Computers & Graphics 35 (2011) 677–684

Contents lists available at ScienceDirect

Computers & Graphics

journal homepage: www.elsevier.com/locate/cag

Short Communication to SMI 2011

Computation of the minimum distance between two Bézier curves/surfaces ☆

Jung-Woo Chang [a], Yi-King Choi [a], Myung-Soo Kim [b], Wenping Wang [a,*]

[a] Department of Computer Science, The University of Hong Kong, Hong Kong
[b] School of Computer Science and Engineering, Seoul National University, Republic of Korea

ARTICLE INFO

Article history:
Received 10 December 2010
Received in revised form
13 March 2011
Accepted 14 March 2011
Available online 6 April 2011

Keywords:
Minimum distance computation
Bézier curve and surface
Bounding volume hierarchy

ABSTRACT

We present an efficient and robust method based on the culling approach for computing the minimum distance between two Bézier curves or Bézier surfaces. Our contribution is a novel dynamic subdivision scheme that enables our method to converge faster than previous methods based on binary subdivision.

1. Introduction

The computation of minimum distance between two models is an important geometric query which is used in various areas including robotics, computer graphics, computer games, simulation, virtual reality and haptic prototyping [7]. In this paper, we propose a minimum distance computation algorithm for two Bézier curves/surfaces.

Existing approaches that compute the minimum distance between two free-form curves/surfaces fall in two groups: root-finding-based methods and culling-based methods. The root-finding-based methods compute the solution of geometric queries by solving a set of non-linear equations. Root-finding-based methods, when used alone, tend to suffer from inefficiency, since it needs to compute all local minima between two curves/surfaces and most of them are redundant.

The general framework of the culling-based approach for distance computation of two curves or surfaces [7] uses a hierarchical structure of subdivision and invokes three operations: estimation of the lower bound of minimum distance between nodes, estimation of the upper bound of minimum distance between nodes, and subdivision of nodes where a node represents a curve segment or a surface patch as a result of subdivision.

During the subdivision process, a global upper bound of the minimum distance is always maintained, which is defined as the smallest of the upper bounds for the minimum distance for all pairs of nodes. If the lower bound of the minimum distance for a pair of nodes is greater than the global upper bound, then the pair is pruned and it is excluded from further consideration in minimum distance computation. Otherwise the pair is kept and its upper bound is found and used to improve the global upper bound. The difference between the global upper bound and the lower bound of each node pair is used to guide the subdivision procedure—if the difference is smaller than a pre-specified threshold, the computation for the pair is terminated; otherwise the nodes of the pair are subdivided. The whole procedure is terminated when the differences of all the existing pairs are smaller than the threshold. The convergence of such an algorithm is measured by how fast this difference between the global upper bound and the lower bound for a pair of nodes converges to zero as the subdivision level increases.

In this paper, we propose an algorithm based on the culling approach for computing the minimum distance between Bézier curves or Bézier surfaces. Our contribution is a new scheme of node subdivision that makes use of the closest pair of points on the convex hulls of control points. We shall show that this new subdivision scheme provides a much faster convergence than binary subdivision.

The rest of this paper is organized as follows. We review the previous work on minimum distance computation in Section 2. Section 3 describes our algorithm. In Section 4 we show the efficiency of our algorithm by giving several experimental results. Finally, we conclude this paper in Section 5.

2. Preliminaries

2.1. Minimum distance between polyhedral models

Early works about minimum distance computation often focused on the minimum distance between rather simple primitives such as

☆The work of W. Wang was partially supported by the Research Grant Council of Hong Kong (718209 and 718010), and the State Key Program of NSFC project (60933008). This work was also supported in part by NRF Research Grant (2010-0014351).

* Corresponding author. Tel.: +852 28597074; fax: +852 25598447.
E-mail addresses: jungwoochang@gmail.com (J.-W. Chang), wenping@cs.hku.hk (W. Wang).

0097-8493/$ - see front matter © 2011 Elsevier Ltd. All rights reserved.
doi:10.1016/j.cag.2011.03.025

convex polyhedral models. Lin and Canny [11] proposed a method that computes the distance between two convex polyhedra by using the Voronoi regions of convex polyhedra. Gilbert et al. [4] proposed an algorithm about minimum distance between two convex objects. The algorithm is well known as the GJK algorithm and is widely used due to its efficiency and simplicity. There are efficient and robust implementations of the GJK algorithm [14].

After the late 90s, algorithms on distance computation between non-convex triangular meshes were proposed. Johnson and Cohen [7] presented a general framework for minimum distance computation. A bounding volume hierarchy-based approach which is widely used to solve collision detection problems [5,6,8] is used to compute the minimum distance between meshes. For distance computation between triangular meshes, Johnson and Cohen [7] used oriented bounding box (OBB) as the bounding volume. Instead of OBB, Larsen et al. [9] proposed to use rectangle swept sphere (RSS) since computing distance between two RSSs is much more efficient than that between two OBBs.

2.2. Minimum distance between curves/surfaces

The minimum distance between two parametric curves can be acquired by solving the following equations [13]:

$$(C_1(u)-C_2(v))\cdot\frac{dC_1(u)}{du}=0,$$

$$(C_1(u)-C_2(v))\cdot\frac{dC_2(v)}{dv}=0.$$

Similarly, the minimum distance between two parametric surfaces can be acquired by solving the following equations:

$$(S_1(u,v)-S_2(s,t))\cdot\frac{\partial S_1(u,v)}{\partial u}=0,$$

$$(S_1(u,v)-S_2(s,t))\cdot\frac{\partial S_1(u,v)}{\partial v}=0,$$

$$(S_1(u,v)-S_2(s,t))\cdot\frac{\partial S_2(s,t)}{\partial s}=0,$$

$$(S_1(u,v)-S_2(s,t))\cdot\frac{\partial S_2(s,t)}{\partial t}=0.$$

The methods for solving a set of non-linear equations as given above can be found in [3,12].

Elber and Grandine [2] presented a method for finding the minimum and Hausdorff distance between two parametric curves or surfaces by solving a set of non-linear equations. Lennerz and Schömer [10] proposed a method that can reduce a set of non-linear equations to univariate polynomials for finding minimum distance between two quadratic curves or surfaces.

The general framework of Johnson and Cohen [7] which is mentioned above can also be used to compute minimum distance between B-Spline surfaces. On the other hand, Chen et al. [1] proposed a culling-based algorithm which computes the minimum distance between two Bézier curves. Given two Bézier curves $C_1(u)$ and $C_2(v)$, the bivariate function $S(u,v)=(C_1(u)-C_2(v))^2$ can be expressed in Bernstein form. The minimum distance between $C_1(u)$ and $C_2(v)$ is then computed by using subdivision of bivariate Bernstein polynomial and the sphere clipping method.

3. Algorithm

In this section, we will present the details of our method. We will first introduce the general framework for computing the minimum distance. Then we will proceed to describe the method that computes the minimum distance between two Bézier curves, and finally describe the method for dealing with two Bézier surfaces.

3.1. General framework

We adopt the culling-based framework to compute the minimum distance between two Bézier curves/surfaces. We will describe the following basic operations: the estimation of lower bound of the minimum distance between two nodes, the estimation of the upper bound of the minimum distance, and node subdivision. A pruning scheme based on these operations is shown in Algorithm 1. Throughout the algorithm, a global upper bound of the minimum distance is maintained which converges to the actual minimum distance between two curves/surfaces.

Algorithm 1 (*Computing the minimum distance between a given pair of nodes*).

Require: The global upper bound of minimum distance α, two nodes N_1, N_2, and a tolerance ε
  if upperbound(N_1,N_2) $\le\alpha$ **then**
    $\alpha\leftarrow$ upperbound(N_1,N_2)
  end if
  if lowerbound(N_1,N_2) $\ge\alpha(1-\varepsilon)$ **then**
    return α
  else
    Subdivide N_1 into N_{11} and N_{12}
    Subdivide N_2 into N_{21} and N_{22}
    $\alpha\leftarrow\min(\alpha$,Algorithm 1($N_{11}$,$N_{21}$,$\alpha$))
    $\alpha\leftarrow\min(\alpha$,Algorithm 1($N_{11}$,$N_{22}$,$\alpha$))
    $\alpha\leftarrow\min(\alpha$,Algorithm 1($N_{12}$,$N_{21}$,$\alpha$))
    $\alpha\leftarrow\min(\alpha$,Algorithm 1($N_{12}$,$N_{22}$,$\alpha$))
  end if
  return α

Given any two nodes, each representing a subdivided part of a curve or a surface, we first compute the upper bound of the minimum distance between two nodes. If the computed upper bound is less than the global upper bound of minimum distance, the global upper bound is updated by the computed upper bound. After that, the lower bound of minimum distance between two nodes is computed. If the computed lower bound is greater than the global upper bound, we can then guarantee that the minimum distance between the two objects does not occur in the given pair of nodes. Thus, the given pair of nodes can be culled out. This culling is the main source of performance improvement of this approach compared to root-finding-based methods. If the computed lower bound is not greater than the global upper bound, we need to further subdivide the given nodes and perform the distance computation for the subdivided nodes.

The minimum distance computation between two objects can be solved by calling Algorithm 1 with two root nodes that correspond to the entire curves or surfaces, and an initial α which is larger than the minimum distance between the two objects.

3.2. Minimum distance between two Bézier curves

Let two Bézier curves be defined as follows:

$$C_1(u)=\sum_{i=0}^{n}P_iB_i^n(u),\quad C_2(v)=\sum_{j=0}^{m}Q_jB_j^m(v),$$

where $u,v\in[0,1]$ and $B_i^n(u)$ and $B_j^m(v)$ are the nth and mth degree Bézier basis functions. In this paper, we assume $C_1(u)$ and $C_2(v)$ are two space curves in $\mathbb{R}^3$. Thus, the control points P_i and Q_j are points in 3D space.