Massively Parallel Nearest Neighbor Queries for Dynamic Point Clouds on the GPU

Pedro Leite, João M. Teixeira, Thiago Farias, Veronica Teichrieb and Judith Kelner
Federal University of Pernambuco, Computer Science Center
Virtual Reality and Multimedia Research Group
Recife, Brazil
{pjsl, jmxnt, tsmcf, vt, jk}@cin.ufpe.br

Abstract—We introduce a parallel algorithm to solve approximate and exact nearest neighbor queries on the GPU, exploiting its massively parallel processing power. Both data structure construction and nearest neighbor queries are performed on the GPU, avoiding memory copies from system memory to device memory. This algorithm achieves real-time performance, enabling its usage in dynamic scenarios, by minimizing the sorting comparisons needed for a large \( K \) value. The underlying data structure for spatial subdivision handles 3D points and is based on grid spatial hashing. Users can specify the grid size interactively. Comparisons were done with other nearest neighbor algorithms implemented on both CPU and GPU. Our approach clearly surpasses CPU implementations regarding processing time, while it presents a competitive solution to GPU ones. Real-time results were obtained with ANN searches (\( K = 10 \)) for data sets up to 163K points and the potential of our algorithm is demonstrated through a point-based rendering application.

Keywords-nearest neighbor query; massive parallel programming; KNN; ANN;

I. INTRODUCTION

Nearest neighbor queries have their roots on the post-office problem, in which residences are assigned to their nearest post office [1]. Nowadays, a vast number of problems relies on nearest neighbor queries for a solution, including pattern classification [2], mobile information systems [3], implicit surfaces definition [4] [5], simplification of point-sampled surfaces [6], nearest photon queries in photon mapping [7], nearest neighbor search in point cloud modeling and particle-based fluid simulation [8] [9], normal estimation [10] and finite element modeling [11], among others. Furthermore, nearest neighbor queries are only efficient if the space (in which the subject data is within) is well subdivided.

Spatial subdivision is a well-known technique for improving performance used in a variety of applications. As an example, ray tracers can benefit from kd-trees, by using them as an acceleration structure for efficient triangle culling in ray-triangle intersection tests [12], for nearest neighbor search in photon mapping [7] or in point cloud modeling [8]. In addition, octrees have been used for smoke and water simulation [13] and appearance preserving [14], while collision detection of deformable objects can be implemented with spatial hashes [15] or representative triangles [16]. Finally, different types of spatial subdivision data structures ease the task of culling non visible objects from a scene [17].

In this paper, we present a massively parallel nearest neighbor query algorithm that achieves real-time performance with both KNN (K nearest neighbors) and ANN (approximate nearest neighbors) searches. The underlying data structure for spatial subdivision handles 3D points (but it can be easily adapted for supporting another number of dimensions) and is based on grid spatial hashing. Its construction shows real-time performance and the user can specify the grid size interactively. Furthermore, we show that our data structure and NN queries are suitable for real-time dynamic scenarios, i.e. each iteration of our solution takes up to 33 milliseconds.

In order to achieve real-time performance, both query and data structure construction were entirely implemented on the GPU taking advantage of its inherently parallel architecture and exploiting its memory model. Our main contributions are performance improvement of previous solutions with a massively parallel approach, exploiting recent graphics card’s processing power and delivering a real-time performance solution for existing applications. In addition, results obtained are independent of the input data set distribution, so our approach is not limited to a small set of applications. Finally, dynamic scenarios can benefit not only from our NN queries, but also from a fast grid-based data structure construction, as demonstrated on a Point-Based Rendering (PBR) application.

The remainder of the paper is organized as follows. In the next section we discuss some related work regarding KNN and ANN queries on CPU and GPU. Section III describes our algorithm in more details and presents the results obtained with our approach compared to previous works. Section IV shows some experimental results obtained with a PBR application, exposing a practical real-time usage for our data structure and ANN queries algorithm. Section V draws a conclusion and points some future works.

II. RELATED WORK

This section discusses KNN and ANN queries. The former is a generalization of the post-office problem, where the solution to the problem involves the K nearest post-offices instead of the nearest one only, while the later can tolerate minimum errors, i.e. there can be post-offices that are farther