Algorithms for Multiresolution Databases*

Robert L. Read, Donald S. Fussell, and Avi Silberschatz
Department of Computer Sciences,
University of Texas,
Austin, TX 78712-1188
Email: {read, fussell, avi}@cs.utexas.edu

February 18, 1994

Abstract

The use of data at different levels of resolution or *multiresolution* is a common technique for improving performance of multimedia and scientific database applications, but is not yet systematically supported by database management systems. The *sandbag* construct, which has been previously suggested as a basis for a data model of multimedia and scientific database, is a generalization of histograms that can model diverse types of data at different levels of resolutions or qualities. A sandbag can hold multimedia data such as motion video and represent incremental improvement in that data. It can be used conveniently to communicate multiresolution information between a standard DBMS and an application by adding a software layer between the DBMS and application. A mechanism for incrementally constructing sandbags of varying descriptiveness from individual facts is presented. A restriction is described that allows sandbags to be efficiently computed. A data structure and algorithms that implement fully incremental construction of sandbags under this restriction is provided. These algorithms have low-order polynomial worst case complexity and may be expected to be efficient in a practical implementation.

1 Introduction

Scientific and multimedia data such as images, sounds and motion video take up so much space that retrieving them and communicating them to a user or application fast enough is a serious problem. An effective way to deal with this problem is to trade a slight decrease in perceived quality for a large decrease in data volume, and hence a large increase in performance. The JPEG [1] and MPEG [2] protocols are a prime example of schemes which are based on this idea, employing a lossy compression scheme.

Current day database management systems (DBMSs) do not directly support this type of tradeoff. We believe that DBMSs and data retrieval systems that systematically support this tradeoff for general data types will be an important part of distributed heterogeneous computing environments of the future. We call such systems *multiresolution* systems.

A fundamental problem in constructing and using such multiresolution systems is organizing the transfer of data at different levels of compression, quality, or resolution from the data retrieval system to the application. A read-only query in a relational or object-oriented setting is relatively simple because it can be thought of as a question that has only one right answer, and the format of that

*This material is based in part upon work supported by the Texas Advanced Technology Program under Grant No. ATP-024, the National Science Foundation under Grant No. IRI-9312003, and grants from the IBM and Hewlett-Packard corporations.

answer is completely determined by the question. The point of a multiresolution data retrieval system, however, is to allow the system to manage resolution for the user by deciding the best resolution to return, subject to constraints that the user defines, such as real-time or other performance constraints. Therefore the application and DBMS must agree on a flexible means of transferring data at different resolutions to the application.

Existing mechanisms such as relations and classes do not solve this problem of organization well, for three reasons. First, these mechanisms do not support a useful notion of incremental improvement. Incremental improvement up to a deadline is an approach to meeting real-time constraints [3], which are ubiquitous in multimedia applications. Its importance is indicated by the fact that it is included directly in the JPEG lossy compression protocol. Second, relations and classes can only support element-by-element multiresolution. For example, a relation of highly-compressed pictures could be substituted for a relation of perfect pictures in some database applications. However, relations could not represent a compression of a large number of integers, such as is occasionally done with histograms or other statistical methods. Third, the possibility of the same query resulting in different answers on exactly the same database is technically a violation of most data models. A long-term solution to the problem of representing multiresolution data must provide clean semantics as well as conveniently and efficiently organizing the transfer of diverse types of multiresolution data between the database and the application.

This paper provides such a solution. The *sandbag* construct [4] is a generalization of the histogram construct. It models multiresolution sets, relations, or classes of multimedia objects at various resolutions, remains consistent while being incrementally improved, and supports a limited but convenient form of statistical approximation. A multiresolution data model based on the sandbag has previously been suggested [4]. Fully general sandbags require approximate solutions. However, this paper focuses on a restricted case that can be made to apply in most database situations that allows the sandbag to be implemented efficiently. We have implemented algorithms for this special case and approximate algorithms for the more general case in about 6000 lines of C++, but we have not yet tested them extensively enough to report performance data [5]. Other approaches to representing multiresolution data inspired our develop of the sandbag [6, 7], and may also be useful.

The sandbag is inherently interesting as a means of representing incomplete information about sets. It may be very useful for computing multiresolution answers to queries. However, the issues of constructing a multiresolution DBMS, which are mostly open, are beyond the scope of this paper, which focuses on the efficient construction and convenient use of sandbags. Although we hope this technology will be part of a comprehensive multiresolution system eventually, it could be used immediately to organize an *ad hoc* use of multiresolution by adding a relatively simple software layer between a data retrieval system and an application.

To explain the algorithms and concepts we introduce in this paper, we use a running example application. Imagine a program implementing a map of Earth that a user can pan across. When a user comes to a particular region containing a point representing a distinctive feature such as an animal species or a restaurant, a short motion video of that feature, or a still photograph, or a sound clip can be presented to the user. If a given region is crowded, the application program must decide how to present this data to the user. For instance, if 16 photographs of the lemurs of Madagascar are available, the application must decide whether to place all of them on the screen, perhaps at reduced resolution, or to present each in full glory one at a time. If implemented on a portable networked via wireless telephony with a sufficiently detailed database, this would constitute a universal field guide and/or travel guide that, for instance, gives sample bird calls and movies of birds flying in addition to text descriptions. The data structure and algorithms presented in this paper could directly support this application, as well as a wide variety of other applications.

In Section 2 we give a more formal introduction to the sandbag and describe the condition under