STARS: A SPATIAL ATTRIBUTES RETRIEVAL SYSTEM FOR IMAGES AND VIDEOS

JOHN Z. LI and M. TAMER ÖZSU
Department of Computing Science
University of Alberta
Edmonton, Canada T6G 2H1

Combining both text-based retrieval and content-based retrieval techniques in building multimedia databases is the ultimate goal of this work. We describe an object-oriented multimedia database which supports such a combination. In supporting content-based image and video retrieval, we focus on spatial properties which is an essential part of any image and video retrieval system. We deal with both spatial similarity and spatial relationships. The system is further enhanced by a powerful multimedia query language and by incorporating two level spatial attributes: precise spatial attributes and their approximation.

1 Introduction

Database management system (DBMS) technology has started to play a central role in the management of multimedia data [1,2,3,4,5,6,7]. The widening use of multimedia information systems has increased the need for efficient management of multimedia data, such as images, video and audio. A multimedia DBMS (MDBMS) is a software system that enables the acquisition, storage, manipulation and transmission of large amount of multimedia data. The types of information that are stored in a MDBMS include iconic data, which are the digitized object; alpha-numeric information consisting of media related data such as the resolution and media content description; media extracted information such as numeric or topological features and component relationships; relationships between the media and the real-world entities modeled in the application, as well as the spatio-temporal relationships; and application information, which describes the application and the environment that is related to that application.

To reduce data volume, a video is normally expressed in a database by a series of representative frames extracted from a set of shots. Each representative frame can be viewed as a still image. Therefore, we focus on images in this paper since the extensions to video are obvious.

Two retrieving approaches have been widely used in MDBMSs: *text-based* (or *keyword-based*) and *content-based*. Each approach has its own advantages and disadvantages [4]. Supporting *content-based* retrieval in an MDBMS has received a lot of research attention because it can be done fully automatically [8]. Experiments have shown that the best query result is obtained when the two approaches are combined [5]. We have designed and implemented an MDBMS called STARS (SpaTial Attributes Retrieval System) to incorporate both text-based and content-based technology. STARS is our first prototype of DISIMA (DIStributed Image database MAnagement system) project [9] and the ultimate goal of DISIMA is to support large and heterogeneous multimedia systems over a distributed environment. In this paper we focus on STARS's system architecture, type system, and spatial properties in dealing with images and videos.

Although object-oriented approach is considered the most promising technique for multimedia DBMS [10,11], surprisingly all content-based retrieval systems are based on relational DBMSs. Part of the reason could be that these systems payed little or no attention to the semantics of the media, i.e., the modeling of real world. This is certainly not acceptable for any full-fledged MDBMS. For example, a common query, such as "*find all images in which* Paul *is to the left of* John", cannot be answered by most of the existing systems.

An important feature of images and videos is the spatial properties among objects. The *spatial properties* (or *attributes*) include shapes and locations of objects, and spatial relationships between them. Most content-based retrieval systems only approximate objects by minimum bounding rectangles (MBR). Such an approximation imposes many restrictions making it difficult to extend these systems to a wide range of applications, such as satellite images or medical images. Furthermore, MBR approach may return false results. For example, two object may be disjoint while their MBRs are overlapped. In STARS the exact geometries and locations of objects are stored using an innovative point-set approach [12] which significantly reduces the amount of storage requirement for object spatial properties in a database. Furthermore, the system supports MBR to provide fast query processing. In supporting content-based spatial queries, we incorporate an existing technique of spatial similarity [13] plus our own efficient way of computing spatial relationships. STARS can handle sophisticated spatial queries. The major contributions of this work are:

- A new type system to model the real world objects by using *logical salient objects* and *physical salient objects* concepts. An advantage of this model is its extensibility so further application can be easily built on top of this system;
- A new multimedia object query language facilitating user queries is incorporated into the system;
- A comprehensive model to process spatial properties allows a set of broad applications; and

- A complete prototype system has been designed and implemented based on the features elaborated above; the system is accessible via the WWW.

The rest of the paper is organized as follows. Section 2 reviews the related work in spatial representation, multimedia modeling and querying. Section 3 introduces the STARS architecture and its type system. Section 4 describes spatial query processing in STARS including both exact match and similarity match queries. Section 5 discusses the detailed implementation of the system over an object database system. Section 6 summarizes our research and points out possible future work.

2 Related Work

QBIC (Query By Image Content)[3] is one of the earliest image retrieval systems and uses image analysis techniques to process queries for an image database. The system has also been extended to handle videos. QBIC lets users find pictorial information in large image and video databases based on colors, shapes, textures, and sketches. A typical query could be "*find more pictures that looks similar to this one*". A user can also sketch a shape, select colors and color distributions from a color panel, or select textures from a predetermined range. The system returns a ranked list of best matches to the user query. However, QBIC has no knowledge of particular objects (such as a person named *John*) and their exact spatial information. Therefore, you cannot ask any queries related to these.

Another content-based image retrieval system using both color and spatial attributes is described in[4]. This approach consists of three steps: the selection of a set of representative colors, the analysis of spatial information of the selected colors, and the retrieval process based on the integrated color-spatial information. After deriving the set of representative colors, spatial knowledge of the selected colors is obtained using a maximum entropy discretization with event covering method. A retrieval process is formulated to make use of the spatial knowledge to retrieve relevant images. However, only MBRs are applied to capture spatial properties.

OVID (Object-oriented Video Information Database)[1] is an object-oriented video model. It introduces the notion of a *video object* which can identify an arbitrary video frame sequence (a meaningful scene) as an independent object and describe its contents dynamically and incrementally. OVID model has no schema and the traditional class hierarchy of Object DBMSs (ODBMS) is not assumed. An inheritance based on an *interval inclusion relationship* is introduced to share description data among video objects. A major problem with OVID model is that it does not support the content-based video retrieval.

Chabot [5] is an image retrieval system based on a relational database. Each image in Chabot is carefully annotated so that it has very nice keyword-matching search ability. It also has limited color feature retrieval to help users identify the *target images*. The annotation and color feature can be combined to make a query. Chabot does not explore other features, such as shapes, textures.

Another system [14] is reported to be able to do content-based video parsing, retrieving, and browsing. The effectiveness of this system lies in its use of video content information provided by a parsing process driven by visual feature analysis. The parsing process temporally segments and abstracts a video source, based on low-level image analyses; then the retrieving and browsing process are proceeded over representative frames selected during the previous process, the spatial-temporal variations of visual features, as well as some shot-level semantics derived from camera operation and motion analysis. However, spatial feature extraction and retrieval are not discussed in this system.

Since populating an MDBMS is a major challenge, much research has been done in developing tools for automatically preprocessing image and video data. The Photobook project [8] seeks to circumvent the issue of predetermined search criteria by storing enough information about each image to make runtime computations possible. Images are classified at load time as having face, shape, or texture properties; techniques have been developed to automate this process – for example, foreground extraction. Virage video engine [15] is another development tool for parsing videos in supporting content-base retrieval. It allows for automatic multi-modal indexing and retrieval of video through the use of media specific primitives.

3 STARS Architecture and Type System

The major objective of this work is to model image and video data using object-oriented techniques and to support efficient content-based spatial queries by using new modeling and indexing techniques. The architecture of STARS should be extensible to allow new features to be added easily. Object-oriented technology provides good extensibility in modeling multimedia data. In this section we first describe the STARS architecture and then the type system.

3.1 STARS System Architecture

Figure 1 shows the STARS architecture and the shaded boxes are the focus of this paper. Raw image and video data are processed by an *Image/Video Analyzing* module which uses image and video processing techniques via an *Image and Video Tool Library* to recognize *salient (physical) objects*, video shots, etc.

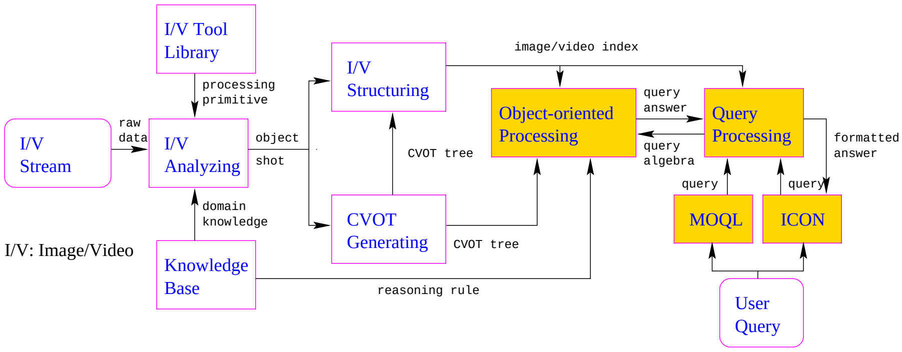

Figure 1: STARS System Architecture

The main function of the *Image and Video Analyzing* module is to make feature extraction. The type of features to be extracted depends on applications. Restricted by the current technology, this analysis has to be domain specific. For video data, the salient objects are encoded in the Common Video Object Tree (CVOT) model [16] by their properties, such as size, location, moving direction, etc. Then a CVOT tree is generated by the *CVOT Generating*. The CVOT tree is an efficient way of indexing salient objects for video data. The *Image and Video Structuring* module builds the necessary indexes from the output of both the *Image and Video Analyzing* module and the *CVOT Generating* module to provide efficient access. A unified model is necessary for users to query the system. An *Object-oriented Processing* module is used because of its powerful representation of the users' views and its suitability for multimedia data. A user posts queries through a Multimedia Object Query Language (MOQL) based on an ODMG's Object Query Language [17] or an Iconic Graphical User Interface (ICON). MOQL, like SQL, allows users to elaborate their queries precisely using a textual language. On the other hand, ICON, like other Graphical User Interfaces (GUI), allows users to sketch iconic objects and to ask the database to return the target images with the closest match according to the spatial properties. All the queries are handled by the *Query Processing* internally. The *Query Processing* module defines translation of the queries into an internal query algebra which can be optimized and executed by the system. During this translation some query optimization can be done; therefore the *Query Processing* module needs to account for the specifics of image and video structuring. While executing user queries, STARS may not have all the essential information, but it can make temporal or spatial inference from the existing relations using the reasoning rules provided by the *Knowl-*

edge Base. The answer to the queries is returned to the end user through a GUI interface.

3.2 STARS Type System

A data model is defined as a collection of mathematically well-defined concepts to express both static and dynamic properties of data intensive applications. STARS model aims at efficient representations of images (videos) to support a wide range of queries. One of the novelties of this model is the introduction of *salient objects*. A *salient object* is an interesting object which an MDBMS designer wants to capture (e.g. persons, houses, cars). A salient object may or may not be in an image or a video frame while each image or video frame can have many salient objects. In order to efficiently access salient objects we distinguish two types[a] of salient objects: *logical salient objects* and *physical salient objects*. *Logical salient objects* correspond to objects independent of their appearance in an image and describe their attributes independent of any media while *physical salient objects* correspond to objects specific to an image and describe their attributes tied to the underline media. For example, suppose *John* is a person which appears in image m. Then, the name and address of *John* are considered the logical properties of the salient object *John*. The spatial, color, and texture attributes of *John* in image m are considered as the physical properties of the salient object *John*. Since *John* might appear in many different images, the relationship between logical salient objects and physical salient objects is one-to-many.

Figure 2 shows a simplified overview. The model is composed of two main blocks: the image block and the salient object block. We define a block as a functional first-level entity that can be broken down into several entities. The *image block* is made up of two layers: the *image layer* and the *image representation layer*. We distinguish an image from its representations to maintain an independence between them, referred to as *representation independence*. In the *image* layer, the user defines image type classification which allows the user to define functional relationships between images.

One of the main problems in an MDBMS is to capture the content of an image. We view the content of an image as a set of salient objects (both logical and physical) with certain spatial relationships to each other. The *salient object block* is designed to handle salient objects. Multiple physical salient objects belonging to images may correspond to one logical salient object. Hence, a logical salient object does not have any representation, but a physical salient object may have one or more. The logical salient object level

[a] In this paper type and class are used interchangeably.

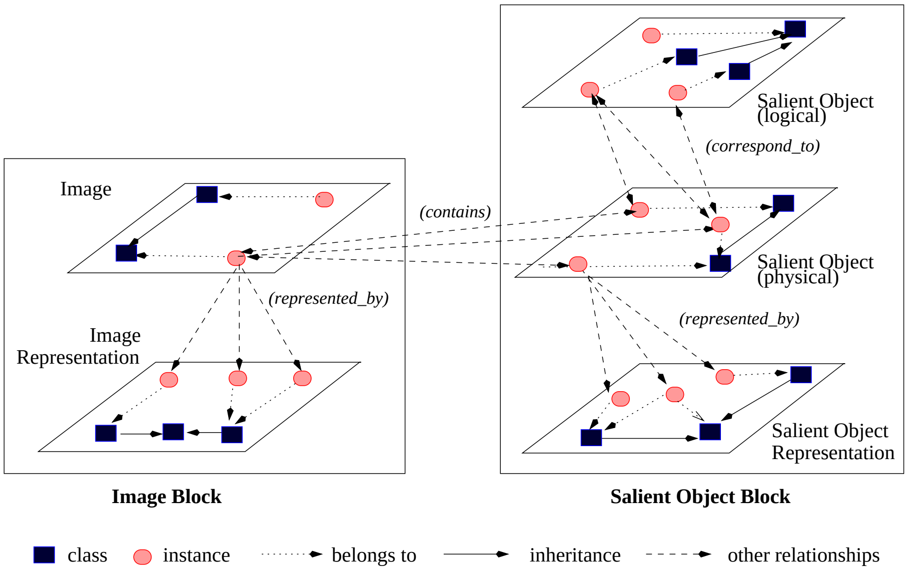

Figure 2: Image and Salient Object Model

models the semantic of physical salient objects. For a given application, salient objects are known and can be defined. The definition of salient objects can lead to a physical salient object which may have one or more representations (a minimum bounding rectangle, a point-set, a vector etc.). Salient object relationships such as spatial relationships (e.g., *left*, *northeast*) make sense only at the physical salient object level. A logical salient object has some functional relationships (is-a) with other logical salient objects. A logical salient object gives a meaning (a semantic) to a physical salient object.

Depending on the application, the user can assign different semantics to the same physical salient object. Detailed relationships between image, logical salient object and physical salient object are given in Figure 3. The two levels of salient objects ensure the semantic independence and multi-representation of salient objects. Only the logical salient object level is used to answer the query "*find all the images in which John appears*". The query "*find all the images in which John is to the left of Paul*" requires the physical salient object level to check the spatial relationship among the two salient objects. The processing can indeed benefit from the existence of spatial indexes. The extension of this model to videos is straightforward.

Figure 3 shows the key classes which are implemented in STARS. This figure is generated directly from Rational Rose [18], which is a design tool for object-oriented development. A line that ends with an arrow represents *subclass* relation while the line ends with a bullet represents *aggregation* relation. Other type of relations are marked explicitly, such as *contains*, *composedOf*, etc. The numbers, such as 0, 1 and n, are the cardinalities of participating classes. *abstract classes* are marked by a small triangle with character "A" inside.

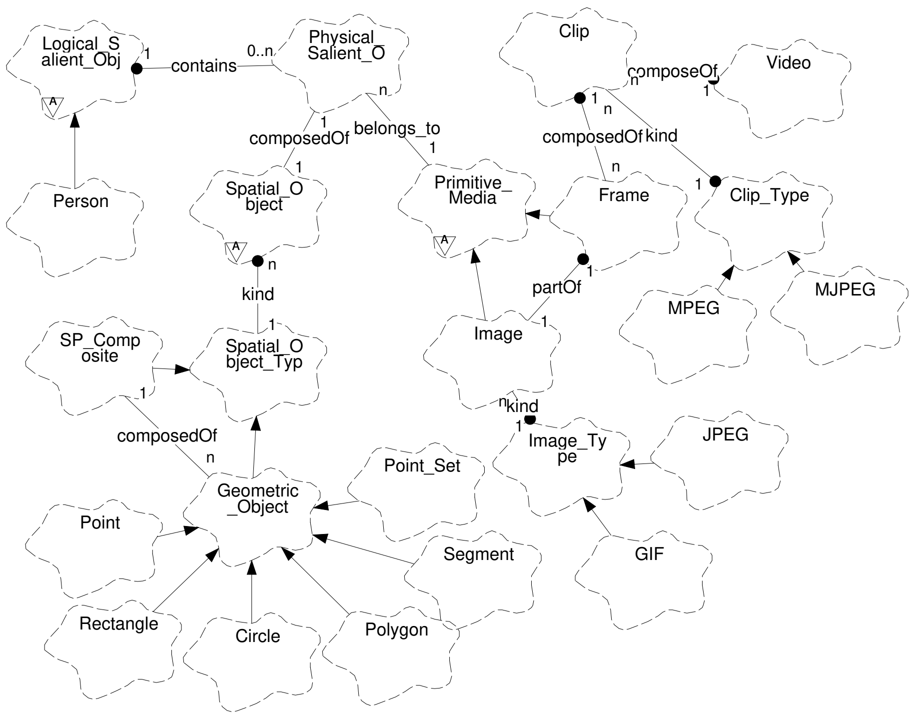

Figure 3: STARS Type System

Two abstract classes deserve elaboration. One is *Primitive_Media* and the other is *Spatial_Object*. *Primitive_Media* is the super class of all media, e.g. images, videos. Only *Primitive_Media* interacts with *Physical_Salient_Object* and their relationship is one-to-one. This will allow us to add any new medium into the system without changing the type system. For example, if we are planing to model animation as a new type of media, then we can simply subtype animation under the *Primitive_Media* without touching any other class.

Spatial_Object is the super class of all geometrical objects which include *Point*, *Line*, *Rectangle*, *Polygon*, *Point_set*, etc. A *Spatial_Object* may consist of one geometric object or multiple geometric objects. For the later, we include a special class *SP_Composite*. This allows the system to model complex objects by simple geometric objects. For example, the spatial object of a person might consist of the head (a circle), the body (a rectangle), and arms (polygons).

Following C++ program reveals more details of some method definitions. The classes *Relationship_1_m* (one-to-many), *Relationship_m_1* (many-to-one), and *Set* are provided by ObjectStore [19], which is a commercial ODBMS – on top of which STARS is built.

```
class Logical_SO {
  Relationship_m_1(Logical_SO,physical_so,
     Physical_SO,logical_so,Set<Physical_SO*>)
             physical_so;
  String annote, type;
public:
  String& get_annotation() const;
  Set<Physical_SO*>& physical_objs();
  void add_physical_obj(Physical_SO* pso);
  void remove_physical_obj(Physical_SO* pso);
  virtual String& get_type() const = 0;
};

class Primitive_Media {
  Relationship_m_1(Primitive_Media,physical_so,
    Physical_SO,pm,Set<Physical_SO*>)
           physical_so;
public:
  Set<Physical_SO*>& physical_objs();
  void add_physical_obj(Physical_SO* pso);
  void remove_physical_obj(Physical_SO* pso);
};
```

```
class Physical_SO {
  Relationship_1_m(Physical_SO,pm,
       Primitive_Media,physical_so,
       Primitive_Media*) pm;
  Relationship_1_m(Physical_SO,logical_so,
       Logical_SO,physical_so,Logical_SO*)
        logical_so;
  Spatial_Obj *spatial_obj;
public:
  Primitive_Media* primitive_media();
     Logical_SO* logical_obj();
     Spatial_Obj* spatial_obj();
     void set_spatial_obj(Spatial_Obj* so);
     void set_logical_obj(Logical_SO* lso);
     void remove_logical_obj(const
              Logical_SO* lso);
     void set_primitive_media(
              Primitive_Media* pm);
     void remove_primitive_media(const
              Primitive_Media* pm);
     Rectangle* get_mbr();
 };
```

4 Spatial Query Processing

One important feature of multimedia data is the inherent spatial properties. *Spatial properties* pertain to the space occupied by objects and include points, lines, regions, etc. Common spatial relations can be classified into two categories: *topological relations* that describe neighborhood as well as incidence (e.g., overlap, disjoint) and *directional relations* that describe order in space (e.g., south, northwest, left).

The point-set technique has been used in spatial databases [20], especially in defining *topological relations*. The description of topological spatial relations in terms of topologically invariant properties of point-sets is fairly simple. As a consequence, the topological spatial relation between two point-sets may be determined with little computational effort. However, point-set approaches are notorious for large storage space requirement. Directly extracting point-sets from raw images is too expensive and almost impossible in practice. The size of raw images can be reduced without losing their spatial relations[7,21]. By using the techniques described in [12] over the reduced images, we are able to store the precise geometrical shapes and locations of salient objects. Then, MBRs of salient objects can be easily computed from this precise information. STARS is perhaps unique in supporting both precise and MBR-based spatial relation computation.

4.1 Spatial Similarity

Retrieval by *spatial similarity* deals with a class of queries that is based on spatial relationships among the domain objects. Spatial queries require selecting target images that satisfy the spatial relationships specified in the query to varying degrees. This degree of conformance is used to rank order target images with respect to the query. A spatial similarity matching algorithm [13] is used in STARS because of its efficiency and robustness. The time complexity of this algorithm is $\Theta(n)$ where n is the total number of salient objects in query and target images. The robustness of this algorithm lies in that it recognizes translation, scale, rotation, and arbitrary variants of an images.

An algorithm for spatial similarity can be defined as a function that takes two images (or the corresponding physical salient objects) as input and returns a value in the range of $[0, 1]$ as output. Value 1 is expected when the input images are spatially identical. The other values must be proportional to the degree of agreement in the spatial relationships between the corresponding physical salient objects in the input images. In other words, spatial similarity algorithms should possess *monotonicity* property. Let t_1 and t_2 be two target images and q be the query image. If t_1 satisfies more spatial relationships in q than t_2, t_1 should be assigned a higher similarity value.

Intuitively, each physical salient object has a centroid which describes its mass location. An edge can be drawn from any two centroids in a 2D space. By comparing the angle of two edges (one is from query image and the other one is from target image), The spatial similarity of the two images can be determined. Formally, let $p(x_p, y_p)$ and $q(x_q, y_q)$ be two points in the 2D Cartesian coordinate space. The notation $\overrightarrow{pq}$ denotes the vector from point p to point q. The magnitude of $\overrightarrow{pq}$, denoted by $|\overrightarrow{pq}|$, is given by $|\overrightarrow{pq}| = \sqrt{(x_q - x_p)^2 + (y_q - y_p)^2}$. Given two vectors $\overrightarrow{pq}$ and $\overrightarrow{p'q'}$, the acute or obtuse angle, θ, between $\overrightarrow{pq}$ and $\overrightarrow{p'q'}$ is computed by the expression

$$\cos\theta = \frac{\overrightarrow{pq} \bullet \overrightarrow{p'q'}}{|\overrightarrow{pq}|\ |\overrightarrow{p'q'}|}, \quad 0 \le \theta \le \pi,$$

where the operator $\bullet$ denotes vector dot product. This approach has been proven to be both efficient and effective [13].

4.2 Multimedia Object Query Language

Some spatial queries are best expressed in the traditional text-based manner because it is simple and precise, e.g., "*find all the images in which* John *is to the left of* Paul". Furthermore, programmable interface to MDBMSs require an embedded text-based query language. We, therefore, propose an object-oriented, general-purpose query language, which we call Multimedia Object Query Language (MOQL) [22], based on OQL. MOQL includes constructs to deal with spatial properties, temporal properties, and presentation properties. OQL (Object Query Language) [17] has been proposed by Object Database Management Group which is a consortium of ODBMS vendors and universities working on standards to allow portability of customer software across ODBMS products. Syntactically, OQL is very similar to SQL. It is currently supported by many ODBMS vendors and its popularity should increase as the ODBMS market grows.

Spatial access methods commonly store object approximations and use these approximations to index the data space in order to efficiently retrieve the potential objects that satisfy user queries. MBR has been used in STARS to approximate objects because an MBR needs only two points for its representation. However, a major disadvantage of using MBR is its *false hit*, i.e., results which do not satisfy user queries are also returned. A good browser might help users to eliminate some false hits. In STARS, if users do not want any false hit, they can choose precise spatial processing. Even in precise spatial processing, MBR processing is still executed first to reduce the potential candidates because precise spatial processing is computationally very expensive. MBR preprocessing is normally called the *filter step*. STARS adopts this two level processing strategy if users require precise spatial processing.

An innovative method [23] for computing spatial relationships based on MBRs is implemented in STARS. The idea is to extend *temporal interval algebra* [24] to model 2D spatial relationships since the temporal interval algebra essentially consists of the topological relations in one dimensional space. In STARS there are 12 directional relations: *north, south, west, east, northeast, southeast, northwest, southwest, above, below, left, right* and eight topological relations: *covers, covered_by, inside, contains, equal, overlap, touch, disjoint.* They have been considered as most commonly used and essential spatial relations. Now we take a look at some MOQL query examples.

Query 1 Find all images in which a person appears.

select	m
from	Images m, Persons p
where	m **contains** p

The predicate **contains** is true if the logical salient object p is in image m. Otherwise, it is false.

Query 2 Find all images in which *John* is to the left of *Paul*.

select m
from Images m, Persons p, q
where m **contains** p **and** m **contains** q **and**
p **left** q **and** p.name = "John" **and** q.name = "Paul"

5 System Implementation

STARS is implemented on top of ObjectStore using C++ for the core system, Yacc/Lex for parsing MOQL, Java for the on-line user interface, and C for generating MOQL query tree and setting the web communication between Java user interface and the core system. Currently there are only 100 images (about 30M bytes data) in STARS database, which are randomly collected from the Internet and they reflect a variety of image domains. Since salient objects to be recognized originate from disparate domains and images contain considerable noise, no known technique is available to do object recognition automatically. This has forced us to identify salient objects manually. However, the spatial properties of salient objects are extracted automatically using the well-known technology. To try STARS on-line demo, please visit http://www.cs.ualberta.ca/~zhong/ web site. We are currently in the process of digitizing 50,000 images from the Photography Service, the University of Alberta. After these images are loaded, we can expect a better estimation of system efficiency.

As shown in Figure 4, the graphic user interface is divided into two major parts: left part is used to display useful information (such help page) and results while right part allows user to post queries. The right part is further divided into two areas: the top area is called **IconSpace** and the bottom area is called **MoqlSpace**. **IconSpace** is designed to help a user to retrieve images and videos by spatial similarity while **MoqlSpace** is designed to help a user to retrieve images and videos by spatial relationships.

In **IconSpace** there is a sketch pad allowing a user to draw any spatial relationships between objects. Each object can be either associated with an icon (such as *Sun, Sea, Person*) or particular salient object (such as *John, my blue-balloon, Paul's bike*). Users can set a *similarity threshold* value (from 1% to 99%) to decrease or increase the qualified results. Lowering this value can result in more returns with less similarity. Figure 4 shows a query image with three icon objects (Sun, Sea, and Tree) spread out like a triangle.

The right bottom of Figure 4 shows the query "*find all the images in which*

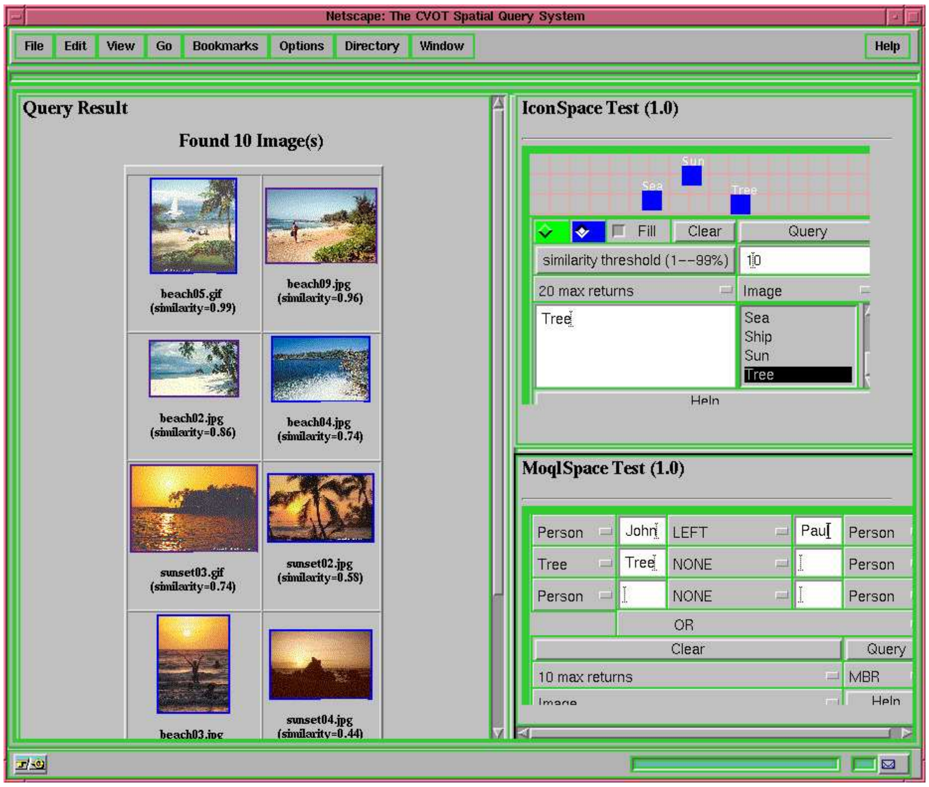

Figure 4: An IconSpace Query

John *is to the left of* Paul or *a tree appears*". Such a query will be translated into following MOQL query:

select m
from Images m, Persons p, q, Tree t
where m **contains** t **or** (m **contains** p **and** m **contains** q **and** p **left** q **and** p.name = "John" **and** q.name = "Paul")

In **MoqlSpace**, if the spatial relation is *NONE*, the system automatically interprets the query as predicate **contains**. For example, the query "*find all the images having a person*" can be simply expressed by selecting a *Person* item and then clicking on *Query* button. Again users can post the queries over images, videos or both. In **MoqlSpace**, either MBRs or precise spatial properties can be specified to compute user queries.

6 Conclusions

The ultimate goal of STARS is to support both text-based and content-based approaches which can result in the best result in dealing with multimedia data retrieval. We have introduced the concept of logical and physical salient objects and shown how these concepts can be applied to the multimedia world. This is accomplished by presenting a unique type (class) system, centered on the accommodation of both logical and physical salient objects, which has been fully implemented. STARS is extensible and future extensions can be easily built on top of it. A unique feature of STARS is that the system has the knowledge of the exact geometries and locations of salient objects by using a new point-set approach. Such a new point-set approach significantly reduces the amount of storage required for storing object spatial attributes in a database. Furthermore the system also supports MBRs to make fast query processing. Queries involving both sophisticated spatial relationships and spatial similarity are supported. A graphical user interface, based on popular web browser such as Netscape, has been built. An on-line demo version is accessible over the Internet.

We are currently in the process of loading large amount of images into STARS. This should provide better opportunity to test the system under large volume of data and to tune the system. Video data will be added into the database soon which we envision as straightforward if the representative frames approach is used. The current functionality of MOQL is minimal. Much more work is needed to make it fully functional in STARS. Another important future work is to incorporating some well-known feature processing techniques, such as color and texture features, into STARS. Such an addition will greatly enhance functionality of STARS.

Acknowledgments

This research is supported by a strategic grant from the Natural Science and Engineering Research Council (NSERC) of Canada.

References

1. E. Oomoto and K. Tanaka. OVID: Design and implementation of a video-object database system. *IEEE Transactions on Knowledge and Data Engineering*, 5(4):629—643, August 1993.

2. S. W. Smoliar and H. J. Zhang. Content-based video indexing and retrieval. *IEEE Multimedia*, 1(2):62—72, Summer 1994.
3. M. Flickner and et al. Query by image and video content: The QBIC system. *IEEE Computer*, 28(9):23—32, September 1995.
4. H. Wynne, T. S. Chua, and H. K. Pung. An integrated color-spatial approach to content-based image retrieval. In *Proceedings of The ACM International Multimedia Conference*, pages 305—313, San Francisco, CA, 1995.
5. V. E. Ogle and M. Stonebraker. Chabot: Retrieval from a relational database of images. *IEEE Computer*, 28(9):40—48, September 1995.
6. S. Marcus and V. S. Subrahmanian. Foundations of multimedia database systems. *Journal of ACM*, 43(3):474—523, 1996.
7. J. Smith and S. F. Chang. VisualSEEK: a fully automated content-based image query system. In *Proceedings of The ACM International Multimedia Conference*, Boston, MA, 1996.
8. A. Pentland, R. Picard, and S. Sclaroff. Photobook: Tools for context-based manipulation of image databases. In *Proceedings of SPIE Symposium*, pages 34—47, San Jose, CA, February 1994.
9. V. Oria, M. T. Özsu, L. Liu, X. Li, J. Z. Li, Y. Niu, and P. J. Iglinski. Modeling images for content-based queries: the DISIMA approach. Proceedings of The 2nd International Conference on Visual Information Systems (in press), December 1997.
10. A. Gupta, T. Weymouth, and R. Jain. The VIMSYS model. In *Proceedings of the 17th International Conference on Very Large Data Bases*, pages 297—308, Barcelona, Spain, 1991.
11. W. Kim. Multimedia information systems: Issues and approaches. In W. Kim, editor, *Modern Database Systems*, pages 5—17. Addison-Wesley, 1995.
12. J. Z. Li and M. T. Özsu. A new way of computing point-set topological spatial relations. Submitted to The First International Forum On Multimedia & Image Processing, May 1998.
13. V. N. Gudivada and G. S. Jung. An algorithm for content-based retrieval in multimedia databases. In *Proceedings of IEEE International Conference on Multimedia Computing and Systems*, pages 56—61, Hiroshima, Japan, June 1996.
14. H. J. Zhang, C. Y. Low, S. W. Smoliar, and J. H. Wu. Video parsing, retrieval and browsing: An integrated and content-based solution. In *Proceedings of The ACM International Multimedia Conference*, pages 15—25, San Francisco, CA, 1995.
15. A. Hampapur and R. Jain. Virage video engine. In *Proceedings of*

Storage and Retrieval for Images and Video Databases V, IS&T/SPIE Symposium on Electronic Imaging Science and Technology, pages 188—197, San Jose, CA, February 1997.

16. J. Z. Li, I. Goralwalla, M. T. Özsu, and D. Szafron. Modeling video temporal relationships in an object database management system. In *IS&T/SPIE International Symposium on Electronic Imaging: Multimedia Computing and Networking*, pages 80—91, San Jose, CA, February 1997.
17. R. Cattell, editor. *The Object Database Standard: ODMG-93*. Morgan Kaufmann, San Francisco, CA, 1994.
18. I. White and M. Goldberg, editors. *Using The Booch Method: A Rational Approach*. The Bejamin/Cummings Publishing Company, Inc., 1994.
19. C. Lamb, G. Landis, J. Orenstein, and D. Weinreb. The ObjectStore database system. *Communications of ACM*, 34(10):19—20, 1991.
20. C. H. Papadimitriou, D. Suciu, and V. Vianu. Topological queries in spatial databases. In *Proceedings of the 15th Symposium on Principles of Database Systems*, pages 81—92, Montreal, Canada, June 1996.
21. D. A. White and R. Jain. ImageGREP: Fast visual pattern matching in image databases. In *Proceedings of Storage and Retrieval for Images and Video Databases V, IS&T/SPIE Symposium on Electronic Imaging Science and Technology*, pages 96—107, San Jose, CA, February 1997.
22. J. Z. Li, M. T. Özsu, D. Szafron, and V. Oria. Moql: A multimedia object query language. In *Proceedgins of The Third International Workshop on Multimedia Information Systems*, Como, Italy, September 1997.
23. J. Z. Li, M. T. Özsu, and D. Szafron. Modeling of video spatial relationships in an object database management system. In *Proceedings of the International Workshop on Multimedia Database Management Systems*, pages 124—133, Blue Mountain Lake, NY, August 1996.
24. J. F. Allen. Maintaining knowledge about temporal intervals. *Communications of ACM*, 26(11):832—843, 1983.