REVIEW: A Real-time Virtual Walkthrough System

L. Shou C. H. Chionh Z. Huang Y. Ruan K.-L. Tan

Department of Computer Science
National University of Singapore
Singapore 117543
(65) 8742862
{shoulida, chionhch, huangzy, ruanyixi, tankl }@comp.nus.edu.sg

1. Real-time Virtual Walkthrough

Data representing virtual environments (VEs) are getting increasingly large in order to better simulate real scenes. This poses interesting challenges to organize, store, and render the data for interactive navigation in VEs, or walkthrough. A large VE usually consists of thousands of 3D objects, each of which can be represented by hundreds of polygons, and may take thousands of megabytes of storage space. The amount of data is so large that it is impossible to store all of them in the main memory. Even for memory resident models, the graphics pipeline can become a bottleneck quickly with a large amount of data and slow down the rendering to an unacceptable frame rate for the walkthrough.

In this demo, we will show the effectiveness of several optimization techniques to address the problems. (1) The data in secondary storage are organized based on their spatial location in an R-tree index and only relevant data are retrieved from the database using a novel R-tree search algorithm. (2) Prefetching of the R-tree index nodes and data is implemented based on a real-time prediction algorithm. (3) A cache replacement policy is used based on the hierarchical structure of the R-tree index and the walkthrough semantics.

2. The REVIEW System

The REVIEW (REal-time VIrtual Environment Walkthrough) system, developed at the Department of Computer Science, National University of Singapore, is a walkthrough system for large virtual environments. REVIEW implemented a novel R-tree search algorithm, the Complement Search algorithm, which allows querying R-tree using concave-shaped regions. This algorithm retrieves objects that have not been stored in the main memory, reduces disk I/Os in the database and minimizes the size of query results. As a result, the rendering frame rate of the walkthrough has been improved. REVIEW also exploited a real-time prefetching algorithm, which predicts the future states of the user based on his/her walkthrough history. During walkthrough, a history list of view frustum is maintained in a data prefetching module. The module is deployed to learn from the recent states and the history list to predict the future state of the view frustum. The prefetching scheme is implemented to predict and retrieve the data, which are the 3D objects likely to appear in the view frustum of the future frames, into memory in advance. For a very large scene whose R-tree index cannot completely fit into memory, we cache it partly using a distance-priority-LRU cache replacement policy. Index nodes at higher levels are initially assigned higher priorities, which may be dynamically adjusted. For example, a higher-level node's priority value will be reduced if its bounding box is distant from the current viewpoint in the walkthrough. In this way, objects indexed by such nodes are less likely to be accessed in the near future.

3. The Demo

In this demo, we shall illustrate the effectiveness of the proposed system in the following ways:

1. Spatial indexing based complement search. To demonstrate the effectiveness of this technique, we will examine two different systems: a traditional approach that implements the standard box search and an optimized approach that implements complement search on the R-tree. The latter approach generates higher frame rate and better visual effect than the previous one.
2. Prefetching of R-tree nodes and object data. The history of the walkthrough is used by the system to prefetch future data from disk. We will compare the systems with and without prefetching enabled. In most cases, the system with prefetching performs better. The improvement depends on the prediction model and the walkthrough semantics.
3. Caching of R-tree index nodes. A novel cache replacement policy, the distance-priority-LRU policy, and the traditional LRU policy will also be compared. The system with the distance-priority-LRU policy performs better in the average search time than that with the LRU scheme.

The REVIEW system incorporating these techniques is implemented on a SGI Octane workstation, with 400MB memory. In one example of walkthrough a virtual city, a dataset of 900 thousand 3D objects is used. We will show that the REVIEW system can achieve real-time walkthrough and has better performance.

ACM SIGMOD 2001 May 21-24, Santa Barbara, California USA
Copyright 2001 ACM 1-58113-332-4/01/05…$5.00