Surface point location by walking algorithm for haptic visualization of triangulated 3D models

Roman Soukal *, Václav Purchart, Ivana Kolingerová

University of West Bohemia, Faculty of Applied Sciences, Department of Computer Science and Engineering, Plzen, Czech Republic

ABSTRACT

Haptic devices are nowadays gaining popularity because of their increasing availability. These special input/output devices provide, unlike mouse or keyboard, a native 3D manipulation, especially a more precise control and a force interaction. With more accurate description of the model, haptics can achieve more realistic force feedback. Therefore, triangulated surface models are often used for an authentic interpretation of 3D models. A common task in haptic visualization using triangulated surface models is to find a triangle which is in the collision trajectory of the haptic probe. Since the render rate of the haptic visualization is relatively high (usually about 1 kHz), the task becomes highly non-trivial for complex mesh models, especially for the meshes which are changing over time. The paper presents a fast and novel location algorithm able to find the triangle which is close to the haptic probe and in the direction of the probe motion vector. The algorithm has negligible additional memory requirements, since it does not need additional searching data structures and uses only the information usually available for triangulated models. Therefore, the algorithm could handle even triangular meshes changing over time. Results show that the proposed algorithm is fast enough to be used in haptic visualization of complex-shaped models with hundreds of thousands of triangles.

© 2014 Elsevier Ltd. All rights reserved.

Introduction

In this paper we focus on the collision detection problem of the haptic device with the surface of 3D model which is defined by a triangular mesh. The goal is to find a triangle (if such a triangle exists) which is in the collision trajectory of the haptic probe to provide appropriate feedback to the user.

Haptic visualization is a tactile feedback method which provides a sense of touch to the user via a haptic device by applying forces, vibrations, or motions while visual perception is usually mediated by a display device. See an example of the haptic device Phantom Omni® used in our experiments in Fig. 1. The haptic visualization finds applications in a variety of areas including haptic surgery simulations [1–3], industry design-based manufacturing [4], or the virtual reality for blind computer users [5]. The majority of the haptic rendering techniques [6,7] require to detect collision (and intersection) of a haptic cursor with the visualized model. However, unlike graphics visualization where a sufficient render rate is about 25 Hz, the render rate required by the haptics is about 1000 Hz to provide an authentic feedback (as it is mentioned by Colgate and Brown [8] a human skin is sensitive to force change of a frequency higher than 500 Hz). Therefore, collisions need to be detected and computed as fast as possible.

Since a higher precision of model representation results in a more realistic perception, many haptic applications [2,6,9,10] use triangulated surface meshes for the representation of models. Existing approaches for the triangulated surface meshes usually deal with static scenes or local changes of the model, where the location methods with search data structures (especially a spatial subdivision techniques) provide adequate results [9]. As mentioned in [10], most of the existing algorithms address collision detection and intersection computation for small models which consist of a few thousands of polygons or they use some kind of down-sampled finite element model [2].

However, in the haptic visualization and interaction used for geometric modeling the model and its topology is often changing, which results in updates of the search data structures. These updates may not be trivial and if the changes in the triangular mesh are frequent, updates of data structures may significantly affect the performance. Moreover, search data structures consume additional memory.

Therefore, the goal is to develop an algorithm which does not utilize additional search data structures and still achieves the

Fig. 1. Example of haptic device – Phantom Omni®.

performance necessary for the application in the haptic visualization. When the haptic probe is moving on the surface of the model, the render rate required by the haptics is about 1000 Hz, thus the longest detection/location should take less than 1 ms. When the haptic probe is moving in a free space, the collisions may be detected in a lower rate, since a low delay of the first touch feedback is not noticeable. The frame rate about 100 Hz is sufficient, thus the longest detection/location should take less than 10 ms.

The problem is more precisely defined as follows. For a given position $\mathbf{q}$ of the haptic probe and its motion vector $\mathbf{m}$ (the current direction of movement of the haptic probe), the goal is to find and return such a triangle from the given surface triangular mesh, which is intersected by the line $\lambda = [\mathbf{q}, \mathbf{m}]$ in the distance from $\mathbf{q}$ lower than or equal to a defined maximal allowable distance $dist_{max}$. Moreover, $\mathbf{m}$ should direct towards the front face of the found target triangle. If such a triangle does not exist, *null* should be returned. Note that $dist_{max}$ should be small enough to ensure unique result. We suppose that the model does not contain errors or holes, all the triangles have uniform orientation and have information about their neighbors.

The proposed solution ranks among walking algorithms, which are popular especially for point location in planar triangular meshes or in tetrahedral meshes. The name of walking algorithms describes their principle: generally, the search goes from a triangle to its neighbor in the direction of the given query point, until the target triangle (which contains the query point) is found. In our case, the walking algorithm is searching in the direction of a triangle, which is intersected by the line defined by the position of the haptic probe and the haptic motion vector (see Fig. 2). Since the next triangle is chosen with respect to local tests, the utilization of a walking algorithm for a point location on the triangulated surface model is not simple and, to the best of our knowledge, no complex walking algorithm for point location on a triangulated surface model has been published.

The proposed algorithm has negligible additional memory requirements since it does not need additional data structures. It only needs the information about neighboring triangles, which is usually required for other purposes as well. Therefore, the algorithm can handle even triangular meshes which are changing over time. Although the algorithm was developed especially for haptic visualization, it is not limited to the haptic collision detection only. It can be used for all point location problems, where the input contains both: a point close to the surface of the triangulated 3D model and a vector directing towards the model. For example, for a parametric description of the model, we can get a point on the surface as well as a vector directing towards the model (it may be the opposite surface normal at this point).

Results show that the proposed algorithm can handle queries on rather complex-shaped models with hundreds of thousands of triangles in a good time and thus it can be successfully used in haptic visualization. The algorithm is suitable also for models changing in time. Although it is not a primary task of the algorithm, it can also handle queries when the model is composed of multiple components. Moreover, the algorithm is easily and effectively parallelizable which can significantly speed up the search process.

The paper is organized as follows. 'State of the art' provides an overview in the task of a collision detection regarding to haptic visualization. 'Proposed method' describes the proposed walking
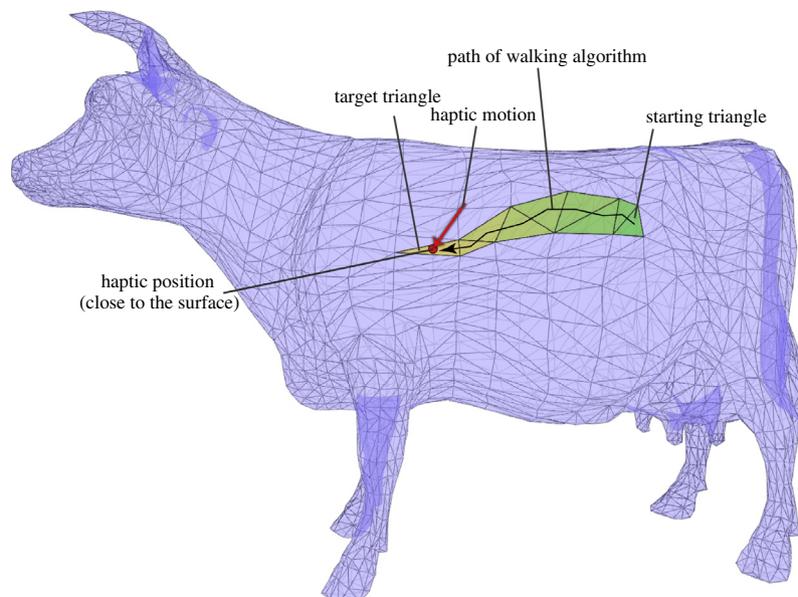

Fig. 2. Illustration of the collision detection on the surface of triangulated 3D model using the walking algorithm. The target triangle is intersected by a line, which is defined by the position of the haptic probe and the haptic motion vector. For a better depiction the model is partially transparent which makes overlapping parts of the mesh a little darker.