Chapter VI

Representing and Processing Screen Space in Augmented Reality

Blaine Bell, Columbia University, USA

Steven Feiner, Columbia University, USA

Abstract

View management involves computing what a user sees within a 3D environment when it is projected onto a display screen. Doing this interactively to create effectively laid out user interfaces for augmented reality, virtual reality, or any other kind of 3D user interface requires representing and processing screen space. In this chapter, we describe how to compute a 2D screen-space representation that corresponds to the visible portions of the projections of 3D objects on the screen. We describe in detail two visible-surface determination algorithms that are used to generate these representations: one based on a binary space partitioning tree, and one based on a hardware accelerated z-buffer and object buffer. We compare the performance and accuracy tradeoffs of these algorithms, and present examples of how to use our representation to satisfy visibility constraints that avoid unwanted occlusions, making it possible to label and annotate objects in 3D environments.

Copyright © 2007, Idea Group Inc. Copying or distributing in print or electronic forms without written permission of Idea Group Inc. is prohibited.
Introduction

Augmented reality (AR) makes it possible to concurrently visualize both the real world and overlaid virtual information. While designing a conventional user interface requires deciding what information should be presented, and how and where it should be shown, designing an AR user interface thus requires addressing another crucial problem: the superimposed virtual information can occlude things that we would otherwise see in the real world. Therefore, to avoid obscuring important objects, it is necessary to determine the position and size of virtual objects within the user interface relative to what is seen in the real world.

*View management* (Bell, Feiner, & Höllerer, 2001) refers to the layout decisions that determine spatial relationships among objects in a 3D user interface. For an application to perform view management, it must compute the visibility of objects of interest in a 3D environment, as seen from a selected 3D viewpoint, taking into account visibility constraints. It also must represent and process the visibility information once it has been projected onto the 2D screen space representing the user’s view.

Figure 1 shows the order in which computations are done in our view management pipeline (Bell, Feiner, & Höllerer, 2005). To precisely maintain the specified visual constraints, this pipeline should get executed for each frame rendered. However, since some of these computations can take a considerable amount of time to compute, asynchronous execution of the pipeline, as in the decoupled simulation model (Shaw, Green, Liang, & Sun, 1993), can preserve interactive rendering frame rates at the expense of view management accuracy. (These tradeoffs might be preferable depending on the user’s preference or task.)

In this chapter, we focus on how to compute the visibility information on which view management is based, as performed in step 3 of Figure 1. This occurs after the viewing specification has been determined (step 2), and the 3D projection has been defined. The resulting visibility information will be used further down the pipeline to satisfy view management constraints, such as avoiding occlusion or determining placements for annotations that are close to the visible portions of associated objects.

Figure 2 demonstrates how we apply view management to a collaborative augmented reality environment. It is photographed through an optical see-through head-worn display, in which
Artificial Neural Network Modelling for Waste: Gas and Wastewater Treatment Applications
www.igi-global.com/chapter/artificial-neural-network-modelling-waste/53308?camid=4v1a