V. S. SUBRAHMANIAN
SUSHIL JAJODIA (Eds.)

Multimedia Database Systems

Issues and Research Directions

多媒体数据库系统

Springer-Verlag
世界图书出版公司

V.S. Subrahmanian
Sushil Jajodia (Eds.)

Multimedia Database Systems

Issues and Research Directions

With 104 Figures and 9 Tables

Prof. V.S. Subrahmanian
University of Maryland
Computer Science Department
College Park, MD 20742
USA

Prof. Sushil Jajodia
George Mason University
Dept. of Information and Software
Systems Engineering
Fairfax, VA 22030
USA

Catalogue in Publication-Data applied for

ISBN 3-540-58710-1 Springer-Verlag Berlin Heidelberg New York

Die Deutsche Bibliothek - CIP-Einheitsaufnahme
Subrahmanian, V.S.: Multimedia database systems: issues and research directions/ V.S. Subrahmanian, Sushil Jajodia;. - Berlin; Heidelberg; New York; Barcelona; Budapest; Hong Kong; London; Mailand; Paris; Santa Clara; Singapore; Tokyo: Springer, 1996 (Artificial Intelligence) ISBN 3-540-58710-1 NE: Subrahmanian, V.S:

Foreword

With the rapid growth in the use of computers to manipulate, process, and reason about multimedia data, the problem of how to store and retrieve such data is becoming increasingly important. Thus, although the field of multimedia database systems is only about 5 years old, it is rapidly becoming a focus for much excitement and research effort.

Multimedia database systems are intended to provide unified frameworks for requesting and integrating information in a wide variety of formats, such as audio and video data, document data, and image data. Such data often have special storage requirements that are closely coupled to the various kinds of devices that are used for recording and presenting the data, and for each form of data there are often multiple representations and multiple standards – all of which make the database integration task quite complex. Some of the problems include:

– what a multimedia database query means
– what kinds of languages to use for posing queries
– how to develop compilers for such languages
– how to develop indexing structures for storing media on ancillary devices
– data compression techniques
– how to present and author presentations based on user queries.

Although approaches are being developed for a number of these problems, they have often been ad hoc in nature, and there is a need to provide a principled theoretical foundation. To address that need, this book brings together a number of respected authors who are developing principled approaches to one or more aspects of the problems described above. It is the first book I know of that does so.

The editors of this book are eminently qualified for such a task. Sushil Jajodia is respected for his work on distributed databases, distributed heterogeneous databases, and database indexing. V. S. Subrahmanian is well known for his work on nonmonotonic reasoning, deductive databases and heterogeneous databases – and also on several different media systems: MACS (Media Abstraction Creation System), and AVIS (Advanced Video Information System), and FIST (Face Information System, currently under development). It has been a pleasure working with them, and I am pleased to have been able to facilitate in some small way the publication of this book.

Preface

With the advent of the information superhighway, a vast amount of data is curently available on the Internet. The concurrent advances in the areas of image, video, and audio capture, and the spectacular explosion of CD-ROM technology has led to a wide array of non-traditional forms of data being available across the network as well. Image data, video data, audio data, all perhaps stored in multiple, heterogeneous formats, traditionally form the "core" of what is known today as multimedia data.

Despite the proliferation of such forms of media data, as well as the proliferation of a number of commercially available tools to manipulate this data, relatively little work has been done on the principles of multimedia information systems. What common characteristics do all these different media-types have in common? Can these characteristics be exploited so as to provide a "common core" skeleton that can be used as a platform on which other multimedia applications can be built? If so, how can this be accomplished? These, and other questions arise in the context of such multimedia systems.

In this book, we bring together a collection of papers that address each of these questions, as well as a number of other related questions.

The first paper, by Marcus and Subrahmanian, provides a basic theoretical foundation for multimedia information systems that is independent of any given application. The authors identify core characteristics common to a variety of media sources. They then show that these core characteristics can be used to build indexing structures and query languages for media data. They argue that query processing can be used as a way of specifying media presentations.

The paper by Gudivada et al. studies a specific kind of multimedia information system – those dealing only with image data. The authors describe various kinds of operations inherent in such systems (e.g. retrieving objects based on shape similarity). They then provide a unified framework, called the AIR model, the treats all these different operations in a unified manner.

The paper by Arya et al. describes the design and implementation of the QBISM system for storing and manipulating medical images. In contrast to the paper of Gudivada et al., in this paper, the authors study issues of logical database design by including two special data types – `VOLUME` and `REGION` to represent spatial information.

In the paper by Sistla and Yu, the authors develop techniques for similarity based retrieval of pictures. Their paper is similar in spirit to that of Gudivada et al. – the difference is that whereas Gudivada et al. attempt to develop a unified data model, Sistla and Yu formalize the process of inexact matching between images and study the mathematical properties resulting from such a formalization.

The paper by Aref et al. studies a unique kind of multimedia data, viz. handwritten data. The authors have developed framework called *Ink* in which a set of handwritten notes may be represented, and queried. The authors describe their representation, their matching/querying algorithms, their implemented system, and the results of experiments based on their system.

In the same spirit as the papers by Gudivada et al. and Sistla and Yu, the issue of retrieval by similarity is studied by Jagadish. However, here, Jagadish develops algorithms to index databases that require retrievals by similarity. He does this by mapping an object (being searched for) as well as the corpus of objects (the database) into a proximity space – two objects are similar if they are near each other in this proximity space.

Belussi et al.'s paper addresses a slightly different query – in geographic information systems, users often want to ask queries of the form: "Find all objects that are as close to (resp. as far from) object O as possible". The authors develop ways of storing GIS data that make the execution of such queries very efficient. They develop a system called Snapshot that they have implemented.

The paper by Ghandeharizadeh addresses a slightly different problem. Once a query has been computed, and we know which video objects must be retrieved and presented to the user, we are still faced with the problem of actually doing so. This issue is further complicated by the fact that video-data must be retrieved from its storage device at a specific rate – if not, the system will exhibit "jitter" or "hiccups". Ghandeharizadeh studies how to present video objects without hiccups.

The paper by Ozden et al. has similar goals to that of Ghandeharizadeh – they too are interested in the storage and retrieval of continuous media data. They develop data structures and algorithms for continuous retrieval of video-data from disk, reducing latency time significantly. They develop algorithms for implementing, in digital disk-based systems, standard analog operations like fast-forward, rewind, etc.

The paper by Marcus revisits the paper by Marcus and Subrahmanian and shows that the query paradigm developed there – which uses a fragment of predicate logic – can just as well be expressed in SQL.

Cutler and Candan study different multimedia authoring systems available on the market, evaluating the pros and cons of each.

Finally, Kashyap et al. develop ideas on the storage of metadata for multimedia applications – in particular, they argue that metadata must be stored at three levels, and that algorithms to manipulate the meta-data must traverse these levels.

The refereeing of the papers by Marcus and Subrahmanian, Jagadish, Ozden et al., Marcus, and Kashyap et al. was handled by Sushil Jajodia. The refereeing process for the other papers was handled by V.S. Subrahmanian. In addition, all but three papers (Ozden et al., Kashyap et al., and Jagadish) were discussed for several hours each in Subrahmanian's Multimedia Database Systems seminar course at the University of Maryland (Spring 1995). We are extremely grateful to those who generously contributed their time, reviewing papers for this book. Furthermore, we are grateful to the authors for their contributions, and for their patience in making revisions. Finally, we are grateful to Kasim Selcuk Candan for his extraordinary patience in helping to typeset the manuscript, and to Sabrina Islam for administrative assistance.

We would like to dedicate this book to our parents.

V.S. Subrahmanian
College Park, MD

Sushil Jajodia
Fairfax, VA

September 1995

Table of Contents

Towards a Theory of Multimedia Database Systems

Sherry Marcus[1] and V.S. Subrahmanian[2]

[1] 21st Century Technologies, Inc. 1903 Ware Road, Falls Church, VA 22043.
E-mail: marcus@nego.umiacs.umd.edu

[2] *Institute for Advanced Computer Studies, Institute for Systems Research*
Department of Computer Science, University of Maryland, College Park, Maryland 20742.
E-mail: vs@cs.umd.edu

Summary. Though there are now numerous examples of multimedia systems in the commercial market, these systems have been developed primarily on a case-by-case basis. The large-scale development of such systems requires a principled characterization of multimedia systems which is independent of any single application. It requires a unified query language framework to access these different structures in a variety of ways. It requires algorithms that are provably correct in processing such queries and whose efficiency can be appropriately evaluated. In this paper, we develop a framework for characterizing multimedia information systems which builds on top of the implementations of individual media, and provides a logical query language that integrates such diverse media. We develop indexing structures and algorithms to process such queries and show that these algorithms are sound and complete and relatively efficient (polynomial-time). We show that the generation of media-events (i.e. generating different states of the different media concurrently) can be viewed as a query processing problem, and that synchronization can be viewed as constraint solving. This observation allows us to introduce the notion of a media presentation as a sequence of media-events that satisfy a sequence of queries. We believe this paper represents a first step towards the development of multimedia theory.

1. Introduction

Though numerous multimedia systems exist in today's booming software market, relatively little work has been done in addressing the following questions:

- What are multimedia database systems and how can they be formally defined so that they are independent of any specific application domain ?
- Can indexing structures for multimedia database systems be defined in a similar uniform, domain-independent manner ?
- Is it possible to uniformly define both query languages and access methods based on these indexing structures ?
- Is it possible to uniformly define the notion of an update in multimedia database systems and to efficiently accomplish such updates using the above-mentioned indexing structures ?
- What constitutes a multimedia presentation and can this be formally defined so that it is independent of any specific application domain ?

In this paper, we develop a set of initial solutions to all the above questions. We provide a formal theoretical framework within which the above questions can be expressed and answered.

The basic concepts characterizing a multimedia system are the following: first, we define the important concept of a **media-instance**. Intuitively, a media-instance (e.g. an instance of video) consists of a body of information (e.g. a set of video-clips) represented using some storage mechanism (e.g. a quadtree, or an R-tree or a bitmap) in some storage medium (e.g. video-tape), together with some functions and/or relations (e.g. next minute of video, or who appears in the video) expressing various aspects, features and/or properties of this media-instance. We show that media-instances can be used to represent a wide variety of data including documents, photographs, geographic information systems, bitmaps, object-oriented databases, and logic programs, to name a few.

Based on the notion of a media-instance, we define a **multimedia system** to be a set of such media-instances. Intuitively, the concatenation of the states of the different media instances in the multimedia system is a snapshot of the global state of the system at a given point in time. Thus, for instance, a multimedia system (at time t) may consist of a snapshot of a particular video-tape, a snapshot of a particular audio-tape, and segments of affiliated (electronic) documentation. In Section 4., we develop a **logical query language** that can be used to express queries requiring multimedia accesses. We show how various "intuitive" queries can be expressed within this language. Subsequently, we define an **indexing structure** to store multimedia systems. The elegant feature of our indexing structure is that it is completely independent of the type of medium being used – in particular, if we are given a *pre-existing* representation/implementation of some information in some medium,. our method shows how various interesting aspects (called "features") of this information can be represented, and efficiently accessed. We show how queries expressed in our logical query language can be efficiently executed using this indexing structure.

Section 5. introduces the important notion of a media presentation based on the notion of a **media-event**. Intuitively, a media-event reflects the global state of the different media at a fixed point in time. For example, if, at time t, we have a picture of George Bush on the screen (i.e. video medium) and an audio-tape of George Bush saying X, then this is a media-event with the video-state being "George Bush" and the audio-state being "George Bush saying X." A **media presentation** is a sequence of media-events. Intuitively, a media-presentation shows how the states of different media-instances change over time. One of the key results in this paper is that any query generates a set of media-events (i.e. those media-events that satisfy the query). Consequently, the problem of specifying a media-presentation can be achieved by specifying a sequence of queries. In other words,

Generation of Media Events = Query Processing.

Finally each media-event (i.e. a global state of the system) must be "on" for a certain period of time (e.g. the audio clip of Bush giving a speech must be "on" when the video shows him speaking). Furthermore, the next media-event must come on immediately upon the completion of the current media-event. We show that this process of synchronizing media-events to achieve a deadline may be viewed as a constraint solving problem, i.e.

$$\textbf{Synchronization} \;=\; \textbf{Constraint Solving.}$$

2. Basic Ideas Underlying the Framework

In this section, we will articulate the basic ideas behind our proposed multimedia information system architecture. For now, we will view a media-source as some, as yet unspecified, representation of information. Exactly how this information is stored physically, or represented conceptually, is completely independent of our framework, thus allowing our framework to be interface with most existing media that we know of.

Suppose $\mathcal{M}$ is a medium and this medium has several "states" representing different bodies of knowledge expressed in that medium – associated with this data is a set of "features" – these capture the salient aspects and objects of importance in that data. In addition, there is logically specified information describing relationships and/or properties between features occurring in a given state. These relationships between features are encoded as a logic program. Last, but not least, when a given medium can assume a multiplicity of states, we assume that there is a corpus of state-transition functions that allow us to smoothly move from one state to another. These are encoded as "inter-state" relationships, specifying relations existing between states taken as a whole. As the implementation of these inter-state transition functions is dependent on the medium, we will assume that there is an existing implementation of these transition functions. As we make no assumptions on this implementation, this poses no restrictions. Figure 2.1 shows the overall architecture for multimedia information systems.

The ideas discussed thus far are studied in detail in Section 4. where we develop a query language to integrate information across these multiple media sources and express queries, and where we develop access structures to efficiently execute these queries.

All the aspects described thus far are independent of time and are relatively static. In real-life multimedia systems, time plays a critical role. For instance, a query pertaining to audio-information may need to be synchronized with a query pertaining to video-information, so that the *presentation* of the answers to these queries have a coherent audio-visual impact. Hence, the data structures used to represent information in the individual media (which so far,

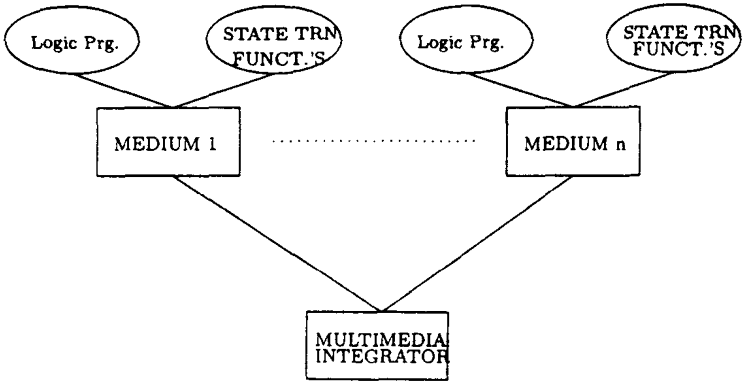

Fig. 2.1. Multimedia Information System Architecture

has been left completely unspecified) must satisfy certain efficiency requirements. We will show that by and large, these requirements can be clearly and concisely expressed as constraints over a given domain, and that based on the design criteria, index structures to organize information within a medium can be efficiently designed.

3. Media Instances

In this section, we formally define the notion of a media-instance, and show how it can be used to represent a wide variety of data stored on different kinds of media. Intuitively, a medium (such as `video`) may have data stored on it in many formats (e.g. `raster`, `bitmap`, `vhs_format`, `pal`, `secam`, etc.). Thus, raster is an example of an instance of the medium `video` because video information may be stored in raster format. However, in addition to just storing information, media instances, as defined below contain information on how to access and manipulate that information.

A *media-instance* is a 7-tuple $\mathrm{mi} = (\mathbf{ST}, \mathrm{fe}, \lambda, \Re, \mathcal{F}, \mathbf{Var}_1, \mathbf{Var}_2)$ where **ST** is a set of objects called *states*, fe is a set of objects called *features*, λ is a map from S to 2^{fe}, $\mathbf{Var}_1$ is a set of objects called *state variables* ranging over states, $\mathbf{Var}_2$ is a set of objects called *feature variables* ranging over features, $\Re$ is a set of *inter-state relations*, i.e. relations (of possibly different arities) on the set ST, and $\mathcal{F}$ is a set of *feature-state relations*. Each relation in $\mathcal{F}$ is a subset of $\mathbf{fe}^i \times \mathbf{ST}$ where $i \geq 1$.