Modeling Images for Content-Based Queries: The DISIMA Approach [*]

Vincent Oria, M. Tamer Özsu, Ling Liu, Xiaobo Li,
John Z. Li, Youping Niu, and Paul J. Iglinski
Department of Computing Science
University of Alberta
Edmonton, Alberta, Canada T6G 2H1
{oria, ozsu, lingliu, li, zhong, niu, iglinski }@cs.ualberta.ca

Abstract

The DISIMA project aims at building an image database system enabling content-based querying. The model, the architecture and a query language have been defined. The prototype is being implemented on top of the ObjectStore DBMS. DISIMA proposes a model for both image and spatial applications. The DISIMA model allows the user to assign different semantics to an image component (semantic independence) and an image representation can be changed without any effect on applications using it (representation independence). The architecture involves different image sources including WWW-servers and file systems. This paper presents the project overview.

1. Introduction

Computers have been primarily used for the management and manipulation of alpha-numeric data. In the last twenty years, database management system (DBMS) technology has matured and started to play a central role in the management of this type of data. The widening use of multimedia information systems has increased the need for efficient management of other types of data such as still images, video and audio. The DISIMA (DIStributed Image database MAnagement system) research project deals specifically with the development of technology to facilitate the management of and access to images using a database management system. Specifically, we propose to develop an image DBMS, which will provide the users with uniform access to multiple, historically separated, and possibly heterogeneous image and spatial repositories.

Our model addresses both image and spatial databases though they differ in significant ways. In spatial database systems, information is more structured. The DBMS has to deal with geo-referenced entities (e.g., cities, roads), attributes or specific properties of entities (e.g., population), and relationships between entities (distance, topological and directional relationships). Applications include geographical information systems, urban planning, distribution networks, and environmental applications. The prototype is being implemented on top of the ObjectStore [LLOW91] DBMS.

In an image database system, users want to query images using not only conventional textual annotation, but also image content which is hard to define. Content-based indexing is harder to build as the content of an image cannot be clearly defined. The query language must be sufficiently sophisticated to allow fuzzy search, and support high-level notions and relationships. There are a number of application domains that require an image database management system. Examples include office automation systems, medical and health-care information systems, and value-added telecommunication services.

The architecture has to be extensible to allow new features and new image processing functions to be added. Several image repositories are now available through World Wide Web (WWW) severs, multimedia databases or file systems. An image DBMS must provide tools to access, in an integrated manner, multiple image sources.

The remainder of this paper is organized as follows: Section 2 gives an overview of the DISIMA model, Section 3 presents the architecture, Section 4 shows how MOQL, an extension of OQL, can be used to query a DISIMA database, and Section 5 states the conclusion.

2. The DISIMA Model

A data model is defined as a collection of mathematically well-defined concepts to express both static and dynamic properties of data intensive applications. This section

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

presents how DISIMA represents the content of an image, the model components, and predicates. Our model has similarities to VIMSYS [GWJ91] in its layered view of image data.

2.1. The Model Components

The DISIMA model aims at efficient representation of images to support a wide range of queries. Typical queries that DISIMA would support include the following:

- Find images that look like this sample image;
- Find images that contain a given object;
- Find images that contains objects o_1 and o_2, and $R(o_1, o_2)$, where R expresses a spatial relationship (e.g., left, right, west, east).

These queries refer both to image and the objects within images, which we call *salient objects*. DISIMA model, as depicted in Figure 1, is composed of two main blocks: the image block and the salient object block. We define a block as a group of semantically related entities.

2.1.1. The Image Block

An image is a basic unit in DISIMA model.

Definition 1 *An image i is defined by a quadruple $< i, R(i), C(i), D(i) >$ where :*
- i is the unique (raw) image identifier;
- $R(i)$ is a set of representations of the raw image in a format such as GIF, JPEG,etc;
- $C(i)$ is the content of i as defined in Definition 3;
- $D(i)$ is a set of descriptive alpha-numeric data associated with i.

The image block is made up of two layers: the *image* layer and the *image representation* layer. As already mentioned, 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 an image type classification similar to object type systems. This layer allows the user to define functional relationships between images. In Figure 2, we give the type of graph for an application which manages news and medical images. These images can be classified according to specific criteria. The *NewsImage* class is specialized by three classes: one for images where a person has been identified (*PersonImage*), another for nature images (*EnvironmentalImage*) and the last one for the others (*MiscImage*).

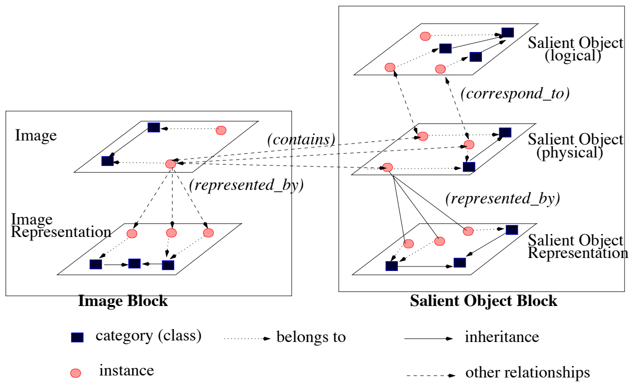

Figure 1. DISIMA Model Overview

Two main representation models exists: the *raster* and the *vector*. Raster representations are mostly used by image applications while vector representations fit well with spatial applications. For the raster representations, we provide the JPEG format as default, while spaghetti modeling [LT92] is used for vector representations.

2.1.2. The Salient Object Block

The main problem in an image database system is to capture the *content* of an image, information which is hard to describe. One way to represent the content is a text description, which does not help too much, as there may be radically different textual descriptions provided by different people. Moreover, it is also difficult for a computer to capture the semantics of a text. DISIMA views the content of an image as a set of *salient objects* (i.e., interesting entities in the image) with certain spatial relationships to each other. The *salient object* block is designed to handle salient object organization.

For a given application, salient objects are known and can be defined. The definition of salient objects can lead to a type lattice as in Figure 2. DISIMA distinguishes two

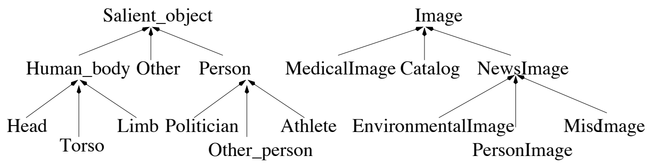

Figure 2. An Example of Image and Logical Salient Object Hierarchies

kinds of salient objects: physical and logical salient objects. A *logical salient object* is an abstraction of a salient object that is relevant to some application. For example, an object may be created as instance of type *Politician* to represent

President Clinton. The object Clinton is created and exists even if there is yet no image in the database in which President Clinton appears. This is called a logical salient object; it maintains the generic information that might be stored about this object of interest (e.g., name, position, spouse)

Particular instances of this object may appear in specific images. There is a set of information (data and relationships) linked to the fact that "Clinton appears in image i_1". The data can be his posture, his localization, or his shape in the image i_1. Examples of relationships are spatial relationships with regards to other salient objects belonging to image i_1. We create a *physical salient object*, (P_objectClinton_1) linked to logical salient object object-Clinton, that refers to image i_1 and gives the additional information. If President Clinton is found in another image i_2, another physical salient object (P_objectClinton_2) will be created.

We now give a formal definition of the content of an image, using physical an logical salient objects.

Definition 2 *A physical salient object is a part of an image and is characterized by a position (i.e., a set of coordinates)in the image space.*
A logical salient object is an object that is used to give semantics to a physical salient object.

Additional properties such as color, texture and shape can be defined for a physical salient object. The shape of a physical salient object can have several representations.

Definition 3 *Let $\mathcal{L}$ be the set of all logical salient objects and $\mathcal{P}$ be the set of all physical salient objects. The content of an image i is defined by a pair $C(i) =< \mathcal{P}^i, s >$ where:*

- $\mathcal{P}^i \subseteq \mathcal{P}$;
- $s : \mathcal{P}^i \longrightarrow \mathcal{L}$: maps each physical salient object to a logical salient object

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 Clinton appears". The query "find all the images in which Clinton is between Arafat and Netanyahu" requires the physical salient object level to check the spatial relationships among the three salient objects. The processing can indeed benefit from the existence of spatial indexes.

Similar to the images, the representation of salient objects is separated from their content information. The representation can be raster or vector. Raster representations of salient objects are useful to access part of images, while vector representations can be used for spatial indexing and spatial relationships computation.

The facility for creating user-defined hierarchies of images and salient objects provides a more flexible and powerful modeling mechanism than having a fixed image hierarchy. Two salient objects belonging to the same image can have different data members. Let us take an image in which we have a well known politician shaking the hand of an athlete. For both of them we may want to know who and what they are. For the politician we may want to add the party he belongs to, while for the athlete we will be interested in the sport he practices. For this purpose, we have created sub-classes (Athlete, Politician) in Figure 2.

2.2. Defining a DISIMA schema

A DISIMA schema is made up of two sub-schemas: an image schema and a salient object schema.

Definition 4 *Let ITG(Image Type Graph) be an image type hierarchy and STG(Salient Object Type graph) be a salient object type hierarchy. A DISIMA schema is defined by (ITG, STG).*

For some pure image applications in which there is no need for spatial searches, images and salient objects may not need to have vector representations. On the other hand, for some pure spatial applications there may not be any raster representations for either images or salient objects. In this case, an image object will, in fact, be a map which does not need any image processing function.

2.3. Image and Salient Object Predicates

Predicates are used in databases to define the characteristics of the objects one is looking for. In classical databases, predicates are limited to $\{=, >, <, \geq, \leq\}$. However, these predicates are not sufficient to handle spatial and functional relationships between salient objects and to compare images. We provide a set of image comparison and salient object relationship predicates.

2.3.1. Salient Object Relationship Predicates

We have two levels of salient object predicates. The first level corresponds to the logical salient object level in the model. We define two predicates: *equal* and *compatible*.

Definition 5 *Two salient objects O_1 and O_2 are equal if $OID(O_1) = OID(O_2)$ (OID: object identifier).*

Definition 6 *Two salient objects O_1 and O_2 are compatible if $type(O_1)$ is an ancestor of $type(O_2)$ or $type(O_2)$ is an ancestor of $type(O_1)$ in the salient object hierarchy graph.*

The second level of salient object predicates corresponds to the physical salient object level. We define some *distances* using color, texture or shape. We also define here the well-known spatial predicates [EF91, CO95]. Spatial relations have been classified into several types, including *topological relations* that describe neighborhood and incidence (e.g., overlap, disjoint); *directional relations* that describe order in space (e.g., south, northwest); and *distance relations* that describe space range between objects (e.g., far, near).

To model spatial relationships, Each physical salient object is represented by a minimum bounding rectangle (MBR) whose projections are taken on the x and y axes. These projections provide intervals which are then mapped to temporal object model that supports Allen temporal interval algebra [All83]. Allen's algebra provide seven basic relations to reason about intervals. There are a number of advantages to representing spatial relationships among objects in terms of intervals that are then mapped to a temporal model designed to handle intervals as primitives. First, interval algebra provides a sufficiently powerful infrastructure to represent spatial relationships. Since temporal models are already equipped to handle intervals properly, no special system infrastructure needs to be built for spatial relations. Second, when DISIMA is extended to deal with video, temporal relationships in addition to spatial ones need to be represented in the database. Using a temporal model for spatial relations provides a uniform basis for the modeling of both types of relations in the database. Finally, the temporal model enables us to build a spatial reasoning system using the interval algebra [LÖS96].

The implementation based on Allen temporal logic includes topological spatial relationships but works only on minimum bounding rectangles. This step has to be seen as a filtering phase and has to be complemented by another computation working on physical salient objects shapes. For example, the MBR's of two physical salient objects might overlap, but analysis of their shapes may reveal that the objects are actually disjoint.More processing algorithms and indexes are being investigated to capture salient objects, color, texture, shapes, and similarity searches.

2.3.2. Image Comparison Predicates

Due to the the difficulty of representing the content of an image, it is also difficult to compare images. What does "image A is similar to image B" mean? This problem is known in the literature as nearest neighbor or similarity search in image databases. Several proposals exist based on the definition of distance using image features [KSF+96, AFS93, GM92].

We provide the *contains* predicate that checks if an image contains a given salient object. We also provide some *distances* using color and texture for the whole image. Assume we are comparing images i_1 and i_2. We can do this in three steps such that step 2 refines step 1 and step 3 refines step 2:

- step 1 - semantic checking: i_1 (or a sub-image of i_1) contains (equal or compatible to) all or part of i_2 salient objects.
- step 2 - feature checking: with or without the same color and/or the same texture and/or the same shape.
- step 3 - spatial relationships checking: with or without the same spatial relationships.

Combining distances defined on whole images and the *contains* predicate, with predicates and distances defined for salient objects, leads to several similarity predicates.

2.3.3. Indexing Images Using $2D$-h Trees

To facilitate querying over spatial relationships, we have developed an indexing technique called $2D$-h tree [NÖL97], based on the 2-D-String techniques [CSY87]. 2-D-Strings are symbolic representations of spatial relationships of salient objects along the x and the y axes.

Standard 2-D-String indexing is suitable for substring matching for a small number of images, but not for large image databases since subsequence matching is done sequentially. We have designed a new indexing technique which builds a B^+-tree type index on top of 2-D-Strings. The non-leaf nodes have a similar layout to non-leaf nodes in a B^+ tree, while the leaf node layout is different. Each non-leaf node in the $2D$-h tree contains $q + 1$ pointers and q entries (q is the threshold). Each pointer points to a child node, which is the root of a sub-tree of the $2D$-h tree. We developed the searching, inserting and deleting algorithms that can be used to efficiently retrieve resulting 2-D Strings from the tree and update the tree. A performance analysis was conducted, and both analytical analysis and experimental results indicate that the $2D$-h tree is an efficient index structure for image retrieval based on directional relationships.

In addition to significant performance improvement, $2D$-h tree indexing incorporates more spatial relationships between salient objects, thereby facilitating more complex spatial queries. We are in the process of extending this indexing technique to video databases.

3. The DISIMA Architecture

This section first focuses on the basic components of a DISIMA single site architecture (Figure 3) and then the distributed one. The single site architecture is composed of the interfaces, the meta-data manager, the image and salient object manager, the image and spatial index manager, and the object index manager. The interfaces provides several ways (visual and alpha-numeric) to define and query image data. The data definition language (DDL) used for the DISIMA project is C++ODL [Cat94] and the query language is an extension of OQL. DISIMA API is a library of low-level functions that allows applications to access to the system services. DISIMA is to be built on top of object repositories (ObjectStore for the current prototype). These object repositories may not have image and spatial indexes. And the object-oriented indexes they provide (if any) may not fit with DISIMA requirements. Moreover, the image and spatial index manager and the object index manager have to dynamically integrate new indexes. The meta-data manager handles meta-information about images and salient objects. The salient object manager implements the DISIMA model, and index managers allow dynamic index management. Indexes include object, image and spatial indexes.

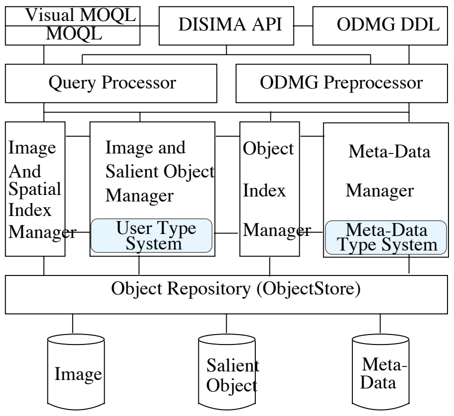

Figure 3. The DISIMA Architecture

3.1. Meta-data Manager

Meta-data is important to improve the availability and the quality of the information to be delivered. It is a kind of on-line documentation. In some image applications [AS94], everything except the raw image data is considered meta-data. Based on object-oriented concepts, the DISIMA model integrates the raw image and the alpha-numeric data linked to it. Meta-data here, stands for meta-types and other types defined to keep track of all classes and behaviors (methods) defined by the user. The idea is to simulate on top of ObjectStore meta-types (i.e. types of types) that will be used by users for schema definitions. DISIMA views the main components of the user-defined schema as objects that can be stored and queried using OQL.

The DISIMA meta-data model separates the definition of object characteristics (a *type*) from the mechanism for maintaining instances of a particular type (a *class*). For example, `T_Image` refers to a type, `C_Image` to its class. In this paper, a reference prefixed by "`T_`" refers to a type, "`C_`" to a class, "`B_`" to a behavior. Meta-types are `T_Type` for type instantiations, `T_Class` for class instantiations with `T_Behavior` for behaviors. Each type is associated with a class. Figure 4 shows an image instance $image_1$ which is of type `T_Image` and belongs to the class `C_Image`. `C_Image`, itself an object, is of type `T_Class` and belongs to `C_Class`. Some of the meta-type relationships are circular. As `C_Class` is a class, it is an instance of itself. Similarly, `T_Type` is of its own type. The Meta-data type system is composed of the three above-mentioned kinds of meta-types and other types for system management purposes.

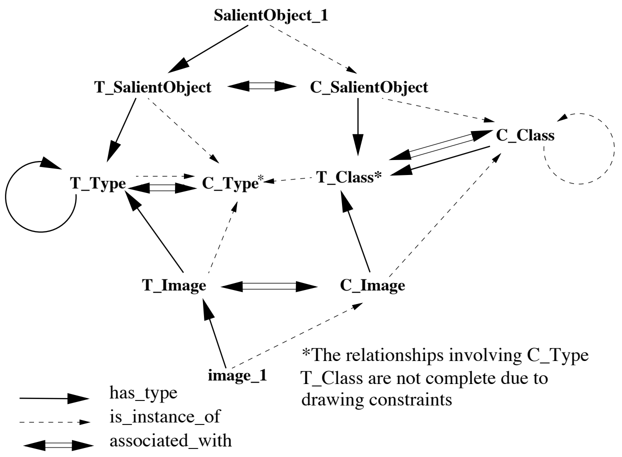

Figure 4. A Class and Instance Structure for Images and Salient Objects

As defined, the The DISIMA meta-data is independent of any underlying object repository. And, as everything is defined as an object, the same query language and the same query processor can be used to query both data and meta-data.

3.2. Image and Salient Object Manager

The image and salient object manager implements the DISIMA model. It contains a set of root types (the user type system) for user-defined image schemas to be derived from. The image root type provides tools to help recognize salient objects. Salient object recognition combines manual and automatic interpretation of images. The automatic interpretation processes several pattern recognition and image analysis algorithms to capture the content of an image. The manual interpretation process complements the automatic one. The user describes the objects he can identify in the image. He links them to predefined logical salient objects. Then, physical salient objects are created for the image components.

The image and salient object manager gives to the user the opportunity to run applications using a classical transaction mode (i.e., with the ACID properties) and a read-only mode. The read-only mode can be used during image object recognition to allow several users to share the logical salient objects. The image and salient object manager watches over the consistency of the stored information.

3.3. The DISIMA Interoperable Architecture

DISIMA provides an interoperable architecture to allows users to query multiple and possibly remote image sources. The DISIMA users may use MOQL [LÖSO97] as a uniform interface. Due to the autonomy of each individual image source, DISIMA needs to have a wrapper built for each participating image data source; the wrapper is responsible for transforming MOQL queries into local queries executable at the individual site. When the individual image source is using a database (relational or object-oriented) model, the transformation is simpler. However, translating content-based queries is not straightforward. When the image source is modeled as file system or hyperlinks, the wrapper needs to provide search capabilities to support queries.

The interoperable architecture is designed using common facilities as defined in the Object Management Architecture (OMA) [Gro93]. CORBA provides transparencies at the platform and the communication levels. There remain two other levels, the database level where different data models can be found and the semantic level to homogenize the meanings of the objects. The distribution architecture involves homogeneous systems and heterogeneous systems. DISIMA builds wrappers that can transform MOQL queries into target queries executable at the target sources for the heterogeneous case.

The mediator provides a global view on data stored in different sources. Although schema integration in general is still an open problem, it is made more difficult by the fact that DISIMA is dealing with images. Two schemas may be totally different with regard to their definitions and yet address the same kind of images from an image content point of view. This is a research problem we are investigating. The interoperable architecture follows the general mediator framework [Wie92] and is given in Figure 5. Wrappers convert queries from each source to MOQL and can be accessed directly.

3.3.1. The DISIMA Integration Mediator

For some sources organized only as image collections, the user may want to extract the features and store them as an auxiliary image feature base in a DISIMA site using the DISIMA mediator. The resulting database is called an auxiliary image feature base. In this case, it will be better to have the DISIMA image extraction tools running on the site where the images are stored. This auxiliary base uses the

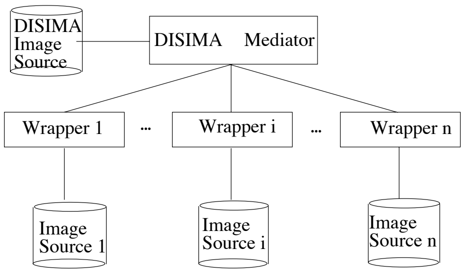

Figure 5. Mediator-Wrapper Architecture

DISIMA model to represent image content and associated data. The difference between an auxiliary image feature base and a DISIMA base is that the auxiliary base does not contain image representations. Only image identifiers are available. When sites involved provide schemas, they are integrated into an *image catalog*. The mediator refers to the image catalog for query decomposition and result assembly (Figure 6).

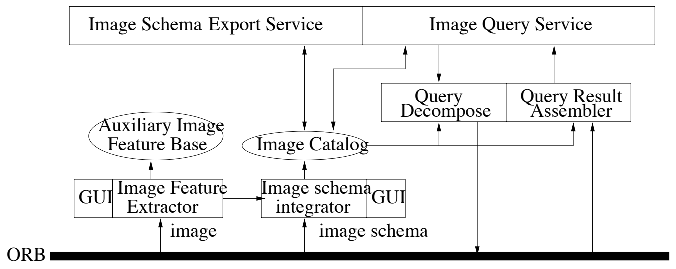

Figure 6. The DISIMA Integration Mediator Architecture

3.3.2. The DISIMA Wrappers

The wrapper has to export the local schema, that is, to translate the local schema using C++ODL or IDL. It also has to translate queries from MOQL into the *ad hoc* language with more or less loss of expressiveness. The wrapper internal structure is given Figure 7.

Global information has to be available to all the sites. It is used by the wrappers during query translations. The challenge will be to find a global solution to integrate file collection hierarchies, hyper-links and schemas based on the content of the images. Another challenge is to find a powerful compression format for raw image data transfer.

4. Querying Images

The user type system in the image and salient object manager provides root types for images and salient objects.

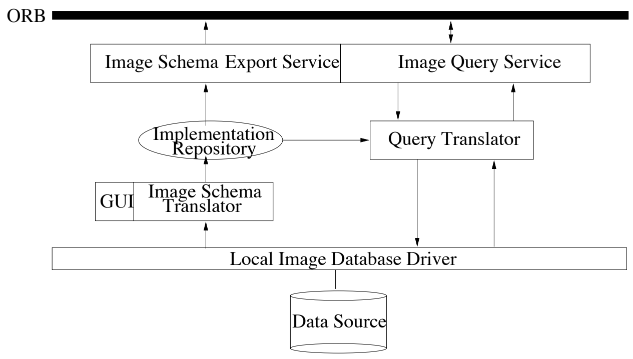
ORB
Image Schema Export Service
Image Query Service
Implementation Repository
Query Translator
GUI
Image Schema Translator
Local Image Database Driver
Data Source

Figure 7. The DISIMA Wrappers

Given a logical salient object and an image, the physical salient object is determined. Therefore, the end-users do not have to be aware of the two levels of salient objects. Behaviors have been defined in the (logical) salient object root type to give the shape (region), the color, and the texture when the image in known. Knowing salient object regions, it is easy to compute their minimum bounding rectangles and corresponding intervals on the axis for spatial reasonning.

The MOQL [LÖSO97] approach extends the standard object query language, OQL [Cat94]. MOQL has been designed for general multimedia queries. It captures temporal and spatial relationships. Most of the extensions have been introduced in the **Where** clause in the form of new predicates. Queries below show how MOQL can be applied to images stored in DISIMA. The queries are based on a schema defined for the application in Figure 2.

Query 1 Find all images in which a person appears.
select *m*
from Images *m*, Persons *p*
where *m* **contains** *p*

Query 2 Find all images in which President Clinton appears.
select *m*
from Images *m*, Politicians *p*
where *p.name* = "Clinton"
and *m* **contains** *p*

Query 3 Find all images in which Clinton wears black.
select *m*
from Images *m*, Politicians *p*
where *p.name* = "Clinton"
and *m* **contains** *p*
and *p.color(m)* = "black"

Query 4 Find all images in which Clinton is between Arafat and Netanyahu.
select *m*
from Images *m*, Politicians *p1*,Politicians *p2*, Politicians *p3*
where *p1.name* = "Clinton"
and *p2.name* = "Arafat"
and *p3.name* = "Netanyahu"
and *m* **contains** *p1*
and *m* **contains** *p2*
and *m* **contains** *p3*
and ((*p1.region(m)* **left** *p2.region(m)*
and *p1.region(m)* **right** *p3.region(m)*)
or (*p1.region(m)* **right** *p2.region(m)*
and *p1.region(m)* **left** *p3.region(m)*))

Query 5 Find all images that look like an image in which Clinton appears.
select *m1*, *m2*
from Images *m1*, Images *m2*, Politicians *p*
where *p.name* = "Clinton"
and *m1* **similar** *m2*
and *m1* != *m2*

The implementation of the *similar* predicate is application dependent. The current implementation of the *similar* predicate in the DISIMA project involves salient objects, color, and spatial relationships among salient objects. However, one of our objectives is to allow to user to set the *similar* predicate to the desirable checking (semantic checking, feature checking and spatial relationship checking). The query processing benefits from the $2D$-h tree.

5. Conclusion

Most of image DBMS do not address spatial applications and some of them are very specialized. In the DISIMA project we think that an image DBMS can be built on top of spatial repositories and then be accessed by both kinds of applications. The DISIMA project aims at developing a model, capturing the behavior and characteristics of image and spatial applications. An image is viewed as being composed of salient objects. The DISIMA model allows independence between image representations and applications (representation independence) and distinguishes the existence and identity of salient objects from their appearance in an Image (semantic independence). These independencies are achieved by introducing an abstraction between an image and its representation and also between a salient object and its appearance in an image.

The DISIMA model is composed of two blocks. The first block deals with salient objects and the second block implements an image model on top of the first one. The salient object block provides tools and predicates to manage user-defined salient object hierarchies. This layered model makes DISIMA easily extensible to handle video data by adding a video block. The video block will use all the functionalities defined on images and salient objects, as the basic element in a video is an image (called a frame). The management of this multi-level modeling requires that DISIMA offers algorithms for quick access to multi-level storage hierarchies.

The architecture provides index managers to allow specific indexes to be added dynamically. The user can add new indexes for both image data and meta-data. Several user interfaces combine query language, browsing, navigation, and application development tools. Several image data sources exist. The distributed DISIMA architecture, using CORBA as its platform, includes multimedia sites as well as WWW-servers and file collections.

We have defined a complete query langage, MOQL, for the DISIMA project. However one of our research objectives is to study algebraic primitives to support optimization of multimedia queries. In our current prototype, each algebraic operator is implemented in terms of ObjectStore functions. This establishes a link between the query processor and the underlying object repository, while enabling independent development of the query processor.

References

[AFS93] R. Agrawal, C. Faloutsos, and A. Swami. Efficient similarity search in sequence databases. In *Foundations of Data Organization and Algorithms (FODO)*, Evanston, Illinois, October 1993.

[All83] J. F. Allen. Maintaining knowledge about temporal intervals. *Communications of ACM*, 26(11):832—843, 1983.

[AS94] J. T. Anderson and M. Stonebraker. Sequoia 2000 metadata schema for satellite images. *SIGMOD RECORD*, 23(4):42—48, December 1994.

[Cat94] R. Cattell. *The Object Database Standard: ODMG-93 (Release 1.1)*. Morgan Kaufmann, San Francisco, CA, 1994.

[CO95] J. P. Cheiney and V. Oria. Spatial database querying with logic languages. In *Proceedings of Fourth International Conference on Database Systems for Advanced Applications (DASFAA'95)*, Singapore, April 1995.

[CSY87] S. K. Chang, Q Shi, and C. Yan. Iconic indexing by 2d-string. *IEEE Transactions on Pattern Analysis and Machine intelligence*, 9:413—428, May 1987.

[EF91] M. Egenhofer and R. Franzosa. Point-set topological spatial relations. *International Journal of Geographical Information Systems*, 5(2):161—174, 1991.

[GM92] Gary and Mehrotra. Shape similarity-based retrieval in image database systems. In *SPIE'92*, 1992.

[Gro93] Object Management Group. The common object request broker: Architecture and specification. *OMG Document*, (93.12.43), December 1993.

[GWJ91] A. Gupta, T. Weymouth, and R. Jain. An extended object oriented data model for large spatial databases. In *Proceedings of the 2nd International Symposium on Design and Implementation of Large Spatial Databases (SSD)*, Barcelona, Spain, September 1991.

[KSF$^+$96] F. Korn, N. Sidiropoulos, C. Faloutsos, E. Siegel, and Z. Propopapas. Fast nearest neighbor search in medical image. In *Proceedings of the 22nd International Conference on Very Large Databases, VLDB*, Bombay, India, September 1996.

[LLOW91] C. Lamb, G. Landis, J. Orenstein, and D. Weinreb. The objectstore database system. *Communications of ACM*, 34(10):19—20, 1991.

[LÖS96] J. Z. Li, M. T. Özsu, and D. Szafron. Spatial reasoning rules in multimedia management systems. In *Proceedings of the International Conference on Multimedia Modeling MMM'96*, Toulouse, France, November 1996.

[LÖSO97] J. Z. Li, M. T. Özsu, D. Szafron, and V. Oria. MOQL: A multimedia object query language. In *Proceedings of the 3rd International Workshop on Multimedia Information Systems*, Como, Italy, September 1997.

[LT92] R. Laurini and D. Thompson. *Fundamentals of Spatial Information Systems*, volume The APIC Series. Academic Press, 1992.

[NÖL97] Y. Niu, M. T. Özsu, and X. Li. $2D-h$ Tree: An index scheme for content-based retrieval of images in multimedia systems. In *IEEE International Conference On Intelligent Processing Systems 1997 (IEEE ICIPS'97)*, Beijing, China, October 1997.

[Wie92] G. Wiederhold. Mediators in the architecture of future information systems. In *IEEE Computers*, pages 38—49, March 1992.