Automation in Construction 42 (2014) 90–99

Contents lists available at ScienceDirect

Automation in Construction

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

High-speed visualization of time-varying data in large-scale structural dynamic analyses with a GPU

Zhen Xu [a], Xinzheng Lu [a,*], Hong Guan [b], Aizhu Ren [a]

[a] Key Laboratory of Civil Engineering Safety and Durability of China Education Ministry, Department of Civil Engineering, Tsinghua University, Beijing 100084, PR China
[b] Griffith School of Engineering, Griffith University Gold Coast Campus, Queensland 4222, Australia

ARTICLE INFO

Article history:
Received 6 September 2013
Received in revised form 16 January 2014
Accepted 22 February 2014
Available online 16 March 2014

Keywords:
High-speed visualization
Time-varying data
Large-scale structural dynamic analyses
Key frame extraction
GPU
Frame interpolation

ABSTRACT

Large-scale structural dynamic analyses generally produce massive amount of time-varying data. Inefficient rendering of these data seriously affects the quality of display of and user interaction with the analysis results. A high-speed visualization solution using a GPU (graphics processing unit) is thus developed in this study. Based on the clustering concept, a key frame extraction algorithm specific to the GPU-based rendering is proposed, which significantly reduces the data size to satisfy the GPU memory requirement. Using the key frames, a GPU-based parallel frame interpolation algorithm is also proposed to reconstruct the complete structural dynamic process. Particularly, a novel data access model considering the features of time-varying data and GPU memory is designed to improve the interpolation efficiency. Two case studies including an arch bridge and a high-rise building are presented, confirming the ability of the proposed solution to provide a high-speed and interactive visualization environment for large-scale structural dynamic analyses.

1. Introduction

An increasing number of large-scale structural dynamic analyses have been performed recently [1–5]. The resulting data of these analyses are time-varying in different time steps. In addition, these analyses often result in dozens of gigabytes (GB) of data, which are massive for 3D visualization [6]. In the post-processor of general structural dynamic analysis software (e.g., MSC.Marc, ANSYS, ABAQUS) [7–9], the rendering process for massive time-varying data is exceedingly slow, sometimes in excess of 1 h for large-scale analyses. To smoothly display the structural dynamic process, most structural analysis software transforms the results to pre-computed animations [7–9]. However, such animations cannot solve the fundamental rendering problems. Most of them merely display the dynamic process at a fixed viewing angle and position, and lack the necessary interactive operations. Consequently, high-speed rendering of such massive amount of time-varying data has become an important issue for large-scale structural dynamic analyses.

Although several methods for accelerating the rendering process of massive static data have been proposed [10,11], rendering time-varying data of large-scale structural dynamic analyses is a much more complicated task [12,13]. In the process of rendering, static data needs to be visited only once, whereas time-varying data requires a continuous visit in order to display a dynamic process. Thus, time-varying data must be stored in a GPU memory for a quick visit in the process of rendering. However, the GPU memory size is in general 1–2 GB, and at most 4 GB [14], therefore the massive amount of time-varying data cannot be completely stored in a GPU memory. In this case, these data need to be continuously transmitted from the host memory to the GPU memory during the rendering process. Such data transmission is relatively slow compared to the direct access in the GPU memory, and most rendering time elapses during this data transmission [15]. These problems are the primary cause for an inefficient rendering of massive time-varying data, which presents a big challenge for high-speed visualization of large-scale structural dynamic analyses.

To minimize the issue of slow data transmission and achieve high-speed rendering, solutions to two key technical problems are necessary to be obtained: Solution 1 — significantly reducing the size of time-varying data to satisfy the capacity limit of a GPU memory; and Solution 2 — efficiently reconstructing a complete structural dynamic process in the process of rendering.

To obtain Solution 1, the key frame extraction method is considered appropriate for reducing massive time-varying data in a structural dynamic analysis. The time steps in such an analysis are referred to as the frames in visualization. Several techniques in relation to key frame extractions are proven to be suitable for identifying the representative time steps in dynamic analyses [16–23]. The extracted time steps are generally a small fraction of the total time steps, thereby significantly

---

* Corresponding author.
E-mail addresses: xuzhen@tsinghua.edu.cn (Z. Xu), luxz@tsinghua.edu.cn (X. Lu), h.guan@griffith.edu.au (H. Guan), raz-dci@tsinghua.edu.cn (A. Ren).

http://dx.doi.org/10.1016/j.autcon.2014.02.020
0926-5805/

reducing the amount of massive time-varying data [20–23]. During the dynamic process, a structure may develop large non-linear deformations, leading to complicated 3D movements. Note that some existing extraction methods developed for 3D data are mainly targeted for motions of a limited number of predefined objects (e.g., some rigid-body motions) or points (e.g., motion capture data at some predefined points) [18,19]. As such they are unsuitable for simulating complicated 3D movements. On the other hand, the clustering method, as one of the widely used key frame extraction methods, has the advantage of handling complicated 3D movements [20–23], and is therefore well suited for structural dynamic analysis. In spite of this, the existing clustering method [20–23] was not originally designed for the GPU-based rendering and cannot be adapted to different GPU platforms, because the size of the extracted key frames may exceed the memory limitation of a GPU. In view of the above, a key frame extraction method specific to the GPU-based rendering is thus necessary to be developed.

To obtain Solution 2, a GPU-based frame interpolation is considered in this study. Note that 3D visualizations generated directly from the extracted key frames are not complete at all. Therefore, frame interpolation is necessary for reconstructing the entire dynamic process of the structures. Note also that due to the GPU hardware limitations in the past, early studies on GPU-based frame interpolations could not achieve satisfactory performance [24]. Since the 2006 release of GPUs with unified architecture, its computational performance and programming convenience have improved significantly [25]. Nevertheless, related studies [26–28] indicated that the access efficiency of data in a GPU memory has become an important bottleneck in the frame interpolation of time-varying data. This is because large amount of data must be continuously accessed from the GPU memory in the process of frame interpolation. It is worth noting that the GPU has a complicated memory system and any non-optimized access model may result in significant memory latency and unacceptable low efficiency. Therefore, a novel data access model is highly desirable for high-efficient GPU-based frame interpolation. Such a model should take a full consideration of the characteristics of time-varying data in structural dynamic analyses as well as the special features of the GPU memory system. Although high-speed 3D visualization in civil engineering and construction has been extensively studied [29–32], an efficient data access model has not yet been proposed.

A complete GPU-based solution for high-speed visualization of massive time-varying data in large-scale structural dynamic analyses is thus developed in this study. To address the issue of GPU memory limitations, a specialized key frame extraction algorithm based on the clustering concept is proposed that is adaptive to different GPU platforms and can significantly reduce the size of time-varying data. Using the key frames, a GPU-based parallel algorithm for frame interpolation is also proposed to reconstruct the complete structural dynamic process. Particularly, a novel data access model considering the features of time-varying data and GPU memory is designed to further improve the interpolation efficiency. Two case studies including a stone arch bridge and a high-rise building are investigated to demonstrate the advantages of the proposed solution.

2. Overall visualization framework

The overall framework of high-speed visualization of massive time-varying data resulted from a large-scale structural dynamic analysis is illustrated in Fig. 1. In this framework, data transmission from the host memory to the GPU memory is implemented once only before rendering. Hence, instead of slow data transmission, key frame extraction and parallel frame interpolation will dominate the rendering efficiency for time-varying data visualization.

Three platforms are used for this framework: (1) a graphics platform, (2) a software development platform and, (3) a hardware platform. An open-source graphics engine, OSG (OpenSceneGraph), is adopted as the graphics platform to implement some in-depth visualization developments [33]. The CUDA (Compute Unified Device Architecture) platform, being the most widely used in general GPU computing development, is adopted as the software development platform [34]. Accordingly, a video card supporting CUDA, e.g., Quadro FX 3800 (192 cores, 1 GB memory, widely used in desktop computers), is adopted as the hardware platform. The osgCompute library developed by the University of Siegen [35] is used to integrate OSG and CUDA. Using these platforms, the entire process of visualization can be fully controlled from software to hardware, which provides a convenient foundation for solving the visualization problems for massive time-varying data.

3. Clustering-based key frame extractions

The time-varying data of a structural dynamic analysis includes displacements, stresses, and velocities. Note that this study focuses on the nodal displacement data which are used as references for visualizing other types of data. The proposed extraction algorithm divides the entire process of movement into several sub-processes (i.e., clusters). Although sub-processes are quite different from each other, there is a significant similarity within each sub-process; thus a few key frames are adequate to represent the entire sub-process. The first, middle and last frames of each cluster are selected as the key frames, which correspond to the beginning, developing and finishing stages of the sub-processes, respectively.

The purpose of key frame extraction is to satisfy the constraints of a GPU memory, i.e., the total volume of key frames must be lower than that of a GPU memory. As mentioned above, a cluster can produce three key frames (two boundaries and one middle), however two adjacent clusters share the same boundary frame. Defining the number of clusters and that of key frames as N_c and k, respectively, we have $k = 2N_c + 1$. The number and volume of the total frames are defined as N_f and V_f, respectively. The variable V_v represents the volume of the GPU

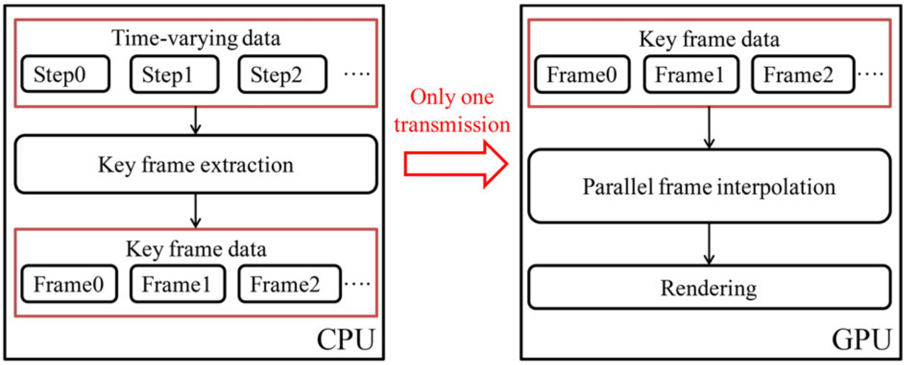

Fig. 1. Overall visualization framework for time-varying data in a large-scale structural dynamic analysis.