

Prototyping Applications For the Physical World Using Integrated Capture/Playback Facilities

Steven Dow¹, Blair MacIntyre¹, Maribeth Gandy² and Jay David Bolter³

¹College of Computing, ²Interactive Media Technology Center

³School of Literature, Communication and Culture

GVU Center, Georgia Institute of Technology

Atlanta, GA 30332, USA

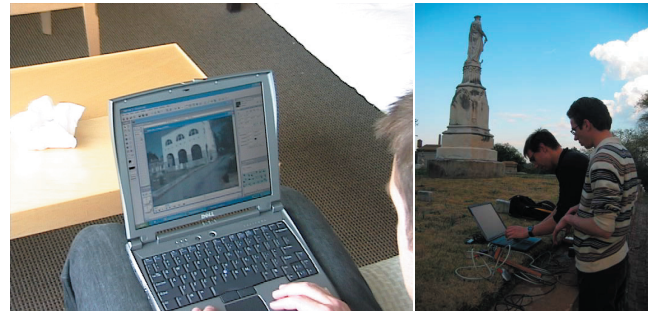
{blair, steven}@cc.gatech.edu, maribeth.gandy@imtc.gatech.edu, jay.bolter@lcc.gatech.edu

Abstract

Designers of digital applications that deal with complicated infrastructures in the physical world often deal with the formidable challenges of working in a specific place. During development and testing, the designer must constantly move around the physical space to debug or test new interactions. In outdoor settings, factors such as weather, poor work ergonomics, and the lack of power and networking are serious impediments for designers. Our strategy is to use a flexible capture/playback infrastructure, which is tightly integrated into a design environment for prototyping physical applications. We are able to capture the sensor data necessary to design and test applications that can then be easily deployed to the real environment. We will demonstrate this approach within the context of DART (the Designer's Augmented Reality Toolkit) by showing how augmented reality applications (such as mocking up the placement of information displays) can be quickly prototyped and tested using captured data sets. The capture/playback method breaks the requirement that sensors be used synchronously, in real-time, in the actual location. The flexible programming environment of Director, used by DART, will allow us to demonstrate this design process to conference attendees.

Introduction

One of the main attractions of Augmented Reality (AR) (as well as other off-the-desktop computing paradigms, such as wearable and ubiquitous computing) is the tight integration of computer-generated material with the physical world. For designers trying to work with AR, this binding between physical and virtual worlds means that the programs they are creating are driven largely by data that is sensed about the world, such as the location of the user and where they are looking, user gesture or speech, and so on. This reliance on sensor data means that, from early design conception through development and user-testing, the designer needs to be physically present in the target environment if they want to see how the application “really” works. The need to work with realistic, live data in the target space seriously hampers the development of experiences that mix physical and virtual worlds.



(a) Working with recorded data

(b) Working live

Figure 1: (a) Working with captured data in a lounge near our lab, (b) dealing with live sensors in Oakland Cemetery. (photograph courtesy of J. Cothran, J. Hoffman, Z. Pousman and M. Norton)

The need to work in the target environment has often been a source of frustration for us. This has been especially true of the outdoor experiences we have built, such as campus tours and our more recent dramatic experiences in historic sites. Over the past four years we have been teaching a class on AR Experience Design to Computer Science and New Media students; the difficulties of our collaborators and students has prompted us to take a serious look at the underlying problems faced by designers prototyping AR experiences. The result of our initial work on this problem is the *Designer's AR Toolkit (DART)*, a collection of AR experience design tools built on top of Macromedia Director. (DART is designed to solve a variety of problems encountered by non-technical designer's when trying to use AR, ranging from the difficulty of the technology to the complexity of the content—for an overview of DART, see [2]). This demonstration focuses on a specific feature of DART designed to free an AR developer from some of the constraints of working with the physical world: the *capture and playback* of the inputs to an AR experience and modular application prototyping on top of the captured input. In general, we seek to understand the process followed by designers in order to allow them to prototype applications for a physical space with different strategies.

The basic idea of this work is straightforward: allow a designer to easily capture all of the inputs (e.g., live video and sensor data) to an experience, and to replay them in the development environment so they appear live to the application. In theory, this should allow designers to work on

the experience's content, seeing how it really looks and works in the target space, without having to be there. While the idea of working with captured data is not new (the computer vision and computational perception communities have been using captured data sets to design, debug and compare new algorithms for many years), the integration of these facilities into an end-user design and development environment presents new opportunities for radically changing the way in which MR, AR and UbiComp systems are designed and built.

System Description

Macromedia Director is a powerful, flexible multimedia-authoring environment built around an object-oriented computer programming language called *Lingo*. Director utilizes the notion of a *stage* (where content is placed), multiple *casts* (where all content elements are stored, including images, video, 3D content, Lingo scripts, text data and so forth), a *score* (the time line of the experience) and *sprites* (cast members that have been placed on the stage or in the score).

Our work on DART integrates various sensors into Director through an external library (or Xtra). The Xtra, written in C++, handles the interfaces with video cameras (we support DirectShow and several custom capture libraries), marker tracking on video frames (currently using ARToolkit [1]), and a large collection of trackers and sensors (via VRPN [4]). This functionality is afforded through modular behavior scripts, which are structured to take advantage of the score's drag and drop ability. Scripts, such as Actors, Events, Physics, etc, are building blocks for interactive, multi-state, AR experiences. These DART building blocks can be stacked on the score in Director, configured using property windows, and manipulated visually to quickly adjust the flow of an application.

There are four scripts that control the capture/playback of video and tracking information: *CaptureVideo*, *CaptureTracker*, *PlaybackVideo*, and *PlaybackTracker*. These are managed on the score in Director so that a user can determine which trackers to capture and for what length of time. All of the data is saved into cast libraries, where it can be viewed and modified if necessary. All captured data is stored in independent cast members, but has synchronized timestamps stored as offsets from the capture start time. Any VRPN devices (e.g., Buttons, Analog, shared memory objects, etc), ARToolkit reports, or other application specific sensor data can be captured.

The obvious reason for integrating capture/playback tools into a prototyping environment is to facilitate offsite application development using time-synchronized sensor data (onsite development is also more ergonomic since the designer doesn't have to repeatedly move around a space to test ideas). Additionally, integrated capture/playback facilities allow other strategies to improve the design process:

- Captured data sets appear no different than live data to the application. This makes quick substitution (particularly

within the modular, drag & drop environment of Director) of pre-captured data sets for live data simple.

- Specific sensor input can be used repeatedly, so that applications can be incrementally developed, debugged, evaluated against other designs on the exact same input, and demonstrated to collaborators in any location.
- The captured data does not have to be played back at the original speed. A key to supporting capture/playback is a global abstract clock (*DARTclock*) used to control the time in DART; all DART entities use this clock rather than Director's built in clock. The designer can pause, step through the data slowly, and rewind the data. This is useful for jumping to a particular time within a segment or for quickly reviewing a captured segment.
- Live and captured data can be mixed so that while some data enters the application live, other data can be simulated. For example, an application could combine live user input with simulated data; this could be useful for gathering user feedback on a particular design.
- Applications can be designed without functioning sensors. If sensors have not been deployed (or if they are broken) the data can be manually created to simulate a captured data set.

Related Work

The Phidget toolkit [3], aimed at making tangible devices available to designers (via COM and ActiveX), provides wizard-of-oz graphical proxies to allow development of applications without access to sensors. In contrast, capture/playback facilities provide realistic sensor data, enabling a design strategy beyond WoZ graphical interfaces. In the UbiComp community, there have been a number of projects involving capture and access systems primarily used for knowledge acquisition within a particular physical space (e.g. [5]). Our work uses captured data sets to design multimedia applications in a prototyping environment. When a design is complete, the application can be easily deployed in the actual physical location with actual sensor data.

References

1. Billingham, M., Bowskill, J., Jessop, M., and Morphett, J., "A Wearable Spatial Conferencing Space", In *Proc. ISWC '98*, 1998, pp. 76-83.
2. MacIntyre, B., Gandy, M., Dow, S., Bolter, J.D., "DART: A Toolkit for Rapid Design Exploration of Augmented Reality Experiences", *To appear in UIST'04*, 2004.
3. Greenberg, S. and Fitchett, C., "Phidgets: Easy Development of Physical Interfaces through Physical Widgets." In *Proc. UIST'01*, November 11-14, 2001, pp. 209-218.
4. Taylor, R., T. Hudson, A. Seeger, H. Weber, J. Juliano, A. Helser, "VRPN - A Device-Independent, Network-Transparent VR Peripheral System", In *Proc. VRST'01*, Nov. 15-17, 2001, pp. 55-61.
5. Truong, K.N., Abowd, G.D., Brotherton, J.A., "Who, What, When, Where, How: Design Issues of Capture & Access Applications", *UBICOMP'01*, 2001, pp. 209-224.