Incorporating Speech Recognition into a Natural User Interface

Author: Nicholas Chapa
Augmented/Virtual Reality Lab, Kennedy Space Center, Merritt Island, FL

Abstract: The Augmented/ Virtual Reality (AVR) Lab has been working to study the applicability of recent virtual and augmented reality hardware and software to KSC operations. This includes the Oculus Rift, HTC Vive, Microsoft HoloLens, and Unity game engine. My project in this lab is to integrate voice recognition and voice commands into an easy to modify system that can be added to an existing portion of a Natural User Interface (NUI). A NUI is an intuitive and simple to use interface incorporating visual, touch, and speech recognition. The inclusion of speech recognition capability will allow users to perform actions or make inquiries using only their voice. The simplicity of needing only to speak to control an on-screen object or enact some digital action means that any user can quickly become accustomed to using this system.

Multiple programs were tested for use in a speech command and recognition system. Sphinx4 translates speech to text using a Hidden Markov Model (HMM) based Language Model, an Acoustic Model, and a word Dictionary running on Java. PocketSphinx had similar functionality to Sphinx4 but instead ran on C. However, neither of these programs were ideal as building a Java or C wrapper slowed performance. The most ideal speech recognition system tested was the Unity Engine Grammar Recognizer. A Context Free Grammar (CFG) structure is written in an XML file to specify the structure of phrases and words that will be recognized by Unity Grammar Recognizer. Using Speech Recognition Grammar Specification (SRGS) 1.0 makes modifying the recognized combinations of words and phrases very simple and quick to do. With SRGS 1.0, semantic information can also be added to the XML file, which allows for even more control over how spoken words and phrases are interpreted by Unity. Additionally, using a CFG with SRGS 1.0 produces a Finite State Machine (FSM) functionality limiting the potential for incorrectly heard words or phrases.

The purpose of my project was to investigate options for a Speech Recognition System. To that end I attempted to integrate Sphinx4 into a user interface. Sphinx4 had great accuracy and is the only free program able to perform offline speech dictation. However it had a limited dictionary of words that could be recognized, single syllable words were almost impossible for it to hear, and since it ran on Java it could not be integrated into the Unity based NUI. PocketSphinx ran much faster than Sphinx4 which would've made it ideal as a plugin to the Unity NUI, unfortunately creating a C# wrapper for the C code made the program unusable with Unity due to the wrapper slowing code execution and class files becoming unreachable. Unity Grammar Recognizer is the ideal speech recognition interface, it is flexible in recognizing multiple variations of the same command. It is also the most accurate program in recognizing speech due to using an XML grammar to specify speech structure instead of relying solely on a

Dictionary and Language model. The Unity Grammar Recognizer will be used with the NUI for these reasons as well as being written in C# which further simplifies the incorporation.