# Augmented Reality Haptics: Using ARToolKit for Display of Haptic Applications

Matt Adcock, Matthew Hutchins, Chris Gunn
*CSIRO Australia*
*Matt.Adcock@csiro.au*

## Abstract

*This paper describes the integration of the ARToolKit with the Reachin Core Technology API. The result is a system capable of providing a coherent mix of real world video, computer haptics and computer graphics. A key feature is that new applications can be rapidly developed. Ultimately, this system will be used to support rich object based collaboration between face-to-face and remote participants.*

## 1. Introduction

Our project goal is to develop a system that combines haptic interaction, 3D computer graphics and auditory interfaces to support collaborative tasks such as design, mentoring and data exploration.

The majority of haptic/graphic system configurations are designed for single user experiences. Usually they consist of a haptic device sitting next to a computer monitor. For a more co-located configuration, we have combined our PHANToM [8] haptic devices with stereo shutter glasses and a mirror [9] (see Fig. 1).

Collaborative haptic applications have now started to emerge [2][4]. However, the majority of these systems have been targeted at telecollaboration rather than face-to-face situations. We believe that Augmented Reality (AR) technologies are well suited to enabling haptic applications to be developed to support and enhance face-to-face collaborative tasks.

There have been some previous investigations that have combined the PHANToM device with AR technologies. Vallino and Brown [10] have investigated the merger of AR and computer haptics. They demonstrated a system that allowed the user to touch virtual objects as displayed in an Augmented Reality system. Their main goal was to accurately align the coordinate systems of the real world, the graphical display, and the haptic display.

Inami et al describe another example of a PHANToM being combined with AR technology [5]. In their system, the haptic device was covered with retro reflective material. A (tracked) head mounted projector was then used to display a virtual scene in such a way as to make the haptic device seem semitransparent.

## 2. The Reachin API and the ARToolKit

The Reachin Core Technology API [7] (formerly Magma [9]) was created in order to better integrate haptics and graphics rendering. It is based on a haptic/graphic scene-graph that borrows a lot of its structure from VRML97 [3]. Applications can be developed rapidly by creating a file that specifies and connects a number of Reachin scene-graph nodes through a system of 'routes'. These files are interpreted at run time. If more complex behaviour is required, new nodes may be created in C++ by sub-classing from other nodes.

The ARToolKit [1] allows developers to include vision-based marker tracking in their applications. It is primarily used to render virtual objects in such a way that they are perceived to be co-located with specific fiducial markers. Inversely, it is also capable of tracking the camera location with respect to the marker.

Reachin requires that new nodes be compiled using Borland CBuilder[1]. We used Borland's bundled conversion tools to create wrappers for the ARToolKit version 2.65 (with Direct Show support) libraries. Using these, we have created a Reachin node that provides ARToolKit functions to Reachin applications through its VRML-like interface.

## 3. The ARTracker Node

The ARTracker node is a Reachin Child node that encapsulates the ARToolKit routines needed for camera tracking. Its field interface and default values are:

```
ARTracker
  numActiveMarkers    0
  position            0 0 0
  orientation         0 0 1 0
```

The behaviour of this node is best considered in three parts: its constructor, renderer and the update() function of the numActiveMarkers field.

The Constructor calls an initialisation function that acts similarly to the init() functions in the ARToolKit example programs. It first loads in from files the camera parameters and the fiducial marker pattern data. At this point the camera parameters are modified for the real world tracking context, as described by Piekarski and

[1] We understand that this may not be required in future versions of the Reachin API.

Thomas [6]. An ARFrameGrabber object is then initialised and used to grab an initial video frame.

The Renderer function is called by Reachin as it traverses the scene graph during its graphics rendering loop. In this function, we switch to an orthographic projection mode and render the latest video frame image on an OpenGL Quad. This Quad is positioned just in front of the rear clipping plane and takes up the whole window, which is itself usually filling the screen. The ARToolKit can 'correct' the camera distortion in the video image but, in this context, we have not yet found it is needed.

Reachin provides a mechanism for overriding a field's update() function. This function is called whenever the field's value is required. We have overridden the behaviour of the numActiveMarkers field such that it also determines new values for the position and orientation fields. It is 'hot wired' to require updating every frame.

The update() function first uses the node's ARFrameGrabber object to grab the latest video frame. It then calls arDetectMarker(), arGetTransMat() and arUtilMatInv() to produce the location of the camera relative to the marker. In order to make this location matrix information more compatible with Reachin, we use arUtilMat2QuatPos(), then negate the resulting quaternion's vector and rotate it 90º about the x-axis.

Finally, these values are modified according to the location of the marker, relative to the origin of the Reachin global coordinate system. The resulting position and orientation information is inserted into the respective fields of the ARTracker node.

The ARTracker node can be used in most Reachin applications simply by routing its position and orientation fields to the Viewpoint node. It can also, just as easily, be routed to other objects in the scene-graph. We have found (by trial) that setting the Viewpoint node's 'field of view' to 0.58 works well for our Logitech QuickCam Pro 3000.

Fig. 2 shows an ARTracker example in which we have offset the real haptic stylus from the virtual one. It is not possible to seamlessly align these (as was Vallino's pursuit) due to the temporal latency of the video image and the spatial tricks used by Reachin's haptics rendering.

## 4. Future Work

Shortly, we will increase the video resolution, incorporate multi-marker tracking (for robustness) and resolve a memory usage bug observed when the program exits. We then intend to integrate our collaborative haptic software tools [4] to enable more than one person to interact with the virtual objects.

In the longer term, we hope to include the ability to render in stereo. We would also like to provide a more versatile interface to the ARTracker node and extend it such that it can support additional markers or groups of marker for tasks such as mode switching and navigation.

## 5. Conclusion

We have integrated the ARToolKit with the Reachin API to create a system that allows the rapid development of rich multimodal experiences. It has the potential to support rich object based collaboration between face-to-face and remote participants.

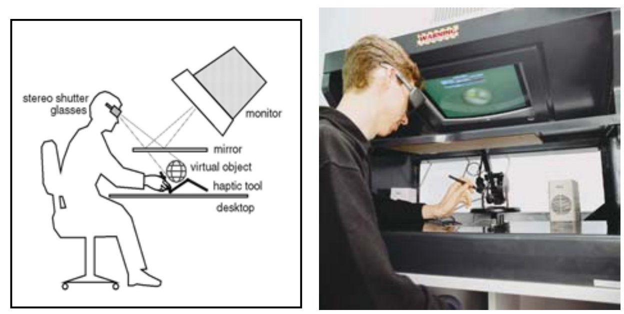


**Figure 1. The CSIRO Haptic Workbench.**

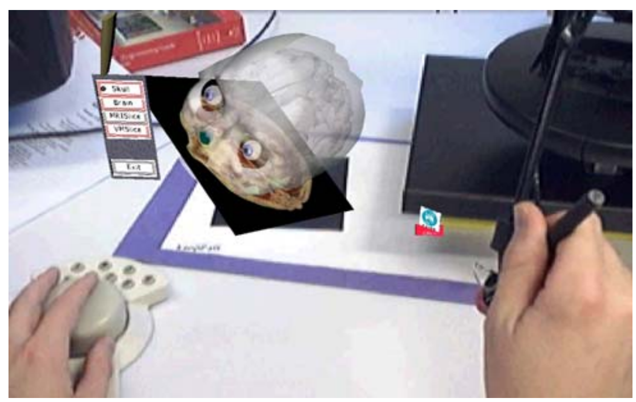

**Figure 2. Interacting with medical data (the virtual interaction tool is accessing the menu, top left.).**

## 6. References


[1] ARToolKit (http://www.hitl.washington.edu/research/ shared_space/download/)

[2] Basdogan, C., C.-H. Ho, M. A. Srinivasan and M. Slater (2000). "An experimental study on the role of touch in shared virtual environments." *ACM Transactions on Computer-Human Interactions* 7(4): 443-460.

[3] Cary, R. and Bell, G. *The Annotated VRML 2.0 Reference Manual*, Developers Press, USA,1997

[4] Gunn, C., Hutchins, M., Adcock, M. and Hawkins, R. "Trans-World Haptic Collaboration", *SIGGRAPH 2003, Conference Abstracts and Applications*, San Diego, July 2003

[5] Inami, M., Kawakami N., Sekiguchi, D., Yanagida, Y., Maeda, T. and Tachi, S., "Visuo-Haptic Display Using Head-Mounted Projector", *Proceedings of IEEE Virtual Reality 2000*, pp.233-240, 2000

[6] Piekarski, W. and Thomas, B. "Using ARToolKit for 3D Hand Position Tracking in Mobile Outdoor Environments", In *2nd International Augmented Reality Toolkit Workshop*, Darmstadt, September 2002.

[7] Reachin Technologies AB (http://www.reachin.se)

[8] SensAble Technologies Inc. (http://www.sensable.com)

[9] Stevenson, D., Smith, K., Veldkamp, P., McLaughlin, J., Gunn, C. and Dixon, M., "Haptic Workbench: A Multisensory Virtual Environment", *The Engineering Reality of Virtual Reality, Electronic Imaging '99*, San Jose, January 1999.

[10] Vallino, J. and Brown, C., "Haptics in Augmented Reality", *Proceedings of the IEEE International Conference on Multimedia Computing and Systems,*, Florence, June 1999.