ICAT 2005
December 5-8, Christchurch, New Zealand

OSGARToolKit: Tangible + Transitional 3D Collaborative Mixed Reality Framework

Raphaël Grasset, Julian Looser, Mark Billinghurst
Hit Lab NZ, University of Canterbury, Private Bag 4800, Christchurch, New Zealand
{*raphael.grasset*|*julian.looser*|*mark.billinghurst*} *@hitlabnz.org*

Abstract
In recent years there have been a number of software frameworks proposed for developing augmented reality (AR) applications. However these frameworks typically do not address three types of interfaces; tangible, transitional and collaborative user interfaces. In this paper we describe a new software framework weve developed that focuses on these three characteristics.

Firstly, we introduce a simple way to build complex tangible applications using a rapid-prototyping approach. Secondly, we offer an environment to build and support transitional interface. Finally, our framework offers a simple way to support collaborative requirements combined with the use of the GPU for improving user awareness.

Keywords: Augmented Reality, Mixed Reality, Transitional, Tangible, Toolkit

1. Introduction
Recently, many research groups have become interested in providing high level tools for creating AR applications. By combining virtual and physical elements AR interfaces completely redefine the way that software applications are developed. For example, the MagicBook project [1], is a mixed reality (MR) book that needs a large integration of complex media content (3D animated virtual models, spatial sound) while using a range of natural interaction metaphors (e.g. turning the page, tangible interaction).

We can distinguish three categories of existing toolkits for creating AR applications: low level libraries/toolkits, high level frameworks/middleware, or rapid-prototyping tools and design applications.

A low level toolkit (generally based on C programming) provides a high degree of flexibility but has a long development time for complex applications (e.g. MXRToolKit [5], and ARToolKit [1]).

A high level framework generally integrates a large number of services, has better portability, and can be easily extended to creating new AR applications (e.g. ARTHUR [2]).

Rapid prototyping tools can be a way for non-programmers to create simple AR applications (e.g. language or script based) or, can be complete AR authoring tools (e.g. DART [4]).

We have developed a new Mixed Reality framework, OSGARToolKit, that not only supports standard MR requirements but also allows the development of the next generation of 3D MR applications. Our work focus on three main elements (Figure 1): tangibility, transitionality [1] and collaboration awareness.

These elements have been individually demonstrated on MR prototypes before, however, there have been few efforts to provide a dedicated high level framework supporting all of these elements together (like being able to create a complex application like 3D Live! [5] from a high level description). At its most basic level our framework is similar to OSGAR [3], however we add elements directly related to AR interaction and collaboration.

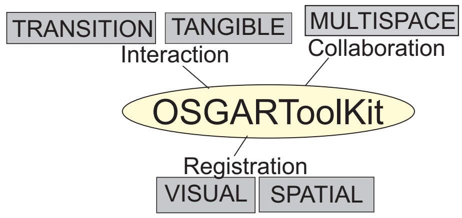

Figure 1: AR specific functionalities of the framework.

2. Approach
Based on a requirements analysis for AR application, we present our current work on focused services.

2.1. Requirements Analysis

Mixed reality applications cover a large multi-disciplinary continuum which encompasses other domains, like traditional CHI, virtual reality or pervasive computing. We try to identify requirements based on this consideration and on a two level viewpoint: the developer and user stage. In developer stage, in addition to traditional requirements (e.g. genericity, modularity, extensibility), the focus needs to be on the authoring of AR content: providing tools to setup and configure the AR content. For the user, the following services needed to be available:

- *Devices*: support of desktop interfaces (mouse, keyboard), VR devices (input/output). Services dedicated to MR: support for tangible input (vision tracking, physical devices).
- *Content*: support of multimedia data: image, video, 3d content and animation. Services dedicated to MR: support for registration of the content (visually and spatially) and also support of physical content (3D Mockup, real text, illustrations).
- *Interaction Mechanisms*: support for desktop metaphors (WIMP), postWIMP (speech, gesture) traditional VR interaction (direct manipulation, navigation mode). Services dedicated to MR: support for tangible and transitional metaphors.
- *Collaboration*: user presence and copresence, interaction awareness (e.g. telepointer, gaze). Services dedicated to MR: mixed real and virtual feedback, multiple spaces of interaction (real, virtual, augmented virtuality (AV), augmented reality (AR)).

Based on these requirements, we choose to combine existing tools to respond to traditional requirements.

2.2. Manipulation:Tangible

We based our approach on a 'connect the box' solution where two elements (physical and virtual) can be easily associated with a simple typed relationship (e.g. 3D spatial relationship). As shown in figure 2, we actually use programming mechanisms similar to the 'signal and slot' approach recently used in Qt Graphical ToolKit. For ease of rapid prototyping, we choose to use high definition of the association (contrary to [2]). In a post-processing stage, the library tries to integrate when possible the coupling directly inside the scene-graph of the application (reducing the complexity of managing all relationships).

```
...
3DVirtualModel model;
model.Load("car.xml");

SpatialObject* mark1=ARToolKit::GetMarkerRef(0);

COUPLING_LOCATION(mark1,model);
...
```

Figure 2: Code example for creating simply a tangible spatial interface.

2.3. Navigation:Transitional

Our framework supports definition of multiple type of spaces (AR, VR, AV, Real), multiple views and also functionalities to define transitions between these elements (illustrated Figure 3 with a MagicBook application). During a viewpoint changing (transition), it's important to give feedback and awareness to the user. For this purpose we maximize the used of GPU programming, proposing hardware-based awareness transition, e.g. fading between the two views.

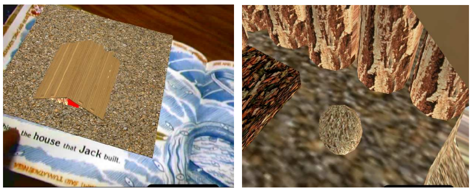

Figure 3: Transition support inside a virtual hutt: from an AR viewpoint to a VR viewpoint.

2.4. Collaboration:Awareness

The focus of our framework is dedicated to closely coupled cooperation, but it also supports multiple space collaboration at different scales ($AR \leftrightarrow AR$,$AR \leftrightarrow VR$, $VR \leftrightarrow AV$, etc.).

Our framework supports a broad range of video streaming: webcam video, movie file and also network video (RTP, TCP/UDP). The video offers us a large number of ways to create virtual avatars, from a simple 2D billboard, to a chroma-keying avatar, and also 3D video avatars. We use the GPU to provide most of the image-processing stage. We offer to the developer a wide choice of integrated classes for rapidly prototype collaborative support: tele-pointers, head avatar, gaze awareness, etc. The library not only furnishes awareness from the remote users, but also augments the awareness of co-located users (e.g. the real gaze direction of other users difficult to observe with HMD).

3. Results

The library is mainly based on the OpenSceneGraph framework, delivering us a pre-existing large choice of multimedia content but also the ability to import from professional designing tools (e.g. Maya, 3D Studio Max). The tracking is mainly based on computer vision, using ARToolKit (extended also for supporting inertial tracking). The physical inputs are supported by a homemade micro-controller connected to the serial port, offering the use of a wide choice of sensors/actuators (switch, pressure, LEDs, motor, etc.) for prototyping tangible AR interface elements.

Figure 4: Collaboration support (video avatar) and tangible support (interactive paddle with physical controls).

Figure 5 shows a MagicBook application developed with the framework that integrates 3D animation, sound and graphics effects. Our library supports traditional spatial registration but also visual registration, like shadow, occlusion or integrating real lighting conditions (image-based lighting).

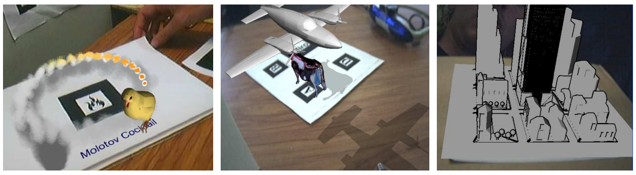

Figure 5: AR applications with high quality rendering.

4. Conclusion

This paper introduces a new framework for developing high level AR application. The advantages of our toolkit over others one are:

- *tangibility*: high level functionalities for fast prototyping interaction metaphors for 3D AR application.
- *transitionality*: first high level framework for creating transitional multispaces applications.
- *collaboration awareness*: easy integration of MR collaborative cues (e.g. gaze).

The software is in its final development stage, and soon we are hoping to produce a development workshop for teaching others how to use it. Based on applied projects, we are also interested in the AR design pipeline and trying to produce new authoring tools for easily configuring AR applications.

References

[1] Mark Billinghurst, Hirokazu Kato, and Ivan Poupyrev. The magicbook: A transitional ar interface. *Elsevier Computers and Graphics*, pages 745–753, 2001.

[2] Wolfgang Broll, Irma Lindt, Jan Ohlenburg, Iris Herbst, Michael Wittkmper, and Thomas Novotny. An infrastructure for realizing custom-tailored augmented reality user interfaces. *IEEE Transactions on Visualization and Computer Graphics*, 11(6), 2005.

[3] 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.

[4] 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.

[5] Simon Prince, Adrian David Cheok, Farzam Farbiz, Todd Williamson, Nikolas Johnson, Mark Billinghurst, and Hirokazu Kato. 3d live: Real time captured content for mixed reality. In *ISMAR 2002*, 2002.