Journal of Network and Computer Applications 42 (2014) 1–11

Contents lists available at ScienceDirect

Journal of Network and Computer Applications

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

An efficient caching algorithm for peer-to-peer 3D streaming in distributed virtual environments ☆

Jinyuan Jia [a], Wei Wang [a], Xiaojun Hei [b,*]

[a] School of Software Engineering, Tongji University, Shanghai, China
[b] Department of Electronics and Information Engineering, Huazhong University of Science and Technology, 1037 Luoyu Road, Wuhan, Hubei 430074, China

ARTICLE INFO

Article history:
Received 24 July 2013
Received in revised form
14 February 2014
Accepted 8 March 2014
Available online 24 March 2014

Keywords:
Caching
Peer-to-peer
Distributed virtual environment
3D streaming

ABSTRACT

Recent technical progress on the Internet and virtual reality has enabled the proliferation of the applications of distributed virtual environments (DVEs). In a DVE, high-resolution 3D contents may generate huge data while peer-to-peer (P2P) streaming takes advantages to carry these huge traffic in a cost-effective manner. In this P2P paradigm, peers can cache and share DVE data cooperatively to reduce server workload and improve streaming quality. Nevertheless, it is critical to maintain and update the cached contents in each peer efficiently. In this paper, we propose an efficient caching algorithm for a P2P 3D content streaming framework. The proposed caching algorithm is based on a new preservation metric that is defined for balancing visual saliency, reusability and potential relevance of cached 3D objects. Then, these cached 3D objects in each peer are updated adaptively with the ascendant order in importance quantified using this new metric. We implemented the proposed caching algorithm in a simulated DVE platform for P2P-based 3D streaming. We conducted a comprehensive simulation study and our experimental results demonstrate that the proposed peer-to-peer streaming method outperforms the classic 3D streaming methods (including FLoD and MRM) in terms of fill ratio, base latency, requests by nodes and requests to the server.

© 2014 Elsevier Ltd. All rights reserved.

1. Introduction

With the advance of the Internet and virtual reality technologies, distributed virtual environments (DVEs) have become popular in recent years. By joining together in a virtual environment, users who are geographically dispersed are able to communicate and interact with each other on the Internet (Steed and Oliveira, 2009). For a high-resolution 3D visual environment system, the 3D content data are huge (e.g., more than 34 TB data in *Second Life* in 2008, http://secondlife.com, Liang et al., 2008 and 70.5 TB data in *Google Earth* in 2006, http://www.google.com/earth/index.html). Nevertheless, it is impossible to preload all these data into a client before a user starts to walkthrough in a virtual scene. Note that a user can only observe a small portion of the whole virtual scene due to the occlusions among 3D objects, a practical way is to download and render the data of the limited visible scene at the current viewpoint in a client. This strategy is particularly useful for thin clients, such as smart phones, pad and other mobile devices.

Various 3D content streaming techniques have been proposed to transfer DVE data, including client–server (Chim et al., 2003; Van Den Bossche et al., 2009; Ta et al., 2011) and peer-to-peer (Botev et al., 2008; Cavagna et al., 2009; Hu et al., 2006, 2008; Royan et al., 2007; Zhu et al., 2011) architectures. It is not unusual that millions of clients join a DVE simultaneously. If they are all connected to the server that stores the whole copy of the virtual environment data, the server is likely to be the bottleneck due to its limited bandwidth. On the other hand, in a large-scale DVE system virtual components are redundant for each user. Each user caches these virtual components and transfers them cooperatively in a naive P2P manner. When a user navigates in a virtual environment, her/his viewpoints exhibit strong time and temporal correlations. In a finer granularity, 3D virtual scenes can be transferred using P2P mechanisms for the clients having similar viewpoints in the virtual environment. These clients are clustered as neighbors and a client can send requests to its neighbors for data transmission of the shared visible scenes. By distributing and reusing 3D objects among clients, P2P DVE systems are advantageous in alleviating the burden of the server and enhancing quality-of-experience of users. On the other side, the virtual environment consistency and the network latency are two major performance issues in DVE applications (Delaney et al., 2006).

☆ Parts of the results in this paper have been presented in the international IFIP conference on networking (Networking 2013), Brooklyn, New York, May 22–24, 2013 (Wang et al., 2013).

* Corresponding author. Tel.: +86 2787544704.
E-mail addresses: jyjia@tongji.edu.cn (J. Jia), ww.cs.tj@gmail.com (W. Wang), heixj@hust.edu.cn (X. Hei).
URLS: http://sse.tongji.edu.cn/jiajinyuan (J. Jia), http://itec.hust.edu.cn/~heixj (X. Hei).

http://dx.doi.org/10.1016/j.jnca.2014.03.005
1084-8045/© 2014 Elsevier Ltd. All rights reserved.

The users' streaming experience in DVEs is crucial for a successful service deployment (Choukair and Retailleau, 2000). If users experience frequent freezes in the navigation or significant delays for startup after switching viewpoints, then the users may abandon the service. In the past decade, P2P video streaming has achieved much success in both academia and industry on the best-effort Internet (Hei et al., 2007, 2008). Those P2P video streaming techniques, such as caching and prefetching, which are suitable for linearly streamed contents, require further research and practice in 3D streaming because 3D streaming is highly interactive and nonlinear in nature with the streaming sequence based on each users' unique visibility or interest area in DVEs (Sung et al., 2008; Hu et al., 2010; Maamar et al., 2010; Chien et al., 2010; Zhao and Ooi, 2013). Recently, Zhao et al. proposed, Joserlin, a joint request and service scheduling scheme that aims to alleviate request contentions and also schedule the prefetch requests by considering their contributions to potential reduction of server load in Zhao and Ooi (2013). Our research focuses on the efficient caching for 3D streaming by jointly considering visual saliency, reusability and potential relevance of cached 3D objects.

Given the limited cache capacity of individual clients, only the visible portion of 3D scene can be loaded. When a user navigates a 3D virtual environment, certain objects must be removed and updated in the cache due to the changes of viewpoints. In this paper, we propose a progressive caching method for real-time navigation in large-scale P2P DVEs. This proposed caching method is based on three factors: in addition to the visual attention that is widely used in cache updating in client/server DVEs, we also propose two new factors of reusability and potential relevance. Some preliminary results on the progressive replacement for cached scenes based on potential relevance were presented to support the interactive walk-through in P2P-based large-scale WebVR worlds in Wang et al. (2010). We further conducted a comprehensive study on various factors on the caching schemes for P2P-based DVEs in Wang et al. (2013) and provided more extensive simulation results in this paper. Our experimental results show that a weighted combination of these three factors provides a satisfied performance measure in updating cached objects.

The rest of this paper is organized as follows. In Section 2, we summarized the related work on progressive transmission and caching methods for 3D streaming in DVEs. In Section 3, we proposed a P2P 3D content streaming framework. Within this framework, we also proposed an efficient caching algorithm based on a new preservation metric that is defined for balancing visual saliency, reusability and potential relevance of cached 3D objects in Section 4 and discussed the experimental results in Section 5. Finally, we presented the concluding remarks and future work in Section 6.

2. Related work

For 3D content streaming in a large-scale DVE, three key ingredients, area-of-interest (AOI) of 3D scene, progressive transmission and cache updating, are involved and their related work is summarized below.

2.1. AOI of complex 3D scenes

Given a limited cache capacity in a client, only a small portion of a complex 3D scene can be preloaded in DVE. Nevertheless, a virtual viewer can only see a small area of the whole virtual environment. The concept of the *area-of-interest* (AOI) has been proposed to characterize this application feature and been used widely in designing DVE systems (Chim et al., 2003; Li et al., 2011; Popescu and Codella, 2002). AOI is a circular area whose center is coincided with the current viewpoint and whose radius is proportional to the visible distance. All 3D objects enclosed in AOI can be reasonably thought as objects in the currently visible scene. When the viewpoint moves, the new added visible scenes have to be incrementally updated and downloaded from the server or other clients. An efficient scene culling algorithm is proposed in Jia et al. (2007), which partitions the entire scene using an axis-aligned mesh. Visible and invisible grids are dynamically maintained according to the temporal coherence between two consecutive AOIs. An improved scalable multi-layer AOI scheduling algorithm is proposed for the large-scale WebVR scenes in Wang and Jia (2009). Two scheduling mechanisms, area-based and cell-based scheduling, are studied in Pan et al. (2010), showing that a hybrid scheduling achieves the best performance.

2.2. Progressive transmission of 3D contents

Given a fixed viewpoint, AOI can greatly reduce the necessary 3D contents for browsing. The 3D objects contained in an AOI, however, could still be of huge data size if they are presented in a very high resolution. Level-of-details (LOD) techniques (Luebke et al., 2002) have been developed, which model a 3D object O by a base shape $\mathcal{B}$ and a set of hierarchical details $\mathcal{D}=\{D^1, D^2, \ldots\}$. Then the object $O=\mathcal{B}\cup\mathcal{D}$ can be represented hierarchically by $O^1\subset O^2\cdots\subset O$, where $O^i=\mathcal{B}\oplus D^1\oplus D^2\oplus\cdots\oplus D^i$, $\oplus$ is a synthesis operator. Usually $\mathcal{B}$ is an overall simplified shape that can be transformed very rapidly. When a user watches and manipulates (e.g., translates and rotates) 3D objects, more and more details in $\mathcal{D}$ can be received by the client and smoothly synthesized into the model $\mathcal{B}$. Notably, two classes exist for generating a LOD representation: surface simplification (Garland and Heckbert, 1997) and progressive mesh (Hoppe, 1996). In this paper, based on progressive representation (Hoppe, 1996) of 3D objects, we use the method in Chim et al. (2003) to compute an optimal resolution for each 3D object located in an AOI that is suitable for the current viewpoint. Given the object progressive representation, a general problem formulation of P2P transmission for 3D contents is conducted in Cheng et al. (2009) and Hu et al. (2010).

2.3. Caching methods of massive 3D contents

When the limited local cache in a client has been crammed by constantly downloaded scene (made up of 3D objects), some previously cached 3D objects have to be updated and replaced by new downloaded 3D objects (to constitute a new scene). Data replacement in the cache is an important issue that not only exists in 3D content streaming, but also in other well-studied applications as we summarize below.

2.3.1. Caching based on temporal factor

Page-based data replacement strategies, such as least recently used (LRU) and most recently used (MRU), have been widely used in database applications (Franklin et al., 1992). The merits of page-based replacement strategies rely heavily on the *principle of locality*, i.e., the higher degree of locality, the better performance can be achieved. However, it is shown in Si and Leong (1997) that these strategies are not suitable for 3D content replacements since 3D objects that are accessed by a client might change frequently over time. Video streaming is another popular application in which caching is widely studied. In P2P video streaming, video contents are regarded as one-dimensional (frames change in the dimension of discrete time) and thus can be accessed sequentially. A general caching strategy used in many P2P video streaming systems is that a receiver caches the recently played contents and supplies them to the requesters. However, P2P 3D content streaming is much more complex since the 3D objects are indexed in a 3D space and accessed according to viewer's interactive navigation behaviors.

Download English Version:

https://daneshyari.com/en/article/6885081

Download Persian Version:

https://daneshyari.com/article/6885081

Daneshyari.com