ABSTRACT

In modern society, many companies offer product customization services to their customers. There are two major issues in providing customized products. First, product manufacturers need to effectively present their products to the customers who may be located in any geographical area. Second, customers need to be able to provide their feedbacks on the product in real-time. However, the traditional presentation approaches cannot effectively convey sufficient information for the product or efficiently adjust product design according to customers’ real-time feedbacks. In order to address these issues, we propose vPresent, a cloud based 3D virtual presentation environment, in this paper. In vPresent, the product expert can show the 3D virtual product to the remote customers and dynamically customize the product based on customers’ feedbacks, while customers can provide their opinions in real time when they are viewing a vivid 3D visualization of the product. Since the proposed vPresent is a cloud based system, the customers are able to access the customized virtual products from anywhere at any time, via desktop, laptop, or even smart phone. The proposed vPresent is expected to effectively deliver 3D visual information to customers and provide an interactive design platform for the development of customized products.

Keywords: Virtual reality, Cave Automatic Virtual Environment (CAVE), cloud computing, finger interactions, resource allocation

1. INTRODUCTION

With the increased popularity of virtual environment applications, Virtual Reality (VR) has received more and more attentions from industry and academia in the past few years. VR refers to computer-generated Three-Dimensional (3D) environments which allow users to perceive and interact with virtual objects. According to the level of user immersion, VR systems can be classified into non-immersive VR, semi-immersive VR, and fully immersive VR systems. As the name suggests, the non-immersive VR system, also called desktop VR, uses a conventional graphic workstation to display a 3D environment on a Two-Dimensional (2D) monitor. The non-immersive VR system is an economical solution. However, it cannot provide the realistic perception to users. The semi-immersive VR system supports the feeling of "looking at" the virtual environment, and allows users to be partially immersed into it. The fully immersive VR system supports the feeling of "being in" the virtual environment. The Cave Automatic Virtual Environment (CAVE) at Ryerson University, as shown in Fig. 1, is a fully immersive VR system, which provides users a wider field of view and a larger freedom of interactions. In the CAVE, four stereoscopic projectors are mounted to project virtual environments into a room-sized cube which consists of three walls and one floor. In order to achieve realistic interactions, an optical tracking system is used to track the user’s motion and location in real time. Images projected on screens are controlled by the user’s perspective parameters captured from the tracking system. With a pair of stereoscopic shutter glasses, the user inside the CAVE can perceive virtual objects as in the real world and manipulate them with interaction tools.

We plan to develop a series of CAVE-based applications and the whole project is named as vProject. The umbrella of vProject contains virtual design (vDesign), virtual health (vHealth), virtual work (vWork), virtual presentation (vPresent), and others. In vDesign, we proposed a CAVE-based virtual design environment, in which users can manipulate virtual furnitures, segment interested regions in an image, and compose multiple regions into one texture using finger interactions. With vDesign, a designer can easily implement his or her idea in a virtual environment and obtain a realistic perception. In vHealth, we developed a real-time health monitoring system with the bidirectional information exchange between the virtual world and the real world. The patient’s physiological parameters (e.g. electrocardiogram, heart rate, and so on) are transmitted to the CAVE, in which the doctor can visualize these parameters on the patient’s 3D Avatar and send alert emails to the patient when necessary. If the doctor is interested in examining the patient’s medical history, he or she can also visualize the patient’s medical record, like brain Magnetic Resonance Imaging (MRI) data. In vWork, we will develop...
a virtual office, in which the user can deal with regular office work in the CAVE, such as sending or receiving emails, and having an immersive teleconference with other colleagues. In this paper, we focus on vPresent, a cloud based 3D virtual presentation environment for interactive product customization.

In order to attract and retain customers, many companies provide product customization services to most valuable customers. For example, Rolls-Royce Limited, as a famous luxury car manufacturer, launched a “customize your own Phantom” service, through which potential buyers can customize the exterior paint and interior design of a Phantom car according to their preferences. In product customization, there are two major issues for manufacturers: 1) how to effectively present a 360-degree view of the product to the customers who are distributed around the world? 2) how to efficiently receive the feedbacks from the customers and accordingly adjust the product? The conventional approaches include text-based product presentation, image-based product presentation, and prototype-based product presentation. However, these conventional approaches are either inconvenient or too expensive. The text-based product presentation can provide the customers a clear and vivid picture. The image-based product presentation can provide the customers the pre-captured images. But the customers cannot freely inspect the product from any angle. With the prototype-based product presentation, the manufacturer makes a miniature replica of the product and sends it to the customers to get their feedbacks. However, the prototype-based product presentation introduces extra cost and delays the product development. Therefore, an efficient product presentation approach, which can fully present a product to the customers and receive the feedbacks in real time, is desired for manufacturers.

To provide a high-quality presentation experience, we propose vPresent, a cloud based 3D virtual presentation environment, in this paper. In vPresent, product experts are able to manipulate the 3D virtual product in the CAVE, present it to the remote customers, and dynamically customize the product according to real-time feedbacks from the customers; meanwhile, the customers can provide their opinions in real time when they are viewing a realistic presentation of the product from any angle. Since the proposed vPresent is a cloud based system, the data is stored and processed at the cloud data center. Therefore, the customers are able to access their customized virtual products from anywhere at any time. Fig. 2 illustrates the proposed vPresent. In Fig. 2(a), the product expert can manipulate a virtual car with finger interactions in the CAVE, while at the same time the customer can inspect the customized car through a smart phone, as shown in Fig. 2(b).

There are three key technical challenges in vPresent. The first key challenge is how to efficiently transmit a complex 3D virtual product model to the remote customers over bandwidth-limited networks. Generally, a 3D model contains a large amount of geometry data and plentiful texture information, which consume considerable bandwidth resources. In order to transmit a 3D model efficiently, we employ a scalable delivery scheme which adaptively transmits the 3D model in a progressive manner. Besides the delivery challenge, the second key challenge is how to efficiently allocate cloud resources for multiple sessions. In vPresent, the latency is a critical factor for the Quality of Experience (QoE). Therefore, manufacturers need to satisfy the service response time requirements for all customers. However, it is challenging to
optimally allocate cloud resources to provide satisfactory services at a minimal cost. In order to address this challenge, we propose a queueing model based resource allocation scheme, which can minimize resource cost while guaranteeing service response time satisfactions for all customers. Last but not least, it is a challenge to develop a natural user interaction interface for product experts in the CAVE. Currently, the conventional interaction tool is the wand, which is a 6 Degrees-Of-Freedom (DOF) interaction tool as shown in Fig. 3(a). However, the wand interaction is neither natural nor intuitive. It requires a learning process, such as memorizing button functions. Aiming for a natural Human Computer Interaction (HCI), we propose finger interactions, which is shown in Fig. 3(b). With finger interactions, the experts in vPresent can perform a variety of actions, such as menu navigation and object manipulations, in an intuitive way. We implemented vPresent prototype and conducted experiments to evaluate the performance. Experimental results demonstrated that the proposed vPresent can effectively deliver 3D visual information to the remote customers, optimally allocate cloud resources for providers, and provide natural interactions for product customization.

2. RELATED WORK

In the literature, the technique of VR has always been a hot research topic. Authors in Ref. 7 utilized VR in the heritage protection field. They developed a 3D modeling environment for cultural heritages using the image-based 3D modeling and laser scanning-based 3D modeling techniques. To present the cultural heritages to visitors at different locations, they set up a web-based 3D model system for the interactive remote presentation. Similarly, Zara et al.\(^8\) presented Czech experience for cultural heritage presentation in virtual environment. In Ref. 8, they introduced the Virtual Old Prague (VOP) project, which is a web application that allows visitors walk through a virtual old Prague city stored in a remote database. The data is progressively transferred to visitors. By checking visitors’ movements, new 3D models are dynamically requested from the server so that a high frame rate can be achieved. Li et al.\(^9\) presented a 3D digital home appliance customization system based on Cult3D API.\(^10\) The system is a web-based application, and users can view a 3D model through a web browser.
However, the work in Ref. 7–9 is desktop based virtual presentation. In contrast to Ref. 7–9, the proposed vPresent is based on the CAVE system, thus able to provide users a 3D immersive and realistic virtual experience.

Various human computer interaction tools have been proposed and tested for VR environment. Traditionally, the so called Flystick, a wand type remote control, is used with various buttons. Individual buttons and a combo of buttons correspond to certain actions such as menu selection, zoom in/out, etc. Abramyan et al.\textsuperscript{11} used two types of wands (Nintendo Wii controller and Nunchuk joystick) in angle viewing and manipulation control. Wand was also used in a virtual table tennis game as the hand tracker to mimic the racket in Li et al.’s work.\textsuperscript{12} Other researchers proposed various approaches in transferring command from 2D touch screens to the 3D CAVE. Kim et al. clicked button and drew arrows on the iPhone/iPod for touch screen to command the CAVE in menu selection and navigation.\textsuperscript{13} In data mining, Prachyabrued et al. developed an interface based on iPod touch technology, in order to achieve complicated and cluttered 3D data visualization and manipulation in the CAVE environment.\textsuperscript{14}

As a cloud based system, an efficient resource allocation scheme is also a major concern in the proposed vPresent. Nan et al.\textsuperscript{15} presented an optimal cloud resource allocation scheme for the priority service. They utilized a priority queueing system to model the service process in cloud data center and optimized resource allocation to minimize the resource cost. Chaisiri et al.\textsuperscript{16} presented an optimal Virtual Machine (VM) placement algorithm to minimize the VM cost based on the stochastic integer programming. The proposed algorithm considered future resource demand and price uncertainty so that it can make decisions based on the optimal solution obtained from the stochastic integer programming. Shi et al.\textsuperscript{17} proposed a cloud resource allocation approach to maximize the steady-state throughput of data center.

3. CLOUD BASED 3D VIRTUAL PRESENTATION ENVIRONMENT

3.1 Overview of the Proposed vPresent Architecture

The proposed vPresent architecture is illustrated in Fig. 4. The service process can be summarized as follows. The customers can launch a product customization service by sending requests to the cloud data center. The resource manager will schedule the request to a product server. If the current product server cannot meet the Quality of Service (QoS) requirements for the customers, the resource manager will allocate more resources. Once the service session is set up, 3D virtual models will be transmitted to the customers. In order to provide efficient transmissions over the bandwidth-limited networks, we employ a scalable delivery scheme in vPresent. Each 3D model is encoded into a base model and a series of vertex splits. With more vertex splits received, the customers can view the 3D model at a higher quality, at the expense of a higher bandwidth consumption. At the manufacturer side, the product expert in the CAVE is able to manipulate the virtual product and display it to the remote customers. If the customers are not satisfied with the current design, they can send their feedbacks to the expert, who will accordingly make changes in real time. To provide a natural interaction interface, we employ finger interactions to perform menu navigation and object manipulation in vPresent. Specifically, the expert wears a marker on each hand. Both markers will be tracked by the optical tracking system. Based on the real-time positions of the
markers, we can calculate the distance between the two markers or the distance between a marker and a virtual object. The distance can be then used to trigger different operations, including menu selection and object manipulations. Compared to the traditional wand interactions, the proposed finger interactions can provide more easier and natural operations in the CAVE.

In the following three subsections, we will address the three technical challenges: 1) scalable delivery of 3D models, 2) optimal resource allocation at the cloud, and 3) natural interactions in the CAVE, respectively.

3.2 Scalable Delivery of 3D Models

To efficiently transmit 3D models to the customers, we apply a scalable delivery scheme in vPresent. The scalable delivery of 3D models requires the progressive transmission and multi-resolution rendering of the data. Hoppe et al. proposed a progressive mesh representation to do a multi-resolution rendering of 3D meshes. The core operations in progressive mesh are edge collapse and vertex split. As shown in Fig. 5, the edge collapse operation combines two adjacent vertices $V_s$ and $V_t$ into one single vertex $V_s$. With the edge collapse, the vertex $V_t$ and two faces $\{V_s, V_t, V_l\}$ and $\{V_s, V_r, V_t\}$ can be removed. More importantly, the simplified 3D mesh by the edge collapse operations can be reversed by the vertex split operations. As illustrated in Fig. 5, the vertex split operation can add a new vertex $V_t$ near the vertex $V_s$, thus creating two new faces $\{V_s, V_t, V_l\}$ and $\{V_s, V_r, V_t\}$.

Based on the edge collapse and vertex split operations, a 3D model can be rendered progressively. Given an initial 3D model $M_{ini}$, we can apply a series of edge collapse operations to simplify the model by reducing the number of vertices and faces. This process can be denoted as $M_{ini} \xrightarrow{edge\ collapse} \cdots \xrightarrow{edge\ collapse} M_{base}$, where the finally simplified model $M_{base}$ is called the base model. Given the base model, the original 3D model can be reconstructed by a series of vertex split operations, denoted as $M_{base} \xrightarrow{vertex\ split} \cdots \xrightarrow{vertex\ split} M_{ini}$. Therefore, a 3D model can be represented by the base model and a series of vertex splits.

In vPresent, each 3D model is encoded into a base model and a series of vertex splits. During transmissions, a customer can render the 3D model at a coarse quality when only the base model is received, and incrementally enhance the rendering quality by adding more vertices and faces. The model is rendered progressively until it is in its full resolution or the bandwidth capacity is achieved.

3.3 Optimal Resource Allocation at Cloud

The remote virtual presentation service has a strict requirement on the service response time, which is defined as the duration from the time when a request arrives at the data center to the time when the request is completely served. In this subsection, we optimize cloud resource allocation to minimize the resource cost for product providers while satisfying service response time requirements for all customers.

We propose a queueing model to study the optimal resource allocation at the cloud. The proposed queueing model for cloud data center is shown in Fig. 6. The model consists of three concatenated queueing systems, which are the scheduling queue, the computation queue, and the transmission queue. The schedule server maintains the scheduling queue, receiving all requests and then distributing them to the computing servers. Each computing server has a computation queue and processes the scheduled requests with the allocated computation resource. The schedule server and the computing servers are connected with high-speed communication links. Therefore, the internal latency between the schedule server
and the computing servers is assumed to be negligible. The results from the computing servers will be transmitted by the transmission server. Due to the time-varying workload, the resources at the cloud need to be dynamically adjusted. Therefore, we divide the time into time slots. The cloud resources are dynamically allocated in every time slot $t$. Since every two consecutive requests may be sent from different customers, the inter-arrival time is a random variable, which can be modeled as an exponential random variable. According to the relationship between the Poisson and exponential distributions, the arrivals of packages therefore follow a Poisson Process. The average arrival rate at the time slot $t$ is denoted by $\lambda(t)$. Since no request is dropped during service, the number of results transmitted by the transmission server is equal to the number of received requests.

The allocated cloud resources include the resource at the schedule server, the computing servers, and the transmission server. At the schedule server, the resource is represented by the scheduling rate $S(t)$ in terms of the number of requests scheduled per second. Suppose that there are $N$ computing servers. The computation resource at computing server $i$ is represented by the processing rate $C_i(t)$ in terms of the number of instructions executed per second. At the transmission server, the bandwidth resource is represented by the transmission rate $B(t)$ in terms of the number of bits transmitted per second. We employ a linear function to model the relationship of the resource cost and the allocated resources. The total resource cost $\Psi_{tot}(t)$ at time slot $t$ can be formulated as

$$\Psi_{tot}(t) = \left( \alpha S(t) + \beta \sum_{i=1}^{N} C_i(t) + \gamma B(t) \right) \Gamma,$$

where $\Gamma$ is the time slot length, $S(t)$ is the allocated schedule resource, $C_i(t)$ is the allocated computation resource at the computing server $i$, $B(t)$ is the allocated bandwidth resource, $\alpha$, $\beta$, and $\gamma$ are the cost rates for scheduling, computation, and transmission, respectively. The linear cost model in Equation (1) has been justified by the numerical analysis in Ref. 21.

Based on the queueing model, we formulate and solve the resource cost minimization problem to minimize the resource cost for product providers. When requests arrive at the data center, all requests enter into the schedule queue first. The schedule queue can be modeled as an $M/M/1$ queueing system with a mean service rate $S(t)$. In order to maintain a stable queue, $\lambda(t) < S(t)$ is required. The response time of the schedule queue is given by $T_{sch}^{(t)} = \frac{1}{\lambda(t) S(t)}$. Since every computing server has the same service procedure, a weighted scheduling scheme is employed. A request is assigned to the computing server $i$ with a probability $p_i$, and we have $\sum_{i=1}^{N} p_i = 1$. According to the decomposition property of Poisson process, the average arrival rate at the computing server $i$ is $p_i \lambda(t)$. Suppose that the average task size is denoted by $F$ in terms of the number of instructions. According to Ref. 22, the execution time can be approximated as the exponential distribution with an average of $F/C_i(t)$. Therefore, the service at the computing server $i$ is modeled as an $M/M/1$ queueing system. To maintain a stable queue, the constraint $p_i \lambda(t) < C_i(t)/F$ should be satisfied. The response time at the computing
The coordinates $z_f$ fingers. At any time, we can get the 6 DOF tracking data in the format of $(x, y, z, \eta, \theta, \phi)$, for any marker on the hand. The coordinates $(x, y, z)$ represent the position of the marker in the 3D space, and the Euler angles $(\eta, \theta, \phi)$ represent the rotation of the marker around its local coordinate system. In vPresent, the trigger of an action is determined by the

$$T^{(t)}_{\text{com}} = \frac{F/C^{(t)}_{\text{com}}}{1 - p_i \lambda^{(t)}/C^{(t)}_{\text{com}}}.$$  

Thus, the mean response time in the computation phase is formulated by

$$T^{(t)}_{\text{com}} = \sum_{i=1}^{N} p_i T^{(t)}_{\text{com}} = \sum_{i=1}^{N} \frac{p_i F/C_{i}^{(t)}}{1 - p_i \lambda^{(t)}/C_{i}^{(t)}}.$$  

Since no request is dropped during service, the arrivals of results at the transmission queue is still $\lambda^{(t)}$. To keep the transmission server stable, the incoming results should be no more than the bandwidth capacity. Thus, $\lambda^{(t)} < B^{(t)}/D$ is required, where $D$ is the average result size. Therefore, the response time in the transmission phase is given by

$$T^{(t)}_{\text{tra}} = \frac{D/B^{(t)}}{1 - \lambda^{(t)}/B^{(t)}}.$$  

Based on the above analysis, the total service response time in cloud data center is the summation of response time in the three phases, which can be formulated as

$$T^{(t)}_{\text{tot}} = T^{(t)}_{\text{sch}} + T^{(t)}_{\text{com}} + T^{(t)}_{\text{tra}} = \frac{1}{S^{(t)}} + \frac{1}{\lambda^{(t)}/S^{(t)}} + \sum_{i=1}^{N} \frac{p_i F/C_{i}^{(t)}}{1 - p_i \lambda^{(t)}/C_{i}^{(t)}} + \frac{D/B^{(t)}}{1 - \lambda^{(t)}/B^{(t)}}.$$  

We optimize the resource allocation at cloud to provide satisfactory services at the minimal resource cost. The resource cost minimization problem can be stated as: to minimize the total resource cost at the cloud by jointly optimizing the allocated schedule resource, the computation resource, and the bandwidth resource, subject to the queueing stability constraint in each queueing system and the service response time constraint. Mathematically, the problem can be formulated as follows.

Minimize

$$\psi^{(t)}_{\text{tot}} = \left( \alpha S^{(t)} + \beta \sum_{i=1}^{N} C_{i}^{(t)} + \gamma B^{(t)} \right) \Gamma$$

subject to

$$\lambda^{(t)} < S^{(t)},$$

$$p_i \lambda^{(t)} < C_{i}^{(t)}/F, \forall i = 1, \ldots, N,$$

$$\lambda^{(t)} < B^{(t)}/D,$$

$$T^{(t)}_{\text{tot}} \leq \tau,$$

where $T^{(t)}_{\text{tot}}$ is given by Equation (2) and $\tau$ is the upper bound of the service response time for vPresent.

We employ the Lagrange multiplier method\(^2\) to solve the optimization problem (3), and get the optimal analytical solution as follows.

$$S^{(t)*} = \frac{\sqrt{\alpha} + \sqrt{\beta} \sum_{i=1}^{N} \sqrt{p_i} + \sqrt{\gamma} D}{\sqrt{\alpha \tau}} + \lambda^{(t)},$$

$$C_{i}^{(t)*} = \frac{\sqrt{\alpha} + \sqrt{\beta} \sum_{i=1}^{N} \sqrt{p_i} + \sqrt{\gamma} D}{\sqrt{\beta \tau}} + \frac{p_i \lambda^{(t)} F}{\sqrt{\tau}},$$

$$B^{(t)*} = \frac{\sqrt{\alpha} + \sqrt{\beta} \sum_{i=1}^{N} \sqrt{p_i} + \sqrt{\gamma} D}{\sqrt{\tau}} + \lambda^{(t)} D.$$  

### 3.4 Natural Interactions in CAVE

The interactions in vPresent system can be classified into two classes: \textit{menu navigation} and \textit{object manipulations}. In this subsection, we will present the natural interactions in vPresent system.

In vPresent, the main menu is activated by the pull-down action performed by the right finger. At current time $t$, the main menu is triggered if $z_t - z_p \geq \rho_t^{(h)}$, where $z_t$ and $z_p$ are the coordinates along the $z$-axis, defined as perpendicular to the floor, at the current time and the previous time, respectively, and $\rho_t^{(h)}$ is a distance threshold which is used to determine if the activation occurs. Once the main menu is triggered, it will appear in front of the user. The user can navigate the menu and choose a function by touching the corresponding menu item, similar to the touch operation in a tablet.

Object manipulations in vPresent are used to modify the design of the virtual product, and they are also performed via fingers. At any time, we can get the 6 DOF tracking data in the format of $(x, y, z, \eta, \theta, \phi)$, for any marker on the hand. The coordinates $(x, y, z)$ represent the position of the marker in the 3D space, and the Euler angles $(\eta, \theta, \phi)$ represent the rotation of the marker around its local coordinate system.
Figure 7. The performance of the progressive mesh compression for: (a) the cow model, and (b) the Ford Focus convertible model. In each figure, the upper level is the mesh representation; the middle level is the model rendered with texture; and the bottom level gives the number of vertices.

positions of the two markers and the position of the virtual object to be manipulated. We define three basic operations for model manipulations, which are moving, rotating, and scaling a 3D model. The moving operation is controlled by the midpoint between the two markers. In other words, the object is moved along a path which is parallel to the moving path of the midpoint. The rotating operation is determined by four factors: rotation plane, rotation axis, rotation direction, and rotation angle. The model manipulation is performed in a discrete-time manner. The rotation plane is determined by the two intersected lines: the line (denoted as $L_c$) passing through the two markers at the current time, and the line (denoted as $L_p$) passing through the two markers at the previous time. The rotation axis is the line perpendicular to the rotation plane and through intersection point. The rotation direction is the same to the rotation direction of the two markers. The rotation angle is the angle through which line $L_p$ is rotated to coincide with line $L_c$ around the rotation axis along the rotation direction. Scaling in vPresent is uniform scaling, which means that the object is enlarged or shrunk with the the same scaling factor in all directions. The scaling factor is defined as the ratio between the distance between the two markers at the current time and that at the previous time. By using finger interactions, the product expert inside the CAVE can naturally manipulate virtual product model and change the product design.

4. EXPERIMENTS

We implemented the prototype of vPresent in the CAVE at Ryerson University, Canada. The prototype is developed on Windows system in C++ language based on the libraries of VR Juggler24 and OpenSceneGraph.25 We conducted experiments to evaluate the performance of the proposed vPresent system.

Firstly, we evaluated the scalable delivery of 3D models in the proposed vPresent. The experiments are conducted on two 3D models: the cow provided in OpenSceneGraph data set and the Ford Focus convertible downloaded from Google 3D Warehouse.26 Fig. 7 shows the performance of the progressive mesh compression. From Fig. 7, we can see that the original 3D model can be gradually simplified by edge collapse operations and the number of vertices to be transmitted can
be greatly reduced. With such a scalable delivery scheme, the customers with a low bandwidth can receive a 3D model at a coarse quality, while the customers with a high bandwidth can receive the 3D model at an improved quality.

Next, we perform simulations to evaluate the proposed optimal resource allocation scheme at the cloud. In the simulation, we set the parameters based on Windows Azure, which provides on-demand computation, storage, and bandwidth resources as utilities through Microsoft data centers. We compare the performance between the proposed optimal allocation scheme, in which the cloud resources are allocated optimally by solving the resource cost minimization problem (3), and the equal allocation scheme, in which the resource cost for scheduling, computation, and database are allocated equally. The comparison of resource cost is shown in Fig. 8, from which we can see that the proposed optimal allocation scheme achieves a much lower resource cost than the equal allocation scheme under the same package arrival rate. In Fig. 8, the maximal difference of resource cost between the two schemes is $2583.5.

Lastly, we conducted user tests to compare the proposed finger interactions and the traditional wand interactions in vPresent. Specifically, we designed two user tasks: 1) menu navigation, and 2) object manipulations. An expert, who has rich experience in using wand and finger interactions, is invited to participate in the test. Each user task is performed 20 times with fingers and 20 times with wand. Fig. 9(a) shows the comparison of the average time on a 6-level menu navigation. From Fig. 9(a), we can find that the expert completes the menu navigation using the proposed finger interactions within 6.19 seconds, much faster than the wand interactions. Fig. 9(b) shows the distortion at different time when the expert manipulate a 3D model from its initial state to a given target state. The manipulation requires moving, rotating, and scaling operations. We use distortion to represent the deviation from the target state. As shown in Fig. 9(b), the object manipulation with finger interactions can achieve the satisfactory state, at which the distortion is less than 0.1, within 1.35 seconds, which is faster than the traditional wand interactions by 30.8%.

5. CONCLUSION

In this paper, we proposed vPresent, a cloud based 3D virtual presentation environment. In vPresent, the product expert can present the 3D virtual product to the remote customers and dynamically customize the product according to the feedbacks from the customers. The customers can acquire a vivid 3D visualization of the product and provide their opinions in real time. We addressed the key technical challenges in vPresent. First of all, we apply the scalable delivery scheme to progressively transmit 3D models to the customers. Each 3D model is encoded into a base model and a series of vertex splits. With more vertex splits received, a higher quality of the 3D model can be reconstructed. In addition, we propose a queueing model based resource allocation scheme, which can minimize resource cost while guaranteeing service response time requirements for all customers. Last but not least, we present natural finger interactions for product customization in vPresent. The finger interactions allow the expert to perform menu navigation and object manipulation in a more intuitive way in the CAVE. We implemented vPresent prototype and conducted experiments to evaluate the performance. Experimental results demonstrated that the proposed vPresent can progressively deliver 3D visual information to customers, allocate cloud resources at a minimal cost, and provide natural interactions for product customization.
Figure 9. Comparison results between the proposed finger interactions and the wand interactions: (a) comparison of the time for menu navigation, and (b) comparison of the time-distortion relationship for the object manipulations.

REFERENCES