OSGART - A pragmatic approach to MR

Julian Looser* Raphaël Grasset† Hartmut Seichter‡ Mark Billinghurst§

Human Interface Technology Laboratory New Zealand
University of Canterbury, Christchurch, New Zealand

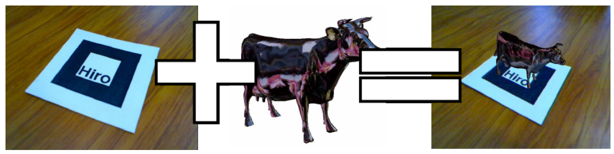

Figure 1: AR Toolkit + OpenSceneGraph = High Quality AR

ABSTRACT

We are presenting a software development framework called *OSGART* for Rapid Application Developement (RAD) in the domain of Mixed Reality (MR). This toolkit is being developed as an extension to OpenSceneGraph [3]. It implements a hierarchical, scenegraph based approach to marker based AR using the AR Toolkit [4]. Due to its' tight integration with OpenSceneGraph it provides a valuable test environment for bi-directional transition between immersive Virtual Environments (IVE) and Augmented Reality (AR).

CR Categories: H.5.1 [Information Interfaces and Presentation]: Multimedia Information Systems—Artificial, augmented, and virtual realities; D.2.2 [Software Engineering]: Design Tools and Techniques—Software libraries

Keywords: Augmented Reality, Mixed Reality, Content Creation, API, Framework

1 INTRODUCTION

Creating content for Mixed Reality (MR) and specifically Augmented Reality (AR) applications requires the definition of the relationship between real world and virtual world [7]. This inherent dependency of virtual to real world objects defines the interaction, perceived affordances and subsequently can create a new meaning. In the authoring process these relationships can be defined externally through a programming toolkit. The level of programmatic abstraction determine the differentiation between a low-level toolkit and a authoring system. Low level toolkits like *AR Toolkit* [4] are focused on computer vision and registration. Frameworks like *Studierstube* [10] provide important infrastructure, whereas authoring systems try to encapsulate the development system either through a graphical user interface [8, 1] or a meta description language [6]. A previous implementation called *OSGAR* [5] demonstrated the feasability of a similar approach with all the complexity involved using a 3D graphics visualisation framework that prefers predictable transformations. Our framework builds upon the experiences reported, but takes a more pragmatic way to provide a toolkit to author interaction and visualisation techniques. The *OSGART* framework follows the same principles of AR Toolkit by providing an easy way for prototyping AR applications with lowering the hurdle of intergrating AR with other visualisation and simulation toolkits. It therefore acts as a testbed for studies of perception and in Human Computer Interface (HCI) research. The conceptual structure of the *OSGART* framework demonstrates a systematic approach for user centric transition within the Reality-Virtuality Continuum [9] which enables novel ways of experiencing MR applications. The *OSGART* framework is largley been used in the *HIT Lab NZ* for various projects including industrial demonstrations, art installations, museum exhibits, student projects and experiments with user evaluations.

2 COMPONENTS

AR is being used in various domains and therefore needs to adapt to the specific needs of these domains. Hence, the *OSGART* framework generalises the available technological concepts but does not provide task specific utilities. The components within our framework provides a simplified handling of concepts that are crucial for an MR system.

2.1 Devices

Hardware components are generalised with various layers for traditional desktop interfaces (e.g. mouse, keyboard etc.) and VR I/O devices (e.g. tracking support). Unlike other toolkits *OSGART* specifically provides support for tangible user interfaces (TUI) and physical elements (e.g. gauges, sliders and motors). This directly enables the prototyping of media rich AR applications.

2.2 Content

An important issue in creating rich sensorial experiences in AR is the possibility to import existing content without further conversion. Because *OSGART* is based on OpenSceneGraph we can directly foster a comprehensive support for various media (e.g. images, video, 3D models and animation). The workflow from industry standard applications into OpenSceneGraph and therefore also into *OSGART* has been proven and is well received with industry partners. Specifically for MR we also support visual and spatial registration of existing physical content (e.g. 3D mockup, real text, illustrations).

*e-mail: julian.looser@hitlabnz.org
†e-mail:raphael.grasset@hitlabnz.org
‡e-mail:hartmut.seichter@hitlabnz.org
§e-mail:mark.billinghurst@hitlabnz.org

2.3 Interaction Mechanisms

In addition to existing desktop metaphors based on (WIMP), post-WIMP (e.g. speech, gesture) our framework supports traditional VR interaction techniques (e.g. ray based interaction, various navigation modes). A rich set of existing interaction techniques for tangible and transitional metaphors is included to directly support AR applications.

2.4 Collaboration

Inherently, AR application afford multiple users. This is important for review and design systems as well as educational applications. Therefore, user presence and copresence, interaction awareness (e.g. telepointer, gaze) are seamlessly integrated. The components are made network transparent through ICE [2] middleware. The *OSGART* framework provides a variety of techniques for mixing real and virtual feedback, multiple spaces and interaction within and across these spaces.

3 IMPLEMENTATION

OSGART is implemented in C++ and portable across various platforms. External bindings provide the possibility to create AR content with scripting languages. One of the main problems in AR is the aquisition of video for the visual registration. A module within our toolkit is covering video sources like files, live streams and generated video.

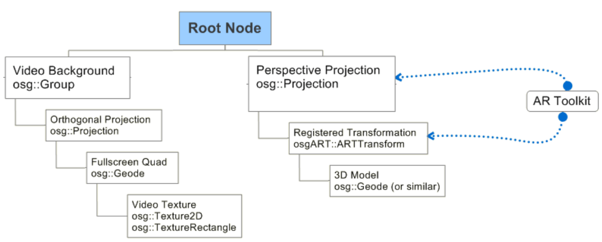

Figure 2: Structure of OSGART

In addition to the abstraction of various AR concepts the *OSGART* framework provides a high level programming approach is possible through the usage of a scripting environment like Python, Ruby or Lua. Runtime environments like Python provide a comprehensive infrastructure which can be fostered to integrate functions into a prototype application. Additional thin lowlevel scripting languages like Lua can be integrated in order to change runtime behaviour for computing intensive and often updated branches in the framework. This way we make optimal use of the different capabilities of runtime interpreted programming without compromising functionality or speed. This level of abstraction also provides us the necessary functionality that is necessary for an authoring tool based on the *OSGART* toolkit.

4 CONCLUSION

In this position paper we introduced a new framework for developing AR applications with a high level API. The advantages of our toolkit over others are the direct conceptualisation of tangibility, transitionality and collaboration awareness. The *OSGART* is already been used in various projects in our laboratory and with industry partners. In the workshop an in-depth overview about the techniques and coding examples will be given. Furthermore, the

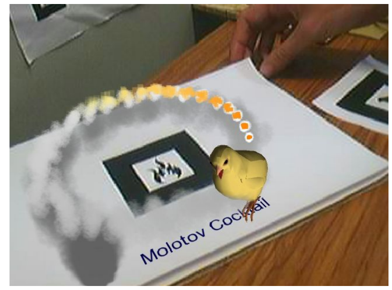

Figure 3: MagicBook art installation presented at *Experimenta Vanishing Project (Australia)* using particle systems and shadow.

Figure 4: MagicLens interaction technique with different rendering techniques.

creation of this framework enables us to create a novel way of authoring and configuring AR applications without the explicit use of a programming environment.

REFERENCES

[1] AMIRE - authoring mixed reality. `http://www.amire.net`.

[2] ICE middleware toolkit. `http://www.zeroc.com`.

[3] OpenSceneGraph. `http://openscenegraph.org`.

[4] M. Billinghurst, S. Campbell, W. Chinthammit, D. Hendrickson, I. Poupyrev, K. Takahashi, and H. Kato. Magic book: Exploring transitions in collaborative ar interfaces. In *Emerging Technologies Proposal, ACM SIGGRAPH.*, 2000.

[5] Enylton Machado Coelho, Blair MacIntyre, and Simon Julier. OSGAR: A scene graph with uncertain transformations. In *International Symposium on Mixed and Augmented Reality (ISMAR04)*, 2004.

[6] F. Ledermann and D. Schmalstieg. April: A high-level framework for creating augmented reality presentations. In *IEEE Virtual Reality 2005*, 2005.

[7] Blair MacIntyre. Authoring 3d mixed reality experiences: Managing the relationship between the physical and virtual worlds. In *ACM SIGGRAPH and Eurographics Campfire: Production Process of 3D Computer Graphics Applications - Structures, Roles and Tools*, 2002.

[8] Blair MacIntyre, Maribeth Gandy, Steven Dow, and Jay David Bolter. Dart: A toolkit for rapid design exploration of augmented reality experiences. In *User Interface Software and Technology (UIST'04)*, 2004.

[9] P. Milgram and F. Kishin. A taxonomy of mixed reality visual displays. *IEICE Transactions on Information and Systems*, 77(9):1321–1329, 1994.

[10] Dieter Schmalstieg, Anton Fuhrmann, Gerd Hesina, Zsolft Szalavari, L. Miguel Encarnao, Michael Gervautz, and Werner Purgathofer. The studierstube augmented reality project. *Presence - Teleoperators and Virtual Environments*, 11(1):33–54, February 2002.