Adaptive Optimization Framework for Mobile Virtual Scenery Application

Qianyi Gu[1,2], Wudao Yang[2], Yongyan Yang[2]
[1]Visual Computing and Virtual Reality Key Laboratory of Sichuan Province
[2]College of Computer Science, Sichuan Normal University, Chengdu, China
qianyigu@foxmail.com

***Abstract*—With the performance improvement of mobile devices, researchers have brought virtual reality applications to mobile world and have created prospective applications including mobile tourist guide, popularized historical heritage education and offsite navigation. The major difficulty of such applications is the limit and diversity of computing capacities of the mobile devices. To address this issue, this paper proposes a mobile virtual scenery publishing platform optimizing the complex 3D data to adapt the computing capacities and network conditions. The optimization algorithm is developed based on a pervasive computing model built from the attributes of the mobile devices and a semantic model describing the virtual scenery world. The experiment data shows that the adaptive optimization can achieve a satisfying rendering speed on various levels of mobile devices while preserve the visual quality and vividness.**

Keywords-virtual reality; optimization; semantics; adaptation; mobile computing

I. Introduction

Equipped with powerful commercial developing tools, the virtual reality experts bring 3D visualization to end users with immersive and interactive experience. The high fidelity and vividness of such visualizations are achieved through huge number of geometry objects and appearance information. Due to the distinct user experience, the virtual reality technology has been successfully applied in many domains such as entertainment, education, advertisement and tourism. Among those applications, many instances of virtual scenery have been developed such as historical heritage, natural landscape, museums and architectures. These virtual scenes bring users the opportunities to have lively experience on the scenery without physical presenting to the site. Thus, virtual scenery applications have achieved great success across different domains.

With the rapidly growth of mobile technology, the performance of mobile devices such as intelligent phones and PDAs has allowed these devices to support complex virtual scenery applications. The appearance of high fidelity visualization of large 3D scenery on mobile devices would significantly benefit applications such as mobile tourist guides, popularized historical heritage education, mobile indoor navigation, offsite annotation of architectural designs, and offline evacuation training. With the self positioning functionality provided by the mobile devices, these highly interactive visualizations can be location-aware and providing information based on the users' physical position in the real world [1].

Even though mobile devices are equipped with powerful computing capacities, it is still a difficult task for them to render virtual scenery containing huge number of geometry data [2]. Most of the mobile devices are not able to handle virtual scenery at an acceptable frame rate which can be smoothly rendered at desktop systems. Compared to the desktop systems, the mobile devices are still characterized as: limited CPU computing facilities, limited memory space, absence of graphical accelerators, low resolution in the screen and unstable network connection [3].

Thus, many approaches have been reported rendering virtual reality data on mobile devices to address the limited capacities problem. One popular research direction is to develop the software architectures to carry out the entire virtual reality data rendering process on mobile devices. Chang and colleagues have proposed exploit techniques to obtain a lower complexity for the 3D rendering process without losing much quality in the rendered scene [4]. Kumar and other researchers have developed a mechanism called visibility culling [5]. The techniques can be used to selectively remove part of the virtual scenery before it is sent through the rendering process to minimize the number of polygons to be rendered. Another research direction to address the capacity issue is to describe the semantics of the virtual scenes to apply transformation before deployment [6]. Interaction Locus is an optimization mechanism proposed by Pitarello by defining a set of geometry elements contained in a real-world space [7]. Thus, the semantic relationship between real-world objects and geometry elements are identified in the interaction locus and adaptation process is applied according to the predefined semantics. While these approaches successfully decrease the complexity of the virtual scenery and preserve the visual quality at the same time, they deliver the same optimized result regardless the diversity of the mobile devices' computing capacities. The mobile industry has grown into the stage that the popular device models range from high-end ones with graphics accelerator to the mediocre ones with very limited memory and computing resources. This paper proposes a mobile virtual scenery publishing platform to adaptively optimize the virtual scenery data based on the mobile clients' computing capacities and network conditions. The optimized data can maximally preserve the visualization quality while achieve a satisfying rendering speed on the mobile devices.

The paper is organized as follows. Section 2 gives the overview of the research framework to adaptively optimize the virtual scenery for the mobile devices. Section 3 introduces a spring based system to detect mobile devices' capacities and construct a pervasive computing model based on these devices attributes. The model is used by the optimization algorithm in the following sections. Section 4 describes the construction of a semantic model to describe the virtual scenery. Section 5 describes in detail the algorithm we implemented to adaptively optimize large X3D models for mobile devices. Section 6 analyzes system performance.

II. RESEARCH FRAMEWORK

Figure 1 partially depicts the framework of the mobile virtual scenery publish platform which adaptively optimize the 3D data based on the client's computing capacities. The mobile devices which request the virtual scenery are monitored by the device detection module and devices' attributes are collected to form a pervasive computing model. The model is used by the 3D optimization algorithm to adaptively decrease the geometry complexity based on the devices' capacities. The semantic information of the virtual scenery is articulated by the domain experts to construct a semantic model to connect the real world information with geometry elements in the virtual scenery. The algorithm utilizes this semantic information to selectively apply optimization on parts of the geometry data in the virtual scenery to achieve a decreased complexity version. Finally, the optimized results best matched the clients' capacities are delivered to the requesting mobile device.

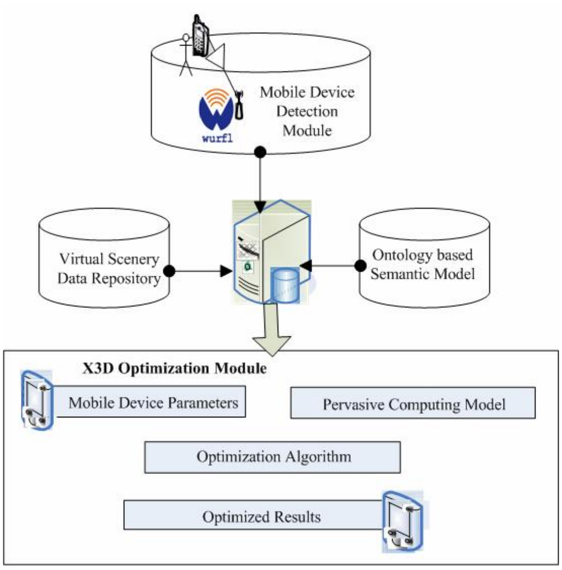

Figure1. Research Framework

III. MOBILE DEVICE DETECTION AND PEVASIVE MODEL

A. *Mobile device attribute collection*

In order to deliver the adapted virtual reality scenery that matched the mobile device computing capacities and network conditions, the system need to collect the associated attributes of the mobile devices when they send the requests for the virtual scenes. The examples of such attributes are operating system, screen resolution, CPU parameters and memory capacity. This paper proposes to use Wireless Universal Resource File (Wurfl) to make the collection.

Wurfl is an open source XML file which contains basic information and related data of mobile devices from all over the world. Wurfl is composed of Jar resources and spring based framework. The framework provides mobile device interaction components. Those components are used by our mobile device detection module to collect the clients' parameters to build our pervasive data model through the web services when the clients are making the network requests.

B. *Pervasive Computing Model of the Mobile Devices*

After the mobile device detection module interacting with various types of mobile devices, we have collected large number of mobile devices data. This paper proposes an approach to build a mathematical structure to represents statistics patterns of the mobile devices' attributes. Based on the patterns, a pervasive computing model is developed. The model is used in the following section as basis to optimized 3D data suited for the corresponding mobile devices. Mobile devices attribute data aggregate as:

$$E = \{\alpha, \beta, \gamma, \chi, \delta \in \Re\} \qquad (1)$$

(α : operating system; β : CPU parameters; γ : memory capacities; χ : screen resolution; δ : network bandwidth)

When a mobile device is requesting the virtual scenery, the device and the server forms a catenation component. Because the computing capacities of the mobile devices differ from each other, the server needs to deliver different optimized 3D data that match the client's capacity levels. The pervasive computing model is to provide a universal responding mechanism to satisfy this request. In our approach, each catenation is defined as a function

$$\omega = function(s, c)_{c \in E} \qquad (2)$$

c is defined as client component in mobile aggregate and s as server component. Then we have the definition of responding vector as responding vector which represents a set of mobile devices and corresponding servers' interactions:

$$V = [\omega_1 \quad \omega_2 \quad K \quad \omega_n] \qquad (3)$$

While the requests for the same virtual scenery could have multiple times, different mobile capacities and network conditions make the server respond with different 3D data. Thus, a dynamic vector σ for the same source data is formed:

$$\sigma_i = \omega_i - \sum_{k=1}^{i-1} \frac{(\omega_i \cdot \omega_k)}{(\omega_i^2 + \omega_k^2)} (1 \le i \le n) \qquad (4)$$

The pervasive model P is formed as a vector:

$$P = [\sigma_1 \quad \sigma_2 \quad L \quad \sigma_n] \qquad (5)$$

IV. SEMANTIC MODEL CONSTRUCTION

This paper presents a semantic model to describe the corresponding relationship between geometric primitives in X3D and real-world objects. These relationships and semantic information is basis for optimizing the virtual reality scenery for the mobile devices. The semantic model is defined as the virtual sceneries ontology which sustains the W3C standards. The ontology is formatted as X3D metadata and OWL (Web Ontology Language) and designed as scene-independent to be incorporated in different domain applications.

A. *Domain knowledge acquisition and analysis*

It is a complex task to establish a thorough domain ontology describing the heterogeneous virtual reality scenes. The quality of the ontology can vary heavily due to the different classification perspective and granularity selected by the ontology engineers. In order to guarantee the soundness and completeness of the ontology, an iterative process is carried out in our approach. In this process, the domain concepts are gathered as a group to identify the classes in the ontology. The real world objects in the domain are collected as the instances of the selected classes. The different types of relationships are articulated by the experts from the pairs of the classes to form a complete ontology. There are three types of semantic relationships in the model: conceptual relationships, spatial relationships and semantic annotations. The examples are shown in figure 2.

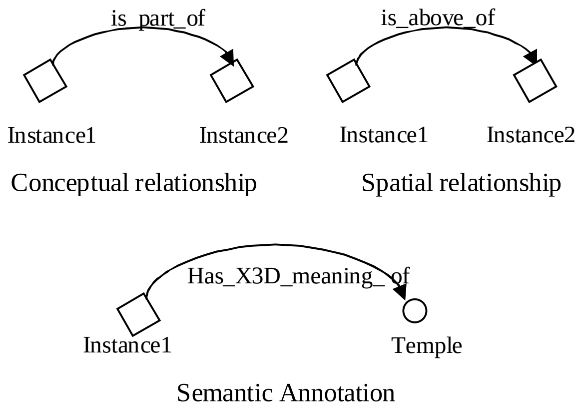

Figure2. Relationships in the Semantic Model

B. *Stabilized concepts identification*

The characteristics of the virtual scenery objects are quite different in the real world. Some of such objects are volatile, and vary from different scenes and applications. In order to articulate reliable and stabilized semantic information describing the virtual scenery in different domains, our research propose the ontology structure to map the volatile objects to the central set of general concepts suitable for all various domains and scenes as stabilized concepts.

C. *Hierarchical concept structure*

In order to precisely identify the relationships between concepts of the virtual scenery and to construct a systematic network of those concepts, this paper presents a systematic methodology to build a hierarchical concept structure as the virtual scenery ontology. A complete list are created cover all major real-world objects in the virtual scenery at the first stage. Then the domain experts are asked to tag each object with abstract meaning and granularity value. After the tagging process, the objects with highest granularity value are selected and mapped with domain meaning as current layer concepts. The process is iteratively carried on until a hierarchical concept structure is built up.

D. *Connect semantic model with virtual reality data*

X3D provides an effective mechanism to link semantic model with geometry data of the virtual scenes through a set of power nodes. Those nodes construct the inline of geometry information with semantic properties through switch functions and intelligent perceptions. With such connections, the semantic model represented as OWL ontology is integrated into X3D data of the virtual reality scenery.

V. X3D OPTIMAZATION

This paper presents an automatic inference engine to adaptively select the optimized X3D data to maximally preserve the geometry information and vividness of the virtual scenery while achieve good performance based on the mobile device computing resources and network bandwidth.

The inference engine use a three dimensional vector as adaptation model: $\eta = \{R, A, P\}$. In the vector, R stands for the responding time of the mobile device; A stands for the adaptation results in the candidate pool; P stands for the pervasive computing model derived from section 3. The inference engine uses a combination model to integrate the uncertainty factors into the three dimensional vector to execute the reasoning process. The combination model consequentially uses the responding time, adaptation results and pervasive model and thus it gets the equation as:

$$R_2r(C_2S) = \{R_2r\,\eta(C_2S)\} \qquad (6)$$

R stands for the request from the mobile device while r stands for the response from the server. C stands for the client and S stands for the server. The equation represents the time, result and model factors of the process when the request is sent from the mobile device for the virtual reality scenery and the optimized data is returned from the server. Three types of the optimization process are proposed in this paper. Which process is selected by the engine is depended on the disturbance factors in the environments and available computing resources. They are described in detail below:

Sequential process:

In the general situation when the disturbance factors are trivial, the sequential optimization process is taken. In this process, several adaptation procedures are consequentially conducted. The mathematic representation of the sequential process is as follows:

$$\{\mathrm{R2r}\,\eta(\mathrm{C2S})\} \Rightarrow \begin{cases} R2r_{\eta(R)}(C2S) = \sum_{i=1}^{n} \eta(R)_i \\ R2r_{\eta(A)}(C2S) = \sum_{i=1}^{n} \eta(A)_i \\ R2r_{\eta(P)}(C2S) = \sum_{i=1}^{n} \eta(P)_i \end{cases} \quad (7)$$

Iterative process:

When there are visible disturbance factors interfering with the adaptation results, the iterative process is taken. In this process, the system is trained iteratively until an optimized solution is identified:

$$\{\mathrm{R2r}\,\eta(\mathrm{C2S})\} \Rightarrow \begin{cases} R2r_{\eta(R)}(C2S) = \sum_{i=1}^{n} \eta(R)_i \\ R2r_{\eta(A,P)}(C2S) = \prod_{i=1}^{n} \eta(A,P)_i \\ R2r_{\eta(R,A,P)}(C2S) = \sum_{i=1}^{n} (1-\eta_i) \end{cases} \quad (8)$$

Iteratively concurrent process:

When the disturbance factors severely interfere with the adaptation procedure, it is unlikely to obtain a satisfying optimization result using the above two mechanisms. The iteratively concurrent process is taken in such cases. Every two procedures are paired with each other and are executed in parallel while every pair are consequently repeated as training data before the optimized solution is reached. The executions are disorderly and the satisfying results are obtained through large number of execution times. The mathematic representation of the sequential process is as follows:

$$\begin{cases} R2r_{\eta(R,A,P)}(C2S) = \sum_{i=1}^{n} V \times \eta(R,A)_i \\ R2r_{\eta 2(R,A,P)}(C2S) = \sum_{i=1}^{n} 1-\eta(R,A,P)_i \end{cases} \quad (9)$$

$$\min(R2r_{\eta(R,A,P)}(C2S), R2r_{\eta 2(R,A,P)}(C2S)) \quad (10)$$

VI. PERFORMANCE ANALYSIS

The performance of the adaptive optimization is tested on various mobile devices with different levels of computing capacities against a simple benchmark containing regular geometry nodes. The rendering speed is compared with the speed before optimization and the speed with unified optimization. The test results show in Figure 3 reveal that our algorithm can not only improve the rendering performance but also can achieve better improvements than unified approach on different types of mobile devices.

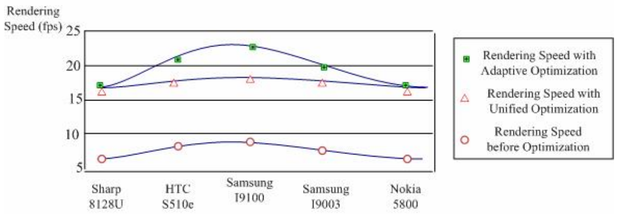

Figure3. Performance Results

VII. SUMMARY

To bring virtual scenery applications into the mobile devices with limited computing resources, this paper proposes a mobile virtual scenery publishing platform optimizing the complex 3D data to adapt the computing capacities and network conditions. The experiment results show that the adaptive optimization mechanism is able to achieve maximum retention of virtual scenery fidelity while ensure the satisfying rendering speed on the mobile devices. The major contribution includes: a mechanism to automatic detect mobile devices parameters through web services, a novel pervasive computing model, a systematic annotation mechanism to connect geometry elements with semantic features and the adaptive optimization algorithm based on semantics.

ACKNOWLEDGMENT

This paper is based on the work partially supported by Research Grant from Visual Computing and Virtual Reality Key Laboratory of Sichuan Province, China (Grant No. Y2010N01) and by the Scientific Research Foundation of the Education Department of Sichuan Province, China (Grant No. 09ZC080).

REFERENCES

[1] A. Mulloni, D. Nadalutti, L. Chittaro, “Interactive Walkthrough of Large 3D Models of Buildings on Mobile Devices,” Proceedings of Web3D Symposium, Perugia, Itlay, p. 17-25, 2007

[2] D. Nadalutti, L. Chittaro, F. Buttussi, “Rendering of X3D Content on Mobile Devices with OpenGL ES,” Proceedings of Web3D Symposium, Columbia, Maryland, p. 19-26, 2006

[3] D. Billsus, C.A. Brunk, C. Evans, B. Gladish and M. Pazzani, “Adaptive interfaces for ubiquitous web access,” Communications of the ACM, 45(5), p. 34 -38, 2002

[4] C.F. Chang, and S.H. Ger, ＂Enhancing 3D Graphics on Mobile Devices by Image-Based Rendering,＂ In Proceedings of the Third IEEE Pacific Rim Conference on Multimedia, Springer-Verlag, Berlin, Germany, 1105 -1111, 2002

[5] S. Kumar, D. Manocha, W. Garrett, and M. Lin, ＂Hierarchical back-face computation,＂ In Proceedings of the Euro-graphics Workshop on Rendering Techniques, Springer-Verlag, Berlin, Germany, 235 – 244,1996

[6] I.M. Bilasco, J.Gensel, M. Villanova-Oliver and H. Martin, “An MPEG-7 framework enhancing the reuse of 3D models,” in Proceedings of Web3D Symposium, Columbia, Maryland, p. 65-74, 2006

[7] F. Pitarello, A. De Faveri, “Description of 3D Environments: a Proposal Based on Web Standards,” Proceedings of Web3D Symposium, Columbia, Maryland, p. 85-95, 2006