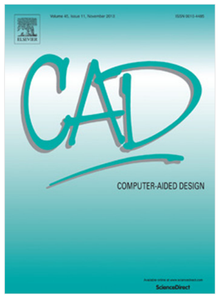

Lightweight preprocessing and fast query of geodesic distance via proximity graph

Shiqing Xin, Wenping Wang, Ying He, Yuanfeng Zhou, Shuangmin Chen, Changhe Tu, Zhenyu Shu

PII: S0010-4485(18)30230-6
DOI: https://doi.org/10.1016/j.cad.2018.04.021
Reference: JCAD 2614

To appear in: *Computer-Aided Design*

Lightweight Preprocessing and Fast Query of Geodesic Distance via Proximity Graph

Shiqing Xin[a], Wenping Wang[b], Ying He[c], Yuanfeng Zhou[a], Shuangmin Chen[d,*], Changhe Tu[a], Zhenyu Shu[e]

[a]*School of Computer Science and Technology, Shandong University*
[b]*Department of Computer Science, The University of Hongkong*
[c]*School of Computer Engineering, Nanyang Technological University*
[d]*School of Information Science and Technology, Qingdao University of Science and Technology*
[e]*School of Information Science and Engineering, Ningbo Institute of Technology, Zhejiang University*

Abstract

Computing geodesic distance on a mesh surface $\mathcal{S}$ efficiently and accurately is a central task in numerous computer graphics applications. In order to deal with high-resolution mesh surfaces, a lightweight preprocessing is a proper choice to make a balance between query accuracy and speed. In the preprocessing stage, we build a proximity graph $\mathcal{G}$ with regard to a set of sample points and keep the exact geodesic distance between any pair of nearby sample points. In the query stage, given two query points s and t, we augment the proximity graph $\mathcal{G}$ by adding s and t on-the-fly, and then use the shortest path between s and t on the augmented proximity graph to approximate the exact geodesic path between s and t. We establish an empirical relationship between the number of samples and expected accuracy (measured in relative error), which facilitates fast and accurate query of geodesic distance with a lightweight processing cost. We exhibit the uses of the new approach in two applications – real-time computation of discrete exponential map for texture mapping and interactive design of spline curves on surfaces.

Keywords: proximity graph, geodesic distance, real-time interaction, exponential map, spline curves

1. Introduction

Fast and accurate query of distance map [1] is central to many computer graphics applications, such as surface segmentation and edit [2, 3], mesh skinning [4] and watermarking [5], Poisson sampling [6], meshing [7], isometry-invariant shape retrieval [8], intrinsic Voronoi tessellation [9, 10, 11, 12], minimum-distortion parametrization [13, 14, 15] and non-rigid correspondence [16]. Also, the research topic has a close relationship with many other research fields [17], such as medical imaging [18], and robot motion planning [19], and architectural geometry [20].

In light of the fact that exact geodesic algorithms [21, 22, 23, 24, 25, 26, 27] are time consuming and hard to be applied to large mesh surfaces, one has to make a balance between accuracy and speed in many real computer graphics applications. In the last decades, there is a large body of literature on approximate geodesic computation [28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39]. But the status quo is that it is still hard to find a suitable geodesic algorithm in practice. A natural question arises: Can we achieve an accurate and fast distance query with a lightweight processing (roughly linear to the model complexity)? Furthermore, in many computer graphics applications, e.g., discrete exponential map generation, we hope the reported geodesic distance is as accurate as possible if measured with relative error. Obviously, the existing approximation approaches cannot meet the couple of requirements.

In this paper we shall present a fast and accurate method for geodesic computation on a 2D manifold surface. The key idea of the method is to use a sequence of precomputed short and exact geodesic paths to approximate the geodesic path between any two given points on a given base surface. Specifically, in a processing step, we generate a set of evenly distributed sample points on the base surface and compute the exact geodesic distance from each sample point to those in its neighborhood of some specified range. This produces a proximity graph of all the sample points. In the query step, when two query points s and t are given, we add s and t to the set of sample points to generate an augmented proximity graph and then compute the shortest path from s to t on this augmented graph to approximate the exact geodesic between the two points, as illustrated in Figure 1.

*Corresponding author: csmqq@163.com (S. Chen).