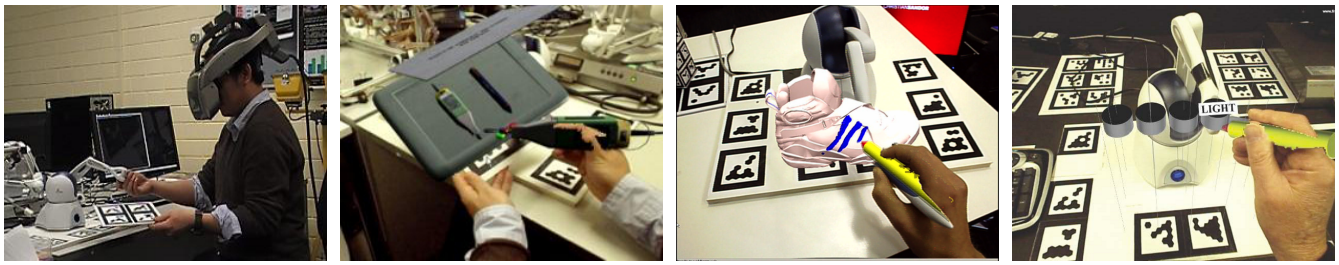
HARP: A Framework for Visuo-Haptic Augmented Reality

Ulrich Eck* Christian Sandor†

Magic Vision Lab
University of South Australia



(a) (b) (c) (d)

Figure 1: Visuo-haptic augmented reality allows users to see and touch virtual objects, which are embedded in the real world. (a) shows our setup with a head-worn display and a haptic device; screenshots from selected applications using our framework: (b) virtual tools are overlaid onto the phantom [9], (c) rapid prototyping demo, which enables users to paint on arbitrary objects [4], and (d) verification of psychophysical phenomena like the Stroop effect in VHAR [8].

Abstract

Visuo-Haptic Augmented Reality (VHAR) is a technology, which enables users to see and touch virtual objects. It poses unique problems to developers, including the need for precise augmentations, accurate colocation of haptic devices, and efficient processing of multiple, realtime sensor inputs to achieve low latency.

We have designed and implemented the software framework HARP, which addresses these issues and simplifies the creation of haptic-enabled Augmented Reality (AR) applications. It allows developers to compose their applications on a high level of abstraction and hides most of the complexity of VHAR.

Our contribution is the initial design and implementation of the framework, which we have validated in nine VHAR research projects ranging from simple interaction design to psychophysical experiments. We discuss limitations of our approach and future developments. These insights will be helpful to researchers and framework designers in the field of VHAR.

Index Terms: H.5.1. [Information Interfaces and Presentation]: Multimedia Information Systems—[Artificial, augmented and virtual realities]; H.5.2. [Information Interfaces and Presentation]: User Interfaces—[Haptic I/O]; D.3.2 [Language Classifications]: Python, C, C++—; D.3.3 [Language Constructs and Features]: Frameworks—; D.2.10 [Design]: Rapid Prototyping—

1 Introduction

Augmented Reality is a technology that displays real and virtual objects aligned with each other in a real world scene [1]. When co-located haptic feedback is integrated into an AR environment, we refer to it as Visuo-Haptic Augmented Reality [5].

The fusion of AR and Haptics into one application is complex. Both subsystems interact and depend on each other on many levels. Several independent processing units need to run in parallel for haptic rendering, tracking, and visual output. The rendering speed ranges from 30 frames per second (FPS) for graphical rendering to 1000 FPS for haptic rendering. Every visual and haptic display and their spatial relations need to be calibrated carefully, to achieve high precision augmentations [2]. We are not aware of any software framework, which allows inexperienced researchers to create VHAR applications with accurate augmentations and haptic feedback. Using a specialized framework, the required knowledge and time to create VHAR applications can be reduced significantly.

In this paper, we present the design of HARP, a software framework for VHAR projects. HARP is designed to support application developers with limited programming skills when creating high-fidelity VHAR applications, but it also allows building more advanced systems through custom extensions. Developers can construct virtual scenes using 3D authoring tools, compose them using a simple scene graph, and add behavior using scripted nodes. The framework was used in several VHAR research projects at the Magic Vision Lab, where new students were able to create sophisticated VHAR applications within weeks or months. HARP simplifies the task of creating VHAR applications and allows developers or artists to work on haptic-enabled AR projects, without requiring in-depth knowledge about the underlying technology.

2 System Design

Our main design goal for HARP was to simplify the creation of VHAR applications, which run in realtime with low latency, display precise augmentations, and accurately co-locate haptic devices. We have designed the framework for two user groups: novice application developers like undergraduate students with limited programming abilities and expert developers who want to create advanced VHAR applications and will extend and customize the framework. HARP aims at hiding the complexity of building VHAR applications through a high level of abstraction and reasonable default settings, but still allow precise customization if needed.

The framework builds on an extensible scene graph, which provides scripting support, the ability to load scenes from X3D files, and viewers to run applications on various displays. Haptic rendering and collision detection is integrated using a haptic library. The library supports many commercial haptic devices. We include support for tracking, sensor fusion, and hand segmentation by using a mixed reality toolkit. A library of extension modules is available for advanced use cases, which includes: tools for calibration, profiling, and workspace configuration.

*e-mail: ulrich.eck@magicvisionlab.com
†e-mail: christian.sandor@magicvisionlab.com

IEEE Virtual Reality 2013
16 - 20 March, Orlando, FL, USA
978-1-4673-4796-9/13/$31.00 ©2013 IEEE

Project	LOC X3D	LOC Python	Amount of Work
Virtual Tools	200	202	1 Month
Object Painting	61	112	1 Week
Stroop Effect	143	500	3 Weeks

Table 1: Project Statistics: Lines Of Code (LOC) exclude mesh descriptions for application-specific X3D and Python files.

An important aspect of a VHAR architecture is the ability to process sensor input concurrently and in realtime. Haptic rendering should run at 1000 FPS, whereas the visual rendering speed is limited to the frame rate in which video images are captured and processed, typically 30 FPS. In the selected scene graph library, the haptic rendering for each haptic device is executed in a separate thread. Synchronization and data exchange between haptic threads and the scene graph only happens at the visual rendering frame rate. When images are processed within such a single threaded visual rendering loop, it can cause slow visual frame rates. In order to utilize all available resources, computational expensive tasks like computer vision should run in separate threads or processes. We added support for executing disconnected subgraphs in separate threads, which communicate via an explicit messaging layer. This extension enables us to utilize more CPU cores for tasks like image stream processing.

To allow developers to test their applications, we integrated record and playback of data streams into HARP. Selected channels of the data flow are recorded into a structured file database. Recorded data streams can then replace missing input for testing. The record/playback functionally is also suited to gather information during user studies or to analyze algorithms.

3 Implementation

HARP uses the haptic-enabled scene graph H3DAPI[6] as core dependency. The scene graph is defined using nodes, which contain fields, and connections between these fields define the data flow. To support efficient streaming and processing of video images within the H3DAPI data flow, we added support for shared image buffer nodes, which avoid copying of data and unneeded memory allocation. Based on the new image streaming capabilities, the framework provides a wrapper for the mixed reality software development toolkit MR-Platform by Canon [7], which integrates the Canon VH-2007 stereo video see-through Head-Worn Display (HWD), fiducial marker tracking, and color-based hand segmentation. Precise visual augmentations require calibration, which MR-Platform provides through offline tools for camera intrinsics and fiducial markers. Since users should be able to create VHAR applications on Linux and Mac OSX computers and MR-Platform is only available for Windows, we added support for IIDC firewire cameras and ARToolkitPlus.

4 Evaluation

Students at Magic Vision Lab have used HARP to develop VHAR applications in nine projects ranging from haptic device calibration to interaction design for tool selection and psychophysical experiments. Figure 1b shows a scene with 3D content, which was generated using standard authoring tools. The hand tool models can be attached to the haptic device to provide visual feedback about the selected interaction mode [9]. Figure 1c shows a rapid prototyping demo, which enables users to paint on arbitrary objects with haptic feedback. An object texture is modified based on the contact reported by the haptic device [4]. Figure 1d shows the user's view while participating in a study to verify a psychophysical experiment in AR [8].

In Table 1, we present project statistics in terms of effort needed to create the application. The number of Lines Of Code (LOC) for the scene graph description and logic include application specific files edited by the developer, excluding mesh descriptions. All applications run at 1000 FPS in the haptic loop and 30 FPS for graphical rendering under normal conditions.

5 Conclusions and Future Work

We have presented the design and implementation of our VHAR software framework. HARP has been used in nine research projects at Magic Vision Lab and it has proven to be simple enough to allow undergraduate students to create VHAR applications. We shield developers from unnecessary complexity by: providing a straightforward workflow for scene creation, Python script support for application logic, and a high level of abstraction for haptic interaction and augmentations.

Nevertheless, HARP needs to be improved to support more advanced VHAR systems. The integration of live video into H3DAPI limits the maximum rate of change propagation within the scene graph to 30 FPS. This causes jitter for haptic interaction with dynamic or deformable objects. H3DAPI also lack mesh to mesh collision detection and modern haptic rendering algorithms with implicit coupling [3]. Furthermore, physics-based simulation and advanced rendering techniques like ray tracing would be desirable to create VHAR applications with state of the art graphics and haptic interaction, but they cannot easily be integrated due to the architectural defects of H3DAPI.

Acknowledgements

Canon Inc., for HMD and MR Platform. eResearch SA, for loan of the Phantom Omni. SenseGraphics AB for H3DAPI and H3D developers for their contributions. All HARP application developers for their useful feedback.

References

[1] R. Azuma. A Survey of Augmented Reality. *Presence: Teleoperators and Virtual Environments*, 6(4):355–385, Jan. 1997.

[2] M. Harders, G. Bianchi, B. Knoerlein, and G. Szekely. Calibration, Registration, and Synchronization for High Precision Augmented Reality Haptics. *IEEE Transactions on Visualization and Computer Graphics*, 15(1):138–149, 2009.

[3] M. Ortega, S. Redon, and S. Coquillart. A Six Degree-of-Freedom God-Object Method for Haptic Display of Rigid Bodies with Surface Properties. *IEEE Transactions on Visualization and Computer Graphics*, 13(3):458–469, 2007.

[4] C. Sandor. Talk at TEDxAdelaide: The Ultimate Display, 2010. Video: `http://youtu.be/3meAlle8kZs`. Slides: `http://www.slideshare.net/ChristianSandor/tedx10-sandor`. Last accessed on 20 November 2012, 2010.

[5] C. Sandor, S. Uchiyama, and H. Yamamoto. Visuo-Haptic Systems: Half-Mirrors Considered Harmful. In *Proc. of the EuroHaptics 2007 Conference and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, pages 292–297. Tsukuba, Japan, 2007.

[6] SenseGraphics. H3DAPI - Open Source Haptics. `http://www.h3dapi.org`. Last accessed on 20 November 2012.

[7] S. Uchiyama, K. Takemoto, K. Satoh, and H. Yamamoto. MR Platform: A Basic Body on Which Mixed Reality Applications Are Built. In *Proceedings of the 1st IEEE and ACM International Symposium on Mixed and Augmented Reality*, pages 246–320, 2002.

[8] P. Weir. Evaluation of Psychophysical Effects in High Quality Augmented Reality Environments. Master's thesis, University of South Australia, Adelaide, Australia, 2012.

[9] K. William-Nyallau. Designing 3D Content for Visuo-Haptic Augmented Reality Systems. Master's thesis, University of South Australia, Adelaide, Australia, 2011.