GLOBAL JOURNAL OF ENGINEERING SCIENCE AND RESEARCHES

A STUDY ON SHORTEST DISTANCE BETWEEN TWO ALGORITHMS CURVES

Mr.CH.Murali Krishna

Department of General Engineering and Basic science, Sri vidyanikethan Engineering college, Tirupathi

ABSTRACT

In this paper, we present the geodesic-like algorithm for the computation of the shortest path between two objects on NURBS surfaces and periodic surfaces. This method can improve the distance problem not only on surfaces but in R 3 . Moreover, the geodesic-like algorithm also provides an efficient approach to simulate the minimal geodesic between two holes on a NURBS surfaces. In this paper I find the shortest distance between two curves by "minima and maxima of functions of f(x,y)".A branch of mathematics that is a sort of generalization of calculus. Calculus of variations seeks to find the path, curve, surface, etc., for which a given function has a stationary value (which, in physical problems, is usually a minimum or maximum). Mathematically, this involves finding stationary values of integrals of the form.

$$I = \int_b^a f(y, \dot{y}, x)\, dx.$$

And it can finds applications in many diverse fields, such as aeronautics (maximizing the lift of an airplane wing), sporting equipment design (minimizing air resistance on a bicycle helmet, optimizing the shape of a ski), mechanical engineering (maximizing the strength of a column, a dam, or anarchy), boat design (optimizing the shape of a boat hull), physics (calculating and geodesics in both classical mechanics and general relativity).We begin with two examples illustrating the types of problems that may be solved using calculus of variations. Techniques in calculus of variation can represents the shortest path (i.e., geodesic) between two given points on a surface. Find the curve between two given points in the plane that yields a surface of revolution of minimum area when revolved around a given axis. Find the curve along which a bead will slide (under the effect of gravity) in the shortest time. A study has taken on various linear and non- linear models to determine the shortest distance between two curves

Keywords- Distance, Geodesic-like curves, Orthogonal projection, Parametric surface, Shape analysis,. Bezier curves.

I. INTRODUCTION

Finding shortest paths is a fundamental problem with numerous applications. There are several variations, including single-source, point-to-point, and all-pairs shortest paths. The single-source problem with non negative arc lengths has been studied most extensively [4, 6, 7, 8, 15, 16, 17, 18, 20, 25, 30, 38, 41]. For this problem, near-optimal algorithms are known both in theory, with near-linear time bounds, and in practice, where running times are within a small constant factor of the breadth-first search time. A local maximum point on a function is a point (x,y) on the graph of the function whose y coordinate is larger than all other y coordinates on the graph at points "close to" (x,y). More precisely, (x,f(x)) is a local maximum if there is an interval (a,b) with $a<x<b$ and $f(x)\geq f(z)$ for every z in (a,b). Similarly, (x,y) is a local minimum point if it has locally the smallest y coordinate. Again being more precise: (x,f(x)) is a local .minimum if there is an interval (a,b) with $a<x<b$ and $f(x)\leq f(z)$ for every z in (a,b). A local extreme is either a local minimum or a local maximum. Now a days Linked Data is a method of publishing structured data so that it can be interlinked and become more meaningful. It builds on standard Web technologies such as HTTP for deducing the distance between two curves. There is a significant sweeping sphere clipping method is presented for minimum distance computation problem between two object O1 and O2 can be described as to find the nearest point pair (p, q) such that $p \in O1$, $q \in O2$, and the distance between p and q is minimum. It is an important problem in many fields such as geometric modeling [4,8], computer graphics [5,11], and computer vision [4,8,11,13]. In geometric modeling, the distance information between a point and a curve or surface is essential for interactively selecting curves and surfaces. Distance information is needed for collision detection in CAD/CAM, computer graphics and computer vision. Moreover, the minimum distance information and the location where the minimum distance occurs are very useful in the location design of a bridge or undersea tunnel. The shorter the length of a bridge, the lower the cost and the less the corrosion of waves. The basic formula for Minimizing the distance between two points is equivalent to minimizing the square of the distance between two points. In that case

the quantity we want to minimize which we will denote by s^2 is $s^2 = \Delta x^2 + \Delta y^2$. In the case you specified, where we are calculating the distance between any point on the curve $y = x^2$ and $(1, 4)$, the distance formula takes on the form:

$$s^2 = \left(4 - x^2\right)^2 + (1 - x)^2$$

Let's go ahead and minimize this value with respect to x.

$$\frac{ds^2}{dx} = 2\left(4 - x^2\right)(-2x) - 2(1 - x) = 0$$

II. COMPUTING THE SHORTEST DISTANCE BETWEEN TWO NORMAL CURVES

The following steps can be involved to find shortest distance between two curves
1) You can calculate the distance between any two points on the curves and then minimize this. It will give you the minimum distance between the curves Let (x1, y1) be any point on f(x) and let (x2, y2) be any point on g(x). Now, the distance between them can be found using distance formula. Then minimize this to get the minimum which is what you are looking for this example is very simple and we already know the answer. However, formalizing it will be of help later. The problem consists in finding the shortest path. Between two points in the plane, A=(x1, y1) and B=(x2, y2).We already know that the answer is simply the straight line connecting the two points, but we will go through this solution using the framework of calculus of variations. Suppose that x1≠x2 and that it is possible to write t

2) You have f(x) and g(x) with you. So you can write y1 in terms of x1 and y2 in terms of x2.
Now you have the distance formula in only two variables x1 and x2. Now you need to identify

III. COMPUTING THE SHORTEST DISTANCE BETWEEN TWO BEZIER CURVES

A sweeping sphere clipping method is presented for computing the minimum distance between two Bezier curves. The sweeping sphere is constructed by rolling a sphere with its center point along a curve. The initial radius of the sweeping sphere can be set as the minimum distance between an end point and the other curve. The nearest point on a curve must be contained in the sweeping sphere along the other curve, and all of the parts outside the sweeping sphere can be eliminated. A simple sufficient condition when the nearest point is one of the two end points of a curve is provided, which turns the curve/curve case into a point/curve case and leads to higher efficiency. Examples are shown to illustrate efficiency and robustness of the new method.

IV. COMPUTING THE SHORTEST DISTANCE BETWEEN TWO FRECHET CURVES

1) As a measure for the resemblance of curves in arbitrary dimensions we consider the so-called Fréchet-distance, which is compatible with parameterizations of the curves.. It is the minimum length of a leash required to connect a dog and its owner, constrained on two separate paths, as they walk without backtracking along their respective curves from one endpoint to the other. The definition is symmetric with respect to the two curves. Imagine a dog walking along one curve and the dog's owner walking along the other curve, connected by a leash. Both walk continuously along their respective curve from the prescribed start point to the prescribed end point of the curve. Both may vary their speed, and even stop, at arbitrary positions and for arbitrarily long periods of time. However, neither can backtrack. The Fréchet distance between the two curves is the length of the shortest leash (not the shortest leash that is sufficient for all walks, but the shortest leash of all the leashes) that is sufficient for traversing both curves in this manner. Let A and B be two given curves in S. Then, the Fréchet distance between A and B is defined as the infimum over all re-parameterizations α and β of $[0, 1]$ of the maximum over all $t \in [0, 1]$ of the distance in S between $A(\alpha(t))$ and $B(\beta(t))$. In mathematical notation, the Fréchet distance $F(A, B)$ is

$$F(A,B) = \inf_{\alpha,\beta} \max_{t\in[0,1]} \left\{ d\Big(A(\alpha(t)), B(\beta(t))\Big) \right\}$$

2) Algorithm using an approximated distance given a possibly infinite sequence of points defining a polygonal path, maintain in a streaming setting, a simplification of this set of points with 2k points and a bounded error. The error is measured using the Hausdorff distance or the Fr´echet distance. The Fr´echet distance being computationally too costly for this framework, they show that error(i, j) can be upper and lower bounded by functions of two values, namely ω(i, j) and b(i, j). ω(i, j) is the width of the points of P(i, j) in the direction pipj . b(i, j) is the length of the longest back path in the direction −−→pipj . Its precise definition requires the following other definitions:
Definition 1. Let l be a straight line of directional vector −→d . α is the angle between l and the abscissa axis. projα(p) denotes the orthogonal projection of p onto the line of angle α. If −−→plpm. −→d < 0, then projα(pl) is "after" projα(pm) in direction α, and we denote projα(pl) >> projα(pm) (see Figure 1(a)).
Definition 2. −−→plpm is a positive shift if and only if −−→plpm. −→d > 0, negative otherwise. Definition 3. A backpath in direction α is a negative shift −−→plpm such that l < m (see Figure 1(b)). pl is the origin of the backpath. The length of the backpath is equal to d(projα(pl), projα(pm)). Lemma 2 relates ω(i, j) and b(i, j) to error(i, j):

V. COMPUTING THE SHORTEST DISTANCE BETWEEN TWO PARAMETRIC CURVES

This paper describes an algorithm for computation of the Hausdorff distance between sets of plane algebraic rational parametric curves. The Hausdorff distance is one of the frequently used similarity measures in geometric pattern matching algorithms. It is defined for arbitrary non-empty bounded and closed sets A and B. Hausdorff distance assigns to each point of one set the distance to its closest point on the other and takes the maximum over all these values. We work with Euclidean distance as point to point distance measure. A plane parametric curve is a mapping f : [a, b] → R 2 , which assings to each parameter value t ∈ [a, b] a point in the plane R 2 . The curve is defined by two functions x(t) : I → R and y(t) : I → R and a parameter values interval I = [a, b]. The algorithm presented here is restricted to the curves with rational parameterization, ie. x(t), y(t) may be polynomials or rational functions, furtheremore the rational parameterezation function must be continuously differentiable on the definition interval I. We also describe a polygon line approximation algorithm, based on Douglas-Peucker polygon line simplification algorithm.

Example : Let A, B ⊂ R 2 – compact, we define the one-sided Hausdorff distance from A to B as ˜δH(A, B) = max a∈A min b∈B d(a, b). (1) . The (bidirectional) Hausdorff distance between A and B is defined as δH (A, B) = max(˜δH (A, B), ˜δH(B, A)) (2) Notice that δH (A, B) = inf{ε > 0|A ⊆ Uε(B) and B ⊆ Uε(A)},

VI. CURVE MATCHING SHORTEST DISTANCE

The problem of curve matching appears in many application domains, like time series analysis, shape matching, speech recognition, and signature verification, among others. Curve matching has been studied extensively by computational geometers, and many measures of similarity have been examined, among them being the Frechet distance (sometimes referred in folklore as the "dog- ´ man" distance). A measure that is very closely related to the Frechet distance but has never been studied in a geometric ´ context is the Dynamic Time Warping measure (DTW), first used in the context of speech recognition. This measure is ubiquitous across different domains, a surprising fact because notions of similarity usually vary significantly depending on the application. However, this measure suffers from some drawbacks, most importantly the fact that it is defined between sequences of points rather than curves. Thus, the way in which a curve is sampled to yield such a sequence can dramatically affect the quality of the result. Some attempts have been made to generalize the DTW to continuous domains, but the resulting algorithms have exponential complexity. we propose similarity measures that attempt to capture the "spirit" of dynamic time warping while being defined over continuous domains, and present efficient algorithms for computing them. Our formulation leads to a very interesting connection with finding short paths in a combinatorial manifold defined on the input chains, and in a deeper sense relates to the way light travels in a medium of variable refractivity.

VII. PARAMETRIC VUBIC SPINE CURVES

These curves are commonly used to model the geometry of road surfaces in real-time driving simulators. Roads are represented by space curves that define a curvilinear frame of reference in which three-dimensional points are expressed in coordinates of distance along the curve, offset from the central axis, and loft from the road surface. Simulators must map from global Cartesian coordinates to local road coordinates at very high frequencies. A key component in this mapping is the computation of the closest point on the central axis of the road to a three-dimensional point expressed in Cartesian coordinates. The paper investigates a two-step method that exploits the complementary strengths of two optimization techniques: Newton's method and quadratic minimization. The Problem A parametric cubic spline curve modeling the centerline of a curved road can be expressed as [5], (x(s), y(s), z(s)), $0 \le s \le L$, where s denotes arc length, L is the arc length of the entire spline curve, and x(s), y(s), and z(s) are cubic spline functions with equally spaced breakpoints {s0, s1, ..., sn} with s0 = 0 and sn = L. The functions x, y, and z are C2 on [0, L]. At each time step of a simulation, the dynamics module computes a new position in Cartesian coordinates for every moving object. Given an object's location in Cartesian coordinates, our problem is to find the closest point on a road centerline to the object. Let p0 = (x0, y0, z0) be the position of an object (see Figure 1). The square of the distance between position p0 and position (x(s), y(s), z(s))

VIII. CONCLUSION

In this paper, we considered the problem of computing the discrete Fr´echet distance between two polygonal curves either approximately or exactly. Our main contribution is a simple approximation framework that leads to efficient $(1 + \varepsilon)$-approximation algorithms for two families of common curves: κ-bounded curves and backbone curves. We also considered the exact algorithm for general curves, and proposed a pseudo–output-sensitive algorithm by observing that only a subset of the white cells from the free-space diagram are necessary for the decision problem. It will be interesting to investigate whether there are families of curves that are guaranteed to have small Φ, which is the maximum number of boundary cells, over all possible values of the threshold δ. We are currently working on extending the pseudo-output-sensitive algorithms to the continuous weak Fr´echet distance. It might be hard to develop algorithms that are significantly sub-quadratic in the worst case, given that no such algorithm exists for the related and widely studied problem of computing the edit distance for strings. Hence our future directions will focus on practical variants of the Fr´echet distance that can handle outliers, partial matching, and/or efficient multiple-curve alignment. Another important direction is to develop efficient (approximation) algorithms for minimizing the Fr´echet distance under transformations such as rigid motions. There is a another view -closest point computation is a core component of real-time ground vehicle simulation. It forms an essential step in the process of mapping from Cartesian coordinates to road coordinates needed to place synthetic agents on the road network. To satisfy the requirements of real-time simulation, the closest point computation must be efficient and extremely robust. Finally A novel method is proposed to compute the minimum distance between two 2D or 3D NURBS curves using control polygons in an efficient and robust way to determine short distance between two curves .

IX. ACKNOWLEDGEMENTS

We would like to thank Dr R.Srinivasa Rao for many discussions on the topic of curve matching measures, vector calculus and differential geometry. We also would like to thanks the anonymous reviewers for very fruit full advice's.

REFERENCES

1. *P. K. Agarwal, K. Varadarajan. Efficient algorithms for approximating polygonal chains Discrete Comput. Geom. 23 (2000), 273–291.*
2. *H. Alt, B. Behrends, and J. Blomer. Approximate matching of polygonal shapes. ¨ Ann. Math. Artif. Intell., 13:251–266, 1995.*
3. *K. Buchin, M. Buchin and C. Wenk, Computing the Frechet Distance Between Simple Polygons in P*
4. *Helmut Alt, Bernd Behrens, and Johannes Blömer. Approximate matching of polygonal shapes. Annals of Mathematics and Artificial Intelligence, 13:251– 265, 1995.*
5. *Helmut Alt and Leonidas J. Guibas. Discrete geometric shapes: Matching, interpolation, and approximation. In Handbook of computational geometry. Elsiever Science B.V., 1999.*
6. *O. Aichholzer, F. Aurenhammer, C. Icking, R. Klein, E. Langetepe, and G. Rote. Generalized self-approaching curves. Discrete Applied Mathematics, 109:3–24, 2001.*
7. *X.D. Chen, Y. Zhou, Z. Shu, H. Su, J.C. Paul, Improved algebraic algorithm on point projection for Bézier curves. International Multi-Symposiums on Computer and Computational Sciences, 17 (2007), pp. 158–163*

8. *H. Alt, A. Efrat, G. Rote, and C. Wenk. Matching planar maps. J. Algorithms, 49:262–283, 2003.*
9. *Helmut Alt and Maike Buchin. Semi-computability of the Fr´echet distance between surfaces. In Proc. 21st European Workshop on Computational Geometry, pages 45–48, March 2005.*
10. *Atkinson, K., Modelling a road using spline interpolation, Reports on Computational Mathematics # 145, Department of Mathematics, The University of Iowa, (2002).*