

CS6710 Advanced Database Systems

Multimedia Databases

Instructor: Yi-Shin Chen
Office: EECS 3201
Email: yishin@cs.nthu.usc.edu
Office Hour: Monday 3-4PM, Wed. 11-12pm

Possible Multimedia Applications

URL: http://imsc.usc.edu/movies/demos/collections/collect1.wmv

Outline

- Introduction & Definition
- Modeling, Indexing
- Querying

Introduction

- Various fields/applications require management of multimedia data

- Challenges
  - Create easy ways to:
    - Analyze
    - Summarize
    - Search
    - View
  - Require better facilities for managing multimedia information

Initial Thoughts

- Can be applied in
  - Relational architectures
  - OO architectures
- An image and its contents ⇒ sets of tuples
- Initially believed that classic relational techniques can work
  - Indexing
  - Query optimization
  - Buffer management
  - Concurrency control
  - Security
  - Recovery

However...

- The previous approach has an inherent weakness
  - A mismatch between:
    - The nature of the data
    - The way both the user and the system were forced to query and operate

Course ID	Name	Instructor	Time
CS671000	Advanced Database Systems	Yi-Shin Chen	M3M4W3
CS342301	Operations System	Tai-Yi Huang	M3M4W2
CS554000	Pattern Recognition	Chaur-Chin Chen	M7M8T6




Moreover

- Object SQL queries and operations won't do
  - Browsing is an important paradigm for multimedia data
- Standard indexing approaches do not work
  - For content-based queries of multimedia data
- Other modules of database systems must be changed

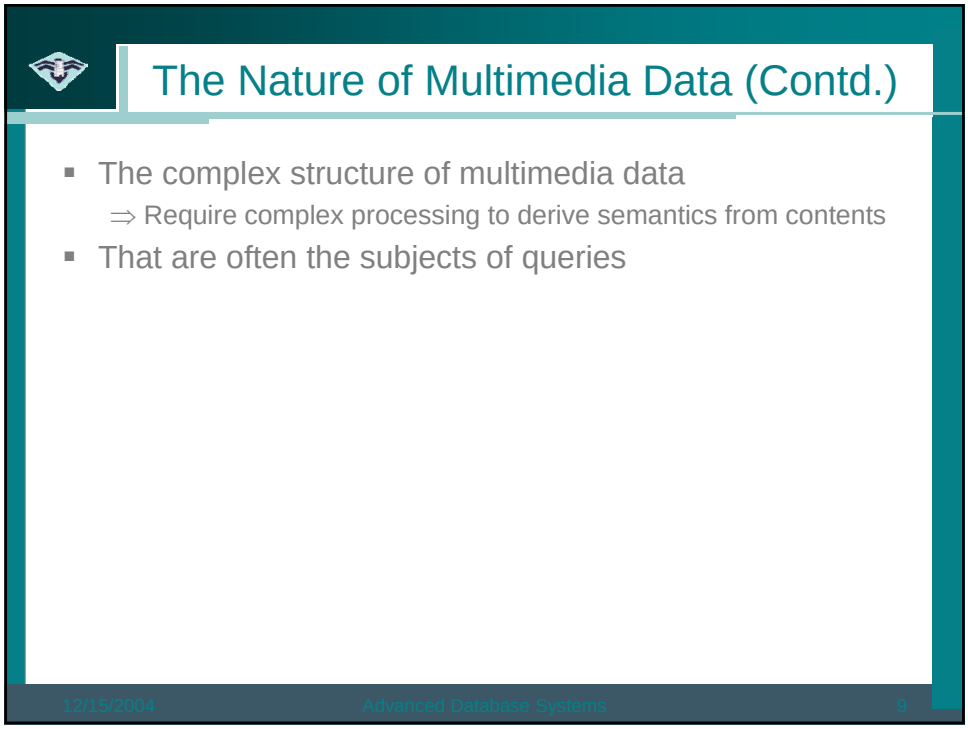
  ⇒ in order to manage multimedia data efficiently

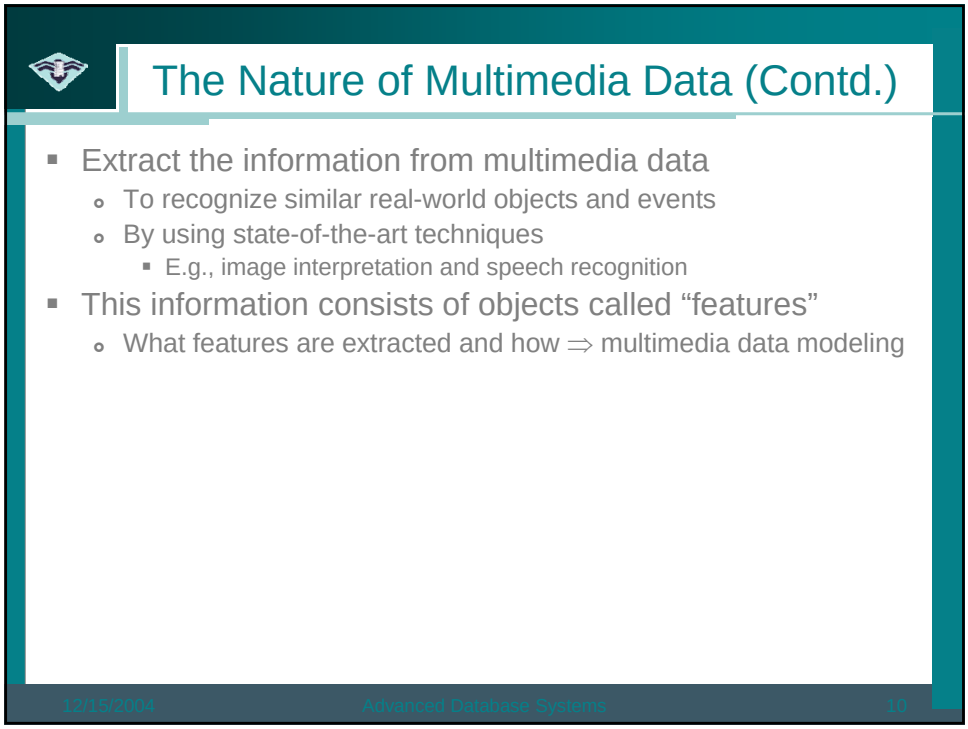



The Nature of Multimedia Data

- From a presentation viewpoint:
  - Multimedia data is huge
  - It involves time dependent characteristics that must be adhered to for coherent viewing

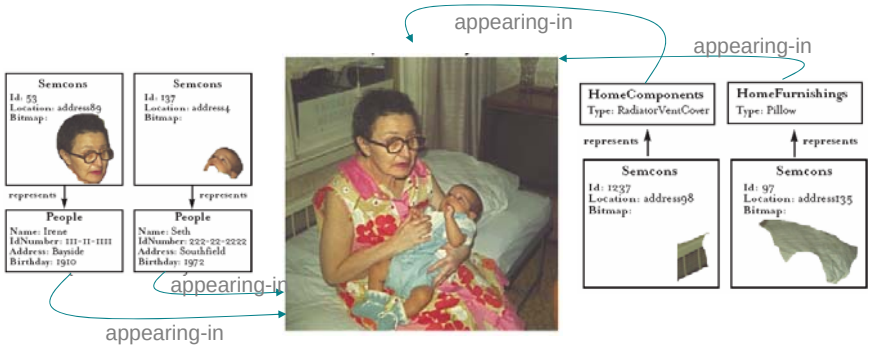
  ∴ Its presentation and the user's subsequent interaction with it push the boundaries of traditional database systems


The Nature of Multimedia Data (Contd.)
The complex structure of multimedia data
⇒ Require complex processing to derive semantics from contents
That are often the subjects of queries
12/15/2004
Advanced Database Systems
9


The Nature of Multimedia Data (Contd.)
Extract the information from multimedia data
To recognize similar real-world objects and events
By using state-of-the-art techniques
E.g., image interpretation and speech recognition
This information consists of objects called “features”
What features are extracted and how ⇒ multimedia data modeling
12/15/2004
Advanced Database Systems
10

Multimedia Data Modeling

- Multimedia objects
  - Completely defined in the database
  - Contain references to other real world objects



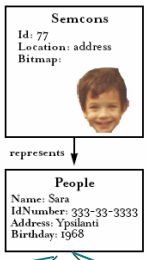
Semcon: for iconic data with semantics

Multimedia Data Modeling (Contd.)

- Information should be captured in a multimedia data model:
  - The detailed structure of the various multimedia objects
  - Structure-dependent operations on multimedia objects
  - Properties of multimedia objects
  - Relationships between multimedia objects and real-world objects
  - Portions of multimedia objects with representation relationships with
    - Real-world objects
    - The representation relationships themselves
    - The methods used to determine them
  - Properties, relationships, and operations on real-world objects

Multimedia Data Modeling (Contd.)

- For images
  - The structure should include:
    - The image format
    - The image resolution
    - The number of bits per pixel …etc.
  - The operations are used to create
    - Derived multimedia objects for similarity matching
    - Various composite multimedia objects from individual component objects



appearing-in

appearing-in is a derived relationship constructed from the above semcon and an **above threshold similarity** match

Query Processing

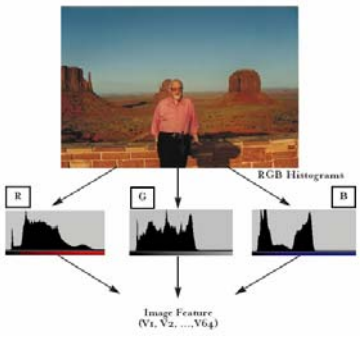
- Querying in a multimedia database is quite different
  - Browsing takes on added importance in a multimedia environment
  - Queries can contain multimedia objects input by the user
  - The results of these queries are based not on perfect matches but on degrees of similarity
- Examples of similarity queries are:
  - Retrieve all video shots showing my friend Tom dancing, given a photograph of Tom
  - Show me all mug shots of criminals resembling this sketch

Indexing

- A generic indexing technique
  - Extract *n* numerical-valued features from a multimedia object
  - Represent these n values by an n-dimensional point
- A spatial index that supports nearest-neighbor searching is used for similarity matching
- These *n* features may be independent of each other or derived from a composite global feature
- Features:
  - Physical Features ⇒ Can be extracted from multimedia objects
  - Semantic Features ⇒ Need human input

Indexing (Contd.)

- An example of a composite feature technique:
  - Representation of the color histogram of an image as a high-dimensional point used in the QBIC database system



- Physical features: texture, shape, color, keywords, …etc.

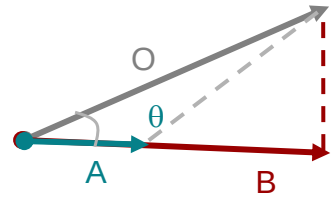
Dimensionality Curse

- Multimedia data often have high dimensionalities
  - There are commonly 64 or more color features
- Multidimensional indexing techniques
  - Quadtrees and grid files grow exponentially with the dimensionality
  - R+-Trees tend to be more robust, at least for dimensions up to around 20
- Possible solutions
  - Break down into small feature vectors
  - Pre-compute the values

Feature Vectors

- Once you can use a feature vector to represent a multimedia object

  ⇒Can compare two objects by calculating the similarity values



cos(A,O)=cos(B,O) Ed(A,O)>Ed(B,O)

Cosine measure can only evaluate the difference in direction

In order to evaluate the magnitude, ***Euclidean distance*** should be employed

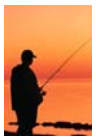
⇒Usually, the similarity value is a float point between 0 to 1

- 0: totally different
- 1: totally identical

Gap

- Gap between physical features and semantic features



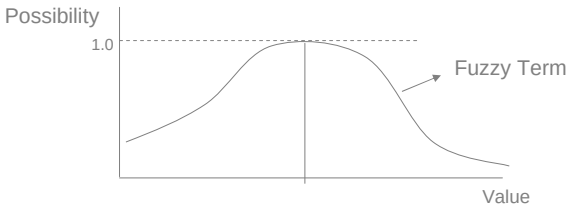


  - Physical features: color histograms, shape
  - Semantic features: fishing, enjoying, getting away
    - Perception & Subjective

Fuzzy

- Examples:
  - Show me the red pictures
  - Show me all mug shots of criminals resembling this sketch

- Fuzzy Logic
  - To deal the uncertain situations



Fuzzy Logic

- $\Pi(\varnothing) = 0$
- $\Pi(\cup) = 1$
- Union rule
  - $\Pi(A \cup SB) = \max(\Pi(A), \Pi(B))$
- Conjunction rule
  - $\Pi(A \vee B) = \max(\Pi(A), \Pi(B))$
- Intersection rule
  - $\Pi(A \cap B) = \min(\Pi(A), \Pi(B))$
- Disjunction rule
  - $\Pi(A \wedge B) = \min(\Pi(A), \Pi(B))$

Query Optimization

- The process of choosing the optimal access path to answer a query
  - Traditional: On a single sever
  - Distributed Environment: Consider network bandwidth
  - Multimedia Databases: Distributed && fuzzy characteristics
    - A query could be combined from several atomic queries
      - E.g., Color="red" && Shape="round"
    - The atomic queries could come from several sub-systems

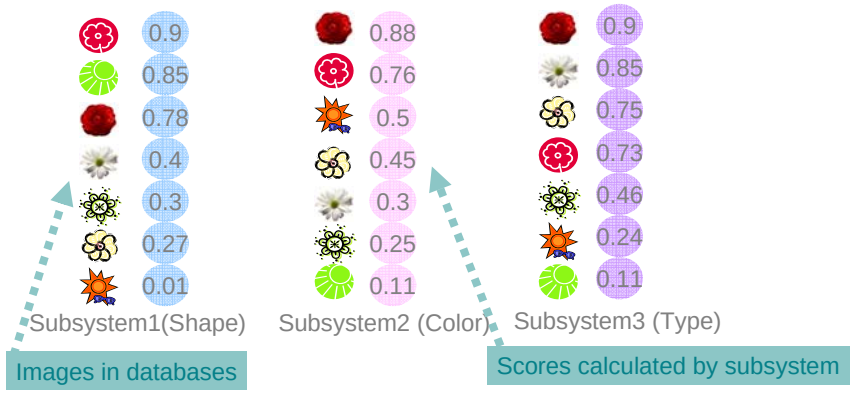
Fagin's Algorithm A_0 (FA)

- The most famous algorithm for evaluation monotone queries
  - Proc. Fifteenth ACM Symp. on Principles of Database Systems, Montreal, 1996, pp. 216-226
  - Proc. Seventeenth ACM Symp. on Principles of Database Systems, Seattle, 1998, pp. 1-10
- Consist of three phases
  - Sorted access phase: for each i, give subsystem i the query A_i under sorted access. Subsystem i begins to output, one by one in sorted order based on grade. Wait until there is a set L of at least k objects such that each subsystem has output all of the members of L
  - Random access phase: for each object x that has been seen, do random access to each subsystem j
  - Computation phase: compute the grade

Fagin's Algorithm

Query: Find out the top-2 objects where Shape = "Round" and Color = "Red" and Type = "Flower"

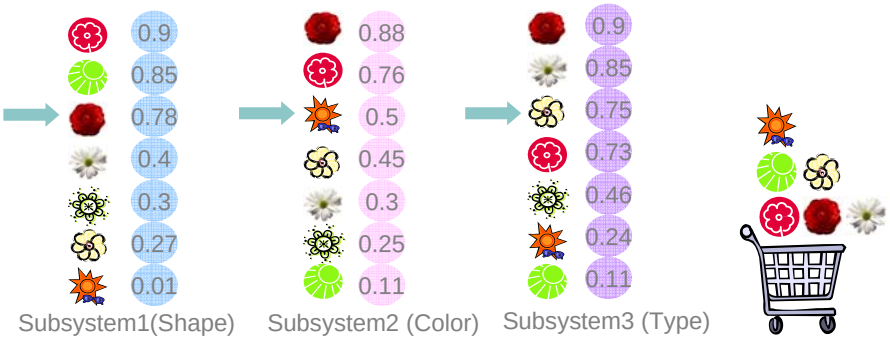
Step 1: Each subsystem produce a ranked (sorted) list according to the query



Fagin's Algorithm

Query: Find out the top-2 objects where Shape = "Round" and Color = "Red" and Type = "Flower"

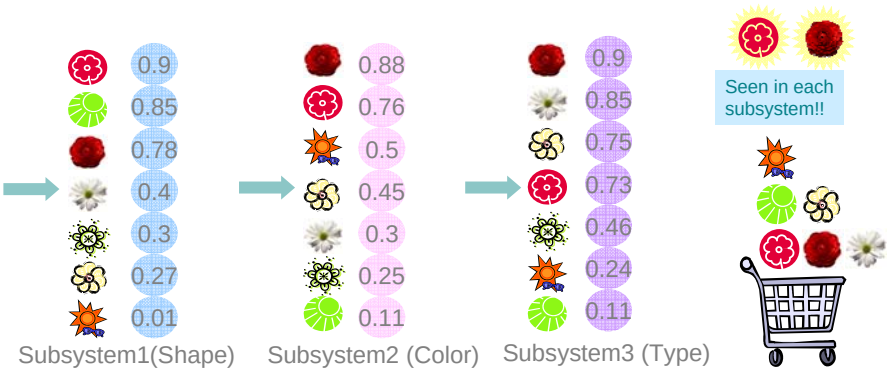
Step 2: Do sorted access in each subsystem and remember those seen so far



Fagin's Algorithm

Query: Find out the top-2 objects where Shape = "Round" and Color = "Red" and Type = "Flower"

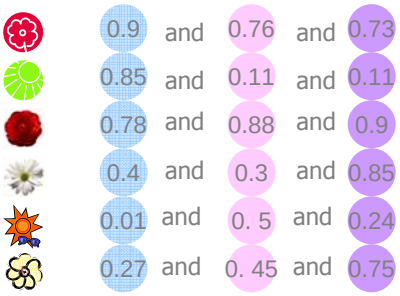
Step 3: Stop while there are at least 2 objects appeared in each subsystem



Fagin's Algorithm

Query: Find out the top-2 objects where Shape = "Round" and Color = "Red" and Type = "Flower"

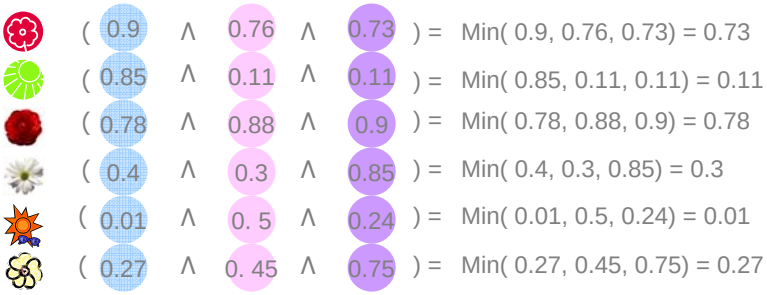
Step 4: For each object have been seen, do random access to get their scores



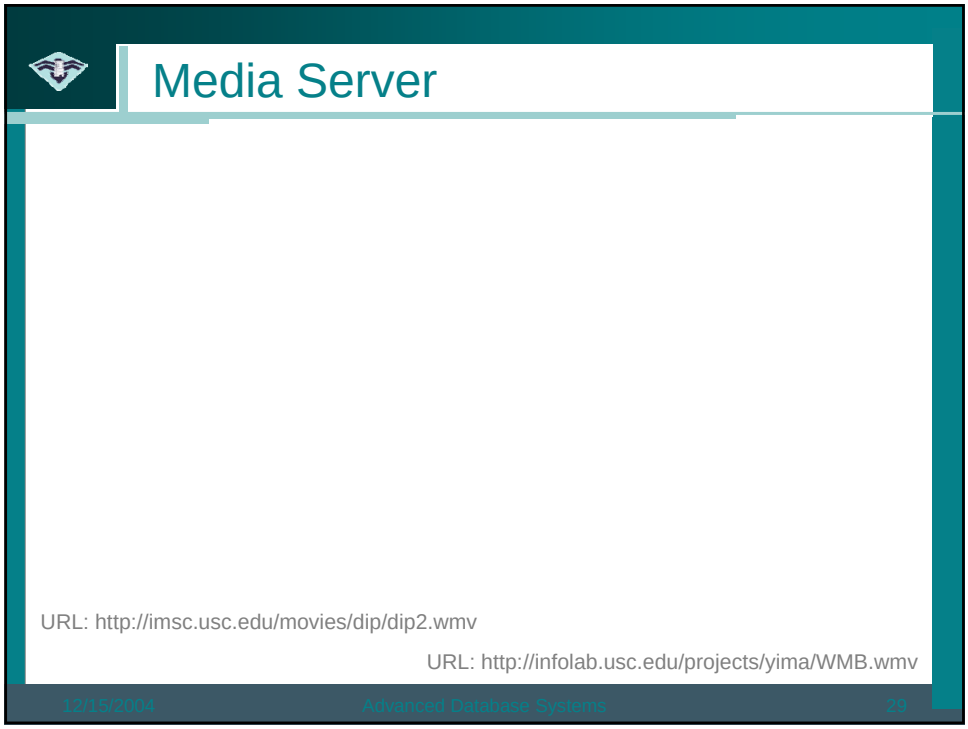
Fagin's Algorithm

Query: Find out the top-2 objects where Shape = "Round" and Color = "Red" and Type = "Flower"

Step 5:Compute the query and get the results



The top 2 are and


Media Server
URL: http://imsc.usc.edu/movies/dip/dip2.wmv
URL: http://infolab.usc.edu/projects/yima/WMB.wmv
12/15/2004
Advanced Database Systems
29