quaternion orientations $\Delta\theta[i]$. If such approximation is not valid, then the distance approximation is done using other equations.

The rotation sequence arc-length corresponds to the quaternion cumulative distance. Once it is known, data is treated in the same manner as it was for the position case:

1. Expression of data according to their cumulated arc-length.
2. Calculation of the quaternion curvature $\kappa(s)$ with respect to the cumulated arc-length.
3. Application of a sliding window for the anisotropic filter in order to smooth data, and
4. Calculation of the correlation coefficient from the smoothed curves of the studied path.

4.2.3 Algorithm of the method

The following algorithm resumes the general solution to compare two $3D$-gestures using curvature signatures: The algorithm is available for analyzing position and orientation data necessary to compare two gestures and thus to evaluate a gesture relative to a reference one set by the experts.

4.3 Experimental Results

The main issue of using curvature as a way to identify $3D$-trajectories is finding the true curvature signature at the corresponding neighborhood size. Often,

Table 4.1: General Algorithm for Curvature Comparison

1	$\boldsymbol{r}(t) = [x(t),\, y(t),\, z(t)]^{\mathsf{T}}$ $\boldsymbol{\theta}(t) = [\phi(t),\, \psi(t),\, \varphi(t)]^{\mathsf{T}}$	Data from sensors: position orientation data.
	$\Downarrow$	
2	$\boldsymbol{r}(s) = [x(s),\, y(s),\, z(s)]^{\mathsf{T}}$ $\boldsymbol{\theta}(s) = [\phi(s),\, \psi(s),\, \varphi(s)]^{\mathsf{T}}$	Data parametrization with respect to cumulative arc-length.
	$\Downarrow$	
3	κ_{pos} κ_{ang}	Compute curvature using second order finite differences with respect to the arc-length.
	$\Downarrow$	
4	$S_{pos}(s)$ $S_{ang}(s)$	Filtered data using the anisotropic filter or quantized curvature estimates using cicle approximations.
	$\Downarrow$	
5	ρ_{pr}	Compute correlation coefficient with displacements in arc-length to compare curvature signatures.

sampling noise and numerical approximations yield curvature values that are deceiving. Curvature estimates are obtained from series of consecutive points that may not be uniformly distributed in the space, creating numerical errors in the derivative estimates. Yet, any curvature estimate has no meaning if it is not related to the size of the neighborhood or scale. It is in this sense that time series smoothing for non-linear parameters is critical. A multi-scale approach similar to wavelet analysis is necessary to evaluate the main curvature and torsion of different section sizes.

In order to determine adequate values for a correlation function ρ_{pr} to measure similarity, we need to use a multi-scale approach to measure the curvature of different sections of a trajectory in $3D$-space. We experiment with two dynamic procedures in which several estimated values of $\hat{\kappa}_i$ are studied for a given estimated window size $\hat{\mathrm{w}}_j$ along the trajectory. The hypothesis is that distinct smoothed or quantized curvature values along the cumulative arc-length will create a template that yields different correlation values.

Our first dynamic method is similar approach to Haar wavelets. The algorithm is described in more detail at Section (3.4.1). It is a non-linear multiscale approach that approximates local curvature values by fitting constant curvature shapes to a $3D$-curve. It is a least squares approximation method. Unfortunately, the method proves to be difficult to implement for $3D$ orientation spaces. The difficulty essentially lies in the fact that quaternions are spherical spaces. Determining a principal direction for large signal scopes inside a sphere is difficult using planar projections.

Our second dynamic method is an nonlinear averaging smoothing filter. The filter coefficients are assigned according to an intrinsic measure for a distance d. Since intrinsic distances may be anisotropic in nature, the filter is also known as anisotropic smoothing filter. It can be best described from the signal processing standpoint, as a low pass discrete infinite impulse response (I.I.R.) filter. However, its Fourier spectral characteristic is non-linear and dependant on the geometry of the distance function.

4.3.1 The Experiment

In order to test the gesture evolution and the differences between experts and trainees, an experiment was conducted. The experiment consisted in analyzing the gesture of trainees at four training sessions:

1. Initial session,
2. Observation session,
3. Explanation session, and
4. Practice session.

During the initial session, trainees have not received any information on the handling technique of obstetric forceps. They have been asked, however, to execute the gesture after a brief informative session on how to locate anatomic key references and they have been shown the target areas on the fetus prop. At this particular point, the students receive no help from any expert nor they have seen the experts during the gesture execution and perform the gesture without any guidance.

In the observation session, trainees were allowed to observe an expert performing the handling gesture. Using the interactive features of BirthSIM, the trainees follow the expert's gesture remotely. No detail explanation of the handling technique was presented, only the expert's motion was observed.

At the explanation session, trainees were given a detailed explanation on the forceps handling gesture by an expert at the same time that the gesture was performed using BirthSIM. However, the trainees were not allowed time to performed the gesture on their own.

At the practice session, trainees had time to perform the handling gesture on their own accord after the previous session. At this phase, the experts were present only as observers, not as mentors.

4.3.2 Constant Curvature Fitting

4.3.2.1 Multi-scale Curvature Radii Profiles

The profiles for the radii of curvature were obtained using the constant curvature fitting algorithm with only three decomposition levels. Hereafter, we call such profiles Multi-scale Curvature Radii Profiles (MCRP) and denoted as $r_\kappa[i]$ where $i = \{1, \dots, M\}$ $M \in \mathbb{Z}$. A MCRP is composed of the individual values $r_{k_\ell}[i]$ for level $\ell = \{1, \dots, N\}$ $N \in \mathbb{Z}$. A summated MCRP is obtained with the product of the radii of curvature at the different decomposition levels:

$$r_\kappa[i] = \prod_{\ell=1}^{N} r_{\kappa_\ell}[i]. \tag{4.3.1}$$

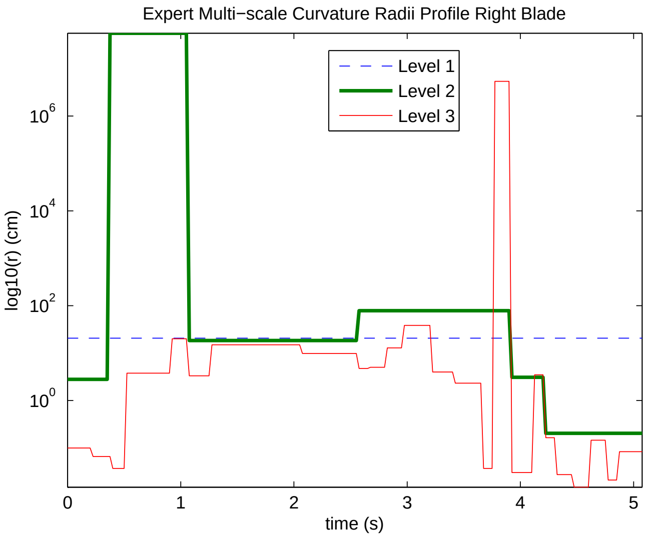
In order to emphasize and avoid round-off errors in the MCRPs, we applied a logarithmic transformation to each level ℓ such that

$$\log_{10}(r_\kappa[i]) = \sum_{\ell=1}^{N} \log_{10}(r_{\kappa_\ell}[i]) \tag{4.3.2}$$

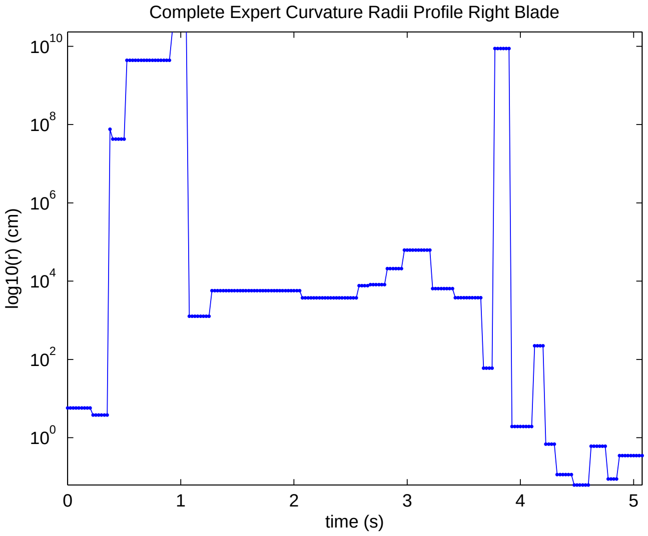
and the summated MCRP is kept as an extra level in the profile.

Figures (4.4) and (4.5) illustrate MCRPs for two experts' gestures for the right blade. As in any multi-scale representation the trade-off is individual accuracy for generalization. The quantized representation of a general radius of curvature for a number of points in a neighbor is inaccurate for individual points within the neighbor.

From a qualitative standpoint, Figures (4.4) and (4.5) also illustrate a typical conduct in the expert's gesture for forceps handling. There are three observable stages: initiation, approximation and placing. The initiation phase is characterized in the expert's gesture by combined high and low curvature radii values, as shown in the first third of the MCRP at decomposition levels 2 and 3. Between the second third or to half of the gesture execution, the middle decomposition level is characterized by low to middle curvature radii values while the detail level can contain middle to high curvature radii values, mostly due to corrections in the trajectory for approximation. During the final stage,



(a) Expert 1 Multi-scale curvature radii profile

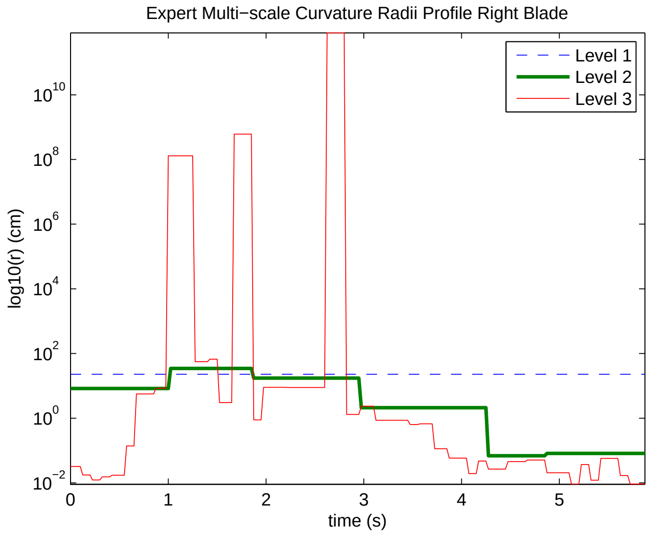


(b) Expert 1's Discrete curvature radii profile

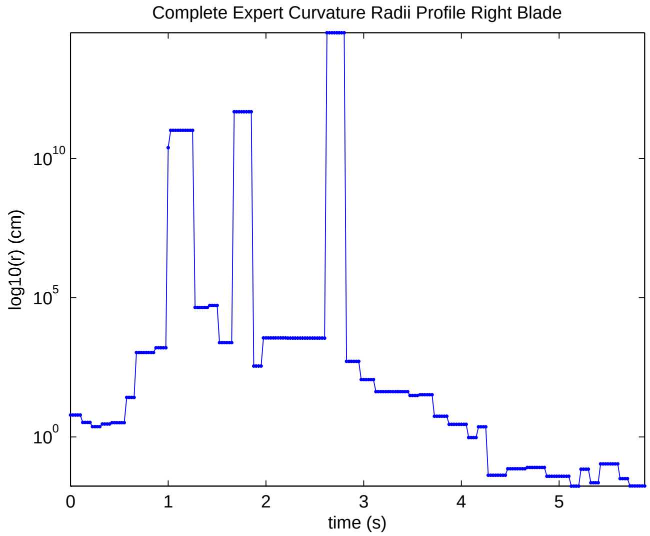
Figure 4.4: Expert 1's multilevel cumulative curvature radii profile for a right blade handling gesture. At Figure (4.4a) three levels are shown. Figure (4.4b) illustrates the summated profile.

both the middle and detail levels, 2 and 3 respectively, are characterized with low curvature radii values.

On the other hand, the MCRP for trainees varies noticeably even in a simple visual inspection depending on the training stage. This is an important



(a) Expert 2's Multi-scale curvature radii profile



(b) Expert 2's multilevel cumulative curvature radii profile

Figure 4.5: Expert 2's multilevel cumulative curvature radii profile for a right blade handling gesture. At Figure (4.5a) three levels are shown. Figure (4.5b) illustrates the summated profile.

result because it shows that the method can be used to evaluate the evolution of the gesture at different stages.

Figure (4.6a) illustrates a common MCCRP of a trainee at the very first training session. Aside the length of the gesture, the MCCRP has a slight tendency to be periodic, specially at the middle decomposition level. The finer curvature radii neighborhood, there are oscillations between high and low curvature radii. But the most important tendency is the final placing of the sensor. The values of the radius of curvature tend to be high, indicating that the final motion reaches the placing target in a straight line. Observing the same stage in the experts, the final motion is not a straight line, but a gentle notch on the target area that is noticeable in medium and finer scales.

During the second phase of training, the students are allowed to observe an expert performing the procedure, but they are not taught directly. An MCRP at this stage is shown in Figure (4.6b). The length of the gesture performance has been shortened, but the fines scales show still some oscillation between high and low curvature radii. The final placement is gentler but hastened at the end.

In the third phase of training, trainees are introduced to the handling with a detail explanation of the technique, but they are not given the time to hone the gesture. An typical MCRP at this stage is illustrated at Figure (4.7a). The length of the gesture remains similar to the observation part at Figure (4.6b). However, the improvement is noticeable in all three stages of the gesture performance. The all important final placement stage is gentler with finer curves.

Figure (4.7b) illustrates a typical MCRP after the trainees had been allowed to hone the gesture. The gesture length is the shortest of all the previous performances with fewer high values in both scales, indicating a smoother gesture curve.

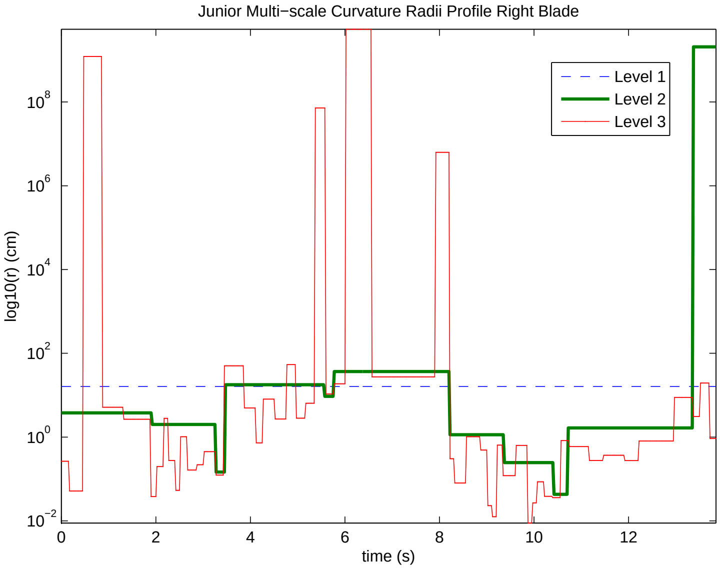
4.3.2.2 Differences Between Experts and Trainees

In order to determine which are the dominant features on a multi-scale curvature radii profile, we applied an autocorrelation function to every decomposition level of multi-scale profile. Unlike random signals, the correlation coefficients do not decay quickly on deterministic signals and the influence between coefficients is strong even when the signals' overlapping is partial.

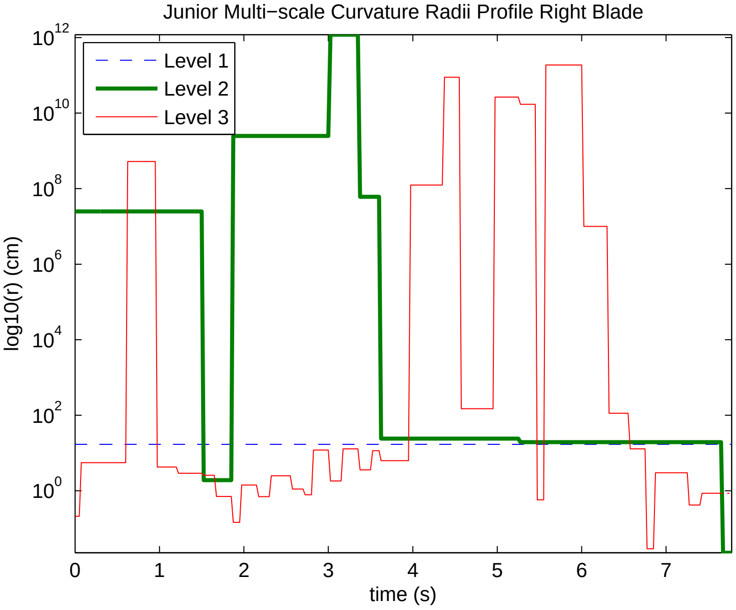
However, in the case of discrete curvature profiles, the correlation between coefficients is dissimilar at different scales. It is expected that in the autocorrelation at different scales will have dominant modes that influence the summed up autocorrelation profile.

4.3.2.2.1 Autocorrelation for Experts

The autocorrelation of the MCRP for experts allow us to analyze further an expert's performance. The autocorrelation for two different experts is illustrated at Figure (4.8).

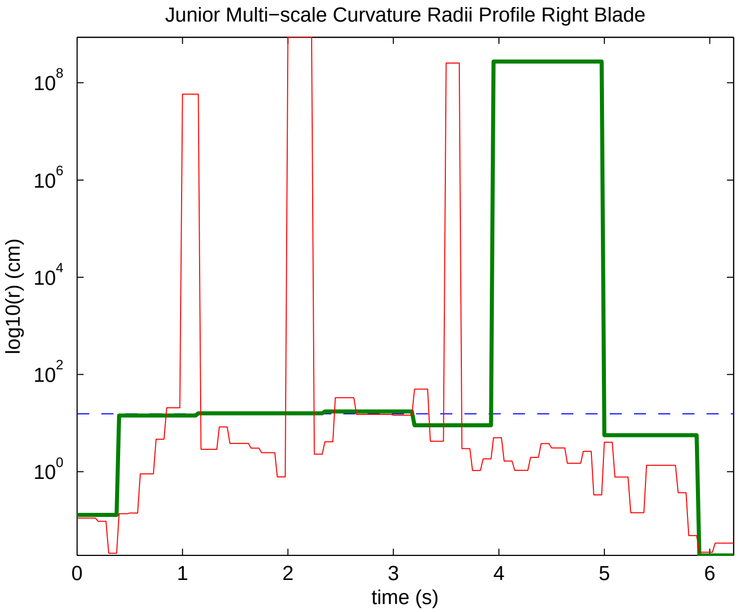


(a) MCRP for a trainee at the first session of training.

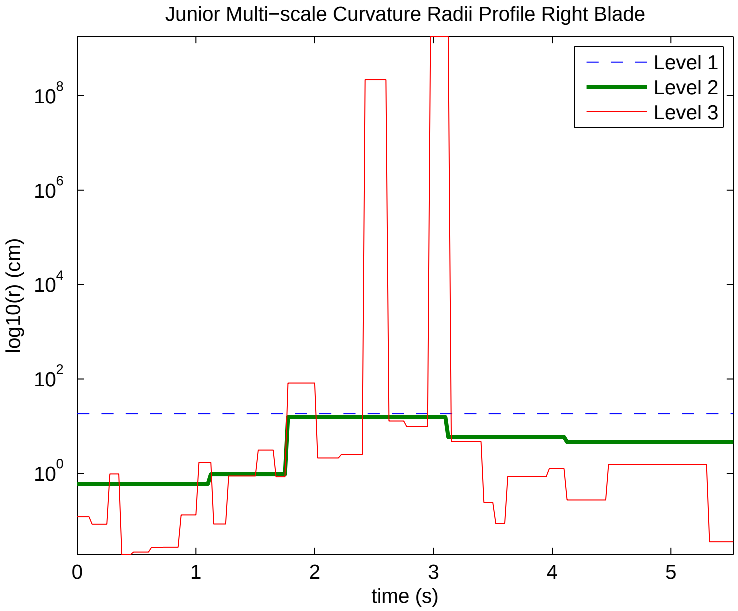


(b) MCRP for a trainee after observing an expert execution of the gesture, but no explanation nor correction are given.

Figure 4.6: MCRPs at two initial training phases. (4.6a) illustrates the initial gesture performance of a trainee without references of the gesture performance. (4.6b) illustrates the same gesture after being allowed to observe the performance of the gesture.

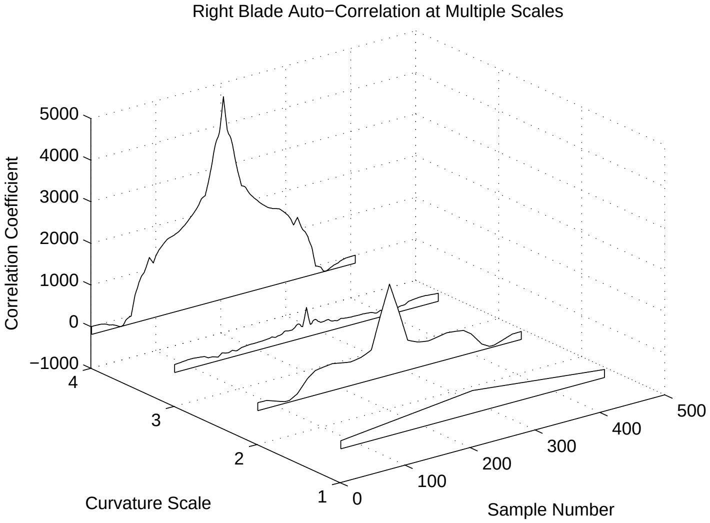


(a) MCRP for a trainee after the gesture performance had been explained.

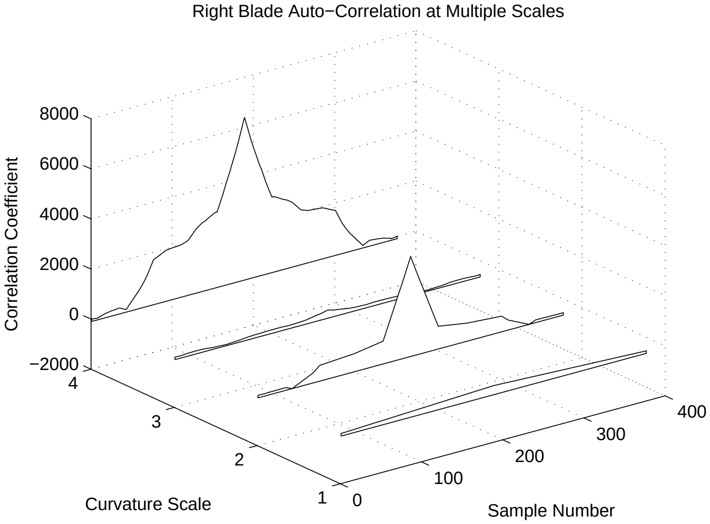


(b) MCRP for a trainee after practicing the gesture.

Figure 4.7: MCRPs at two final training phases. (4.7a) illustrates the handling gesture performance right after receiving a detailed explanation on the gesture performance. (4.7b) illustrates the final gesture performance after being allowed time to practice the handling gesture.



(a) Expert 1 Autocorrelation function for a sample trajectory on the right blade



(b) Expert 2 Autocorrelation function for a sample trajectory on the right blade

Figure 4.8: Autocorrelation functions for the curvature radii profiles using the constant curvature fitting algorithm. The autocorrelation functions operate at each decomposition level and the general curvature profile.

In both, Figure (4.8a) and Figure (4.8b) one can see that the shape of the autocorrelation of the summed up profile is mostly determined by the approximation of the curvature radii at the middle level of decomposition, not the finer detail level. This is key since it shows that the coarser level dominates the shape of the autocorrelation, no the fine detail level. The detail level is usually associated with fast changes in the trajectory.

4.3.2.2.2 Autocorrelation for Trainees at Different Stages

Next we introduce a qualitative explanation of the autocorrelation results at different stages of training for the forceps gesture.

Initial Session

Figure (4.9) shows two typical autocorrelation profiles for the multi-scale curvature radii of two trainees at this stage. The most notable observation aside the length of the autocorrelation is that the shape of the summed up profile is strongly influenced by the correlation in the details level.

Observation Session

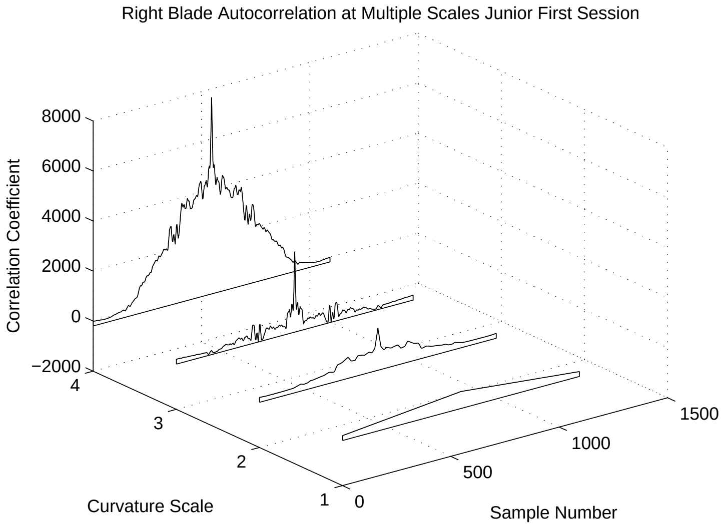
Figure (4.10) illustrates two autocorrelation for the MCRP of two juniors at after observing the gesture performance from an expert. The length of the autocorrelation is shorter but the summed up profile is still strongly influenced by the correlation in the details level.

Explanation Session

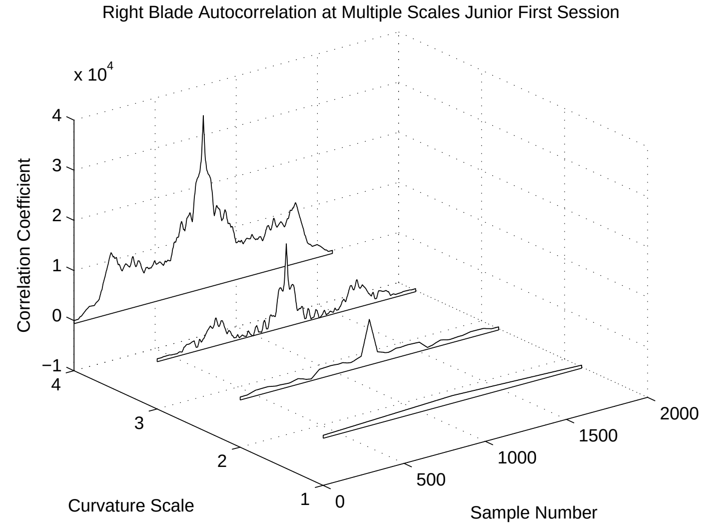
Figure (4.11) shows two autocorrelation for the MCRP after the explanation session. The length of the autocorrelation is similar to the observation session, however the shape of the summed up profile is now influenced by the correlation in the coarser curvature radii level.

Practice Session

Figure (4.12) illustrates two typical autocorrelation profiles for the multi-scale curvature radii of two juniors. The length of the autocorrelation is the shortest of all training sessions. They are the most similar to the autocorrelation from the experts'. The autocorrelation is now clearly dominated by the coarse decomposition level instead of the details level.

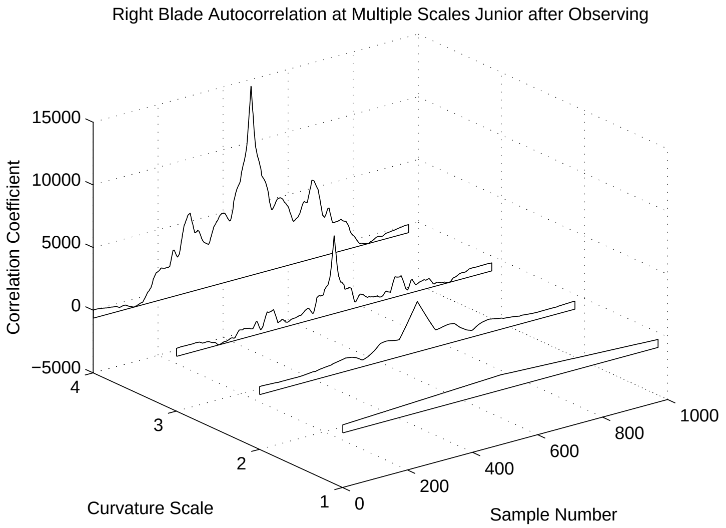


(a) Junior 1 Autocorrelation function for right blade at the initial session

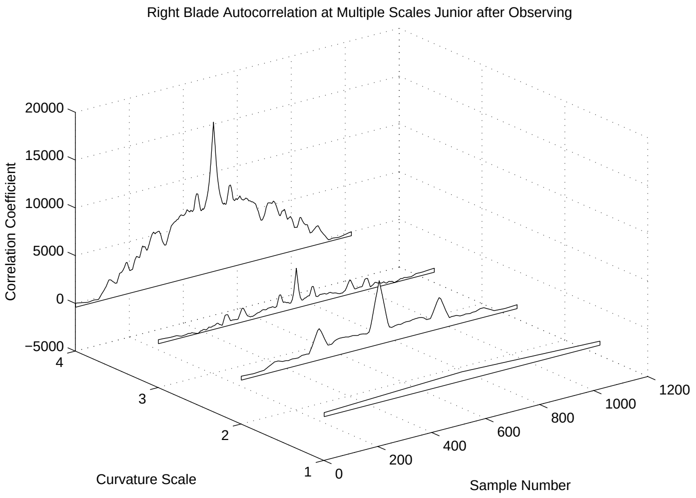


(b) Junior 2 Autocorrelation function for right blade at the initial session

Figure 4.9: Autocorrelation functions for multilevel curvature radii profiles. (4.9a) Junior 1 and (4.9b) Junior 2 first session. The technique has not been shown nor explained to the trainees.

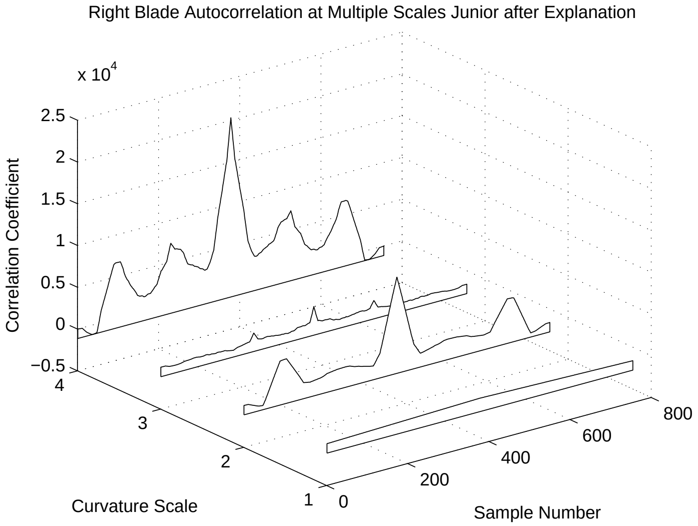


(a) Junior 1 Autocorrelation function for right blade at the observation session

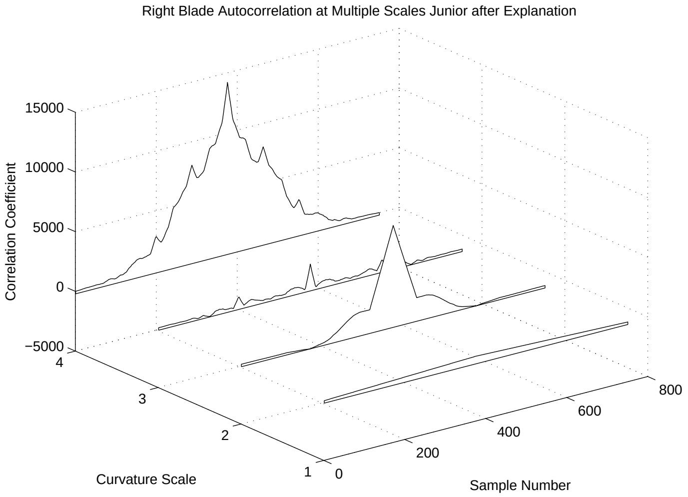


(b) Junior 2 Autocorrelation function for right blade at the observation session

Figure 4.10: Autocorrelation functions for multilevel curvature radii . (4.9a) Junior 1 and (4.9b) Junior 2 observation session. The technique has been shown but not explained to the trainees.

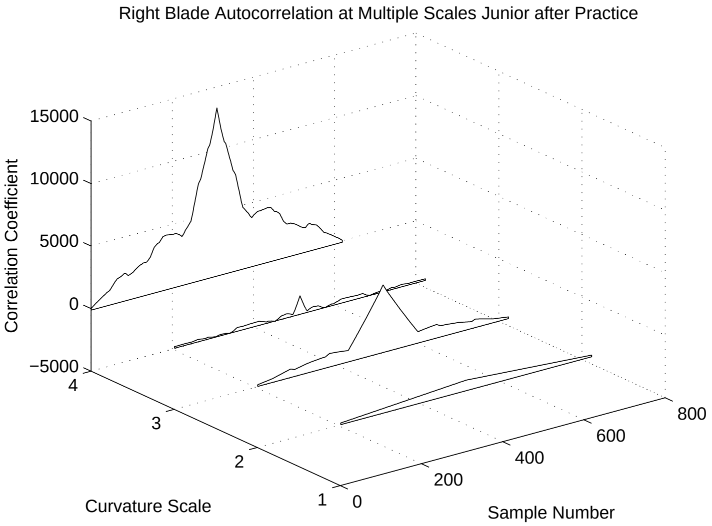


(a) Junior 1 Autocorrelation function for right blade at the explanation session

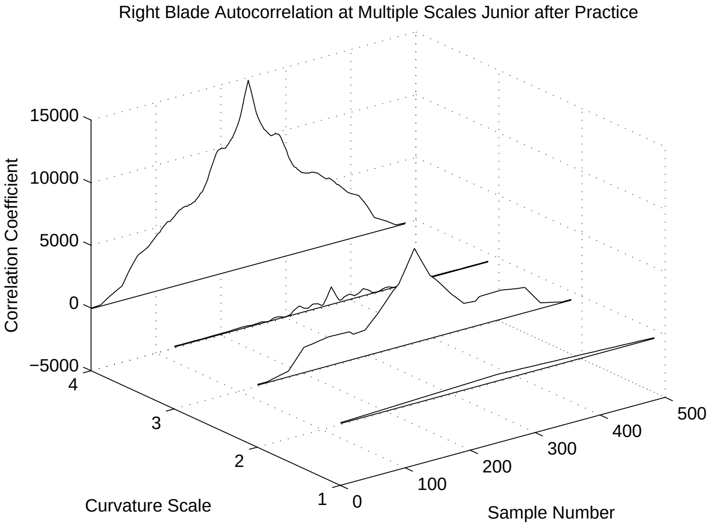


(b) Junior 2 Autocorrelation function for right blade at the explanation session

Figure 4.11: Autocorrelation functions at for multi-scale curvature radii. (4.9a) Junior 1 and (4.9b) Junior 2 after receiving instruction. The technique has been explained to the trainees.



(a) Junior 1 Autocorrelation function for right blade at the practice session



(b) Junior 2 Autocorrelation function for right blade at the practice session

Figure 4.12: Autocorrelation functions at for multi-scale curvature radii. (4.9a) Junior 1 and (4.9b) Junior 2 last session. The technique has been explained and the trainees had time to practice the handling gesture.

4.3.3 Anisotropic Kernel Smoothing

4.3.3.1 Optimal Anisotropic Kernel Filter Parameters

The two main parameters for the anisotropic filter, scale σ and window support k, were determined statistically by constructing an estimation set from randomly picked samples from experts and juniors, see Section 3.4.4. The parameter tuple σ and k was found using single factor ANOVA for the curvature approximation sum of squares error (SSE). The statistical results show that σ is an independent parameter with respect to the curvature's SSE for $\sigma > 0.09$ with p-values of $p_{0.05} = 0.714$ and $p_{0.01} = 0.916$ for position and orientation respectively.

However, the curvature SSE is strongly dependent on the window size k with p-values $p_{0.01} = 4.52 \times 10^{-6}$ and $p_{0.01} = 8.43 \times 10^{-6}$ for position and orientation trajectories respectively. High correlation coefficients are expected for parameter tuples $\sigma \geq 0.1$ and $k = 5$ for position trajectories and $\sigma \geq 0.05$ and $k = 3$ for orientation trajectories.

The significance is clearly that multi–scale curvature analysis using our anisotropic filter relies on the window support k and less on the scaling factor σ.

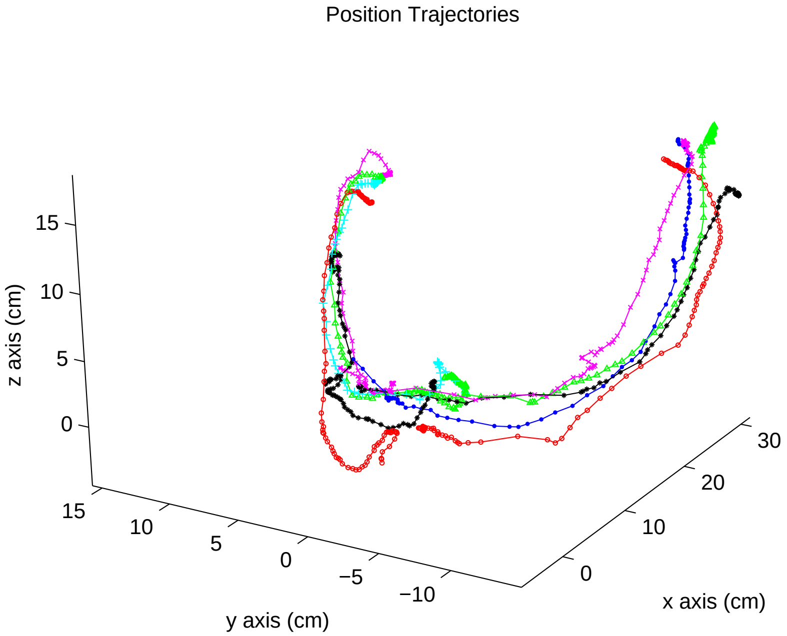
4.3.3.2 Pearson's Correlation

By calculating Pearson's correlation coefficient between the curvatures of the trainee paths and the reference curvature during their training, it is possible to quantify the progression of the trainee skills.

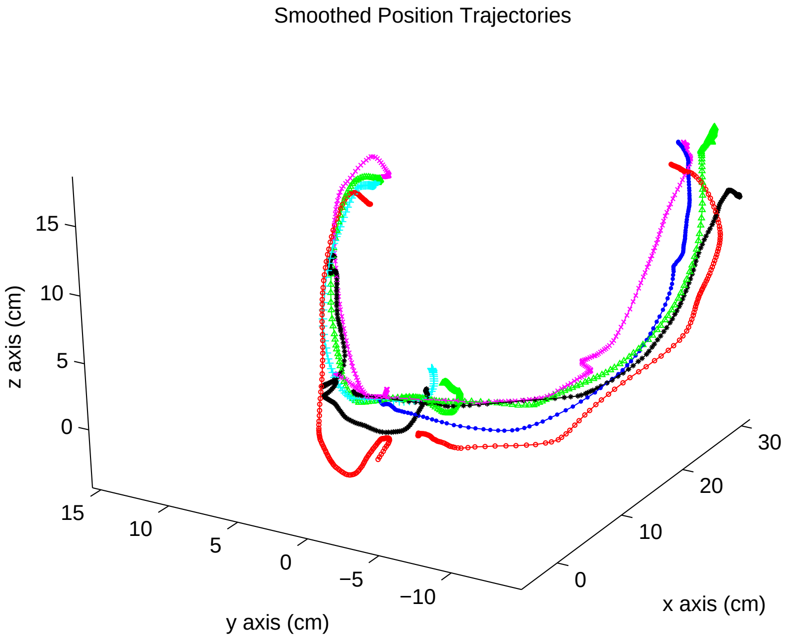
Figures (4.13) and (4.14) are the analyzed curves for the expert trajectories which has been used as reference. The two other paths correspond to the forceps blade placement carried out by two novices at different training stages.

On these figures, from a qualitative point of view, the paths after the training are more similar to the expert one than before the training, in the next section the study of the correlation coefficient between the curvatures allows to quantify this similarity. Figures (4.17) and (4.18) summarize results for the training evolution of a single novice at different training stages.

At different stages the number of sampled trajectories varies and thus the number of correlation coefficient varies as well. One can observe that for the initial training sessions, where no explanation and no observation are given, the trajectories are long with comparatively low correlation between the expert and the novice. At the second session, the novices were allowed to observed the expert, the correlation coefficients where higher, but still long sequences. During the third session, the novices were also given a detailed explanation on the handling of forceps and the gesture. The correlation coefficients remained more or less the same, but the sequences where shorter. In the final day, where

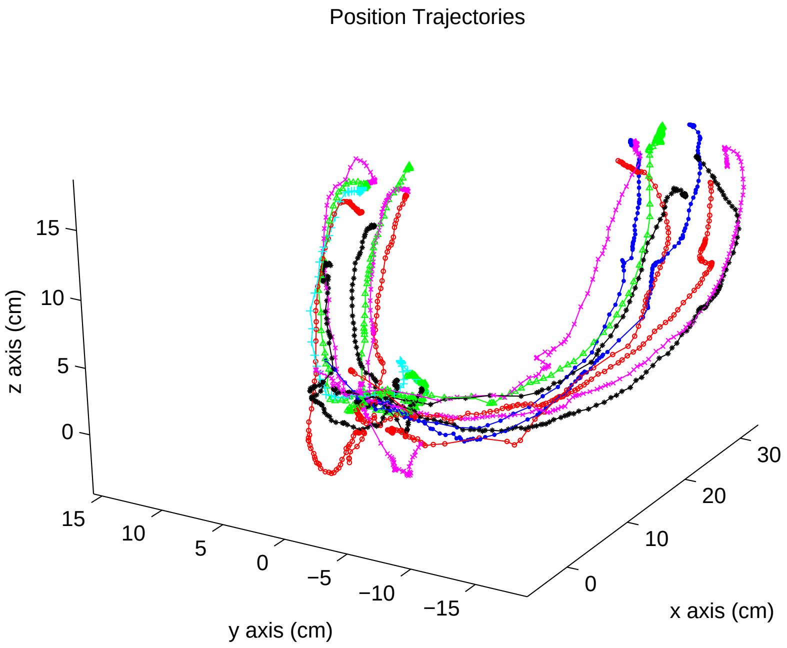


(a) Raw Expert 1

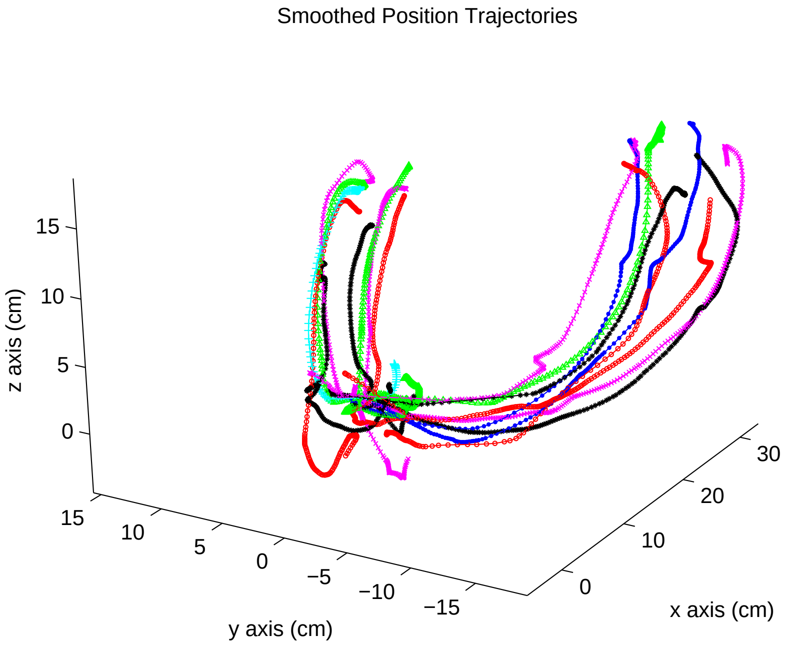


(b) Anisotropic-smoothed Expert 1.

Figure 4.13: Raw and smoothed trajectories with right and left Forceps blade position in $3D$ for Expert 1.

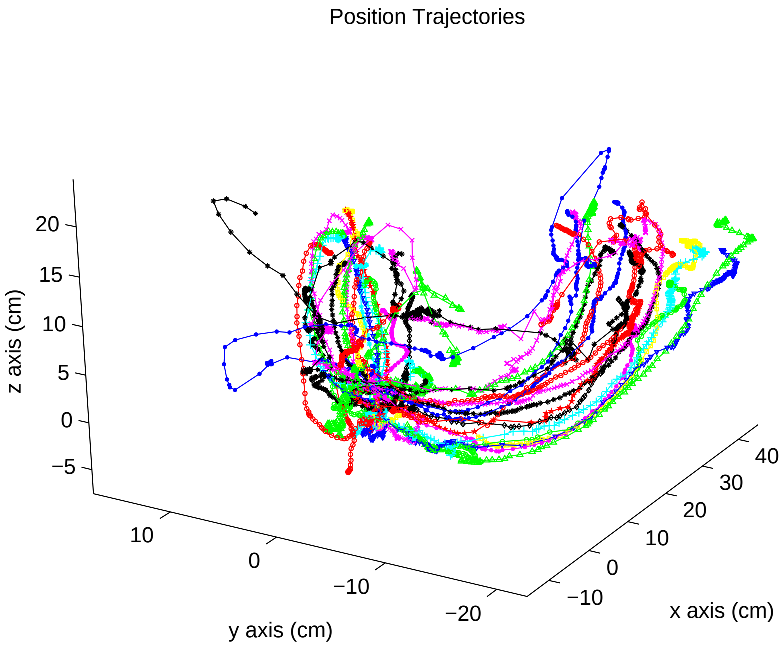


(a) Raw Expert 2

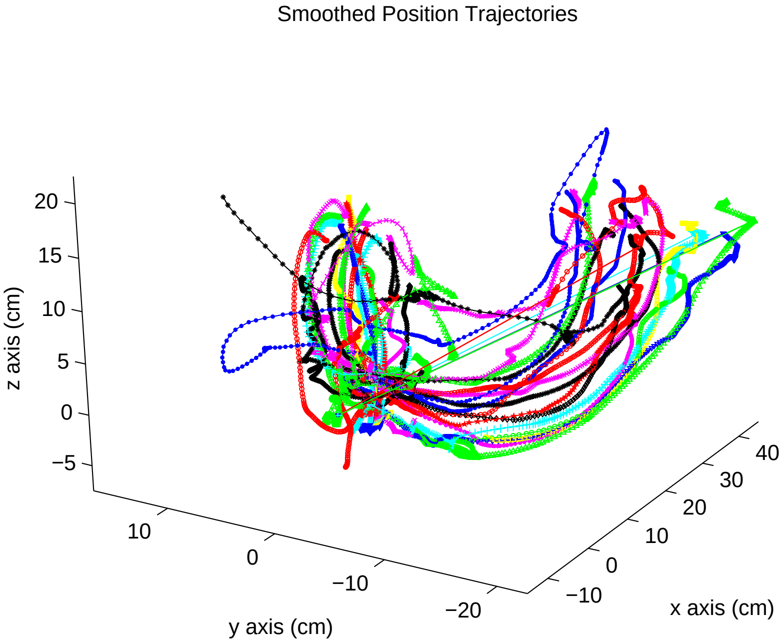


(b) Anisotropic-smoothed Expert 2.

Figure 4.14: Raw and anisotropic-smoothed trajectories with right and left Forceps blade position in $3D$ for Expert 2.

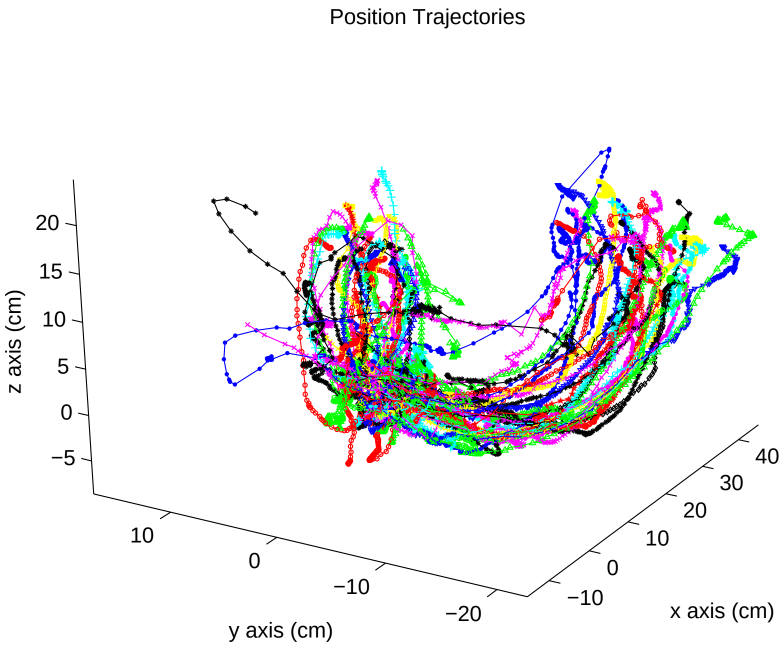


(a) Raw Junior 2

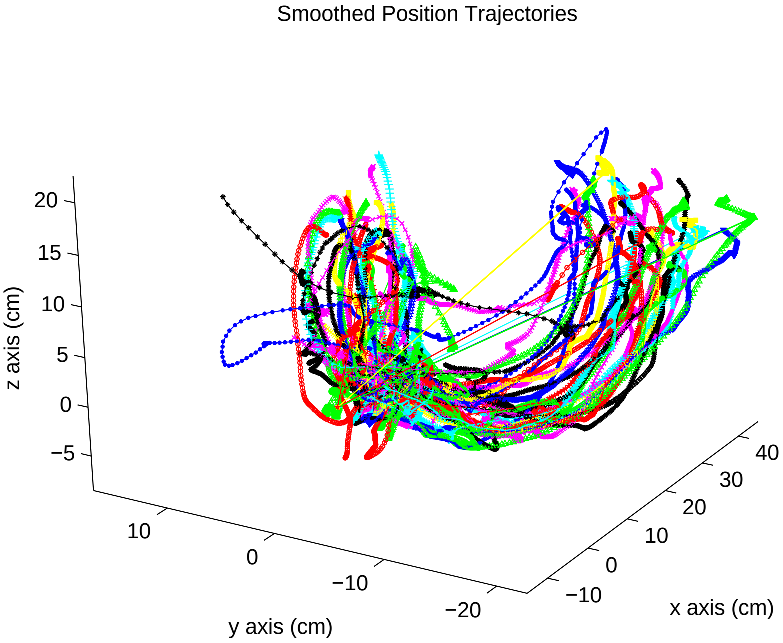


(b) Anisotropic-smoothed Junior 2.

Figure 4.15: Raw and anisotropic-smoothed trajectories with right and left Forceps blade position in $3D$ for Junior 2.

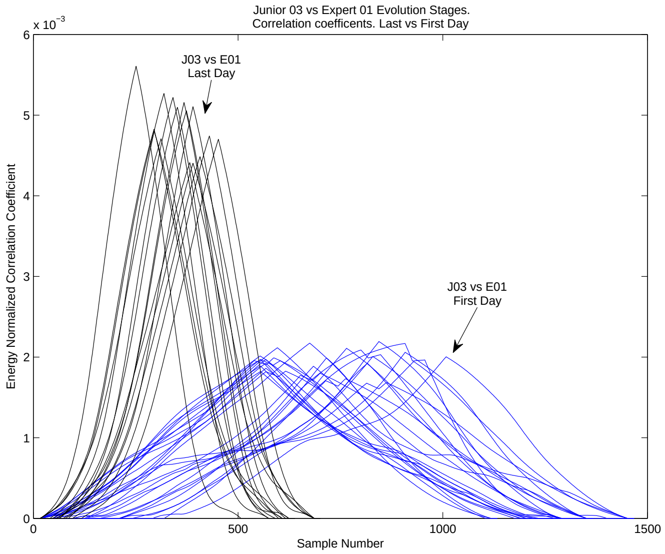


(a) Raw Junior 3

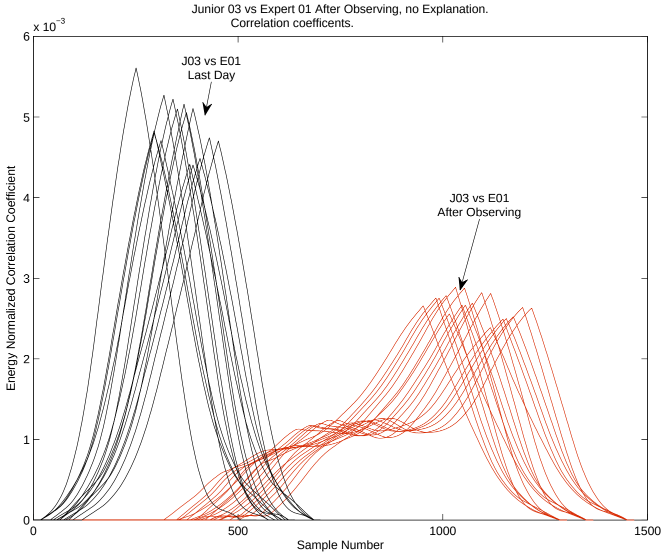


(b) Anisotropic-smoothed Junior 3

Figure 4.16: Raw and Anisotropic-smoothed trajectories with right and left Forceps blade position in $3D$ for Junior 3.



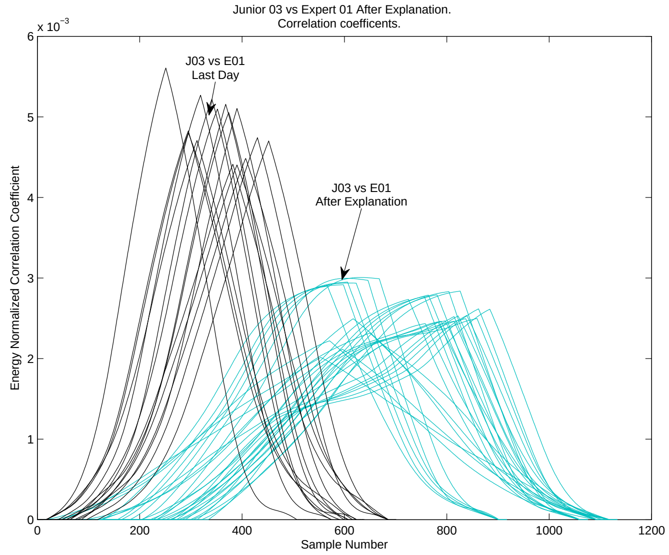
(a) Junior 03 First Day. No direct observation, no explanation



(b) Junior 03 Second Day. After observing expert, no explanation

Figure 4.17: Cross-correlation functions comparing Junior 3 vs. Expert 1. Different stages in training. Figure (4.17a) shows the correlation coefficients between the first and last training days. Figure(4.17b) shows the correlation coefficients between after observing the expert but no explanation was given.

the trainees were allowed some practice. During this stage, the sequences showed the higher correlation against the expert and the shortest arc-length.



(a) Junior 03 Third Day. After Explanation

Figure 4.18: Correlation coefficients comparing Junior 03 vs. Expert 01. Different stages in training. Figure (4.18a) shows the correlation coefficients between after the trainee receives detail explanation, but no practice. All coefficients are compared against the last day in training where the trainee has given time to practice after receiving a thorough explanation in the use of forceps

4.3.4 Analysis for the Sensor Positions Trajectory

Table 4.2 is a brief summary of results for different training days (from 1 to 3). Each individual result is the Pearson's cross-correlation coefficient ρ between the trainee and an expert. It indicates the similarity to an expert path used as reference. This result corresponds to the average of the three recorded paths at the end of the training day (except for the first day where the first forceps blade placements were recorded to know their skill before the training).

In Table 4.2 the acronyms LFB, RFB, T and TD refers to the terms: Left Forceps Blade, Right Forceps Blade, Trainee and Training Day, respectively. These notations are used hereafter. In Table (4.2), one can observe a raise in the correlation coefficient between the path curvature of the trainee and the expert one during their training. At the end of the training, all trainees manage to obtain more than $\rho_{pr} = 0.44$ of similarity except for the right blade

Table 4.2: Evolution of the correlation coefficient of the curvature for the positions in % for trainees according to the training day

Correlation Coefficient ρ_{pr} for Curvature Signatures		TD 1	TD 2	TD 3
T 1	LFB	0.34	0.11	0.46
	RFB	0.28	0.25	0.53
T 2	LFB	0.18	0.38	0.44
	RFB	0.03	0.44	0.33
T 3	LFB	0.29	0.17	0.35
	RFB	0.28	0.26	0.61
Trainee's Average	LFB	0.27	0.22	0.42
	RFB	0.20	0.32	0.49

of Trainee 2 and the left blade of Trainee 3. Lets arbitrarily consider the correlation coefficient ρ_{pr} as:

excellent:	if ρ_{pr} is beyond 0.70,
good:	if ρ_{pr} is between 0.50 and 0.70,
fair:	if ρ_{pr} is between 0.30 and 0.50,
poor:	if ρ_{pr} is between 0.10 and 0.30,
very poor:	if ρ_{pr} is less than 0.10.

Table 4.3 is obtained by rearranging the results according to this arbitrary grouping.

Table 4.3: Distribution of the correlation coefficient for the position curvature according to the training day

Correlation Coefficient ρ_{pr} for Curvature Signatures		TD 1	TD 2	TD 3
Excellent	$(\rho_{pr} > 0.70)$	0	0	0
Good	$(0.50 < \rho_{pr} < 0.70)$	0	0	2
Fair	$(0.30 < \rho_{pr} < 0.50)$	0	2	4
Poor	$(0.10 < \rho_{pr} < 0.30)$	5	4	0
Very Poor	$(\rho_{pr} < 0.10)$	1	0	0

This table shows the progression of the trainees according to the training day, all their results are *fair* and *good* at the end of the training whereas they only have *poor* and *very poor* results before the training.

It is then possible to find where high variations of the curvature occur. These variations represent fast modifications of the direction which could lead to injuries for the fetus and the mother during real deliveries. They could also represent hesitations during the forceps blade placement or simply a withdrawal of the forceps blade in order to replace it correctly.

These curvature peaks appear when their values are beyond the threshold of high curvature variations. This threshold is fixed at a value of 5 but could be changed according to the expert choice. For increased clarity, only the path and the curvature of left forceps blade are represented on these figures.

In Table 4.4, the number of the curvature points beyond the threshold, fixed at value 5, are represented according to the training day. This number represents the average of the number of peaks for the three paths recorded during each training day. The smaller this number is, the smoother the path is, and the less dangerous the gesture is. Some of the peaks on the curvature appear at the beginning or at the end of the gestures, *i.e.* when the forceps entered inside the maternal pelvis,or took its final position behind the fetus ear, or due to the fact that the right blade has to be assembled with the left one to complete the placement. One can then count two kind of peaks according to their position during the gesture. In Table 4.4, the column 100% corresponds to the total number of peaks and the column 80% represents the peaks which appears between 10% and 90% of the total chord length in order to avoid taking into account the extremity peaks. The peaks in this area are potentially dangerous because they appear while the blade is displaced along the fetal face.

Table 4.4: Evolution of the number of curvature peaks beyond the threshold for the positions for trainees according to the training day

Number of peaks		T 1		T 2		T 3	
		LFB	RFB	LFB	RFB	LFB	RFB
TD 1	100%	18	17	20	24	29	40
	80%	6	11	14	10	20	28
TD 2	100%	18	29	7	25	27	40
	80%	10	21	2	10	23	24
TD 3	100%	12	35	11	16	17	4
	80%	3	21	2	4	3	2

Except for Trainee 1 with his right blade all trainees manage to reduce their number of peaks, this result means that fast modifications of the direction during forceps blade placement are reduced. At the end of the training, the trainees obtained similar results than the expert. Indeed for both blades on the reference paths there are 9 peaks and only 2 of them appear between 10% and 90% of the total chord length. The visualization component of the BirthSIM simulator could help novice to modify his gesture in order to become smoother and surer. As shown Tables 4.2 to 4.4, trainees manage to reduce their number of peaks, in other terms, their gesture become smoother during the training.

Concerning the position analysis, one can conclude that Trainee 1 needs more training in order to smooth his handling trajectory during the placing phase for the right blade. Otherwise, the trainees have manage to improve their skills and may proceed to the traditional training in order to gain more experience.

4.3.5 Analysis of the $3D$ Orientation Curve

Figure (4.19) illustrates a trainer's orientation path on the unit quaternion space.

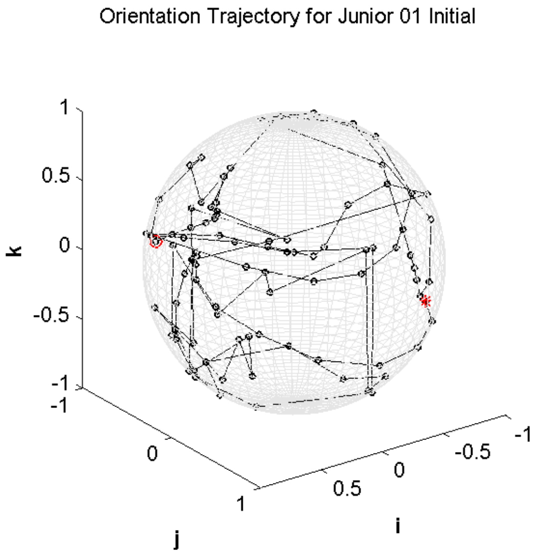
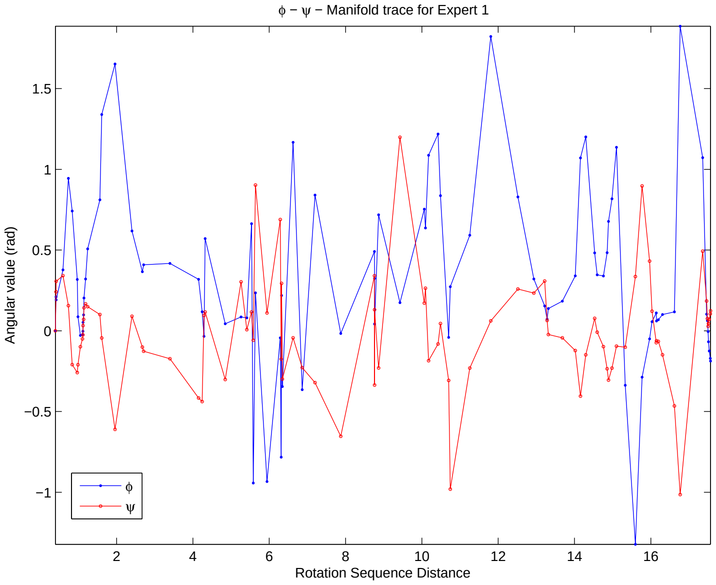
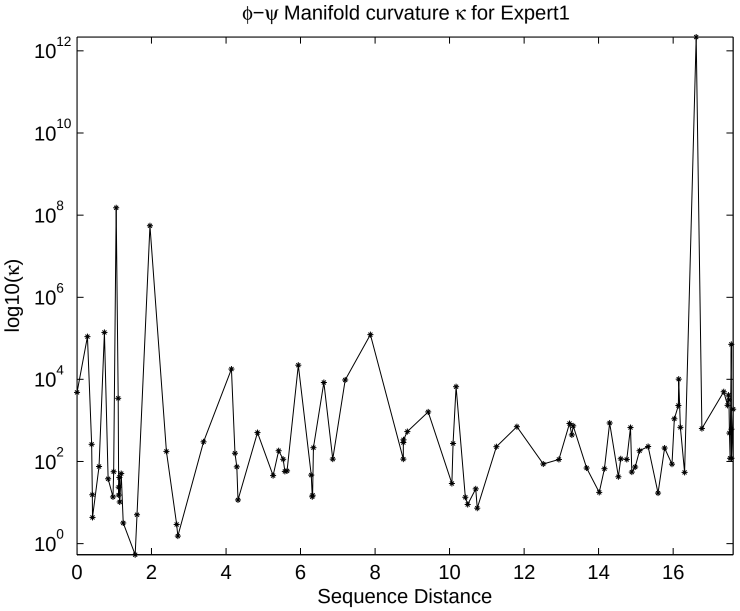


Figure 4.19: Expert 1 forceps handling gesture in the unit quaternion space

The $3D$ orientation trajectory is illustrated as a $2D$ trace on the surface of the unit sphere. However, the orientation trajectory is a $3D$ curve itself. This curve is also approximated using the curvature, see Section 3.4. Both projections are shown at Figures (4.20a) and (4.20b).



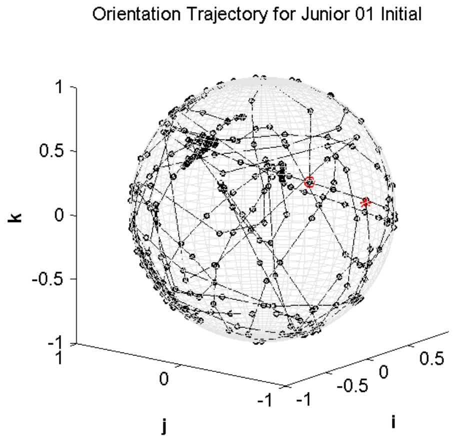
(a) Expert 1 $\phi - \psi$ Manifold trace



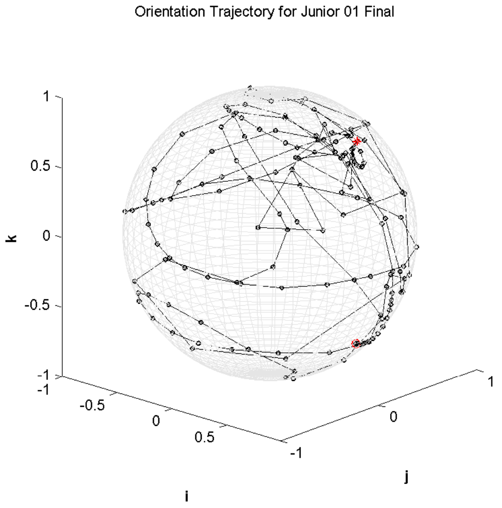
(b) Expert 1 curvature signature for the orientation $\kappa(s)$ trace

Figure 4.20: Expert 1 $\phi - \psi$ orientation trajectory on the manifold and the corresponding curvature trace $\kappa(s)$ in the unit quaternion space.

Figures (4.21a) and (4.21b) show an example of the orientation path on the unit quaternion space at the beginning and at the end of the training for the left forceps blade. Similar results are obtained for the right one.



(a) A trainee orientation path at the beginning of his training.



(b) Trainee orientation paths at the end of his training.

Figure 4.21: Junior 3 orientation path at two different stages of training. Figure (4.21a) shows the orientation path at the initial day of training. Figure (4.21b) shows the orientation path for the same junior at the final day of training.

Both Figures (4.21a) and (4.21b), illustrate how noticeable the trainee improved is at handling the forceps blade as the path becomes smoother and similar to the expert's. This result shows that the quaternion unit space allows to qualitatively quantify the skill progression even to the naked eye with a cleaner and shorter path.

By smoothing the curve base on the curvature, one can apply correlation based methods independently of the geometry of the problem. Orientations are hyper-spherical spaces. Table 4.5 gathers the correlation coefficient for the curvature of the orientation paths.

Table 4.5: Correlation coefficient ρ_{pr} for the orientation curvature according to session number.

Trainee	Forceps handle	Session Number		
		TD 1	TD 2	TD 3
T 1	LFB	0.05	0.08	0.31
	RFB	0.23	0.46	0.40
T 2	LFB	0.23	0.36	0.46
	RFB	0.27	0.33	0.61
T 3	LFB	0.09	0.02	0.19
	RFB	0.41	0.10	0.46

As mentioned previously, the trainee skills raise up to 40% except for the left blade of Trainee 1 and Trainee 3. Concerning the expert results, the correlation coefficient is 0.75 and 0.70 of similarity for the left and right blades respectively.

If the correlation coefficients are gathered according to the distribution presented subsection 4.3.4, the Table 4.6 is obtained.

Table 4.6: Distribution of the correlation coefficient for the orientation curvature according to the training day

Curvature correlation values ρ_{pr} for the orientations		TD 1	TD 2	TD 3
Excellent	$\rho_{pr} > 0.70$	0	0	0
Good	$0.50 < \rho_{pr} < 0.70$	0	0	1
Fair	$0.30 < \rho_{pr} < 0.50$	1	3	4
Poor	$0.10 < \rho_{pr} < 0.30$	3	1	1
Very Poor	$\rho_{pr} < 0.10$	2	2	0

As for the position study, trainees have better results at the end of the training. None of these results are *very poor* and only one is *poor* conversely to the beginning of the training whereas all results are *very poor* and *poor* except one which is *fair*.

The study of the high variations of the curvature concerning the orientation paths are gathered in Table 4.7 where the number of the curvature points beyond the threshold fixed at 5 are represented according to the training day as for the positions. This number represents the average of the number of peaks for the three paths recorded during each training day. Peaks are gathered in two groups to distinguish the extremity peaks and the ones appeared between 10% and 90% of the total chord length.

Table 4.7: Evolution of the number of curvature peaks beyond the threshold for the orientations for trainees according to the training day

Number of peaks		T 1		T 2		T 3	
		LFB	RFB	LFB	RFB	LFB	RFB
TD 1	100%	18	18	20	23	31	45
	80%	10	14	14	17	24	36
TD 2	100%	27	37	12	25	24	43
	80%	11	27	7	14	14	30
TD 3	100%	13	38	13	17	16	23
	80%	5	25	5	9	5	9

As for the positions, except for the right blade of Trainee 1, all of them manage to reduce their number of peaks, this result highlights that important changes of direction during forceps blade placement are reduced. For the reference paths, the numbers of peaks are 16 (2) and 18 (7) respectively for the left and right forceps blade (the number in brackets indicates the number of peaks appeared between 10% and 90% of the total chord length). The trainees finally obtained similar results than expert at the end of the training.

4.4 Summary

The evaluation of a specialized gesture is compulsory in order to validate a training method. A general method based on the six degree of freedom measurement of a medical gesture was developed in order to quantify the progression of novice skills compared to an expert.

By estimating the curvature signature of the forceps blade placement gesture, it is possible to evaluate inexperienced trainees against qualified experts. Time independence of the analysis is ensured by expressing data with respect to the path's arc-length. The orientation trajectory is also analyzed in the unit quaternion space. It provides a curvature analysis for the orientation trajectory that is analogous to Euclidean curvature. The correlation coefficient between invariant signatures measures the similarity between different paths. The statistical frequency of curvature peaks over a threshold is a measure of how smooth the manipulative gesture for forceps blade is.

Simulator training allows to proceed to risk-free training and to reduce training time. Some trainees obtained satisfactory results that allow them to proceed with the traditional training in the delivery ward. However they still have the possibility to increase their performance with continuous training on the BirthSIM simulator. The trainees learnt how to correctly place the forceps by visualizing the expert's path represented with guiding spheres in the visual interface. The trainee's progression was quantitatively qualified with the developed method.

The next series of measurement carried out on the BirthSIM simulator will have to take into account more novices and a longer follow-up period. The objective will explore if a training using a simulator allows trainees to accumulate useful experience before carrying out a childbearing procedure with forceps. In the long term the intent is to quantify the utility of a childbirth simulator both as a learner's tool and qualitative evaluator.

In this chapter two estimation procedures for the curvature signature of a $3D$-gesture were introduced. These estimation procedures quantify the performance of $3D$-gesture for different individuals at different skill levels. An important objective of the current dissertation. Both methods improve on previous estimates of the curvature obtained from either analytic means such as finite differences or by fitting piecewise approximations of linear quadrics. Both methods are different from diffused curvature approximations such as the curvature scale space [137]. The anisotropic kernel filter has an additional advantage over fitting methods. It can not only be used on traditional Euclidean spaces, but it also can be used in spaces such quaternions. Shape fitting methods depend on finding a principal direction which can not be found in spherical geometries, making their application on those particular spaces difficult at best.

Chapter 5

Conclusion and Future Work

The current dissertation is a study in $3D$ gesture comparison in the absence of predictive models. $3D$ gesture comparison without Lagrangian state-space models is challenging not only because of the lack of prediction provided by physical state-space models, but also because of its multi-geometric space, its spatial complexity and the nonlinearity of the invariant spaces associated to $3D$ curves. The multi-geometry treatment for the spatio-temporal trajectory arises from using an Euclidean space to trace position and a hyper-sphere to trace orientations. The spatial complexity arises from the fact that one cannot guarantee that trajectories in a $3D$-space will ever meet at one point. The significance is that one has to assume that each and every trajectory in the space is unique. Then the problem is asserting the similitude between two or more trajectories. And the third issue refers to the use of invariant descriptions or how to ensure that a trajectory is the same from different view points.

We tackled all the above problems both theoretically and practically. We devised a uniform treatment for $3D$ location trajectories that is geometry independent. Location being defined as the composition set of variables in both geometries, Euclidean and hyper-spherical. We define a 3-tuple of vector variables $(\boldsymbol{r}, \boldsymbol{\theta}, t)$ for position, orientation and time respectively. Our work hypothesis states that under an invariant transformation any trajectory in a $3D$ space, Euclidean or hyper-spherical, can be compared to each other with time-tested similitude measures such as correlation. Consequently, any other similitude measure based on geometrical alignment such as the Iterative Closest Point (ICP), dynamic programming such as time warping, or graph models such as Hidden Markov Models (HMMs) should also simplify and improve.

Finding a uniform space description for both position and orientation is challenging. Unlike position, that is described with Euclidean spaces, orientations have been traditionally described in terms of rotation matrices and vector-angle operations. These descriptive spaces for orientations are not linear space and one cannot define a metric between points and thus there are no interpolation functions available. Additionally, there are singularities within these spaces that are unpredictable. Quaternions, on the other hand, are a better descriptive spaces for orientations and rotation operations. They do define an invariant space metric, their singularities are predictable, and they do possess interpolation functions. However, quaternions are nonlinear plane operators with four parameters. Four parameters is not a minimum number of parameters for a $3D$-space.

One may think that instead of reducing the spatial complexity of an orientation curve, quaternions actually increase it. If a common point for two curves in a $3D$ space is not guaranteed, in a $4D$-space it almost certain that they will also have redundancies. As matter of fact, they do. Quaternions have mirror representations in the same space. An orientation trajectory has two antipodal mirror curves.

We have shown that although quaternions are four parameter spaces, quaternion subsets with magnitude constraints abate the number of parameters with projections onto $3D$ manifolds. There are different projections available with advantages and disadvantages of their own. One main contribution in our work is a projection from unit quaternion space onto a $3D$ manifold that is bounded. We denominate such a projection as geodesic projection or cycloid projection. The projection is a consequence of the very own nature of hyper-spherical spaces. Its main advantage is that the projection is both limited and simplified onto a $3D$ vector field. However, a manageable nonlinearity is involved but the space itself can be consider a $3D$ volumetric sphere. Unlike Hanson's proposal for embedding quaternions onto the Frenet-Serret (FS) differential frames [99] on $3D$-Euclidean curves, our proposal is a direct description of the orientation curve in a $3D$ space. The $3D$-curve accepts FS frames since the orientation curve is just another $3D$ curve inside of a sphere volume.

In n-dimensional Euclidean spaces, invariance is achieved using local differential reference frames, mainly FS frames. However, other reference frames such as parallel transport are also possible [99] [98]. The differential reference frames offer two main compelling characteristics: invariance and spatial complexity reduction. However, there are some trade-offs in exchange for these two advantages: nonlinearity and noise sensitivity.

Likelihood methods, mainly graph models using dynamic programming such as HMMs or time warping, consider $3D$-curves as compositions of three non-covariant factors evolving through an independent variable t. The issue with such a view is the dependance on the location of the general reference frame $\mathcal{F}_0$. The significance is that these models are not view-point invariant. Geometric procedures such as the ICP algorithm do consider non-covariant factors as well; however, the basic assumption is that one should align first the curves before comparing them with a series of affine geometric transformations. In this case, the view-invariance is forced upon the comparison.

Descriptions based on differential geometry do not consider non-covariant factors, on contrary, it is assumed that within a sequence of local frames, the location of the following frame depends only on the previous one. This constraint reduces the number of degrees of freedom for the local frames but in exchange a trajectory is represented with one less dimension. The parametrization of $3D$-curves depends now on local distance metrics and angular relationships between the local frames within a sequence. Time stops being an important independent variable and it is re-parameterized in terms of the arc-length for the sequence. This is key when comparing gestures using kinematic models.

Curve arc-length re-parametrization s is view invariant since there is no general external reference frame $\mathcal{F}_0$. As trade-off, the re-parametrization now depends on two nonlinear values $\kappa(s)$ and $\tau(s)$, or curvature signatures. These signatures depend on the $\mathcal{C}^{(n)}$ continuity condition or n-order derivative, which is a major downside.

Although in theory the derivative of parametric curves easy to calculate, in practice, the derivatives are approximated from discrete data sets. Any numerical approximation usually introduces noise and distortion due to anisotropic conditions of the space metric. Using traditional signal processing methods, one must assume that the sensor induces some error in the measurements and in general, the quantity of induced noise is unknown. Depending on the scale of the signal, the noise can be manageable or not. If the signal to noise ratio (SNR)is above one, then simple convolution filters may yield a fair approximation to the signal. Otherwise, if the SNR is below one, then one has to smooth the data based on estimation techniques such as the Kalman filter. The derivative approximation is in general, highly dependent on the smoothing process applied to the data.

In practice, one approximates the curvature κ with ease because the second derivative can be robust to variations in the spacing interval h, even when the differences do not cancel the odd components of the Taylor expansion. The robustness is achieve by linearly compensating the finite difference method against anisotropic distances. However, the torsion τ depends on the third derivative. Nonlinear noise is introduced to the torsion τ through the finite approximations. Usually, one cannot guarantee that the third derivative converges to analytical conditions.

A design decision was made to characterize $3D$-curves mainly on the first curvature κ and consider the lack of approximation to the original curve as an approximation error due to the second curvature τ. The consequence is that the method lacks perfect reconstruction characteristics. In other words, we can not reconstruct the original $3D$-curve from our analysis coefficients. We took the perspective of a principal component analysis in which we considered the first curvature κ as our main description for smoothness. Evidently, we could have considered the approximation error as part of the second curvature τ and achieve perfect reconstruction later on with a similar analysis as for the first curvature.

An important contribution in this thesis is the proposal of multi-scale curvature analysis. We proposed two techniques that are robust to noise due to approximations to the derivative. The first is a least square fitting of constant curvature curves to the data. The second is the anisotropic filter that is able to handle the approximation error for finite difference methods.

The least squares constant curvature fitting algorithm is not novel. It basically consists of fitting circles to thin-plate projections of the $3D$-curves. However, this simple algorithm presents an advantage over other methods, the derivative approximation to real data is not calculated directly from the data but by the fitting of a curve. The approximation error for the circle is a multi-scale process, in similar spirit to wavelet decomposition, specifically the Haar wavelet. In the same manner as a Haar wavelet, constant curvature curves do not approximate a curve perfectly, there will always exist an approximation error and the continuity conditions will be violated at the junction points. However, the advantage is that it provides a natural discretization and grouping of points with similar curvature values. This is advantageous for graph models such as HMMs where the sequences are found with high likelihood in the state machine. Our experiments suggest that the multilevel discretization and grouping of the data into constant curvature sections is fast, providing generalization for up to three decomposition levels. Beyond this point, the number of sections grows and diverges very rapidly until they reach a lower bound imposed by the fitting algorithm. These common curvature sections can be used to train the hidden state machine in the HMMs.

In the case of position trajectories in a $3D$ Euclidean space, the curvature analysis there is enough evidence that indicate the influence of different curvature scales in the assessment of a gesture. In the BirthSIM study case, the differences between experts and trainees were established with the energy content in the detail level with respect to the discrete curvature profile. If the detail level dominates the shape of the autocorrelation function, it shows inexperience in the gesture performance. While on the other hand, if the autocorrelation shape is dominated by the coarse level, then it indicates assurance in the gesture performance. This is a quantitative measure of gesture performance.

The main disadvantage of the curve fitting approach is that it is difficult to apply to orientation spaces. An orientation space has a spherical geometry and it is difficult to find a principal direction for a planar projection except for small sections of the trajectory, *i.e.* the details. The implication is that the projections are not invariant in large scales and thus useless for our purposes.

Our second method is more interesting. The idea is to create a smoothing filter from first curvature κ approximations. The curvature signatures $\kappa(s)$ and $\tau(s)$ cannot be compared directly with any correlation method available. In general, any curvature signature has singularities or inflection point at any given moment within the signature. Any correlation under such conditions is undetermined. However, if we use the reciprocal of the curvature distances to assign the coefficients of an averaging filter, the output signal will in turn be smoother in its corresponding extrinsic $3D$-space. The curve's component themselves are distorted but the combined trace is smooth if we had found the appropriate parameters for window size and weighting function. We determined anisotropic filters for both trajectories in Euclidean and orientation spaces for gesture sets.

Multi-scale curvature analysis using the anisotropic filter is not done directly. In this case, the multi-scale is provided by different window sizes based on the curvature signature. Each signature is unique and numerically stable since it is a FIR filter. In order to establish a form of comparison between different signals, we correlated the smoothed outputs and measure their correlation coefficients. However, a major disadvantage of the method is the window support in Euclidean spaces. The window size is limited only to odd sizes such that the scaling is non-uniform. In order to test increasingly larger neighborhood sizes, the signal tails at the section of interest must be also larger. A non-optimal solution to this problem is to consider a blind decimation method that uses the curvature residuals from the previous smoothing.

In orientation spaces, the window support for the anisotropic filter is enough to show variations of the curvature. The reason is that trajectories in rotation sequences cannot contain large number of points for a single geodesic. Usually, complex rotation sequences in the geodesic-arc projection change direction on the surface of the unit sphere within less than few samples. The anisotropic filter smooths the rotation trajectory such that averages these rapid changes to a point inside a plane defined by only three quaternions. Increasing the window support in orientation spaces has the effect of collapsing smaller geodesics onto a single larger geodesic defined by the extreme points in the sequence. In this sense, the anisotropic filter is the only capable of explaining the trace in orientation spaces at different scales.

We tested our framework on gesture data sets showing the evolution of trainees versus experts. A forceps $3D$ data set was provided by the BirthSIM training. The data set contained several experts and trainees with different number of samples per expert and trainee. The best smoothing using anisotropic filtering was obtained for window size and the weighting factor σ for both position and orientation. The objective was to observe the evolution of trainees at different stages of their training. Our framework showed that simple correlations can be used to quantitatively observe the differences at different stages and determine how close the trainees had become in relation to the experts.

Future Work

We have devised new forms to process geometric signals involving two different geometric spaces and treat them similarly. This has led to new ways of simplifying complex comparisons that we are still exploring.

There are some open questions about how to treat average curvature signatures in order to create a general pattern for expert signatures in form of filters. Our method can also be extended to other problems such as $3D$ geometric representation of objects at different curvature scales. In $2D$ applications, multi-scale curvature signatures can be used to simplify or reduce methodologies in which level set methods require constant calculation of a propagating curvature signature. Instead, one could use the approximations for different window sizes in order to create nonlinear propagation wavefronts.

Our framework was tested only on the birth simulator case. A natural extension of this work is the application to multi-joint mechanisms. Our projection for quaternion spaces satisfies multiple conditions for which different scales can be used to smooth kinematic solutions that are currently solved using constraints on a Jacobi matrix.

Bibliography

[1] J. K. Aggarwal and Q. Cai. Human Motion Analysis: A Review. *Comput. Vis. Image Und.*, 73(3):428–440, 1999. 10.1006/cviu.1998.0744.

[2] T. Ahmad, C. J. Taylor, A. Lanitis, and T. F. Cootes. Tracking and Recognizing Hand Gestures Using Statistical Shape Models. *Image Vision Comput.*, 15(5):345–352, 1997. 10.1016/S0262-8856(96)01136-5.

[3] H. Akaike and G. Kitagawa. *The Practice of Time Series Analysis.* Statistics for Engineering and Physical Science. Springer-Verlag, New York, NY, USA, 1999.

[4] K. Akita. Image Sequence Analysis of Real World Human Motion. *Pattern Recogn.*, 17(1):73–83, 1984. 10.1016/0031-3203(84)90036-0.

[5] R. M. Alexander. A Minimum Energy Cost Hypothesis for Human Arm Trajectories. *Biol. Cybern.*, 76(2):97–105, 1997. 10.1007/s004220050324.

[6] J. Alon, S. Sclaroff, G. Kollios, and V. Pavlović. Discovering Clusters in Motion Time-Series Data. In *Proc. 2003 IEEE Conf. Comput. Vis. and Pattern Recognit. – (CVPR'03)*, volume 1, pages 375–381. IEEE Computer Society Press, Madison, WI, USA, 2003. 10.1109/CVPR.2003.1211378.

[7] Y. Aminov. *The Geometry of Submanifolds.* CRC Press, 2001.

[8] P. Anandan. A Computational Framework and an Algorithm for the Measurement of Visual Motion. *Int. J. Comput. Vis.*, 2:283–310, 1989.

[9] C. G. Atkeson and J. M. Hollerbach. Kinematic Features of Unrestrained Vertical Arm Movements. *J. Neurosci.*, 5(9):2318–2330, 1985.

[10] E. Bayro-Corrochano. *Geometric Algebra with Applications in Science and Engineering.* Birkhäuser, 2001.

[11] E. Bayro Corrochano. *Geometric Computing for Perception Action Systems.* Springer-Verlag, New York, NY., USA., 2001.

[12] P. Ben-Horin and M. Shoham. Application of Grassmann–Cayley Algebra to Geometrical Interpretation of Parallel Robot Singularities. *Int. J. Robot. Res.*, 28(1):127–141, 2009. 10.1177/0278364908095918.

[13] D. Berndt and J. Clifford. Using Dynamic Time Warping to Find Patterns in Time Series. In *AAI Workshop on Knowledge Discovery in Databases*, pages 229–248. 1994.

[14] E. V. Biryoukova, A. Roby-Brami, A. A. Frolov, and M. Mokhtari. Kinematics of Human Arm Reconstructed from Spatial Tracking System Recordings. *J. Biomech.*, 33(8):985 – 995, 2000. 10.1016/S0021-9290(00)00040-3.

[15] M. J. Black, D. J. Fleet, and Y. Yacoob. A Framework for Modeling Appearance Change in Image Sequences. In *Proc. 6th IEEE Int. Conf. on Comput. Vis. – (ICCV'98)*, pages 660–667. Narosa House, Bombay, India, 1998. 10.1109/ICCV.1998.710788.

[16] M. J. Black and A. D. Jepson. A Probabilistic Framework for Matching Temporal Trajectories: CONDENSATION-based Recognition of Gestures and Expressions. In H. Burkhardt and B. Neumann, editors, *Proc. 5th European Conference on Computer Vision – (ECCV'98)*, volume LNCS 1046 of *Lecture Notes in Computer Science*, pages 909–924. Springer-Verlag, Freiburg, Germany, 1998.

[17] M. J. Black and A. D. Jepson. EigenTracking: Robust Matching and Tracking of Articulated Objects Using a View-based Representation. *Int. J. Comput. Vis.*, 26(1):63–84, 1998. 10.1023/A:1007939232436.

[18] M. J. Black and A. D. Jepson. Recognizing Temporal Trajectories Using the CONDENSATION Algorithm. In *Proc. 3rd IEEE Int. Conf. on Automatic Face and Gesture Recognition – (FG 1998)*, volume 1, pages 16–21. IEEE Computer Society, Nara, Japan, 1998. 10.1109/ICCV.1998.710788.

[19] M. J. Black and Y. Yacoob. Recognizing Facial Expressions in Image Sequences Using Local Parameterized Models of Image Motion. *Int. J. Comput. Vis.*, 25(1):23 – 48, 1997. 10.1023/A:1007977618277.

[20] A. Blake, B. Bascle, M. A. Isard, and J. MacCormick. Statistical Models of Visual Shape and Motion. *Phil. Trans. R. Soc. Lond. A.*, 356(1740):1283–1302, 1998. 10.1098/rsta.1998.0222.

[21] A. Blake, R. Curwen, and A. Zisserman. A Framework for Spatiotemporal Control in the Tracking of Visual Contours. *Int. J. Comput. Vis.*, 11(2):127–145, 1993. 10.1007/BF01469225.

[22] A. Blake, M. A. Isard, and D. Reynard. Learning to Track the Visual Motion of Contours. *Artif. Intell.*, 78(1-2):179–212, 1995. 10.1016/0004-3702(95)00032-1.

[23] S.-J. Blakemore and J. Decety. From the Perception of Action to the Understanding of Intention. *Nat. Rev. Neurosci.*, 2(8):561–567, 2001. 10.1038/35086023.

[24] A. F. Bobick. Movement, Activity and Action: The Role of Knowledge in the Perception of Motion. *Phil. Trans. R. Soc. Lond. B.*, 352(1358):1257–1265, 1997.

[25] A. F. Bobick and J. W. Davis. Real-time Recognition of Activity Using Temporal Templates. In *Proc. 3rd IEEE Workshop on Appl. of Comput. Vis. – (WACV '96)*, volume 1, pages 39–42. IEEE Computer Society Press, Sarasota, FL USA, 1996. 10.1109/ACV.1996.571995.

[26] A. F. Bobick and J. W. Davis. The Recognition of Human Movement Using Temporal Templates. *IEEE Trans. Pattern Anal. Mach. Intell.*, 23(3):257–67, 2001. 10.1109/34.910878.

[27] A. F. Bobick and Y. A. Ivanov. Action Recognition Using Probabilistic Parsing. In *Proc. 1998 IEEE Conf. Comput. Vis. and Pattern Recognit. – (CVPR'98)*, pages 196 – 202. IEEE Computer Society, Santa Barbara, CA. USA, 1998. 10.1109/CVPR.1998.698609.

[28] A. F. Bobick and A. Y. Johnson. Gait Recognition Using Static, Activity-Specific Parameters. In *Proc. 2001 IEEE Conf. Comput. Vis. and Pattern Recognit. – (CVPR'01)*, volume 1, pages I–423–30. IEEE Comput. Soc, Los Alamitos, CA, USA Kauai, HI, USA, 2001. 10.1109/CVPR.2001.990506.

[29] A. F. Bobick and A. D. Wilson. A State-based Approach to the Representation and Recognition of Gesture. *IEEE Trans. Pattern Anal. Mach. Intell.*, 19(12):1325–37, 1997. 10.1109/34.643892.

[30] P. Boulanger. *Extraction Multiéchelle d'Éléments Géometriques.* PhD thesis, École Polytechnique, 1994.

[31] P. Boulanger. Multiscale Edge Detection based on a New Geometrically Intrinsic Filter. In S. F. El-Hakim, editor, *Proceedings of the SPIE*, volume 2350 – Videometrics III, pages 264–278. The International Society of Optical Engineering, Boston, MA, USA, 1994. 10.1117/12.189138.

[32] P. Boulanger and P. R. Cohen. Adaptive Smoothing of Range Images Based on Intrinsic Surface Properties. In D. P. Casasent and A. G. Tescher, editors, *Proceedings of the SPIE Technical Symposium on Optical Engieneering and Photonics in Aerospace Sensing.*, volume 1297 – Hybrid Image and Signal Processing II, pages 254–263. The International Society for Optical Engineering, 1990. 10.1117/12.21320.

[33] P. Boulanger, O. Jokinen, and A. Beraldin. Intrinsic Filtering of Range Images Using a Physically Based Noise Model. In *Vision Interface 2002.* Calgary, Alberta, Canada, 2002.

[34] C. A. Bouman. Cluster: An Unsupervised Algorithm for Modeling Gaussian Mixtures, 1997.

[35] G. R. Bradski and J. W. Davis. Motion Segmentation and Pose Recognition with Motion History Gradients. *Mach. Vision Appl.*, 13(3):174–184, 2002. 10.1007/s001380100064.

[36] M. Brand and V. Kettnaker. Discovery and Segmentation of Activities in Video. *IEEE Trans. Pattern Anal. Mach. Intell.*, 22(8):844–851, 2000. 10.1109/34.868685.

[37] M. Brand, N. M. Oliver, and A. P. Pentland. Coupled Hidden Markov Models for Complex Action Recognition. In *Proc. 1997 IEEE Conf. on Comput. Vis. and Pattern Recognit. – (CVPR'97)*, pages 994–999. IEEE Computer Society Press, San Juan Puerto Rico, USA, 1997. 10.1109/CVPR.1997.609450.

[38] C. Bregler. Learning and Recognizing Human Dynamics in Video Sequences. In *Proc. 1997 IEEE Conf. on Comput. Vis. and Pattern Recognit. – (CVPR'97)*, pages 568 – 574. IEEE Computer Society Press, Los Alamitos, CA, USA San Juan, Puerto Rico, 1997. 10.1109/CVPR.1997.609382.

[39] C. Bregler, A. Hertzmann, and H. Biermann. Recovering Non-Rigid 3*D* Shape from Image Streams. In *Proc. 2000 IEEE Conf. on Comput. Vis. and Pattern Recognit. – (CVPR'00)*, pages 690–6 vol.2. IEEE Computer Society Press, Los Alamitos, CA, USA Hilton Head Island, SC, USA, 2000. 10.1109/CVPR.2000.854941.

[40] C. Bregler and J. Malik. Learning Appearance based Models: Mixtures of Second Moment Experts. In M. C. Mozer, M. I. Jordan, and T. Petsche, editors, *Neural Information Processing Systems 9 – (NIPS 1996)*, pages 845–851. MIT Press, Denver, Colorado, USA, 1996.

[41] C. Bregler and J. Malik. Tracking People with Twists and Exponential Maps. In *Proc. 1998 IEEE Conf. on Comput. Vis. and Pattern Recognit. – (CVPR'98)*, pages 8–15. IEEE Computer Society Press, Santa Barbara, CA, USA, 1998. 10.1109/CVPR.1998.698581.

[42] C. Bregler, J. Malik, and K. Pullen. Twist based Acquisition and Tracking of Animal and Human Kinematics. *Int. J. Comput. Vis.*, 56(3):179 – 194, 2004. 10.1023/B:VISI.0000011203.00237.9b.

[43] C. Bregler and S. M. Omohundro. Nonlinear Manifold Learning for Visual Speech Recognition. In *Proc. 1995 IEEE Int. Conf. on Comput. Vis. – (ICCV'95)*, pages 494 – 499. IEEE Computer Society Press, Los Alamitos, CA, USA Cambridge, MA, USA, 1995. 10.1109/ICCV.1995.466899.

[44] C. Bregler, S. M. Omohundro, and Y. Konig. A Hybrid Approach to Bimodal Speech Recognition. In A. Singh, editor, *Proc. 28th IEEE Asilomar*

Conf. on Signals, Syst. and Comput., volume 1, pages 556 – 560. IEEE Computer Society Press, Los Alamitos, CA, USA Pacific Grove, CA, USA, 1994. 10.1109/ACSSC.1994.471514.

[45] P. M. T. Broersen. Estimation of the Accuracy of Mean and Variance of Correlated Data. *IEEE Trans. Instrum. Meas.*, 47(5):1085–1091, 1998. 10.1109/19.746561.

[46] P. M. T. Broersen. The Quality of Models for ARMA Processes. *IEEE Trans. Signal Process.*, 46(6):1749–1752, 1998. 10.1109/78.678516.

[47] M. Browne, N. Mayer, and T. R. H. Cutmore. A Multiscale Polynomial Filter for Adaptive Smoothing. *Digit. Signal Process.*, 17(1):69–75, 2007. 10.1016/j.dsp.2006.01.006.

[48] S. Bryson. Measurement and calibration of static distortion of position data from 3D trackers. Technical Report RNR-92-011, National Aeronautical and Space Administration, 1992.

[49] S. R. Buss and J. P. Fillmore. Spherical Averages and Applications to Spherical Splines and Interpolation. *ACM T. Graphic*, 20(2):95–126, 2001. 10.1145/502122.502124.

[50] S. R. Buss and J.-S. Kim. Selectively Damped Least Squares for Inverse Kinematics. *JGT*, 10(3):37–49, 2005.

[51] T. Caelli. Shape Boundary Tracking with Hidden Markov Models. In F. J. Ferri, J. M. Iñesta Quereda, A. Amin, and P. Pudil, editors, *Proc. Jt. IAPR Int. Workshops SSPR 2000 and SPR 2000 on Advanc. in Pattern Recognit. [8th Int. Workshop on Structural and Syntactic Pattern Recognit.; 3rd Int. Workshop on Statistical Techniques in Pattern Recognit.]*, volume LNCS 1876 of *Lecture Notes in Computer Science*, pages 308–317. Springer-Verlag, Alicante, Spain, 2000. 10.1007/3-540-44522-6_32.

[52] T. Caelli, A. McCabe, and G. Binsted. On Learning the Shape of Complex Actions. In C. Arcelli, L. P. Cordella, and G. Sanniti di Baja, editors, *Proc. 4th Int. Workshop on Visual Form – (IWVF-4)*, volume 2059 of *Lecture Notes in Computer Science*, pages 24 – 39. Springer-Verlag, Capri, Italy, 2001. 10.1007/3-540-45129-3_3.

[53] L. W. Campbell and A. F. Bobick. Recognition of Human Body Motion Using Phase Constraints. In *Proc. 5th IEEE Int. Conf. on Comput. Vis. – (ICCV'95)*, pages 624–630. Cambridge, MA USA, 1995. 10.1109/ICCV.1995.466880.

[54] S. Carlsson. The Double Algebra: An Effective Tool for Computing Invariants In Computer Vision. In *Applications of Invariance in Computer Vision*, pages 145–164. 1994. 10.1007/3-540-58240-1_8.

[55] C. Cédras and M. Shah. Motion-based Recognition: A Survey. *Image Vision Comput.*, 13(2):129–155, 1995. 10.1016/0262-8856(95)93154-K.

[56] A. D. C. Chan, K. Englehart, B. Hudgins, and D. F. Lovely. Hidden Markov Model Cclassification of Myoelectric Signals in Speech. *IEEE Eng. Med. Biol. Mag.*, 21(5):143–146, 2002. 10.1109/MEMB.2002.1044184.

[57] I.-C. Chang and C.-L. Huang. The Model-based Human Body Motion Analysis System. *Image Vision Comput.*, 18(4):1067–1083, 2000. 10.1016/S0262-8856(00)00046-9.

[58] N. Chernov, C. Lesort, and N. Simnyi. On the Complexity of Curve Fitting Algorithms. *J. Complexity*, 20(4):484–492, 2004. 10.1016/j.jco.2004.01.004.

[59] C. K. Chui, L. Montefusco, and L. Puccio. *Wavelets : Theory, Algorithms, and Applications* . Academic Press, San Diego, CA, USA, 1994.

[60] J. J. Craig. *Introduction to Robotics: Mechanics and Control*. Prentice Hall, third edition, 1989.

[61] P. E. Crouch, G. Kun, and F. Silva Leite. The DeCasteljau Algorithm on Lie Groups and Spheres. *J. Dynam. Control Systems*, 5(3):397–429, 1999. 10.1023/A:1021770717822.

[62] E. B. Dam, M. Koch, and M. Lillholm. Quaternions, Interpolation, and Animation. Technical Report IT-C:758, University of Copenhagen, July 17, 1998.

[63] P. R. Davidson, R. D. Jones, J. H. Andreae, and H. R. Sirisena. Simulating Closed- and Open-loop Voluntary Movement: A Nonlinear Control-Systems Approach. *IEEE Trans. Biomed. Eng.*, 49(11):1242–52, 2002. 10.1109/TBME.2002.804601.

[64] J. W. Davis. Recognizing Movement Using Motion Histograms. Technical Report TR487, The Media Laboratory. Massachusetts Institute of Technology, April 30, 1998.

[65] J. W. Davis. Representing and Recognizing Human Motion: From Motion Templates to Movement Categories. In *Proc. 2001 IEEE/RSJ Int. Conf. on Digital Human Modeling Workshop – (IROS 2001)*. Maui, Hawaii, USA, 2001.

[66] J. W. Davis and A. F. Bobick. The Representation and Recognition of Human Movement Using Temporal Templates. In *Proc. 1997 IEEE Conf. on Comput. Vis. and Pattern Recognit. – (CVPR'97)*, pages 928–934. IEEE Comput. Soc, Los Alamitos, CA, USA San Juan, Puerto Rico, 1997. 10.1109/CVPR.1997.609439.

[67] J. S. Day, D. J. Murdoch, and G. A. Dumas. Calibration of position and angular data from a magnetic tracking device. *J. Biomech.*, 33(8):1039 – 1045, 2000. 10.1016/S0021-9290(00)00044-0 .

[68] D. Del Vecchio, R. M. Murray, and P. Perona. Decomposition of Human Motion into Dynamics based Primitives with Applications to Drawing Tasks. *IFAC Automatica*, 39(12):2085–2098, 2003. 10.1016/S0005-1098(03)00250-4.

[69] J. Deutscher, A. Blake, and I. Reid. Articulated Body Motion Capture by Annealed Particle Filtering. In *Proc. 2000 IEEE Conf. on Comput. Vis. and Pattern Recognit. – (CVPR'00),*, volume 2, pages 126–133. IEEE Computer Society Press, Hilton Head, SC USA, 2000. 10.1109/CVPR.2000.854758.

[70] J. Deutscher, A. Davison, and I. Reid. Automatic Partitioning of High Dimensional Search Spaces Associated with Articulated Body Motion Capture. In *Proc. 2001 IEEE Conf. on Comput. Vis. and Pattern Recognit. – (CVPR'01)*, volume 2, pages II–669 – II–676. IEEE Computer Society Press, Kauai, HI, USA, 2001. 10.1109/CVPR.2001.991028.

[71] J. Deutscher, B. North, B. Bascle, and A. Blake. Tracking Through Singularities and Discontinuities by Random Sampling. In *Proc. 7^{th} IEEE Int. Conf. on Comput. Vis. – (ICCV'99)*, volume 2, pages 1144–1149. IEEE Computer Society Press, Kerkyra, Greece, 1999. 10.1109/ICCV.1999.790409.

[72] G. S. Djordjević, M. Rasić, D. Kostić, and V. Potkonjak. Representation of Robot Motion Control Skill. *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, 30(2):219–238, 2000. 10.1109/5326.868444.

[73] P. M. Djurić and J.-H. Chun. An MCMC Sampling Approach to Estimation of Nonstationary Hidden Markov Models. *IEEE Trans. Signal Process.*, 50(5):1113 – 1123, 2002. 10.1109/78.995067.

[74] C. J. L. Doran and A. N. Lasenby. *Geometric Algebra for Physicists.* Cambridge University Press, 2003.

[75] T. Elfouhaily, S. Guignard, R. Awadallah, and D. R. Thompson. Local and Non-local Curvature Approximation: A New Asymptotic Theory for Wave Scattering. *Wave Random Media*, 13(Copyright 2003, IEE):321–37, 2003.

[76] X. Feng and P. Perona. Human Action Recognition by Sequence of Movelet Codewords. In *Proc. First Int. Symp. on 3D Data Processing Visualization and Transmission*, pages 717–721. IEEE Computer Society, Padova, Italy, 2002. 10.1109/TDPVT.2002.1024148 .

[77] M. Flanders, J. J. Pellegrini, and S. D. Geisler. Basic Features of Phasic Activation for Reaching in Vertical Planes. *Exp. Brain Res.*, 110(1):67–79, 1996. 10.1007/BF00241376.

[78] B. Friedland. *Control System Design: An Introduction to State Space Methods.* Dover Publications, 2005.

[79] J. Funda, R. H. Taylor, and R. P. Paul. On Homogeneous Transforms, Quaternions, and Computational Efficiency. *IEEE Trans. Robot. Autom.*, 6(3):382–388, 1990. 10.1109/70.56658.

[80] A. Galata, N. Johnson, and D. C. Hogg. Learning Behaviour Models of Human Activities. In T. Pridmore and D. Elliman, editors, *Proc. 1999 British Machine Vision Conference – (BMVC'99)*. The British Machine Vision Association, University of Nottingam, Nottingam, UK., 1999.

[81] A. Galata, N. Johnson, and D. C. Hogg. Learning Variable-Length Markov Models of Behavior. *Comput. Vis. Image Und.*, 81(3):398–413, 2001. 10.1006/cviu.2000.0894.

[82] D. M. Gavrila. Hermite Deformable Contours. In *Proc. 13th IEEE Int. Conf. on Pattern Recognit. – (ICPR'96)*, volume 1, pages 130–135. IEEE Computer Society Press, Vienna, Austria, 1996. 10.1109/ICPR.1996.546005.

[83] D. M. Gavrila. *Vision-based 3D Tracking of Humans in Action.* Ph. D. thesis, University of Maryland, 1996.

[84] D. M. Gavrila. The Visual Analysis of Human Movement: A Survey. *Comput. Vis. Image Und.*, 73(1):82–98, 1999. 10.1006/cviu.1998.0716.

[85] D. M. Gavrila and L. S. Davis. 3D Model-based Tracking of Human in Action: A Multi-view Approach. In *Proc. 1996 IEEE Conf. on Comput. Vis. and Pattern Recognit. – (CVPR'96)*, volume 1, pages 73–80. IEEE Computer Society Press, San Francisco, California, USA, 1996. 10.1109/CVPR.1996.517056.

[86] Z. Ghahramani. Factorial Learning and the EM Algorithm. In G. Tesauro, D. S. Touretzky, and T. K. Leen, editors, *Neural Information Processing Systems 7 – (NIPS 1994)*, pages 617–624. Morgan Kaufmann Publishers, Denver, Colorado, USA, 1994.

[87] Z. Ghahramani. An Introduction to Hidden Markov Models and Bayesian Networks. *Int. J. Pattern Recogn.*, 15(1):9–42, 2001. 10.1142/S0218001401000836.

[88] Z. Ghahramani and G. E. Hinton. Variational Learning for Switching State Space Models. *Neural Comput.*, 12(4):831 – 864, 2000. 10.1162/089976600300015619.

[89] Z. Ghahramani and S. T. Roweis. Learning Nonlinear Dynamical Systems Using and EM Algorithm. In M. S. Kearns, S. A. Solla, and D. A. Cohn, editors, *Neural Information Processing Systems 11 – (NIPS 1998)*, volume 11, pages 599–605. MIT Press, Denver, Colorado, USA, 1998.

[90] S. Gibet, P.-F. Marteau, and F. Julliard. Models with Biological Relevance to Control Anthropomorphic Limbs: A Survey. In I. Wachsmuth and T. Sowa, editors, *Proc. 2001 Int. Gesture Workshop: Gesture and Sign Languages in*

Human-Computer Interaction – (GW 2001), volume LNAI 2298 of *Lecture Notes in Computer Science Lecture Notes in Artificial Intelligence*, pages 253–269. Springer-Verlag, London, UK, 2001. 10.1007/3-540-47873-6_12.

[91] S. Gibet, J. Richardson, T. Lebourque, and A. Braffort. Corpus of 3D Natural Movements and Sign Language Primitives of Movement. In I. Wachsmuth and M. Fröhlich, editors, *Proc. 1997 Int. Gesture Workshop: Gesture and Sign Language in Human-Computer Interaction – (GW'97)*, volume LNAI 1371 of *Lecture Notes in Artificial Intelligence*, pages 111–121. Springer-Verlag, Bielefield, Germany, 1997.

[92] M. A. Giese and T. Poggio. Neural Mechanisms for the Recognition of Biological Movements. *Nat. Rev. Neurosci.*, 4(3):179–192, 2003. 10.1038/nrn1057.

[93] N. H. Goddard. The Interpretation of Visual Motion: Recognizing Moving Light Displays. In *Proc. 1989 IEEE Workshop on Visual Motion*, pages 212–220. Irvine, California, USA, 1989. 10.1109/WVM.1989.47112.

[94] N. H. Goddard. *The Perception of Articulated Motion: Recognizing Moving Light Displays.* Ph. D. thesis, University of Rochester, 1992.

[95] A. Gruen and D. Akca. Least Squares 3D Surface And Curve Matching. *ISPRS J. Photogramm.*, 59(3):151–174, 2005. 10.1016/j.isprsjprs.2005.02.006.

[96] R. Grzeszcuk, G. R. Bradski, M. H. Chu, and J.-Y. Bouguet. Stereo based Gesture Recognition Invariant to 3D Pose and Lighting. In *Proc. 2000 IEEE Conf. on Comput. Vis. and Pattern Recognit. – (CVPR 2000)*, volume 1, pages 826–833. 2000. 10.1109/CVPR.2000.855906 .

[97] S. W. R. Hamilton. *Elements of Quaternions.* Google Public Domain, 1866.

[98] A. J. Hanson. Quaternion Frenet Frames: Making Optimal Tubes and Ribbons from Curves. Technical Report TR-407, Indiana University, 1994.

[99] A. J. Hanson. *Visualizing Quaternions* . Series in interactive 3D technology. Morgan Kaufmann, 2006.

[100] I. Haritaoglu, D. Beymer, and M. Flickner. Ghost^{3D}: Detecting Body Posture and Parts Using Stereo. In *Proc. 2002 IEEE Workshop on Motion and Video Computing.*, pages 175–180. IEEE Computer Society, Orlando, Florida, 2002. 10.1109/MOTION.2002.1182231.

[101] I. Haritaoglu, D. Harwood, and L. S. Davis. Ghost: A Human Body Part Labeling System Using Silhouettes. In *Proc. 14th IEEE Int. Conf. on Pattern Recognit. – (ICPR'98)*, volume 1, pages 77–82. IEEE Computer Society Press, Brisbane, Australia, 1998. 10.1109/ICPR.1998.711084.

[102] J. C. Hart, G. K. Francis, and L. H. Kauffman. Visualizing Quaternion Rotation. *ACM T. Graphic*, 13(3):256–276, 1994. 10.1145/195784.197480.

[103] T. Hastie and W. Stuetzle. Principal Curves. *J. Am. Stat. Assoc.*, 84(406):502–516, 1989.

[104] J. B. Hayfron-Acquah, M. S. Nixon, and J. N. Carter. Automatic Gait Recognition by Symmetry Analysis. *Pattern Recogn. Lett.*, 24(13):2175–2183, 2003. 10.1016/S0167-8655(03)00086-2.

[105] S. Haykin. *Adaptive Filter Theory.* Prentice Hall Information and System Science. Prentice Hall, Englewood Cliffs, NJ 07632, USA, first edition, 1991.

[106] B. Heisele and C. Whler. Motion-based Recognition of Pedestrians. In *Proc. 14th IEEE Int. Conf. on Pattern Recognit. – (ICPR'98)*, volume 2, pages 1325–1330. IEEE Computer Society Press, Brisbane, Qld. Australia, 1998. 10.1109/ICPR.1998.711946.

[107] C. Hirsch. *Numerical Computation of Internal and External Flows* , volume 1. Elsevier, 2nd. edition edition, 2007.

[108] F. G. Hofmann, P. Heyer, and G. Hommel. Velocity Profile based Recognition of Dynamic Gestures with Discrete Hidden Markov Models. In I. Wachsmuth and M. Fröhlich, editors, *Proc. 1997 Int. Gesture Workshop on Gesture and Sign Language in Human-Computer Interaction*, volume LNAI 1371 of *Lecture Notes in Artificial Intelligence*, pages 81–95. Springer-Verlag, Bielefeld, Germany, 1997.

[109] M. Ikits, J. D. Brederson, C. D. Hansen, and J. M. Hollerbach. An Improved Calibration Framework for Electromagnetic Tracking Devices. In *Proc. 2001 IEEE Virtual Reality*, pages 63 – 70, Yokohama, Japan, 2001. IEEE Computer Press. 10.1109/VR.2001.913771 .

[110] S. S. Intille and A. F. Bobick. Recognizing Planned, Multiperson Action. *Comput. Vis. Image Und.*, 81(3):414–445, 2001. 10.1006/cviu.2000.0896.

[111] M. A. Isard and A. Blake. Contour Tracking by Stochastic Propagation of Conditional Density. In B. F. Buxton and R. Cipolla, editors, *Proc. 4th European Conference on Computer Vision – (ECCV'96)*, volume LNCS 1064 of *Lecture Notes in Computer Science*, pages 343–356. Springer-Verlag, Cambridge, UK, 1996.

[112] M. A. Isard and A. Blake. A Smoothing Filter for CONDENSATION. In H. Burkhardt and B. Neumann, editors, *Proc. 5th European Conference on Computer Vision – (ECCV'98)*, volume LNCS 1406 of *Lecture Notes in Computer Science*, pages 767–781. Springer-Verlag, Freiburg, Germany, 1998.

[113] Y. Iwai, K. Manjoh, and M. Yachida. Gesture and Posture Estimation by Using Locally Linear Regression. In F. J. Perales López and E. R. Hancock, editors, *Proc. 2nd Int. Workshop on Articulated Motion and Deformable Objects – (AMDO 2002)*, volume LNCS 2492 of *Lecture Notes in Computer*

Science, pages 177–188. Springer-Verlag, Palma de Mallorca, Spain, 2002. 10.1007/3-540-36138-3.

[114] Y. Jiu Lou and H. Han Pang. A Fast and Smooth Walking Pattern Generator of Biped Robot Using Jacobian Inverse Kinematics. In *Proc. 2007 IEEE Workshop on Advanced Robotics and Its Social Impacts – (ARSO 2007)*, pages 1–6. 2007. 10.1109/ARSO.2007.453141.

[115] G. Johansson. Visual Perception of Biological Motion and a Model for its Analysis. *Percep. Psychophys.*, 14(2):210–211, 1973.

[116] A. Y. Johnson and A. F. Bobick. A Multi-view Method for Gait Recognition Using Static Body Parameters. In J. Bigün and F. Smeraldi, editors, *Proc. 3^{rd} Int. Conf. on Audio- and Video-Based Biometric Person Authentication – (AVBPA 2001)*, pages 301–11. Springer-Verlag, Halmstad, Sweden, 2001. 10.1007/3-540-45344-X_44.

[117] S. X. Ju, M. J. Black, and Y. Yacoob. Cardboard People: A Parameterized Model of Articulated Image Motion. In *Proc. 2^{nd} IEEE Int. Conf. on Automatic Face and Gesture Recognition – (FG 1996)*, pages 38–44. IEEE Computer Society Press, Killington, VT, USA, 1996. 10.1109/AFGR.1996.557241.

[118] B. Julesz. Textons, the Elements of Texture Perception and Their Interactions. *Nature*, 290:91 – 97, 1981. 10.1038/290091a0.

[119] I. A. Kakadiaris and D. N. Metaxas. Model-based Estimation of $3D$ Human Motion. *IEEE Trans. Pattern Anal. Mach. Intell.*, 22(12):1453–1459, 2000. 10.1109/34.895978 .

[120] A. Kale, A. N. Rajagopalan, N. Cuntoor, and V. Kruger. Gait-based Recognition of Humans Using Continuous HMMs. In *Proc. 5^{th} Int. Conf. on Automatic Face and Gesture Recognition – (FG 2002)*, pages 321–326. Washington, D.C. USA, 2002. 10.1109/AFGR.2002.1004176.

[121] V. V. Kindratenko. Calibration of Electromagnetic Tracking Devices. *Virtual Reality: Research, Development, and Applications*, 4(2):139–150, 1999.

[122] V. V. Kindratenko. A survey of electromagnetic position tracker calibration techniques. *Virtual Reality: Res., Develop., Appls.*, 5(3):169 – 182, 2000.

[123] G. Kitagawa and H. Akaike. A Procedure for the Modeling of non-Stationary Time Series. *Ann. Inst. Stat. Math.*, 30(2):351 – 363, 1978.

[124] A. Kojima, T. Tamura, and K. Fukunaga. Natural Language Description of Human Activities from Video Images Based on Conceptual Hierarchy of Actions. *Int. J. Comput. Vis.*, 50(2):171–184, 2002. 10.1023/A:1020346032608.

[125] J. B. Kuipers. *Quaternions And Rotations Sequences.* Princenton University Press, 1999.

[126] S. LaScalza, J. Arico, and R. Hughes. Effect of Metal and Sampling Rate on Accuracy of Flock of Birds Electromagnetic Tracking System. *J. Biomech.*, 36(1):141 – 144, 2003. 10.1016/S0021-9290(02)00322-6.

[127] F. Leymarie and M. D. Levine. Tracking Deformable Objects in the Plane Using an Active Contour Model. *IEEE Trans. Pattern Anal. Mach. Intell.*, 15(6):617–34, 1993. 10.1109/34.216733.

[128] Y. Li, T. Wang, and H.-Y. Shum. Motion Texture: A Two-level Statistical Model for Character Motion Synthesis. In *Proceedings of the 2002 ACM Transactions on Graphics (SIGGRAPH 2002)*, volume 21 of *ACM Transactions on Graphics*, pages 465–472. ACM, 2002. 10.1145/566654.566604.

[129] J. Lo, G. Huang, and D. N. Metaxas. Human Motion Planning Based on Recursive Dynamics and Optimal Control Techniques. *Multibody Syst. Dyn.*, 8(4):433–458, 2002. 10.1023/A:1021111421247.

[130] J. B. Maas and G. Johansson. Motion Perception I: 2-Dimensional Motion Perception, 1971. Movie presentation.

[131] J. B. Maas and G. Johansson. Motion Perception II: 3-Dimensional Motion Perception, 1971. Movie presentation.

[132] D. J. C. MacKay. Choice of Basis for Laplace Approximation. *Mach. Learn.*, 33(1):77–86, 1998. 10.1023/A:1007558615313.

[133] R. MacKay. Estimating the Order of a Hidden Markov Model. *Can. J. Stat.*, 30(4):573–589, 2002.

[134] G. Mather, K. Radford, and S. West. Low-level Visual Processing of Biological Motion. *Proc. R. Soc. Lond. B*, 249(1325):149–155, 1992.

[135] C. G. M. Meskers, H. Fraterman, F. C. T. van der Helm, H. M. Vermeulen, and P. M. Rozing. Calibration of the “Flock of Birds” Electromagnetic Tracking Device and Its Application in Shoulder Motion studies. *J. Biomech.*, 32(6):629 – 633, 1999. 10.1016/S0021-9290(99)00011-1 .

[136] T. B. Moeslund and E. Granum. A Survey of Computer Vision-based Human Motion Capture. *Comput. Vis. Image Und.*, 81(3):231–268, 2001. 10.1006/cviu.2000.0897.

[137] F. Mokhtarian and M. Bober. *Curvature Scale Space Representation: Theory, Applications, and MPEG-7 Standardization (Computational Imaging and Vision).* Springer, 2003.

[138] R. Moreau, M. T. Pham, O. Olaby, O. Dupuis, and T. Redarce. Paths Analysis for a Safe Forceps Blades Placement on the BirthSIM Simulator. In *Proc. IEEE International Conference on Robotics and Automation (ICRA'06)*, pages 739–744. Orlando, USA, 2006.

[139] R. Moreau, M. T. Pham, T. Redarce, and O. Dupuis. A New Learning Method For Obstetric Gestures Using The BirthSIM Simulator. In *Proc. IEEE International Conference on Robotics and Automation (ICRA'07)*, pages 2279–2284. Roma, Italy, 2007.

[140] R. Moreau, M. T. Pham, R. Silveira, T. Redarce, X. Brun, and O. Dupuis. Design of a New Instrumented Forceps: Application to Safe Obstetrical Forceps Blade Placement. *IEEE Trans. Biomed. Eng.*, 54(7):1280–1290, 2007.

[141] G. Mullineux. Modeling Spatial Displacements Using Clifford Algebra. *ASME J. Mech. Design*, 126(3):420–424, 2004. 10.1115/1.1701876.

[142] R. M. Neal, M. J. Beal, and S. T. Roweis. Inferring State Sequences for Non-linear Systems with Embedded Hidden Markov Models. In S. Thrun, L. K. Saul, and B. Schölkopf, editors, *Neural Information Processing Systems 16 – (NIPS 2003)*. MIT Press, Vancouver, British Columba, Canada, 2003.

[143] P. Neri, M. C. Morrone, and D. C. Burr. Seeing Biological Motion. *Nature*, 395, 1998. 10.1038/27661.

[144] C. W. Ng and S. Ranganath. Real-time Gesture Recognition System and Application. *Image Vision Comput.*, 20(13-14):993–1007, 2002. 10.1016/S0262-8856(02)00113-0.

[145] K. C. Nishikawa, S. T. Murray, and M. Flanders. Do Arm Postures Vary with the Speed of Reaching? *J. Neurophysiol.*, 81(5):2582, 1999.

[146] V. M. Ochoa Mayorga, P. Boulanger, and M. Garcia. Local Quaternion Weighted Difference Functions for Orientation Calibration on Electromagnetic Trackers. In *Proc 1st IEEE Int. Workshop on Computational Advances in Multi-Sensor Adaptive Processing*, page 4. Puerto Vallarta, Mexico, 2005. 10.1109/CAMAP.2005.1574227.

[147] V. Pavlović, B. J. Frey, and T. S. Huang. Time-series Classification Using Mixed State Dynamic Bayesian Networks. In *Proc. 1999 IEEE Conf. on Comput. Vis. and Pattern Recognit. – (CVPR'99)*, volume 2, page 615. IEEE Computer Society Press, Fort Collins, CO USA, 1999. 10.1109/CVPR.1999.784983.

[148] V. Pavlović and J. M. Rehg. Impact of Dynamic Model Learning on Classification of Human Motion. In *Proc. 2000 IEEE Conf. on Comput. Vis. and Pattern Recognit. – (CVPR'00)*, volume 1, pages 788 – 795. IEEE Computer Society, Hilton Head Island, SC, USA, 2000. 10.1109/CVPR.2000.855901.

[149] V. I. Pavlović, R. Sharma, and T. S. Huang. Visual Interpretation of Hand Gestures for Human-Computer Interaction: A Review. *IEEE Trans. Pattern Anal. Mach. Intell.*, 19(7):677–695, 1997. 10.1109/34.598226.

[150] W. D. Penny and S. J. Roberts. Dynamic Logistic Regression. In *Proc. 1999 IEEE Int. Jt. Conf. on Neural Netw. – (IJCNN '99)*, volume 3, pages 1562–1567. Washington DC, USA, 1999. 10.1109/IJCNN.1999.832603.

[151] W. D. Penny and S. J. Roberts. Dynamic Models for Nonstationary Signal Segmentation. *Comput. Biomed. Res.*, 32(6):483–502, 1999. 10.1006/cbmr.1999.1511.

[152] W. D. Penny and S. J. Roberts. EEG-Based Communication: A Pattern Recognition Approach. *IEEE Trans. Rehabil. Eng.*, 8(2):214–215, 2000. 10.1109/86.847820.

[153] W. D. Penny and S. J. Roberts. Hidden Markov Independent Components for Biosignal Analysis. In *Proc. First Int. Conf. on Advances in Medical Signal and Information Processing*, volume 1, pages 244–250. Bristol, UK, 2000.

[154] W. D. Penny and S. J. Roberts. Variational Bayes for Generalized Autoregressive Models. *IEEE Trans. Signal Process.*, 50(9):2245–2257, 2002. 10.1109/TSP.2002.801921.

[155] P.-O. Persson and G. Strang. Smoothing by Savitzky-Golay and Legrend Filters. In J. Rosenthal, editor, *Mathematical Systems Theory in Biology, Communications, Computation and Finance.* Springer, 2003.

[156] R. W. Picard. *Affective Computing.* MIT Press, 1997.

[157] L. Piegl and W. Tiller. *The NURBS Book.* Monographs in Visual Communication. Springer-Verlag, 1997.

[158] R. Polana and R. Nelson. Detecting Activities. In *Proc. 1993 IEEE Conf. on Comput. Vis. and Pattern Recognit. – (CVPR'93)*, volume 1, pages 2–7. New York, NY USA, 1993. 10.1109/CVPR.1993.341009.

[159] R. Polana and R. Nelson. Low Level Recognition of Human Motion (or How to Get Your Man Without Finding His Body Parts). In *Proc. 1994 IEEE Workshop on Motion of Non-Rigid and Articulated Objects*, volume 1, pages 77–82. Austin, TX USA, 1994. 10.1109/MNRAO.1994.346251.

[160] A. Psarrou, S. Gong, and M. Walter. Recognition of Human Gestures and Behaviour based on Motion Trajectories. *Image Vision Comput.*, 20(5 - 6):349 – 358, 2002. 10.1016/S0262-8856(02)00007-0.

[161] L. R. Rabiner. A Tutorial on Hidden Markov Models and Selected Application in Speech Recognition. *Proc. IEEE*, 77(2):257–286, 1989. 10.1109/5.18626.

[162] R. Ramamoorthi, C. Ball, and A. H. Barr. Dynamic Splines with Constraints for Animation. Technical Report CaltechCSTR:1997.cs-tr-97-03, California Institute of Technology, 1997//03 1997.

[163] L. Reng, T. B. Moeslund, and E. Granum. Finding Motion Primitives in Human Body Gestures. In S. Gibet, N. Courty, and c. Kamp, Jean-Fran editors, *Proc. 6th Int. Gesture Workshop - Gesture in Human-computer Interaction and Simulation – (GW 2005)*, volume 3881 NAI of *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, pages 133–144. Springer Verlag, Heidelberg, D-69121, Germany, Berder Island, France, 2005. 10.1007/11678816.

[164] G. Rigoll, S. Eickeler, and S. Müller. Person Tracking in Real-World Scenarios Using Statistical Methods. In *Proc. 4th IEEE Int. Conf. on Automatic Face and Gesture Recognition – (FG 2000)*, volume 1, pages 342 – 347. Grenoble, France, 2000. 10.1109/AFGR.2000.840657.

[165] G. Rigoll, A. Kosmala, and S. Eickeler. High Performance Real-time Gesture Recognition Using Hidden Markov Models. In I. Wachsmuth and M. Fröhlich, editors, *Proc. 1997 Int. Gesture Workshop: Gesture and Sign Language in Human-Computer Interaction – (GW'97)*, volume LNAI 1371 of *Lecture Notes in Artificial Intelligence*, pages 69–80. Springer-Verlag, Bielefeld, Germany, 1997.

[166] G. Rizzolatti, L. Fogassi, and V. Gallese. Neurophysiological Mechanisms Underlying the Understanding and Imitation of Action. *Nat. Rev. Neurosci.*, 2(9):661–670, 2001. 10.1038/35090060.

[167] S. T. Roweis and Z. Ghahramani. A Unifying Review of Linear Gaussian Models. *Neural Comput.*, 11(2):305–345, 1999. 10.1162/089976699300016674.

[168] S. T. Roweis and Z. Ghahramani. An EM Algorithm for Identification of Nonlinear Dynamical Systems. In S. Haykin, editor, *Kalman Filtering and Neural Networks*. Wiley, 2000.

[169] T. Rydén. Estimating the Order of Hidden Markov Models. *Statistics*, 26(4):345–354, 1995.

[170] A. Savitzky and M. J. E. Golay. Smoothing and Differentiation of Data by Simplified Least Squares Procedures. *Anal. Chem.*, 36(8):1627–1639, 1964. 10.1021/ac60214a047.

[171] L. Sciavicco and B. Siciliano. A Dynamic Soluton to the Inverse Kinematic Problem for Redundant Manipulators. In *Proc. 1987 IEEE Int. Conf. on Robot. and Autom. – (ICRA'87)*, volume 4, pages 1081–1087. IEEE Computer Society Press, Washington, DC, USA, Raleigh, NC, USA, 1987.

[172] C. E. Shannon. Prediction and Entropy of Printed English. *AT&T Labs Tech. J.*, 30:50–64, 1951.

[173] K. Shoemake. Animating Rotation with Quaternion Curves. In *Proc. 12th Annu. SIGGRAPH Int. Conf. on Comput. Graphics and Interact. Tech.*, pages 245–254. ACM Press, San Francisco, CA, USA, 1985. 10.1145/325334.325242.

[174] A. Shokoufandeh, S. Dickinson, C. Jöhnsson, L. Bretzner, and T. Lindeberg. On the Representation and Matching of Qualitative Shape at Multiple Scales. In A. Heyden, G. Sparr, M. Nielsen, and P. Johansen, editors, *Proc. 7^{th} European Conference on Computer Vision – (ECCV 2002)*, volume LNCS 2352 of *Lecture Notes in Computer Science*, pages 759–775. Springer-Verlag, Copenhagen, Denmark, 2002.

[175] C. C. Silva and R. d. A. Martins. Polar and Axial Vectors Versus Quaternions. *Am. J. Phys.*, 70(9):958–963, 2002. 10.1119/1.1475326.

[176] J. F. Soechting, C. A. Buneo, U. Herrmann, and M. Flanders. Moving Effortlessly in Three Dimensions: Does Donders' Law Apply to Arm Movement? *J. Neurosci.*, 15(9):6271–, 1995.

[177] J. Steinier, Y. Termonia, and J. Deltour. Comments on Smoothing and Differentiation of Data by Simplified Least Square Procedure. *Anal. Chem.*, 44(11):1906–1909, 1972. 10.1021/ac60319a045.

[178] J. Sullivan, M. Eriksson, and S. Carlsson. Recognition, Tracking, and Reconstruction of Human Motion. In F. J. Perales López and E. R. Hancock, editors, *Proc. 2^{nd} Int. Workshop on Articulated Motion and Deformable Objects – (AMDO'02)*, volume LNCS 2492 of *Lecture Notes in Computer Science*, pages 142–154. Springer-Verlag, Palma de Mallorca, Spain, 2002.

[179] S. Sumi. Upside-down Presentation of the Johansson Moving light-spot Pattern. *Perception*, 13(3):283–286, 1984.

[180] R. Tanawongsuwan and A. F. Bobick. Gait Recognition from Time-normalized Joint-angle Trajectories in the Walking Plane. In *Proc. 2001 IEEE Conf. on Comput. Vis. and Pattern Recognit. – (CVPR'01)*, volume 2, pages II–726–31 vol.2. IEEE Comput. Soc, Los Alamitos, CA, USA Kauai, HI, USA, 2001. 10.1109/CVPR.2001.991036.

[181] Y. Y. Tang, S.-W. Lee, and C. Y. Suen. Automatic Document Processing: A Survey. *Pattern Recogn.*, 29(12):1931–1952, 1996. 10.1016/S0031-3203(96)00044-1 .

[182] D. Terzopoulos and R. Szeliski. Tracking with Kalman Snakes. In A. Blake and A. L. Yuille, editors, *Active Vision*, Artificial Intelligence, pages 3–20. MIT Press, Cambridge, Massachusetts, USA, 1992.

[183] D. Tolani, A. Goswami, and N. I. Badler. Real-Time Inverse Kinematics Techniques for Anthropomorphic Limbs. *Graph. Models*, 62(5):353, 2000. 10.1006/gmod.2000.0528.

[184] E. B. Torres and D. Zipser. Reaching to Grasp with a Multi-jointed Arm I. Computational Model. *J. Neurophysiol.*, 88:2355–2367, 2002. 10.1152/jn.00030.2002.

[185] E. B. Torres and D. Zipser. Simultaneous Control of Hand Displacements and Rotations in Orientation-matching Experiments. *J. Appl. Physiol.*, 96(5):1978, 2004.

[186] M. K. Viblis and K. J. Kyriakopoulos. Gesture Fecognition: The Gesture Segmentation Problem. *J. Intell. Robot. Syst.*, 28(1-2):151–158, 2000. 10.1023/A:1008101200733.

[187] C. Vogler and D. N. Metaxas. ASL Recognition Based on a Coupling Between HMMs and 3D Motion Analysis. In *Proc. 6th Int. Conf. on Comput. Vis. – (ICCV'98)*, pages 363–369. IEEE Computer Society Press, Bombay, India, 1998. 10.1109/ICCV.1998.710744.

[188] L.-C. T. Wang and C. C. Chen. A Combined Optimization Method for Solving the Inverse Kinematics Problem of Mechanical Manipulators. *IEEE Trans. Robot. Autom.*, 7(4):489–499, 1991. 10.1109/70.86079.

[189] J. A. Webb and J. K. Aggarwal. Structure from Motion of Rigid and Jointed Objects. *Artif. Intell.*, 19(1):107–130, 1982. 10.1016/0004-3702(82)90023-6.

[190] N. L. White. GrassmannCayley Algebra and Robotics. *J. Intell. Robot. Syst.*, 11(1):91–107, 1994. 10.1007/BF01258296.

[191] A. D. Wilson and A. F. Bobick. Nonlinear PHMMs for the Interpretation of Parameterized Gesture. In *Proc. 1998 IEEE Conf. on Comput. Vis. and Pattern Recognit. – (CVPR'98)*, pages 879–884. IEEE Computer Society Press, Los Alamitos, CA, USA Santa Barbara, CA, USA, 1998. 10.1109/ICCV.1998.710739.

[192] A. D. Wilson and A. F. Bobick. Recognition and Interpretation of Parametric Gesture. In *Proc. 6th IEEE Int. Conf. on Comput. Vis. – (ICCV'98)*, pages 329–36. Narosa Publishing House, Bombay, India, 1998. 10.1109/ICCV.1998.710739.

[193] A. D. Wilson and A. F. Bobick. Parametric Hidden Markov Models for Gesture Recognition. *IEEE Trans. Pattern Anal. Mach. Intell.*, 21(9):884–900, 1999. 10.1109/34.790429.

[194] C. R. Wren and A. P. Pentland. Dynamic Models of Human Motion. In *Proc. 3rd IEEE Int. Conf. on Automatic Face and Gesture Recognition – (FG 1998)*, pages 22–27. IEEE Computer Society Press, 1998. 10.1109/AFGR.2000.840662.

[195] C. Xu and J. Prince. Snakes, Shapes, and Gradient Vector Flow. *IEEE Trans. Image Process.*, 7(3):359–69, 1998. 10.1109/83.661186.

[196] C. Xu and J. L. Prince. Gradient Vector Flow: A New External Force for Snakes. In *Proc. 1997 IEEE Conf on Comput. Vis. and Pattern Recognit. –*

(CVPR'97), Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pages 66–71. IEEE, Los Alamitos, CA, USA, San Juan, PR, USA, 1997. 10.1109/CVPR.1997.609299.

[197] G. Zachman. Distortion Correction of Magnetic Fields for Position Tracking. In *Proceedings of the 1997 IEEE International Conference on Computer Graphics.*, pages 213 – 220, Hasselt and Diepenbeek, 1997. 10.1109/CGI.1997.601306.

[198] Z. Zhang. Iterative Point Matching for Registration of Free Form Curves. Technical report, INRIA, March 1992.

[199] S.-C. Zhu, C.-e. Guo, Y. Wang, and Z. Xu. What are Textons? *Int. J. Comput. Vis.*, 62(1-2):121 – 143, 2005. 10.1023/B:VISI.0000046592.70770.61.

[200] Y. Zhu, G. Xu, and D. J. Kriegman. A Real-Time Approach to the Spotting, Representation, and Recognition of Hand Gestures for Human-Computer Interaction. *Comput. Vis. Image Und.*, 85(3):189–208, 2002. 10.1006/cviu.2002.0967.

Appendix A

A.1 Frenet-Serret Frames In Higher Dimensions

One can show by induction and Gram-Schmidt decomposition that the Frenet-Serret formulas can be written in a brief form as:

$$\boldsymbol{\zeta}'_s = \boldsymbol{K}_s \boldsymbol{\zeta}_s \tag{A.1.1}$$

where:

$$\begin{bmatrix} \dfrac{d\boldsymbol{\zeta}_1}{ds} \\ \dfrac{d\boldsymbol{\zeta}_2}{ds} \\ \dfrac{d\boldsymbol{\zeta}_3}{ds} \\ \vdots \\ \dfrac{d\boldsymbol{\zeta}_{n-1}}{ds} \\ \dfrac{d\boldsymbol{\zeta}_n}{ds} \end{bmatrix} = \begin{bmatrix} 0 & \kappa_1 & 0 & \cdots & 0 & 0 & 0 \\ -\kappa_1 & 0 & \kappa_2 & \cdots & 0 & 0 & 0 \\ 0 & -\kappa_2 & 0 & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & -\kappa_{n-2} & 0 & -\kappa_{n-1} \\ 0 & 0 & 0 & \cdots & 0 & -\kappa_{n-1} & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{\zeta}_1 \\ \boldsymbol{\zeta}_2 \\ \boldsymbol{\zeta}_3 \\ \vdots \\ \boldsymbol{\zeta}_{n-1} \\ \boldsymbol{\zeta}_n \end{bmatrix} \tag{A.1.2}$$

Where $\boldsymbol{\zeta}_i$ $(i = 1, \ldots, n)$ denote the vectors of the local ortho-normal Frenet-Serret basis and κ_j $(j = 1, \ldots, n-1)$ the generalized curvatures of the n-dimensional curve respectively.

By applying the properties of the inner product and the collinearity of two multi-vectors, it is easy to demonstrate that the formulas expressing the generalized j^{th}-dimensional curvature κ_j with respect to the arc length parametrization can be express as [7]:

$$\begin{aligned} &\kappa_1 = |\boldsymbol{\gamma}' \wedge \boldsymbol{\gamma}''| = \|\boldsymbol{\zeta}'_1\| \\ &\kappa_2 = \frac{|\boldsymbol{\gamma}' \wedge \boldsymbol{\gamma}'' \wedge \boldsymbol{\gamma}'''|}{|\boldsymbol{\gamma}' \wedge \boldsymbol{\gamma}''|^2} \\ &\vdots \\ &\kappa_{n-1} = |\boldsymbol{\gamma}' \wedge \ldots \wedge \boldsymbol{\gamma}^{(n)}| \frac{|\boldsymbol{\gamma}' \wedge \ldots \wedge \boldsymbol{\gamma}^{(n-2)}|}{|\boldsymbol{\gamma}' \wedge \ldots \wedge \boldsymbol{\gamma}^{(n-1)}|^2} \end{aligned} \tag{A.1.3}$$

where $\wedge$ denotes the outer product for multi-vectors as defined by Grassmann's algebra [12, 54, 190].

Note that a multi-vector of n vectors $\boldsymbol{\gamma}' \wedge \ldots \wedge \boldsymbol{\gamma}^{(n)}$ in $\mathbb{R}^n$ has only one component which is equal to the determinant of the matrix formed by the components of these vectors. That is to say:

$$k_{n-1} = \det\left(\boldsymbol{\gamma}', \ldots, \boldsymbol{\gamma}^{(n)}\right) \frac{|\boldsymbol{\gamma}' \wedge \ldots \wedge \boldsymbol{\gamma}^{(n-2)}|}{|\boldsymbol{\gamma}' \wedge \ldots \wedge \boldsymbol{\gamma}^{(n-1)}|^2} \tag{A.1.4}$$

Let us define the vectors $\dot{\boldsymbol{\gamma}}_t$, $\ddot{\boldsymbol{\gamma}}_t$, ..., $\boldsymbol{\gamma}_t^{(n)}$ as the successive derivatives of $\boldsymbol{\gamma}(t)$ with respect to the time variable t. If the parametrization is not the arc-length but the time, then evidently:

$$|\boldsymbol{\gamma}' \wedge \boldsymbol{\gamma}''| = |\dot{\boldsymbol{\gamma}}_t \wedge \ddot{\boldsymbol{\gamma}}_t| \left|\frac{dt}{ds}\right|^3 \tag{A.1.5}$$

And

$$\|\boldsymbol{\gamma}'\| = 1 = \|\dot{\boldsymbol{\gamma}}_t\| \left|\frac{dt}{ds}\right|^3 \tag{A.1.6}$$

So the formula for k_1 becomes:

$$k_1 = \frac{|\dot{\boldsymbol{\gamma}}_t \wedge \ddot{\boldsymbol{\gamma}}_t|}{\|\dot{\boldsymbol{\gamma}}_t\|^3}. \tag{A.1.7}$$

Furthermore, one have:

$$k_1^2 k_2 = |\dot{\boldsymbol{\gamma}}_t \wedge \ddot{\boldsymbol{\gamma}}_t \wedge \dddot{\boldsymbol{\gamma}}_t| \tag{A.1.8}$$

Hence:

$$k_2 = \frac{|\dot{\boldsymbol{\gamma}}_t \wedge \ddot{\boldsymbol{\gamma}}_t \wedge \dddot{\boldsymbol{\gamma}}_t|}{\|\dot{\boldsymbol{\gamma}}_t\| \, |\dot{\boldsymbol{\gamma}}_t \wedge \ddot{\boldsymbol{\gamma}}_t|} \tag{A.1.9}$$

One can rewrite (A.1.3) and (A.1.4) with a time parametrization as following:

$$\begin{aligned} k_{j-1} &= |\dot{\boldsymbol{\gamma}}_t \wedge \ldots \wedge \boldsymbol{\gamma}_t^{(j)}| \frac{|\dot{\boldsymbol{\gamma}}_t \wedge \ldots \wedge \boldsymbol{\gamma}_t^{(j-2)}|}{\|\dot{\boldsymbol{\gamma}}_t\| \, |\dot{\boldsymbol{\gamma}}_t \wedge \ldots \wedge \boldsymbol{\gamma}_t^{(j-1)}|^2} \\ k_{n-1} &= \det\left(\dot{\boldsymbol{\gamma}}_t, \ldots, \boldsymbol{\gamma}_t^{(n)}\right) \frac{|\dot{\boldsymbol{\gamma}}_t \wedge \ldots \wedge \boldsymbol{\gamma}_t^{(n-2)}|}{\|\dot{\boldsymbol{\gamma}}_t\| \, |\dot{\boldsymbol{\gamma}}_t \wedge \ldots \wedge \boldsymbol{\gamma}_t^{(n-1)}|^2} \end{aligned} \tag{A.1.10}$$

A.2 Geometrical Interpretation of Derivatives with Savitzky-Golay Filter

Although the method has fallen in disuse because numerical instability, it provides a useful geometrical interpretation. Assume a matrix $\boldsymbol{A} = [\boldsymbol{a}_1, \cdots, \boldsymbol{a}_i, \cdots, \boldsymbol{a}_n]$ where $\boldsymbol{a}_i$ is a column-vector of $\boldsymbol{A}$. The projection of a vector $\boldsymbol{a}$ onto $\boldsymbol{e}$ is defined as:

$$\operatorname{proj}_{\boldsymbol{e}} \boldsymbol{a} = \frac{\boldsymbol{e} \cdot \boldsymbol{a}}{\|\boldsymbol{e}\|} \hat{\boldsymbol{e}} \tag{A.2.1}$$

then

$$\begin{aligned}
\boldsymbol{u}_1 &= \boldsymbol{a}_1 & \hat{\boldsymbol{e}}_1 &= \frac{\boldsymbol{u}_1}{\|\boldsymbol{u}_1\|} \\
\boldsymbol{u}_2 &= \boldsymbol{a}_2 - \operatorname{proj}_{\boldsymbol{e}_1} \boldsymbol{a}_2 & \hat{\boldsymbol{e}}_2 &= \frac{\boldsymbol{u}_2}{\|\boldsymbol{u}_2\|} \\
\boldsymbol{u}_3 &= \boldsymbol{a}_3 - \operatorname{proj}_{\boldsymbol{e}_1} \boldsymbol{a}_3 - \operatorname{proj}_{\boldsymbol{e}_2} \boldsymbol{a}_3 & \hat{\boldsymbol{e}}_3 &= \frac{\boldsymbol{u}_3}{\|\boldsymbol{u}_3\|} \\
&\vdots & &\vdots \\
\boldsymbol{u}_n &= \boldsymbol{a}_n - \sum_{i=1}^{n-1} \operatorname{proj}_{\boldsymbol{e}_i} \boldsymbol{a}_i & \hat{\boldsymbol{e}}_n &= \frac{\boldsymbol{u}_n}{\|\boldsymbol{u}_n\|}
\end{aligned} \tag{A.2.2}$$

It follows that

$$\boldsymbol{a}_n = \sum_{i=1}^{n} (\boldsymbol{e}_i \cdot \boldsymbol{a}_i)\, \hat{\boldsymbol{e}}_i \tag{A.2.3}$$

and $\boldsymbol{e}_i \cdot \boldsymbol{a}_i = \|\boldsymbol{u}_i\|$. The matrix form of $\boldsymbol{A}$ is

$$\boldsymbol{A} = \boldsymbol{Q}\boldsymbol{R} \tag{A.2.4}$$

and

$$\boldsymbol{Q} = [\boldsymbol{e}_1, \ldots, \boldsymbol{e}_n] \qquad \boldsymbol{R} = \begin{bmatrix} \|\boldsymbol{u}_{1,1}\| & \|\boldsymbol{u}_{1,2}\| & \|\boldsymbol{u}_{1,3}\| & \cdots \\ 0 & \|\boldsymbol{u}_{2,2}\| & \|\boldsymbol{u}_{2,3}\| & \cdots \\ 0 & 0 & \|\boldsymbol{u}_{3,3}\| & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix} \tag{A.2.5}$$

The off-diagonal values contain the residuals from the vector projections to the corresponding main vector axes. In other words, the inversion of the QR product produces an averaging filter while the corresponding inversion of the residuals in $\boldsymbol{R}$ produces the corresponding difference filter $\boldsymbol{B}(z)$, *i.e.*,

$$\boldsymbol{B}(z) = k\boldsymbol{R}^{-1}(z)\boldsymbol{R}^{-\mathsf{T}}(z)\boldsymbol{J}(z), \tag{A.2.6}$$

the constant value k is an adjusting parameter to adjust the derivative magnitude.

A.3 Quaternions

A.3.1 Invariant Quaternion Metric

A distance function for a geodesic-arc between two quaternions $\boldsymbol{a}$ and $\boldsymbol{b}$ is defined in terms of the quaternion dot product:

$$d(\boldsymbol{a}, \boldsymbol{b}) = \quad \boldsymbol{a}^{-1}\boldsymbol{b} = \boldsymbol{a}^{-1} \cdot \boldsymbol{b} = \boldsymbol{b} \cdot \boldsymbol{a}^{-1} \tag{A.3.1}$$

is invariant for the $4D$-space. Hereafter we will denote the geodesic-arc for any quaternion as the quaternion product $\boldsymbol{a}^{-1}\boldsymbol{b}$.

A.3.2 Gram-Schmidt Quaternion Decomposition

Quaternions are bi-vectors in a $4D$-space, they can also use the same tools for vector analysis. A quaternion $\boldsymbol{a}$ can be decomposed into two orthogonal components using a reference quaternion $\boldsymbol{b}$:

$$\begin{aligned} \boldsymbol{a}_{\parallel} &= (\boldsymbol{a}\cdot\boldsymbol{b})\boldsymbol{b} \\ \boldsymbol{a}_{\perp} &= \boldsymbol{a} - (\boldsymbol{a}\cdot\boldsymbol{b})\boldsymbol{b} \end{aligned} \tag{A.3.2}$$

Using the Clifford algebras for the quaternion product:

$$\begin{aligned} \boldsymbol{a}\boldsymbol{b} &= \boldsymbol{a}\cdot\boldsymbol{b} + I\,\boldsymbol{a}\wedge\boldsymbol{b} \\ \boldsymbol{b}\boldsymbol{a} &= \boldsymbol{a}\cdot\boldsymbol{b} - I\,\boldsymbol{a}\wedge\boldsymbol{b} \end{aligned} \tag{A.3.3}$$

such that

$$\begin{aligned} \boldsymbol{a}\cdot\boldsymbol{b} &= \frac{1}{2}(\boldsymbol{a}\boldsymbol{b} + \boldsymbol{b}\boldsymbol{a}) \\ (\boldsymbol{a}\wedge\boldsymbol{b})\,I &= \frac{1}{2}(\boldsymbol{a}\boldsymbol{b} - \boldsymbol{b}\boldsymbol{a}) \end{aligned} \tag{A.3.4}$$

The distance function using Gram-Schmidt decomposition defines a distance function in terms of rotations of quaternions using vector reflections such that:

$$d(\boldsymbol{a},\boldsymbol{b}) = -\boldsymbol{n}^{*}\boldsymbol{a}\boldsymbol{n} = -\boldsymbol{n}\boldsymbol{a}\boldsymbol{n}^{*} = -\frac{1}{2}\boldsymbol{a}^{-1}\boldsymbol{b}. \tag{A.3.5}$$

where the quaternion $\boldsymbol{n}$ defines the reflection plane between $\boldsymbol{a}$ and $\boldsymbol{b}$.

A.3.3 Quaternion Interpolation Functions

Quaternion interpolation uses the distance definition for quaternions in (A.3.1) along with quaternion exponentiation. The exponentiation of a quaternion $\boldsymbol{a}$ by a value $t \in \mathbb{R}$ is defined as

$$\boldsymbol{a}^{t} = \mathrm{e}^{\ln \boldsymbol{a}^{\mathsf{T}}} = \mathrm{e}^{t\ln \boldsymbol{a}} = \|a\|^{t}\,\mathrm{e}^{t\theta\hat{\boldsymbol{\xi}}} \tag{A.3.6}$$

where the term $\mathrm{e}^{t\theta\hat{\boldsymbol{\xi}}}$ is periodic for $t = (-1)^{n}n\pi$. This implies that the domain is limited to $[0,\pi)$.

The general function for interpolation on the unit quaternion space is the Spherical Linear Interpolation formula for two limit quaternions $\widehat{\boldsymbol{a}}$ and $\widehat{\boldsymbol{b}}$:

$$\widehat{\boldsymbol{a}}(t) = -(\widehat{\boldsymbol{a}}^{-1}\widehat{\boldsymbol{b}})^{t}\widehat{\boldsymbol{a}} \quad \forall\ 0 \le t \le 1,\ \ t \in \mathbb{R} \tag{A.3.7}$$

It is easy to show that Equation (A.3.7) corresponds to Shoemake's original formula [173]:

$$\begin{aligned} \widehat{\boldsymbol{a}}(t) &= \frac{\sin(1-t)\Omega}{\sin\Omega}\widehat{\boldsymbol{b}} - \frac{\sin t\Omega}{\sin\Omega}\widehat{\boldsymbol{a}} \quad 0 \le t \le 1 \\ \Omega &= \frac{1}{2}(\theta_{b} - \theta_{a}) \end{aligned} \tag{A.3.8}$$

Then, an invariant quaternion distance is defined by the interpolation function as

$$d(\boldsymbol{a},\boldsymbol{b}) = -\widehat{\boldsymbol{a}}^{-1}\widehat{\boldsymbol{b}}\widehat{\boldsymbol{a}} = 2\boldsymbol{a}^{-1}\boldsymbol{b} \tag{A.3.9}$$