Poster: Say It To See It: A Speech Based Immersive Model Retrieval System

Ross Tredinnick*   Kevin Ponto†

University of Wisconsin - Madison

Figure 1: Demonstration: The user speaks the term "pizza" and is presented with a visual interface of results (A). The user selects an item and a placeholder box is created that the user can interact with (B). Upon model load, the placeholder box is replaced with the model (C).

Abstract

Virtual spaces have proven to be a valuable means to visualize and inspect 3D environments. Unfortunately, adding objects to 3D scenes while inside of an immersive environment is often difficult as the means used to acquire models from repositories are built for standard computer interfaces and are generally not available during a users session. We develop a novel interface for the insertion of models into a virtual scene through the use of voice, 3D visuals, and a 3D input device. Our interface seamlessly communicates with external model repositories (Trimble 3D Warehouse) enabling models to be acquired and inserted in the scene during a user's virtual session. We see the benefits of our pipeline and interface in the fields of design, architecture, and simulation.

Index Terms: H.5.1 [Information Presentation]: Multimedia Information Systems—Artificial, augmented, and virtual realities;

1 Introduction

This paper introduces an interface and automated pipeline for preparing and exporting content from a 3D design application. The pipeline is combined with a solution of searching for models from an external repository, in our case the Trimble 3D Warehouse, from within a virtual environment (VE). A user can search for a model using speech and choose a model they wish to download to the environment from a GUI showing the search results. The chosen models are downloaded, converted, and automatically loaded into the immersive VE application. This interface could provide benefits for design, architecture, and simulation, particularly at the conceptual stage.

The goal of our work is to develop a method for users to create scenes while entirely inside of a VE. 3D environments are much more difficult to interact with compared to standard 2D user interfaces [5]. Unfortunately, standard methods for interfacing with virtual environments do not work well for input technologies. Takala et al. provide a survey of recent interfaces for 3D environments [9]. Speech has been identified as an intuitive interface for Virtual Environments [4]. Speech can often be used as an augmented interface to direct manipulation [2, 7]. Kim provides a survey of speech interfaces for 3D applications [6].

Model repositories and search engines for them have been available since the early 2000's with Min et al. providing a good listing of several early 3D model databases [8]. Our system utilizes the Trimble 3D Warehouse (formerly the Google Warehouse) inside of SketchUp [3]. Boeykens and Bogani describe the difficulties in using model repositories [1]. The design of our system takes these difficulties into consideration.

2 Method

Our system creates a generalized pipeline to obtain live 3D models from within an immersive display application (Figure 2). The VE application runs in a C6 CAVE. The system utilizes an Intersense IS900 tracking system and a tracked MicroTrax 3D wand input device to interact with the GUI, to manipulate objects and to navigate a scene.

As our goal was to create an input method that did not require a keyboard-like device, we chose to implement our system utilizing speech as an interface. The Microsoft Speech SDK 11 was selected for its ease of use. To prevent unwanted processing, the user is first required to speak the trigger phrase, "model search", to activate the system. After receiving audio confirmation, the user verbally provides a search query. In order to aid the speech recognition software, the grammar is limited to approximately 2,000 common nouns. The result of the speech recognition is forwarded to the web processing application. The phrase "search results" can be used to activate the previous search query.

The passed in spoken term is used to search an online 3D model database. Our system communicates with the Trimble 3D Warehouse as it provides an easy source to acquire both models and thumbnail images. Our system utilizes a program built on top of Ruby to parse HTML, extract model download links and retrieve thumbnails. This approach was undertaken to separate the model downloading and exporting application from the application rendering the VE. Files containing thumbnails and links are saved to a shared file system location accessible from both the PC performing the search as well as our immersive display system cluster.

Upon parsing the links and thumbnails, the VE application creates a GUI for choosing a desired model from the search results. The adapter of this pipeline can customize the appearance and style of the GUI for their own needs depending on their VR display system (i.e. HMD or CAVE, etc.), but at a minimum should present the search results to the user in some way. Our framework adapts a tiled wall panel GUI that shows thumbnail search results across the front wall of the CAVE. The GUI spans the size of the CAVE wall

*e-mail: rdtredinnick@wisc.edu
†e-mail: kbponto@wisc.edu

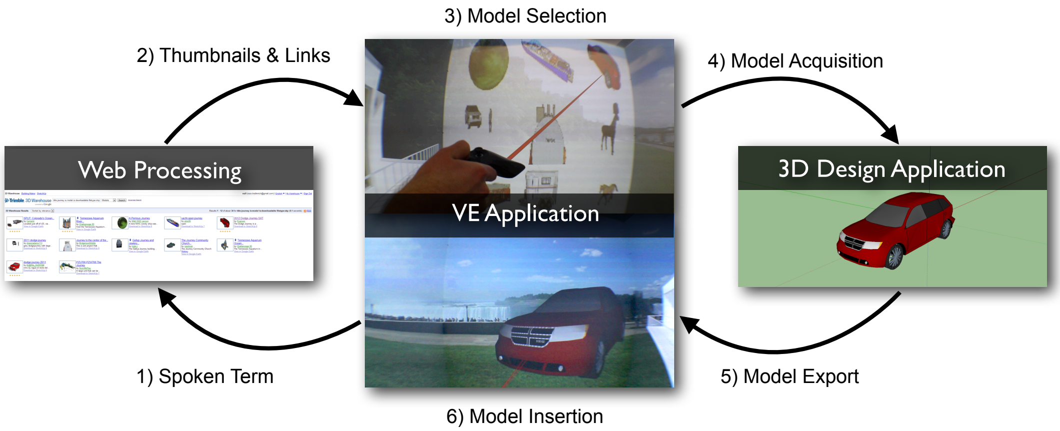

Figure 2: The pipeline for our system.

and shows an application specified number of search results, set to 9 by default. Future work will allow the user to browse additional search results. The tiled display renders with some transparency so that the user can still partially see the environment behind the GUI. The application associates a download link with each thumbnail so that upon choosing a thumbnail, the download link can be handed off to a 3D design application for model acquisition.

As this loading is not instantaneous, a cube textured with the chosen thumbnail drops out of the location of where the user selected the thumbnail on the GUI. The cube can be manipulated, and positioned to a desired location within the VE with the wand. This cube acts as a placeholder to represent the location of where the 3D model will appear after it has finished downloading.

Our pipeline adapts SketchUp as the 3D design application of choice due to its ability to download models directly from the Trimble 3D warehouse into a running instance of the application through a Ruby API. A Ruby script reads the link of the chosen thumbnail and downloads the 3D model from the 3D warehouse. Upon finishing the download, the script exports the model using an open source SketchUp to OSG (OpenSceneGraph) exporter modified to fit our particular needs.

Upon finishing the export, the 3D design application must communicate to the VE application that an export is finished and a model is ready to load. In our pipeline, SketchUp writes one file per cluster machine that contains the location of the exported model to a file system location accessible by the cluster. Our VE application checks for the file and loads up the newly exported OSG file and positions the model in the scene in the location of the previously created placeholder cube.

3 Evaluation

The system, while still in early stages, has produced many rich interactions among lab employees and students. To assess the performance of the system, we tested the timings of bringing in 20 random models through the entire pipeline. All models were acquired using a standard gigabit network interface. The number of triangles in these 20 models ranged from 48 to 111,769 and file sizes ranged from 83 KB to 6.1 MB. The average export time for these 20 models was 12.9455 seconds and the average download time was 2.83 seconds. On average these 20 models appeared in the scene in under 20 seconds. The bottleneck in our pipeline occurs at the export stage where converting between different file formats can be a time consuming process. Methods to alleviate these issues are discussed below.

One issue when our approach was first undertaken was that the scale of objects was not correctly defined. While the software in the VE expected models to be in meters there was no guarantee that models were correctly generated at this scale when uploaded to the 3D model repository. To alleviate these problems the model is first scaled so that its longest dimension matches the length of the same dimension of the placeholder cube. Interactively, the user can uniformly scale the object through button presses. Additionally, the user can toggle between the model's original scale as it was when downloaded from the 3D warehouse and its user defined scale.

A major issue present in our current setup comes from speech recognition errors. Often these errors require a user to repeat a request for the system to understand. The system helps mitigate this problem by giving the user auditory feedback when a command is recognized. We plan to try different audio acquisition methods as well as improving the speech recognition capabilities moving forward.

As mentioned above, the bottleneck in our system appears during the exporting step. The main cause of this comes from the model conversion process. The SketchUp to OSG exporter currently exports the SketchUp file to the COLLADA format before converting to OSG. A future adjustment to the export process of the system will seek to directly export form SketchUp's format to OSG by utilizing the SketchUp C++ SDK Future work will also focus on adding the ability to search other model repositories and supporting other file formats. We plan to create a local repository for models acquired through our system. This local repository would not only expedite the loading process, but would also act a filter by showing the most popular models first by showing local results followed by results from an online 3D model database.

References

[1] S. Boeykens and E. Bogani. Metadata for 3d models. how to search in 3d model repositories? *ICERI 2008 Proceedings*, 2008.

[2] M. Ceňak and A. Sannier. Command speech interface to virtual reality applications. 2002.

[3] A. Chopra. *Google Sketchup 7 for Dummies*. For Dummies, 2009.

[4] S. Everett, K. Wauchope, M. Pérez, et al. A natural language interface for virtual reality systems. Technical report, Citeseer, 1994.

[5] K. P. Herndon, A. van Dam, and M. Gleicher. The challenges of 3d interaction: a chi '94 workshop. *SIGCHI Bull.*, 26(4):36–43, Oct. 1994.

[6] S. Kim. A literature survey on the design of speech interface to 3d applications. *Relation*, 10(1.90):1685, 2008.

[7] S. McGlashan and T. Axling. A speech interface to virtual environments. *Swedish Institute of Computer Science*, 2007.

[8] P. Min, J. A. Halderman, M. Kazhdan, and T. A. Funkhouser. Early experiences with a 3d model search engine. In *Proceedings of the eighth international conference on 3D Web technology*, Web3D '03, pages 7–ff, New York, NY, USA, 2003. ACM.

[9] T. Takala, P. Rauhamaa, and T. Takala. Survey of 3dui applications and development challenges. In *3D User Interfaces (3DUI), 2012 IEEE Symposium on*, pages 89 –96, march 2012.