ImageTclAR: A Blended Script and Compiled Code Development System for Augmented Reality

Charles Owen, Arthur Tang, Fan Xiao

Media Entertainment Technology Laboratory
Michigan State University, East Lansing, Michigan

cbowen@cse.msu.edu, tangkwo1@msu.edu, xiaofan@cse.msu.edu

Abstract

AR systems are usually highly specialized and task-specific, but there are also some system common tasks to be carried out. The ImageTclAR augmented reality (AR) development system is a Tcl/Tk-based development environment targetting different level of system developers with widely different background and programming abilities. The modularity and blended script and compiled code development features of ImageTclAR toolkit significantly aids the process of AR system development on different level, and faciliates he reuse of components. A few application cases of this development environment in rapid prototyping and experimental design by different level of users are also discussed.

1. Introduction

In augmented reality (AR), user perceives the environment around them with the computer-generated virtual world superimposed on or composed with the real environment. Code of different AR systems usually shares some common components such as initial devices driver setup, event scheduling and dispatching, real time 3D graphics rendering, reference frames calibration, stereoscopic display method, and user interface components. Most of the programming tasks of these components require extensive programming skill and experience, and these kinds of programming skill are not typically bared by average application developers, interface designers, behavioral scientists, and experimental psychologist. ImageTclAR toolkit is a system designed to support AR system software development for different level of users, with different level of requirements and programming skill. Its aims are to provide a programming environment for rapid AR applications development and a structured scheme for various levels of development.

The ImageTclAR development environment is an extension of the ImageTcl multimedia algorithm development environment [1]. It is based on the Tool Command Language (Tcl) and the Tk Toolkit [2]. The Tcl/Tk Toolkit has been designed to provide application and algorithm developers with a rapid prototyping environment. It also provides a structured scheme for various levels of development. The highly modular structure design of ImageTclAR toolkit allows standardized units to be reused between different applications, as well as faster compile and link times during the softwae development process.

2. Related Work

There are a few programming libraries developed specially for AR system development. The Grasp augmented vision system developed at European Computer-Industry Research Center [3] is an object-oriented library written in C++ that consists of classes for some common operations in typical AR systems such as calibrations, rendering, file I/O, and input devices. The ARToolkit developed at the Human Interface Technology Laboratory [4, 5] is a C based computer vision tracking library that can be used to calculate camera position and orientation so that 3D virtual objects can be rendered and aligned with a real object. The Virtual Reality Laboratory at Ecole Polytechnique Fédérale de Lausanne developed a C++ based development frame work for virtual reality and augmented reality called VHD++.

Most toolkits or application-programming interface (API) for AR system development have focused on providing low level system libraries for common routines being used by typical AR systems that is linked to an application program. This approach does not favor the software development process that requires extensive testing and adjustment of parameters for rapid prototyping. Using an API to start a complete programming project is also not suitable for novice developers, who have limited programming ability.

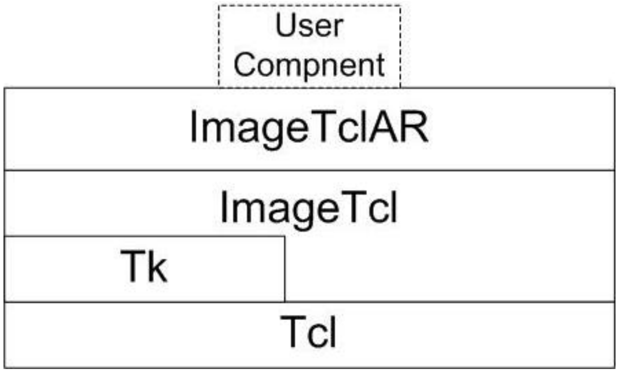

Figure 1. Dependency structure for ImageTclAR Toolkit.

3. Design Criteria

The design of ImageTclAR toolkit, and the basic ImageTcl system, was chosen to ease the software development process for AR system with extensive testing and parameter adjustment. It also aims to support tasks normally undertaken by users with widely different backgrounds and abilities. The major design features have evolved through three major iterations, each attempting to better address these needs. During the design process, the major tasks in the system were fundamentally based on the classes of users.

ImageTclAR toolkit has been designed to support the needs of three major classes of users: novices, application developers, and advanced developers. Novice users are interested in creating augmented reality applications with limited (often negligible) programming skills. Application developers have advanced programming skills including high-level language programming, but do not need to know the internal details of the system, preferring to view the system as an application-programming interface. Advance users are typically researchers who may develop new major system components or replace major portions of the system with experimental components.

4. System Structure

The major software structure of the ImageTclAR toolkit is illustrated in Figure 1. This section describes the components in this figure.

4.1 Tcl

The base system is Tcl. Tcl is a simple textual language, intended primarily for issuing commands to interactive programs such as text editors, debuggers, illustrators, and shells. Tcl can be used to write command procedures to provide more powerful commands than those in the built-in set of commands.

4.2 Tk

Tk is an extension to Tcl that provides the developer with an interface to the windowing system. In the windows scenario developers will encounter Tk by way of the wish command. Wish is a simple windowing shell that permits the developer to write Tcl applications in our prototyping environment.

4.3 ImageTcl

ImageTcl is a package that, as a stand-alone package, is not functional. ImageTcl paired with the core Tcl system with standard commands becomes a highly modular rapid prototyping environment for image, audio, and signal processing.

4.4 ImageTclAR

ImageTclAR is a package built on top of the ImageTcl package to add features that facilitate augmented reality and virtual reality development. ImageTclAR is dependent upon the components beneath it in Figure 1. ImageTcl and ImageTclAR perform low level functions such as processing and rendering at the compiled code level. Control, sequencing and user interfaces components are composed at the Tcl scripting level.

4.5 User Component

User components are components that are created by the developer of an AR system for a specific task. User components are used so that users can get the flexibility they need in order in order to implement any AR system but do not have to make changes to the core ImageTclAR system. An example of a user component that may be developed is with the iartracker module. If the user would like to support a tracker that does not already have a module written for the tracking device, a new user component is required to make the device usable to the system.

4.6 Current User Components

A number of components were created for tasks often being used by a typical AR system. A tracker module is developed to support several different position and orientation tracking systems, including the Polhemus IsoTrak® II and Fastrak®, Ascension Flock of Bird®, and InterSense IS-600 Mark 2 and IS-900. A joystick module is developed for interaction using a standard joystick. A VRML import module is created to provide a way to import virtual objects created in 3D graphic modeling software. A calibration module, with different calibration algorithms and methods, is available for different kinds of calibration, including stylus tip calibration, workspace calibration (affine and rigid body transformation), and optical see-through head-mounted display (HMD) calibration (a variation of the SPAAM algorithm [6]). A stereoscopic display module is also available to support two stereoscopic display methods (frame sequential stereo and dual output) for different HMDs.

5. Blended Script and Compiled Code Development

ImageTclAR toolkit addresses the disparate needs for different levels of developer through the use of mixed script/compiled code development in conjunction with an arsenal of software tools designed to automate common tasks. The most visible design element is the mixture of script-level development in the Tcl programming language with compiled code development in C++. The script-level interface provides tools for application sequencing, parameterization, and component glue.

5.1 Application Sequencing

The novice typically has very limited programming capabilities. Most novices are able to learn the basics of scripting languages, but cannot be expected to understand the intricacies of modern compiled languages such as C++ or Java. Application sequencing and basic graphic design are the most common requirements of novice users. They need to create the graphical elements for an AR application and they need to be able to design the user interaction with the environment. ImageTclAR toolkit provides this capability through the scripting interface.

5.2 Parameterization

In the research environment, there is a general need to change parameters to applications for either experiments or tests. This need is met with the parameterization of ImageTclAR toolkit because many times a researcher’s experiments and tests involve various algorithms that require execution parameters such as user calibration, thresholds, iterations counts, etc. The parameters in ImageTclAR toolkit are defined at script level. Since the parameterization is implemented at the script level in the interpereted Tcl scripts, parameters can be adjusted quickly, in two ways by editing and re-executing the script or by adding Tk user interface components. The easy parameterization of ImageTclAR toolkit for rapid development of of tests and experiments is a direct result of using the Tcl scripting environment.

5.3 Component Glue

Glue code is needed for an application that is composed of loosely coupled components that have no knowledge of which other components are being used in the application. In many cases these components must notify other components when specific events occur, especially when components are used to interface with external devices, user interfaces, or to obtain data from other components in the system. There have been systems that used the extensible markup language (XML) as glue as Bauer’s component-based framework [7]. XML simply provides a powerful and flexible way of representing many different kinds of structured data by itself XML is static. XML has no operational or functional component. The glue code used for ImageTclAR toolkit is Tcl scripts, fragments of interpreted code, to facilitate component interaction with other components in the system by executing the appropriate script to the language interpreter. Tcl has the powerful decision-making capabilities and procedure to abstraction of components

5.4 Facilitate Different Level of Development

Advanced users are researchers that update or replace various components of the ImageTclAR system for experimental purposes. These users typically take advantage of computer vision, image processing, advanced graphics, and animation techniques. In order to use these advanced techniques, this level of user must have the powerful decision making capability along with the speed of C++. The advanced user must have a full understanding of the ImageTclAR system so that they can make the needed changes to the system by adding or replacing components to implement these advanced techniques efficiently.

Development environments commonly are based on the API, wherein the system consists of a library that is linked to an application program. API's are not suitable for novice programmers, as they require the development of complete programming projects in order to use their capabilities and often have a significant learning curve. The use of scripting languages to provide glue and sequencing in applications has been quite popular, particularly in research systems. The OGI Speech Tools are an example of a mixed Tcl and C++ development environment for speech recognition research. These systems often allow for novice programmers at the script level, but complicate development of compiled components at the application and advanced user levels. A design element of ImageTclAR toolkit has been the attempt to facilitate both script-level development and compiled-code development with equal ease, rather than shifting all application development to the script level.

ImageTclAR toolkit is an environment, wherein software tools designed to decrease the complexity of many tasks supplement the API, both at the script and compiled levels. It relieves much of the load for both novice and advanced users with the provision of a suite of software tools. ImageTclAR toolkit has a build utility used to eliminate manual creation of project files, make files, or initialization modules for packages that are added onto the Tcl base system. The interactive component creation utility is used to create the files as a ready to install and compile shell of a new command, data type, or module that the user edits to add functionality. The graph editor is used to graphically put together the sequencing of an ImageTclAR toolkit application. The script-level library is used to encapsulate the commonly used things. The calibration subsystem is used for calibration of an AR system and is routinely done in the initializing of an AR system.

6. Application Scenarios

ImageTclAR toolkit is being used for various AR earch projects in Michigan State University. Users of the toolkit have different background of expertise and level of programming skill.

6.1 Behavioral Science

A series of studies about depth perception, human spatial cognitive capability in AR environment [8, 9], and human factor in optical see-through HMD calibration methods [10] have been conducted. Experimenter designed a series of experiments using the ImageTclAR toolkit to test out different hypotheses of human cognition in an AR environment. Presentation of stimulus materials, audio instructions for subjects, experimental procedure sequencing, and data collection of the experiment are automated so that the experimenter does not need to hand code the experimental results. Experimenter has limited programming experience, and all the programming tasks are being done on the scripting level without dealing with any C++, OpenGL or device driver code. The typical length of code being used is about 100 line of code, and is very manageable for novice developers.

6.2 AR Application Prototyping

A few prototype applications were developed using the ImageTclAR toolkit, including applications in instructional media for object assembly [11, 12], gaming, and other task specific information display. Developers of these applications need to write specific lower level C++ code (e.g. OpenGL, or other device specific code), and they can still utilize the mixed programming model to greatly simplify procedural and glue elements of the program. Testing and manipulations of user interface components and other parameters are available at the scripting level that simplified the development and testing process.

6.3 AR User Interface Prototyping

Interface designers and researchers in human-computer interaction are using the ImageTclAR toolkit for a project about developing general usage AR computer interface [13, 14]. Some special purpose low-level components are developed such as two-hand tracking for bimanual interaction, body position and orientation tracking for body-stabilized information display, and other interaction specific code.

7. Summary

This paper gives an overview of the ImageTclAR toolkit, an AR development system based on Tcl/Tk and ImageTcl. ImageTclAR provides the system developer with a fast and effective prototyping environment in the software development process. The focus of the system is on system development in C++, and testing and user interface development in Tcl. Several applications of ImageTclAR in AR research and prototyping are also described.

8. Note

ImageTcl and ImageTclAR toolkits are distributed free for non-commercial or research uses. More information is available at "http://metlab.cse.msu.edu/imagetclar/".

9. Acknowledgement

The authors would like to acknowledge Michael Malinak, Eric Arnold, and Paul Middin for their participation in the development of the ImageTclAR toolkit project. This work was supported by the National Science Foundation Grants #00-8274, #02-22831.

10. Reference

1. Owen, C. *The ImageTcl Multimedia Algorithm Development System*, in *Fifth Annual Tcl/Tk Workshop*, 1997, Boston, MA.
2. Ousterhout, J.K., *Tcl and the Tk Toolkit*. 1994, Reading, MA.
3. Ahlers, K., Crampton, C., Greer, D., Rose, E., and Tuceryan, M., *Augmented Vision: A Technical Introduction to Grasp 1.2 System*. 1994. Technical Report: European Computer-Industry Research Center. ECRC-94-14-1994.
4. Billinghurst, M., Kato, H., and Poupyrev, I., *The Magic Book: Moving Seamlessly between Reality and Virtuality*. IEEE Computer Graphics and Applications, 2001: p. 2-4.
5. Kato, H., Billinghurst, M., and Poupyrev, I. *Virtual Object Manipulation on a Table-Top AR Environment*, in *IEEE and ACM International Symposium for Augmented Reality (ISAR 2000)*, 2000, Munich, Germany.
6. Tuceryan, M. and Navab, N. *Single Point Active Alignment Method (SPAAM) for Optical See-through HMD Calibration for AR*, in *IEEE and ACM International Symposium for Augmented Reality (ISAR 2000)*, 2000, Munich, Germany.
7. Bauer, M., Bruegge, B., Klinker, G., MacWilliams, A., Reicher, T., Stedan Riß, Christian Sandor, and Wagner, M. *Design of a Component-Based Augmented Reality Framework*, in *IEEE and ACM International Symposium on Augmented Reality (ISAR'01)*, 2001, New York, NY.
8. Owen, C. and Tang, A. *Visualizing Medical Imagery in Situ: Augmented Reality as X-Ray Vision*, in *10th International Conference on Human-Computer Interaction*, 2003, Crete, Greece.
9. Mou, W., Biocca, F., Owen, C., Tang, A., and Fan, X., *Frames of Reference in Mobile Augmented Reality Displays*. Human Factors: The Journal of the Human Factors and Ergonomics Society, submitted.
10. Tang, A., Zhou, J., and Owen, C. *Evaluation of Calibration Methods for Optical See-Through Head-Mounted Displays*, in *The Second IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR '03)*, 2003, Tokyo, Japan.
11. Tang, A., Owen, C., Biocca, F., and Mou, W. *Experimental Evaluation of Augmented Reality in Object Assembly Task*, in *IEEE and ACM International Symposium on Mixed and Augmented Reality*, 2002, Darmstadt, Germany.
12. Tang, A., Owen, C., Biocca, F., and Mou, W. *Comparative Effectiveness of Augmented Reality in Object Assembly*, in *ACM CHI '2003*, 2003, Fort Lauderdale, FL.
13. Mou, W., Biocca, F., Owen, C., and Tang, A., *Spatial Cognition and Spatialized Interface Design of Mobile Augmented Reality Systems*. Human Factors: The Journal of the Human Factors and Ergonomics Society, submitted.
14. Biocca, F., Tang, A., and Lamas, D. *Evolution of the mobile infosphere: Iterative design of a high-information bandwidth, mobile augmented reality interface.*, in *Euroimage 2001: International Conference on Augmented, Virtual Environments and 3D Imaging*, 2001, Mykonos, Greece.