Open Software Platform for Haptic Interaction

Jia Liu*, Donghui Xie
School of Information and Control,Nanjing University of Information Science & Technology, Nanjing, China
E-mail:liujia@nuist.edu.cn

***Abstract*—Based on an analysis of the features of current software for haptic interaction, an extensible software for highly flexible and reconfigurable systems called the open software platform for haptic interaction (OPHI) is presented. The OPHI serves a wide range of functions, including the control of hardware devices and the execution of complicated application programs. Based on general software architecture, standardized components with design patterns, frameworks, and interfaces are established to develop haptic interaction applications easily and efficiently. Additionally, the OPHI subjects all modules and components to proper evaluation and testing, thereby avoiding potential errors caused by incomplete coding. As a comprehensive platform for all stages of development, the OPHI serves a significant function in the standardization of software for haptic interaction.**

***Index Terms*—haptic, component, software platform**

I. INTRODUCTION

The wide application of virtual reality technologies and the remote operation of robots give rise to an urgent need for effective haptic interaction systems. In a virtual reality system, an operator can touch the virtual objects using a haptic device to feel the haptic information (such as flexibility, surface texture, and temperature). In the process of the remote operation of a robot, these systems should be capable of feeding the haptic information on the action object (such as muscles, tissues, and organs in remote surgery) back to the operators, such that the operator can accurately judge and identify the physical characteristics and classifications of the object[1–4].

Researchers have recently started using robotic software platforms[5], such as Evolutionary Robotics Software Package[6], Microsoft Robotics Developer Studio[7], Open Robot Control Software[8], Robot Technology Components[9], and Robot Operating System[10][11][12]. Haptic interaction has been extensively applied, thus increasing the demand for the structural design of relevant software systems. A reusable haptic interaction software platform with precise positioning, strong scalability, and efficient processing mechanism needs to be proposed. Existing software platforms applicable in haptic interaction are inadequate[13]; only CHAI 3D can be considered as a comparatively mature platform[14][15]. CHAI 3D is an open-source set of C++ libraries for computer haptics, visualization, and interactive real-time simulation. This platform supports extensions and a variety of commercialized haptic feedback devices with three and six degrees of freedom. One disadvantage of CHAI 3D is that API-based development is difficult. Another disadvantage is its inadequate support of haptic devices and other hardware resources, as well as algorithm logic extension.

Other interactive instrument software architectures are available. One is the component-oriented software architecture that can divide the virtual instrument into several components with comparatively independent functions. This architecture can also accomplish software assembly using component composition technology, thus preliminarily realizing design and process multiplexes. However, this architecture suffers from several disadvantages, including coarse component granularity, failure to consider interaction monitoring support, and dependence on COM component technology for extension[16]. Another type of architecture is the user interface–model system structure, which simplifies the model–view–controller mode, separates instrument operation from algorithm logic, and provides the foundation for software standardization. However, this kind of hierarchical architecture focuses on the standardization of user interfaces and fails to consider the standardization of software and hardware resources[17]. Another is the open architecture, which provides two kinds of extension points: equipment console and software control. The open architecture provides support for all hardware equipment and software functions but is dependent on specific development tools and frameworks and cannot design the extension mechanism corresponding to haptic interactive characteristics[18].

Based on the survey above, the following major problems clearly exist in haptic interactive software platforms: limited studies on haptic software standardization and improved system structure; dependence of existing architectures on specific components or tools, which limit the configurability and the scalability of the software to some extent; and the standardization and specialization issues faced by developers. When providing functionality without considering application domain requirements, the system design of the software can increase the cost of haptic interactive systems and present difficulties in the scalability of specific functions. Utilizing a customized

Manuscript received May 19, 2013; revised August 15, 2013; accepted September 30, 2013
Project number: National natural science foundation of China (No. 61203316)
*Corresponding author

doi:10.4304/jsw.9.5.1107-1112

infrastructure for each application domain would result in the loss of standardization and the integration of single solutions. These non-standardized implementations would hinder the application and promotion of haptic interactive technology[19][20][21].

In this study, haptic interaction software features and application requirements are first analyzed. An extensible architecture with haptic interaction software developed in the laboratory is then proposed as an example. The design, components, structure, and interface of the proposed software architecture can flexibly expand haptic interaction devices, data models of virtual scenes, and visual display interfaces. These features enable the increase or decrease and replacement of components at run time, thereby improving the scalability and configurability of the system and meeting the needs of various applications. The system is significantly valuable in the standardization of haptic interaction software.

II. Haptic Interactive Software

A. Requirements Analysis

A haptic interaction system comprises the operator, the haptic interaction device, and the computer simulation. In human–virtual environment interactions, the visual information of a virtual scene is observed through the eyes, whereas the interaction between the operators and the environment is facilitated by a haptic interaction device. The operator’s hand is usually mapped to a virtual hand. Operators control the interaction between the virtual hand and the virtual objects within the virtual environment through a haptic display device. Meanwhile, the haptic information about objects in the virtual environment is fed back to operators through haptic interactive devices, thus achieving visual-haptic harmony.

The system software serves several functions: to set up models of various virtual objects and virtual hands, describe the geometric and physical properties of virtual objects, compute the deformation arising from the process of system interaction operation, display real-time three-dimensional graphic scenes through a graphic display system, and synchronize the entire system with the haptic display. Ultimately, the operators are provided realistic operating experiences.

B. Requirements Analysis for Software Platform

Device control, interactive information collection, virtual environment, physical simulation, virtual haptic computing, simulation display, and other software functions can be grouped into three kinds of typical demands: haptic perception, scenario display, and human–computer interaction.

Haptic perception involves command sequence detection and interactive response, as well as the control of haptic interaction devices. Given the variety of software and hardware technologies and tools used in haptic interaction systems, haptic perception as a demand is characterized by the improvement of system scalability to satisfy different hardware and software interfaces. Scenario display involves the computation of virtual haptic simulation and the rendering and refreshing of virtual scenes. This kind of demand is characterized in response to internal commands and the completion of system output to provide end users with a friendly interface and flowing perception effect. Human–computer interaction is the most complex requirement. Such interaction can serve as an input to management systems and internal interactions, including timeliness check of interactive events, command sequence sending and persistence mechanism, collision detection and deformation processing of virtual objects, as well as treatment frequency monitoring and adjustment. This kind of demand is characterized in the abstraction of processing logic algorithm, as well as in the control and optimization of data flow to ensure the highly efficient and stable operation of the whole system.

To achieve the extensible and configurable software design goals, the architecture should not be designed to cover various scaled systems addressing all needs. Instead, the architecture should assume the role of software platform between the application layer and the haptic display hardware device with sound interface. This design will enable developers to extend the system functions and consequently address the aforementioned requirements. Table 1 presents the major requirements that must be satisfied in system design. The table also shows the corresponding extension mechanisms and the features that can be incorporated into the design.

TABLE 1.
SYSTEM REQURIMENTS

Requirements	Extension mechanism	Extension characteristics
Haptic perception	Standardization interface of haptic device Definition of standardized event	Supportive of multiple haptic devices Realization of the recording and replaying of interactive operation Realization of multiple display interfaces for one event
Virtual scenes	Standardized data structure Generics-based sequence	Supportive of models set up by 3DMAX and other tools Capacity to save dynamic changes in the physical simulation algorithm in a variety of files or databases.
Human–machine interaction	Configurable soft panel Interaction/events transformation and aging test	Configuration of the function model of software Adjustable in terms of data acquisition, interface refreshing frequency, and frequency adaptiveness

III. Design Of Haptic Interaction Software Platform

A. Architectural Design

To design an extensible, configurable, and standardized haptic interaction software architecture, the following factors must be considered: depiction of the processing mechanism of interaction; provision of an infrastructure that enables the dynamic increase and replacement of interactive processing functionality (functional requirements); and introduction of the processing of nonfunctional aspects, such as aging tests and frequency adjustments. Based on the above requirements, this study proposes a software platform called the open software platform for haptic interaction (OPHI). The OPHI offers scalability of software components, frameworks, and scenarios (Fig. 1).

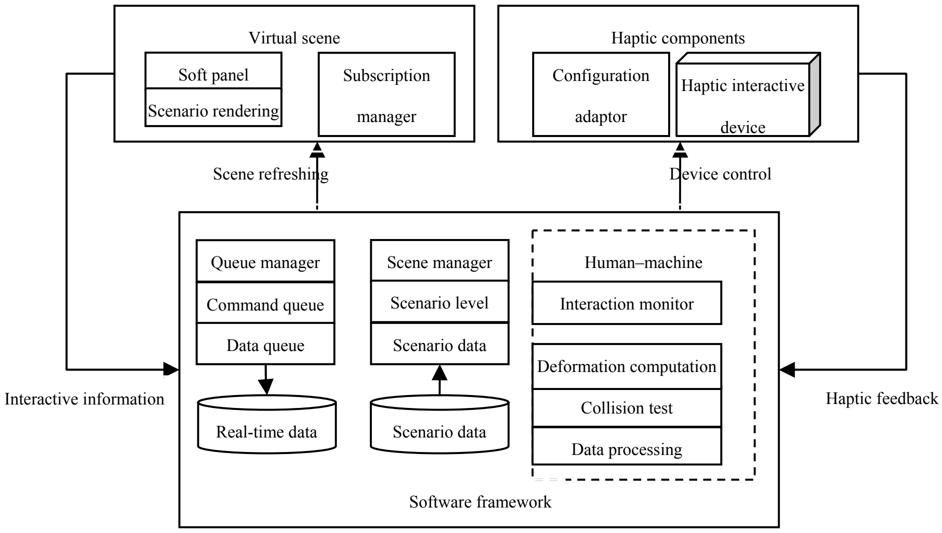


Figure 1. Overall structure of the OPHI

The overall structure of the OPHI can be divided into three parts: software framework, haptic components, and virtual scenes.

The software framework is functional in data management and maintenance. Data on the haptic interaction system can be divided into real-time and virtual scene data. Real-time data, which are managed by multiple circular queues, are data generated during system operation and interaction. Virtual scene data are the model data required in building a virtual environment, which can be divided into object, surface, vertex levels. These levels are managed and loaded as per configuration. Another important function of the software framework is to realize human–machine interaction with the aim of tracking and processing user operation events based on finite state machines (FSMs) using the corresponding interface of the invocation model. The result is an updated current scenario based on simulation results, thereby automatically realizing real-time rendering and haptic feedback of virtual scenes.

Haptic components correspond to haptic perception demand, sending haptic feedback information to software framework, and realizing extension of adaptors to haptic hardware devices.

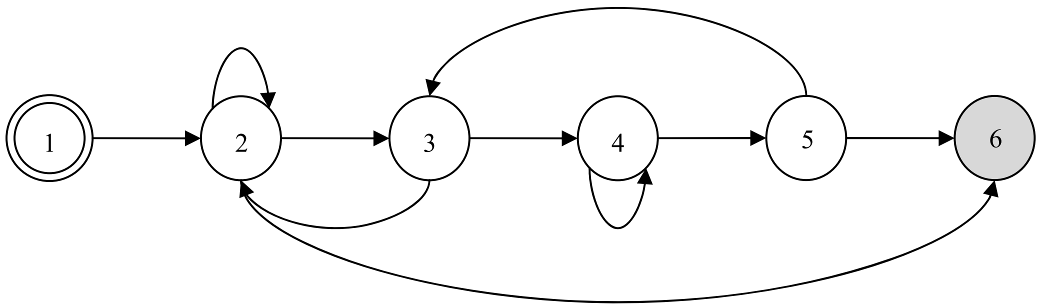
Virtual scene is applied in visual ways, that is, in displaying the state of virtual objects to users and realizing visual/haptic fusion, as well as separating operation/user panels through subscription mechanism in views to facilitate the modification and reuse of interface components.

Based on the software platform described above, the physical device, scenes, events, news, and strategy description of the whole system are configurable. The transformation of fixed reference ways between objects into flexible connection ways reduces the interdependence among objects, thus improving overall scalability and reusability.

B. Working Process of Haptic Components

By determining the current work state of the system and by following the formulated rules, haptic components can judge the action ways of control signals and send them out to haptic devices (hand control, step motor, etc.), such that these devices perform actions correspondingly and achieve the integrated control of haptic interaction.

The running processes of haptic interaction software can be divided into different states. Using an FSM can clearly express the process of haptic components while reducing the difficulty in program implementation. The haptic interaction software described in this article can be divided into device adjustment, experiment data collection, virtual scene adjustment, and virtual hand position adjustment. Different inputs under these states, such as data acquisition of haptic interaction device and mouse operation on the user interface, adopt the queue cache. The corresponding process is performed based on the current state and configuration. This method can improve processing efficiency while enhancing scalability. Fig.2 shows the working process.



1: Initiation; 2: Gear adjustment; 3: Experimental data collection; 4: Virtual scene; 5: Virtual hand position adjustment; 6: End
Figure 2. Working process

An FSM can also coordinate the working frequency. A square relation exists between the calculations of the overhead and those of deformation because the deformation process of virtual objects only considers the affected points. As the deflection intensifies, the calculation of comparatively fine virtual objects lags; adaptive adjustment must thus be performed on the work frequency according to the requirements. The object deformation process within the virtual scene adjustment (State 4) is divided into four states: separation, collision detection, deformation, and separation test. The separation state shifts to the collision detection state as the virtual hand approaches the virtual object. To ensure collision detection amid rapid movement, a high frequency is adopted for the collision test. The deformation state follows the collision. As deformation is an accumulative process, the processing frequency can be decreased appropriately. Meanwhile, the current deformation algorithm can be loaded and changed. As deformation decreases and the virtual object nearly returns to its original shape, the separation test state follows. In this process, a high frequency is also needed.

Another advantage of the FSM is the simplification of data processing. In the different states of the haptic interaction process, the procedures for data input and data processing differ. These procedures include data acquisition of haptic interaction device (State 3), user interaction on the virtual scene (State 4), and so on. All the data are cached in the queue, the corresponding processing is adopted based on the current status and configuration. This process enhances expansibility while improving processing efficiency.

C. Extension Mechanism of Virtual Scenes

In virtual scenes, the publisher-subscriber model is adopted to define the one-to-many relationship between the data model and the scene. After loading the virtual scene, the corresponding subscription interface is registered for each virtual object in the model based on the configuration. When the data object of the model is changed, all the objects are notified and undergo automatic refreshing. As shown in Fig.3, the data subscription object can be the interface view of a software or hardware device.

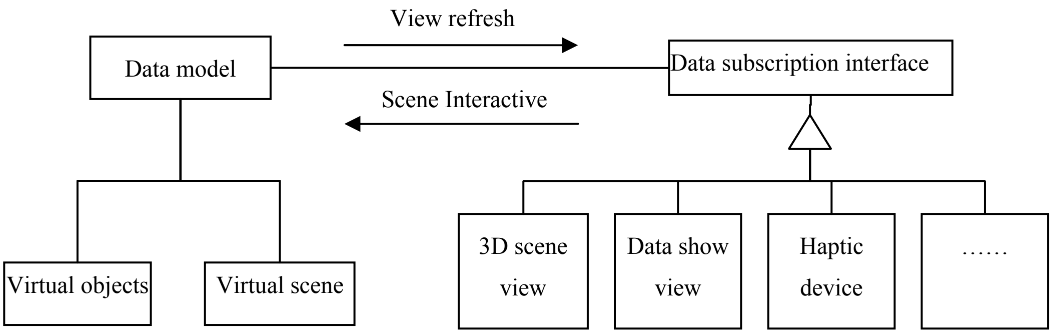


Figure 3. Virtual scene based on publisher-subscriber model

Such extension mechanism solves three major problems. First is the openness of interaction. The movement of a device and the operation of a keyboard/mouse correspond to the position adjustment of virtual hands. That is, multiple operation ways are provided for the actions of identical virtual objects. Second is the decoupling of background data and front desk display. When the virtual hands touch the virtual objects, the effective length of the elastic beam in a haptic reproduction device is calculated based on the stiffness at the surface contact point of virtual objects. The deformation of the virtual object surface is calculated according to the displacement value generated by the finger pressure of the operator on the displacement sensor. Both the haptic device view and the virtual scene view subscribe to these data and are notified via a message upon data updating to control the action of haptic components and to refresh the graphical interface and ultimately achieve haptic-visual coordination. Third is the fusion between video and images. As the publisher-subscriber is based on the configuration, softness settings for newly added virtual objects, virtual environment adjustment, and other operations can conveniently render virtual objects in videos, images, and other backgrounds only by changing the configuration.

D. Data Management in the Software Framework

The data in a haptic interaction system can be divided into two categories: virtual scene and real-time data. The quantity of virtual scene data is relatively fixed. Data are loaded upon system initialization. By contrast, real-time data include sampling data and equipment control instructions. The quantity of this type of data increases as the system operates. Virtual scene and real-time data are preserved by adopting a hierarchical structure and a circular queue structure, respectively. Extension of models involves the extension of data types and data operations.

The extension of data types is mainly based on the realization of virtual scenes, including multiple levels, such as scenes, models, objects, and planes. Each level defines a batch of data types that are used to preserve attribute information, including the structural information of scenes (handle of parent node or child node), model description information (bounding box and coordinates), and feature attribute information (transformation matrix and material). Both data types and data sources are described through plug-in configuration files so that the hierarchical structure model can organize various elements based on a unified management maintenance method. In case of the addition of new virtual objects, the model only needs to add a file path, object types, corresponding menu, and other description information in the configuration files to modify virtual scenes flexibly and amend the goals of the user interface automatically.

The operation of data can be categorized in terms of data type, access, and processing logic. The extension of operation can realize the dynamic loading of data and change the processing of data, including the modification of the shape of the bounding box, the adjustment of algorithm precision, and the extraction of sampling data. Considering the definition of a unified operation interface, the traversal methods of data must be reloaded for operation extension, and the state must be provided in the processing cycle to ensure that the interaction behavior of virtual scenes obtains extensions.

IV. Realization Of OPHI-Based Haptic Interaction System

In the laboratory, a set of small haptic interaction devices is developed, and a haptic interaction system is built. The laboratory output can assist operators in identifying the interaction information both visually and tactually (Fig. 4). The haptic interaction device collects the motions of the operator. In turn, the operator can directly touch the objects in the virtual environment with his fingers and recognize the stiffness on the surface.

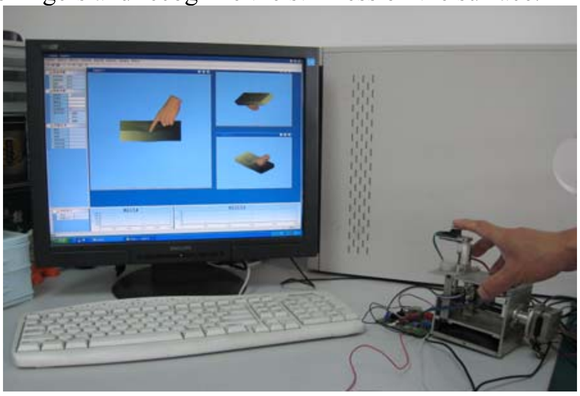


Figure 4. Haptic interaction system based on small haptic interaction device

In the case of realization without extensive plug-in, the system software is equipped only with a basic reference setting function, and no virtual object can be found in the scene. After realization of extension plug-ins, such as virtual objects and deformation algorithm, and after the completion of deployment as per directory specification, the system automatically loads the plug-in and the assembly interface (Fig.5). The software runs stably before and after extension.

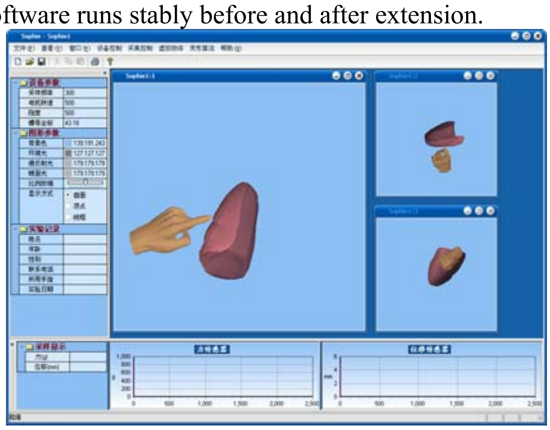


Figure 5. Software interface after realization of extension

The adoption of the OPHI corresponds to the following major advantages:

(1)The reusability of the software is enhanced. Systems with varied functions can be constructed only by modifying configuration files.

(2)Alignment with the modular development approach is achieved, with each module compiled independently. When function extension is necessary, adding or modifying a plug-in will suffice, thus eliminating the need to recompile the whole software.

(3)A third advantage is the coverage of internal details. Interactions are conducted only through interfaces. Dynamically adding or reducing and replacing modules are allowed during run time. Deployment and upgrading are facilitated. Multiple haptic devices and virtual scenes are supported.

(4)Abnormal processing is workable on the infrastructure in a unified way, thus enhancing the reliability of the software.

V. Conclusions

Considering the factors of flexibility, timeliness, reliability, easy maintenance, and scalability, we propose the OPHI and provide examples that can be used to develop flexible and configurable systems. The advantages of the platform software design are as follows:

- Real-time realization of the "plug-and-play" feature of the hardware and software components of the haptic interaction system; support for a variety of haptic devices; and promotion of the standardization of the haptic interaction system interface.
- Maintenance of the long-term stability of the platform framework; support for labor division in the module development; and improvement in the development efficiency and reliability of the system.
- Facilitation of the extension of the function of the haptic interaction system to realize conveniently the application system of the network.

The presented software platform is applied in a haptic interaction system developed in the laboratory. The platform is to be substantiated in the process of practical application and improved constantly with the development of human–machine interaction technology.

Acknowledgements

The paper was supported by the National Natural Science Foundation of China (No. 61203316).

References

[1] Aiguo Song, Dan Morris and J. Edward Colgate, "Real Time Stiffness Display Interface Device for Perception of Virtual Soft Object," *Proc IEEE/RSJ Inter Conference on Intelligent Robots and System,* Edmonton, Canada, pp.139 – 143, August 2005.

[2] Greg R. Luecke, "Haptic Interactions Using Virtual Manipulator Coupling With Applications to Underactuated Systems," *IEEE Transactions on Robotics*, vol.27,pp.730-739, August 2011.

[3] Taylor R H, Stoianovici D, "Medical Robot in Computer-integrated Surgery," *IEEE Trans. on Robotics and Automation*, vol.19,pp.765-781, October 2003.

[4] Bicchi Antonio, Scilingo E P, Rossi D D,"Haptic Discrimination of Softness in Teleoperation: The Role of the Contact Area Spread Rate," *IEEE Trans. on Robotics and Automation*, vol.16, pp.496-504. October 2000.

[5] Soohee han,Mi-sook Kim,Hong Seong Park, "Open Software Platform for Robotic Services," *IEEE Transaction on Automation Science and Engineering*, vol.9, pp.467-481,May 2012.

[6] Evolutionary Robotics Software Package (ERSP), 2013. [Online].Available: http://www.evolution.com

[7] Microsoft Robotics Developer Studio (MSRDS), 2013. [Online].Available: http://msdn.microsoft.com

[8] Open Robot Control Software (OROCOS), 2013. [Online]. Available:http://www.orocos.org

[9] Robot Technology Component (RTC), 2013. [Online]. Available:http://www.is.aist.go.jp/rt/OpenRTM-aist/

[10] Robot Open Source (ROS), 2013. [Online]. Available: http://www.ros.org

[11] S. Cousins,"Exponential growth of ROS," *IEEE Transactions on Robotics and Automation*, vol.18, pp. 19–20,April 2011.

[12] S. Cousins.B. Gerkey, K. Conley, W. Garage,"Sharing software with ROS," *IEEE Transactions on Robotics and Automatio*n, vol.17, pp. 12–14, June 2010.

[13] Atif Alamri, Mohamad Eid and Abdulmotaleb El Saddik, "Towards a Standard Modeling of Haptic Software System," *Proc of IEEE International Workshop on Haptic Audio Visual Environments and their Application*s, Ottawa, Canada, pp. 84-88, November 2006.

[14] Melo J S S,Borges G A and Brasil L M," Interaction haptic multiplatform framework," Pan American Health Care Exchanges, Rio de Janeiro, Brazil, pp.328-329,March 2011.

[15] Quan-Zen Ang,Horan B and Najdovski Z," Grasping Virtual Objects with Multi-point Haptics," Proc of IEEE Virtual Reality Conference, Singapore, pp.189-190, March 2011.

[16] Liangliang Chen,Xiangyan Kong and Tiantian Chen "Developmet Method of Virtual Instrument Based on Architecture and Component Oriented Approach," *Chinese Journal of Scientific Instrument*,vol.28, pp.263-266,April 2007.

[17] Wenbo Niu, Jianqing Huo and Xiaopu Wang,"Virtual Instruments Based on UI-Model Architecture," *Journal of Univerisity of Science and Technology of China*, vol.37,pp.323-326,March 2007.

[18] Jiangbin Zhao, Tielin Shi and Wuxing Lai, "IsVIP:an In-situ Virtual Instrumentation Platform," *Chinese Journal of Scientific Instrument*, vol.28, pp.634-639, April 2007.

[19] Min Xiong,Isabelle Milleville-pennel and Cédric Dumas, "Comparing Haptic and Visual Training Method of Learning Chinese Handwriting with a Haptic Guidance," *Journal of Computers*, vol.8, pp.1815-1819,July 2013.

[20] Yan Zhao,Dajun Zhao and Siyi Wang,"A Metadata-driven Cloud Computing Application Virtualization Model," *Journal of Computers*, vol.8,pp.1571-1579,June 2013.

[21] Hong Men,Ping Zhang and Caiwa Zhang, "Study and Implementation of Spacecraft Integration Test Platform Based on Component Technology," *Journal of Computers*,vol.6,pp.963-968,May 2011.

Jia Liu, born in 1981, is currently an associate professor at Nanjing University of Information Science & Technology. Her research interests cover haptics, virtual reality in general.