FAAST: The Flexible Action and Articulated Skeleton Toolkit

Evan A. Suma[*] Belinda Lange[*] Albert "Skip" Rizzo[*] David M. Krum[*] Mark Bolas[*†]

[*]USC Institute for Creative Technologies [‡]USC School of Cinematic Arts

Abstract

The Flexible Action and Articulated Skeleton Toolkit (FAAST) is middleware to facilitate integration of full-body control with virtual reality applications and video games using OpenNI-compliant depth sensors (currently the PrimeSensor and the Microsoft Kinect). FAAST incorporates a VRPN server for streaming the user's skeleton joints over a network, which provides a convenient interface for custom virtual reality applications and games. This body pose information can be used for goals such as realistically puppeting a virtual avatar or controlling an on-screen mouse cursor. Additionally, the toolkit also provides a configurable input emulator that detects human actions and binds them to virtual mouse and keyboard commands, which are sent to the actively selected window. Thus, FAAST can enable natural interaction for existing off-the-shelf video games that were not explicitly developed to support input from motion sensors. The actions and input bindings are configurable at run-time, allowing the user to customize the controls and sensitivity to adjust for individual body types and preferences. In the future, we plan to substantially expand FAAST's action lexicon, provide support for recording and training custom gestures, and incorporate real-time head tracking using computer vision techniques.

Index Terms: H.5.1 [Information Interfaces and Presentation]: Multimedia Information Systems—Artificial, augmented, and virtual realities; I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Virtual reality; K.8.0 [Personal Computing]: General—Games

Keywords: depth-sensing cameras, gestures, video games, middleware

1 Introduction

Recent advances in video game technology have fueled a proliferation of low-cost commodity devices that can sense the user's motion. These range in capability from handheld controllers that can be used for gesture-based control, such as the Nintendo®Wiimote and the Playstation®Move, to cameras that use computer vision techniques to sense the user's body pose, such as the Playstation®Eye. In the past year, low-cost depth-sensing cameras have also become commercially available, including the widely publicized Microsoft Kinect, which have made it possible to sense the full-body pose for multiple users without the use of markers or handheld devices. The OpenNI™organization has emerged to promote standardization of these natural interaction devices, and has made available an open source framework for developers. To facilitate the rapid development of virtual reality applications using OpenNI-compliant devices (currently the PrimeSensor and the Kinect), as well as to incorporate motion-based control in existing off-the-shelf games, we have developed the Flexible Action and Articulated Skeleton Toolkit (FAAST). FAAST provides convenient access to pose and gesture information provided by the OpenNI framework, and enables customizable body-based control of PC applications and games using an input emulator that generates virtual mouse and keyboard events.

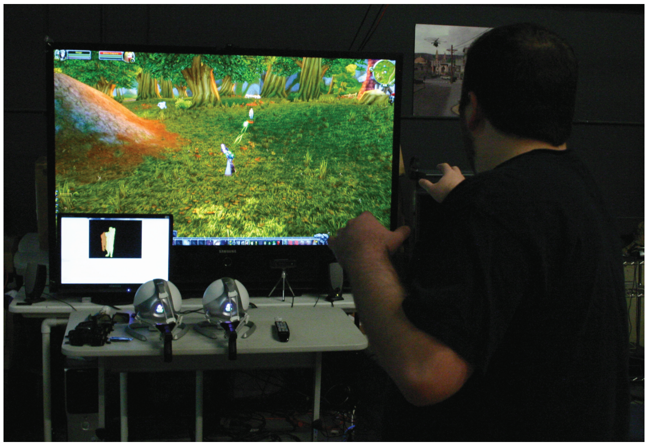

Figure 1: A user casting a spell using a "push" gesture in World of Warcraft, an off-the-shelf online video game that was not developed to support motion sensing devices. FAAST can be custom-configured to detect specific actions and bind them to virtual keyboard and mouse commands that are sent to the active window.

2 FAAST Overview

FAAST was initially developed to provide a convenient and accessible interface for the PrimeSensor™Reference Design, a USB plug-and-play depth-sensing camera developed by PrimeSense. This technology, based on infrared structured light to compute a depth image of the environment, was licensed to Microsoft for the Kinect. The OpenNI software is compatible with both of these sensors, and along with NITE middleware provided by PrimeSense, performs user identification, feature detection, and basic gesture recognition using the depth image from the sensor [2]. FAAST interfaces directly with OpenNI/NITE to access this information and performs additional high-level gesture recognition for generating events based on the user's actions.

FAAST considers two broad categories of information from the sensor: *actions* and *articulated skeletons*. Articulated skeletons consist of the positions and orientations for each joint in a human figure, and are useful for virtual reality and video game applications in allowing direct body-based control of a virtual avatar. FAAST retrieves these skeleton joints from the OpenNI drivers and transmits them to the end-user application using the Virtual Reality Peripheral Network (VRPN), a popular software package in the virtual reality community for interfacing with motion tracking hardware [4]. We built a custom VRPN server into the FAAST application that streams the skeletal information for each joint as a six degree-of-freedom tracker, allowing applications to interface with the sensor as they would any other motion tracking device. Figure 2 shows an example user puppeting a virtual wireframe avatar and a skinned virtual character rendered using an existing virtual reality

[*]e-mail: {suma, lange, arizzo, krum, bolas}@ict.usc.edu

IEEE Virtual Reality 2011
19-23 March, Singapore
978-1-4577-0037-8/11/$26.00 ©2011 IEEE

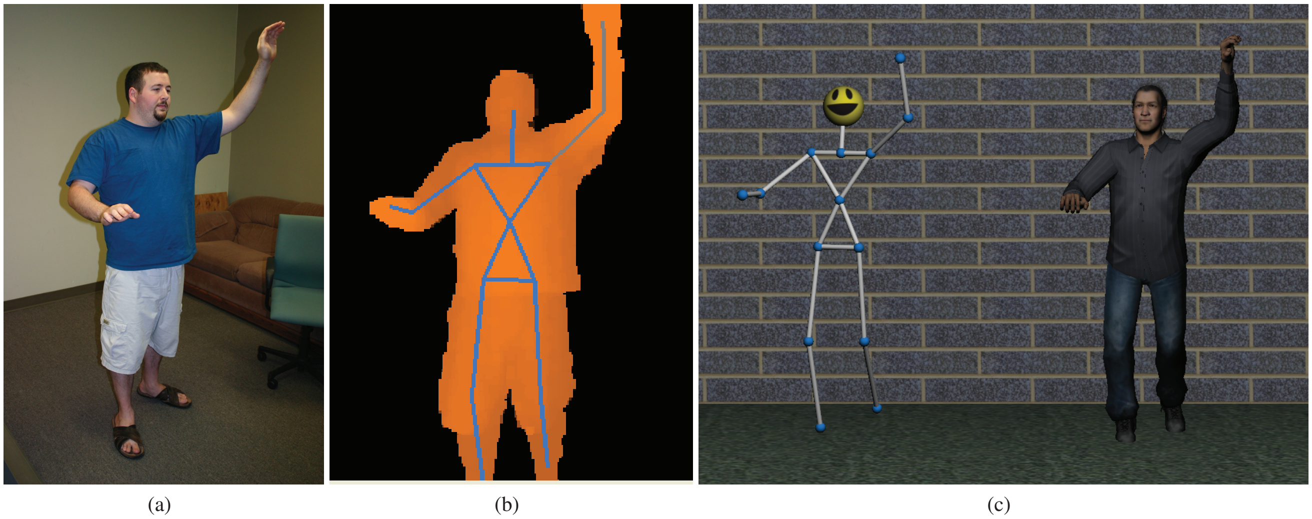

Figure 2: (a) An example user strikes a pose. (b) The PrimeSense NITE software is able to identify and segment the user from the background depth image, and fit an articulated skeleton to the resulting point cloud. (c) Using FAAST, the user is able to puppet a wireframe avatar and an animated virtual character in real-time.

engine with a VRPN client. To interface with existing applications and games, FAAST also can control the Windows mouse cursor by making it follow the user's hand joints with either absolute 1:1 motion over a customizable "active area" or relative motion with velocity determined by hand position.

In constrast to articulated skeleton data, actions are more complicated since they require inferring meaning from the user's pose and their movements over time. NITE provides some basic gesture recognition for certain atomic actions involving the user's hands, such as a push, swipe, circle, and wave. These actions appear well suited for simple 2D interfaces such as media center control, but 3D selection, manipulation, and locomotion requires a much richer gesture set. Thus, we calculate several other relevant actions based on the user's body pose, including lean (forward, backwards, left, and right) and twist (left and right). Within FAAST, the user can adjust the angular threshold for detecting these actions to make them more or less sensitive. Actions were also developed for arm or leg movements in different directions, jumping, and walking-in-place. Any of the actions computed in FAAST, as well as the basic actions from NITE, can be bound to virtual keyboard or mouse commands that are sent to the actively selected window. Thus, FAAST enables these custom sensors to provide input to arbitrary applications, including off-the-shelf games that were not developed to support motion sensing input (see Figure 1). Furthermore, users can customize the bindings and sensitivity for these actions at run-time, thus providing flexible input that can easily be adjusted according to the individual user's body type and preferences.

3 Future Work

FAAST provides an extensible framework with many possibilities for enhancement in the future. First, we plan on substantially expanding the action recognition capabilities with more complicated or higher level actions, which will require the application of machine learning techniques. Furthermore, we also intend to provide an interface that will allow users to record custom actions and train the toolkit to recognize them. We will also expand the capabilities of the input event emulator, for example, by adding virtual joystick devices. VRPN also provides a button wrapper that will be useful for sending action events to networked VR applications that are already connected to the server to read articulated skeleton data. However, it may prove convenient to also provide an input emulator server and client architecture, so that full-body motion can be leveraged in off-the-shelf games even if the sensor is connected to a different computer on the network.

OpenNI and NITE offer a great deal of built-in capabilities, and it was not our intention replicate the functionality that was already available using this natural interface software framework. While the PrimeSensor or Kinect hardware is surprisingly robust in providing the position and orientation of each joint in the skeleton, there are a several areas which we would like to improve. While the sensor is capable of localizing the position of the head, it cannot compute the head orientation based on the infrared coded light. Therefore, we intend to incorporate real-time head tracking by leveraging the sensor's built-in RGB camera along with the corresponding depth image, using an existing library such as Watson, which uses an adaptive view-based appearance model [3]. Additionally, the sensor cannot also reliably estimate the twist of the user's arm, and so this may also be improved by applying computer vision techniques to the RGB image. We are also interested in integrating simultaneous localization and mapping (SLAM) into FAAST, which could be used to provide inside-looking-out tracking reported over the integrated VRPN server, or for 3D model scanning and reconstruction. In the long term, we may also add support for additional motion sensing controllers such as the Nintendo®Wiimote, Playstation®Move, or the upcoming Sixense Truemotion®.

FAAST is currently available as a free software download, and an open-source version will be released to the community in early 2011 [1].

References

[1] http://projects.ict.usc.edu/mxr/faast/.

[2] http://www.primesense.com.

[3] L.-P. Morency, A. Rahimi, and T. Darrell. Adaptive view-based appearance model. In *IEEE Computer Vision and Pattern Recognition*, pages 803–810, 2003.

[4] R. M. Taylor, T. C. Hudson, A. Seeger, H. Weber, J. Juliano, and A. T. Helser. VRPN: a device-independent, network-transparent VR peripheral system. In *ACM Virtual Reality Software & Technology*, pages 55–61, 2001.