



UNIVERSITY OF ALBERTA

# DISIMA: A DISTRIBUTED IMAGE DATABASE MANAGEMENT SYSTEM

---

FINAL REPORT –  
NSERC STRATEGIC GRANT STR181014

23 APRIL 2000

---

## TABLE OF CONTENTS

---

<b>INTRODUCTION.....</b>	<b>3</b>
<b>SUMMARY OF TECHNICAL WORK.....</b>	<b>4</b>
DISIMA ARCHITECTURE AND MODEL.....	4
SYSTEM REQUIREMENTS AND ARCHITECTURE.....	4
DISIMA MODEL.....	7
CONTENT MODELING OF IMAGES.....	9
VIEWS ON IMAGE DATABASES.....	10
QUERYING THE IMAGE DATABASE.....	11
INDEXING OF THE IMAGE DATABASE.....	14
2-D-S TREES.....	14
DYNAMIC BINTREE (DB-TREE):.....	15
COLOR SIMILARITY INDEXING.....	16
IMAGE PROCESSING AND OBJECT RECOGNITION.....	16
IMAGE CODING.....	17
<b>PROTOTYPE DEVELOPMENT AND TECHNOLOGY TRANSFER.....</b>	<b>18</b>
<b>TRAINING OF RESEARCH PERSONNEL.....</b>	<b>20</b>
<b>PUBLICATIONS.....</b>	<b>21</b>
<b>CONCLUSIONS.....</b>	<b>23</b>
<b>OTHER REFERENCES.....</b>	<b>24</b>

---

## INTRODUCTION

---

This is the final report of the NSERC Strategic Grant STR181014 entitled “DISIMA: A Distributed Image Database Management System”. The project started on 1 November 1995, and funding for it terminated on 1 November 1999. The final set of tasks will be completed by May 2000, at which point this phase of the research will terminate.

The project was highly successful, having fulfilled all of its research objectives as described below. It involved three Ph.D. students in various capacities, four M.Sc. students, one post-doctoral fellow, and two research associates. We have given demonstrations at three conferences and a fourth one will be given in mid-May. Most importantly, it has opened up new lines of research that we intend to follow in the future.

Three investigators initially headed the research project: Prof. M. Tamer Özsu (Principle Investigator and Applicant), Prof. Xiaobo Li, and Prof. Ling Liu. Prof. Liu took leave of absence from the University of Alberta on July 1, 1997 and then resigned her position as of July 1, 1998. Thus, her involvement with the research was restricted to the first year of the project.

The total funding for the project was \$327,620, with a distribution according to years as depicted in Table 1. The funds were completely spent. To complement the funding and the research scope, the Principle Investigator obtained funding from IRIS NCE on “Image Database Systems”. That funding was used for complementary work on image indexing.

FUNDING YEAR	REQUESTED	GRANTED
1995-1996	\$66,300	\$66,800
1996-1997	\$84,600	\$82,160
1997-1998	\$84,600	\$99,500
1998-1999	\$84,660	\$82,160

*Table 1. Funding request and allocation*

## RESEARCH OBJECTIVES

The primary objective of this project was to design and develop a distributed image DBMS, which, on the one hand, provides mechanisms and techniques for efficient image compression, fast image transmission, and flexibility in content-based querying of images; and, on the other hand, provides a convenient and uniform interface for global users to access image data from multiple, disparate source repositories. More specifically, the following objectives were established at the outset:

- 1) Design of an interoperable integration framework based on the object-oriented methodology for distributed image database management, which provides typical DBMS services for uniform access to multiple, disparate image repositories.
- 2) Development of new image database indexing techniques to support flexible access to the images from multiple source repositories, and to enable global users to query remote image databases beyond the set of pre-defined query patterns.
- 3) Exploration and development of powerful and efficient image compression algorithms that enable fast and high-quality compression of images.

- 4) Provision of facilities to support progressive transmission and selective transmission of images across networks.

While we worked on all of these objectives to a certain extent, the progress of the research dictated revisions in our objectives as we went along. In particular, the database aspects of image management (objectives 1 and 2) became more of a focus, and took more of our time and effort, than the image processing issues (objectives 3 and 4). Within the context of image processing, our focus shifted slightly outside the originally stated objectives. It became apparent that to facilitate the database processing of images, we needed to make progress on fundamental aspects of image segmentation and recognition. We, therefore, worked on face recognition, as our driving application was a news image repository consisting of images containing people.

---

## SUMMARY OF TECHNICAL WORK

---

Even though the full technical details of the DISIMA system cannot be described in this report, we describe, in this section, the main research accomplishments, and indicate our progress towards the milestones that we had identified. The details can be found in the publications that have resulted from this research.

## DISIMA ARCHITECTURE AND MODEL

### SYSTEM REQUIREMENTS AND ARCHITECTURE

*Milestone 1: Definition of specific model and architecture of the image DBMS, and the determination of the requirements (April 1995)*

The project involved an extensive study of the requirements of a next generation image DBMS. The principles of image modeling are identified as follows:

- 1) Multiple representations of an image (GIF, JPEG, ...) can exist. The system has to be able to deal with these multiple representations.
- 2) Images are categorized according to a user-defined image types hierarchy. This allows for reuse among image characteristics, in addition to their classification.
- 3) An object-oriented image processing library is attached to the image DBMS which defines operations that can be performed on an image.
- 4) An image can have (semantic, functional, spatial) relationships with other images (or documents), which should be represented in the DBMS.
- 5) An image is composed of *salient objects* (image components). These are objects in an image that are of interest and, therefore, need to be individually identified. Multiple representations of a salient object (grid, vector) are allowed. As with images, salient objects are classified according to a user defined type hierarchy.

Consideration of the above requirements were essential in defining both the system architecture (discussed below) and the image database model (described in the next subsection). The architecture of the single-site version of the system that is now operational is shown in Figure 1.

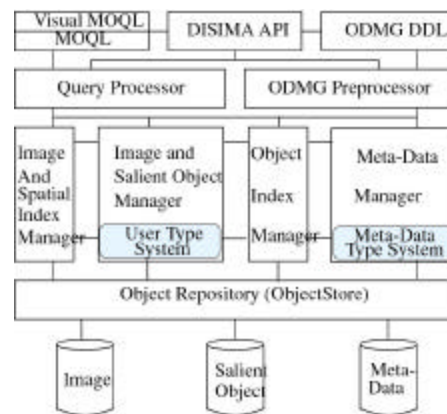


Figure 1. DISIMA Architecture

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 provide several alternatives (visual and alpha-numeric) for defining and querying image data. The data definition language (DDL) used for the DISIMA project is C++ODL [Cat94]<sup>1</sup> and the query language is an extension of OQL. DISIMA API is a library of low-level functions that allows applications to access 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, while the salient object manager implements the DISIMA model, and index managers allow dynamic index management. Indexes include object, image and spatial indexes.

The image and salient object manager implements the DISIMA model that is discussed in the next subsection. 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, and links them to predefined logical salient objects. Then, physical salient objects are created for the image components. The image and salient object manager gives 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.

<sup>1</sup> In this report, citations of the type [Cat94] are to outside publications that are listed in the REFERENCES section, while citations of the type [1] are to our publications that resulted from the DISIMA project, and that are listed in the PUBLICATIONS section.

Meta-data is important for improving the availability and 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.

*Milestone 4: Determination of the interoperability requirements of image storage systems (March 1996).*

*Milestone 6: Definition of the interoperability architecture and first attempt at linking multiple image storage systems (March 1998).*

In addition to the functionality of the single-site system discussed above, DISIMA provides an interoperable architecture (Figure 2) to allow users to query multiple and possibly remote image sources. DISIMA users may use MOQL (Multimedia OQL) [5] 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.

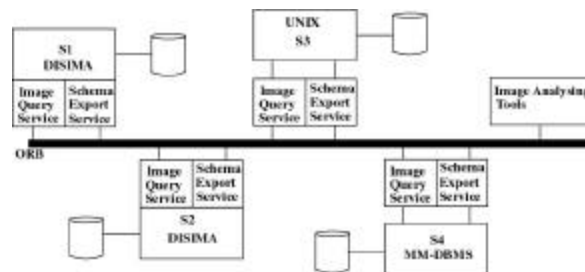


Figure 2. Distributed DISIMA Architecture

The interoperable architecture is designed using common facilities as defined in the Object Management Architecture (OMA) [OMG93]. 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.

*Milestone 3: Develop a complete type system for images and first prototype implementation (March 1996).*

A data model is defined as a collection of mathematically well-defined concepts that express both static and dynamic properties of data intensive applications. Within an image DBMS, this corresponds to the modeling of images and their content. This section (intuitively) presents how DISIMA represents the content of an image, the model components, and predicates. For more formal discussions of the model, we refer to publications [1, 3, 4, 8, 13].

The DISIMA model, as depicted in Figure 3, 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. The image block corresponds to individual images, while the salient object block models the contents of images. This separation allows us to be able to pose queries on both the images themselves (e.g., “Find images that are similar in color to this sample image.”) and on their contents (e.g., “Find images that contain a given object.” or “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).”)

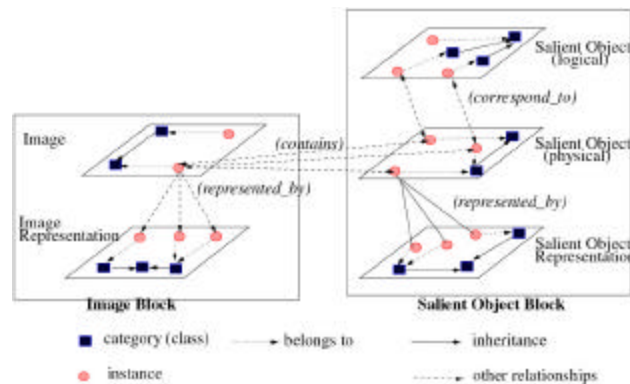


Figure 3. DISIMA Model

The image block is made up of two layers: the *image layer* and the *image representation layer*. We distinguish an image from its representations in order to maintain *representation independence*, whereby the same image can be stored in different representations at the same time.

In the image layer, the user defines an image type classification similar to type systems of object databases. This layer allows the user to define functional relationships between images. Figure 4 depicts, as an example, the type of graph for an application which manages digitized news photographs and medical images. These images can be classified according to specific criteria. The NewsImage class is specialized into three classes: one for images where a person has been identified (PersonImage), another for nature images (EnvironmentallImage) and the last one for the others (MiscImage).

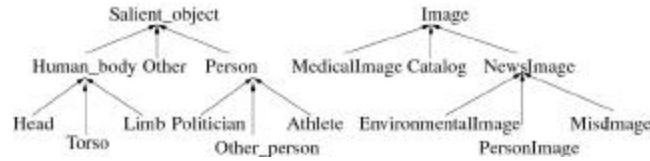


Figure 4. An Example of Image and Logical Salient Object Hierarchies

Two main representation models exist: 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; spaghetti modeling [LT92] is used for vector representations.

The main problem in an image database system is to capture the *content* of an image – information that 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 of the same image provided by different people. Moreover, it is difficult for a computer program 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, the application developer can define salient object classes. The definition of salient objects can lead to a type hierarchy. This allows for user-defined classification of important objects in an image.

DISIMA distinguishes two 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 an instance of type `Politician` to represent Jean Chrétien. The object `L_objectChretien` is created and exists, even if there is yet no image in the database in which Jean Chrétien 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 “Jean Chrétien appears in image  $i_1$ ”. The data can be his posture, his localization, or his shape in this particular image. Examples of relationships are spatial relationships with regards to other salient objects belonging to image  $i_1$ . We create a *physical salient object*, (`P_objectChretien_1`) linked to logical salient object `L_objectChretien`, that refers to image  $i_1$  and gives the additional information. If Jean Chrétien is found in another image  $i_2$ , another physical salient object (`P_objectChretien_2`) will be created.

The two levels of salient objects ensure the semantic independence and multi-representation of salient objects. Even though queries are answered using the physical salient objects, the semantic independence gives different semantics to the same physical salient object using the view mechanism described later. Even though we do not enforce it in the current implementation, it is possible to associate logical salient objects with images in order to be able to answer queries using both logical and physical salient objects. In that case, only the logical salient object level would be used to answer, for example, the query “Find all the images in which Chrétien appears.” The query “Find all the images in which Chrétien is next to Castro”, on the other hand, would be answered by means of the physical salient object level so as to check the spatial relationships between the two salient objects.



As is the case with 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 parts 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.

### CONTENT MODELING OF IMAGES

*Milestone 7: Incorporation of image indexing and content-based access features of the image DBMS (March 1998).*

To achieve the goals of content-based querying of image databases, it was essential to find a suitable content-based model of images. This means that methods had to be developed to represent the physical salient objects in an image, and their spatial relationships. In DISIMA, we represent salient objects with their minimum bounding rectangles (MBRs). MBRs have been used extensively to approximate objects, because only two points are required for their representation. However, MBRs demonstrate some disadvantages when approximating non-convex or diagonal objects. DISIMA, therefore, also supports a more precise and more expensive computation of spatial relationships, based on the geometry of the objects [10]. Consequently, searches over DISIMA databases can be conducted in two steps: the first filtering step is performed using MBR representation of salient objects, while in the second step, finer testing of spatial relationships among salient objects is performed, using the point set representation.

To facilitate spatial querying over the database (e.g., “Find all images which show Jean Chrétien next to Bill Clinton.”), it is necessary to capture and represent the spatial relationships among salient objects. These relationships can be in the form of *topological relationships* or *directional relationships*. Topological ones are neighborhood and incidence relationships, while directional ones represent order in space.

It is well known that eight fundamental topological relations can hold between two planar regions [EF91]. These relations are computed using four intersections over the concepts of boundary and interior of point sets, between two regions embedded in two-dimensional space. The four intersections result in eight topological relations: equal, inside, contains, cover, covered\_by, overlap, externally connected, and disjoint. Among these, contains and inside are inverses of each other, as are cover and covered\_by. Therefore, we work with six basic topological relations.

We consider 12 directional relations in our model, and classify them into the following three categories: *strict directional relations* (north, south, west, and east), *mixed directional relations* (northeast, southeast, northwest, and southwest), and *positional relations* (above, below, left, and right).

To uniformly model these relationships, we use Allen’s temporal interval algebra [All83]. The elements of the algebra are sets of seven basic relations that can hold between two intervals, and their inverse relations: B before C, B meets C, B overlaps C, B during C, B starts C, B

finishes  $C$ , and  $B$  equal  $C$ . Even though Allen has defined these for temporal intervals, they can also be used for spatial intervals. In our setting, the spatial intervals correspond to those obtained by taking the projection of a salient object's MBR over the  $x$  and  $y$  axes. The interval algebra essentially consists of the topological relations in one-dimensional space, enhanced by the distinction of the order of the space. It is sufficiently powerful to capture all of the topological and directional relationships that are supported in DISIMA [3].

The choice of an interval algebra as the basis for representing spatial relationships enables us to have a system kernel that provides algebraic primitives for manipulating intervals without concern for what these intervals represent. This is important, since we would like the system to be extendable to handle video data as well, even though this was not part of the project's mandates. In video modeling and querying, it is important to be able to represent the temporal relationship of clips and frames (e.g., "Find the clip that shows player  $X$  *before* the goal is scored." A system kernel based on interval algebra can easily represent these temporal relationships among clips and frames, as well as the spatial relationships among salient objects in an image.

### VIEWS ON IMAGE DATABASES

As described in the previous section, the content of an image can be seen as the point of view of a user or a group of users. As an example, let us take a picture of the Smith family. The family has four members: Mr. and Mrs. Smith, their young son John, and their daughter Jane. If John is describing this picture to his friends, he will say: "This is me, this is my mother and father and over there is my sister." The description of Mrs. Smith for the same picture will be: "This is me, my husband, my young son John, and my daughter Jane." The example illustrates different descriptions of the same image. They are all valid within the right context.

As another example, let us consider an electronic commerce system with a catalog containing photographs of people modeling clothes and shoes. From the customer's point of view, interesting objects in this catalog are shirts, shorts, dresses, etc. But the company may want to keep track of the models, as well as clothes and shoes. Assume the models come from different modeling agencies. Each of the agencies may be interested in finding only pictures in which their models appear. All these users of the same database (i.e., the catalog) have different interpretations of the content of the same set of images.

This modeling problem can be summarized as follows: Given the fact that we can manually or automatically extract content information from images, how do we organize this information so that an image can be interpreted with regard to a context? That is, if the context of an image changes, the understanding of the image may change as well. We approached this problem by providing a view mechanism for images.

Views have been widely used in relational database management systems to extend modeling capabilities, and to provide data independence. Basically, views in a relational database can be seen as formulae, which define virtual relations that are not produced until the formulae are applied to real relations (view materialization is an implementation/optimization technique). DISIMA uses object-oriented technology, and, despite several research efforts in the object-oriented community, the objective of a view mechanism, as defined for the relational model, has not yet been achieved. The problem is complex and may be too general in the object-oriented environment.

We first proposed a powerful object-oriented view mechanism [16] based on the distinction between class and type that we used in the image view definition. As discussed above, the DISIMA model separates the physical salient objects from the logical salient objects. We developed an image

view mechanism that allows us to give different semantics to the same image by exploiting this separation. For example, a (derived) image class can be defined by deriving new logical salient object classes that give new semantics to the objects contained in an image or by hiding some of the objects by directly defining a derived image class.

### QUERYING THE IMAGE DATABASE

One of the basic functionalities of a DBMS is to be able to efficiently process declarative user queries. DISIMA extends this basic DBMS functionality to image databases. The complex spatial and temporal relationships inherent in the wide range of multimedia data types, make a multimedia query language quite different from its counterpart in traditional DBMSs. For example, the query languages of traditional DBMSs deal only with exact-match queries on conventional data types. Content-based information retrieval requires non-exact-match (fuzzy) queries. The issue is addressed in DISIMA by extending a standard language, OQL [Cat94], by multimedia constructs. The resulting language is called MOQL (Multimedia OQL) [5]. This approach is in contrast to those that develop entirely new and specialized languages (e.g., [CIT+93, HK95, KC96, ATS96]) or languages based on logic or functional programming (e.g., [DG92, MS96]). The choice was made because OQL is better accepted by users, and an extended version of it is easier to learn and use by those who are familiar with the general SQL style querying.

Most of the extensions that are introduced to OQL are in the predicates (the *WHERE* clause) in the form of three new predicate expressions: *spatial\_expression*, *temporal\_expression*, and *contains\_predicate*. The *spatial\_expression* is a spatial extension which includes spatial objects (such as points, lines, circles), spatial functions (such as length, area, intersection) and spatial predicates (such as cover, disjoint, left). The *temporal\_expression* deals with temporal objects, temporal functions, and temporal predicates<sup>2</sup>. The *contains\_predicate* checks whether a particular salient object exists in a particular media\_object (e.g., an image). Even though we do not discuss the details of the language here, some examples of the MOQL queries are the following:

**Query 1.** Find all images in which a person appears.

```
SELECT  m
FROM    Images m, Persons p
WHERE   m contains p
```

**Query 2.** Select all the cities, from a map of Canada, which are within a 500 km range of the longitude 60 and latitude 105 with populations in excess of 50000.

```
SELECT  c
FROM    Maps m, m.cities c
WHERE   m.name='Canada' AND c.location inside circle(point(60, 105), 500) AND
        c.population>50000
```

This query works as follows: For each map named 'Canada', the method *cities* retrieves all the cities in the map. Then, a city's location is checked to see if it is within the required range. **point** is a constructor that accepts either two values or three values to create a two-dimensional (2D) point or

---

<sup>2</sup> The temporal\_expressions are not necessary for image database querying, but have been added to the language as part of our effort to seek extensions into video database systems.

three-dimensional (3D) point, respectively. Here, **point**(60, 105) represents a 2D point; **circle** is a circle object constructor that accepts a spatial point acting as the center of the circle and a radius; **inside** is one of the spatial predicates introduced into the language.

In addition to the text-based language, MOQL, a visual interface to DISIMA was developed, called VisualMOQL [18]. Since the media managed by DISIMA are inherently visual, it makes sense to provide visual querying capability. VisualMOQL implements the image part of MOQL (i.e., it does not support the temporal\_expressions) and allows users to query images by their semantics. The user can query the database by specifying the salient objects in the image. Defining the color, shape, and other attribute values of these salient objects can refine the query. Furthermore, the user can specify the spatial relationships among salient objects in the image – including both topological and directional ones. The user can also specify properties of the image meta-data, such as the name of the photographer and the date the photo was taken.

VisualMOQL has the following features:

- It is a declarative visual query language with a step-by-step construction of queries, close to the way people think in natural languages.
- It has a clearly defined semantics based on object calculus. This feature can be used to conduct a theoretical study of the language, involving concepts such as expressive power and complexity.
- It combines several querying approaches: semantic-based (query image semantics using salient objects), attribute-based (specify and compare attribute values), and cognitive-based (query-by-example). A user can start a query using the semantic and/or attribute-based approach and then choose an image for cognitive-based query.

Even though a full description of the interface is not feasible, we will highlight its basic functioning by referring to Figure 5. The VisualMOQL window consists of a number of components which can be used to design a query. The user specifies a query by choosing the image class he/she wants to query and the salient objects he/she wants to see in images. Several levels of refinement are offered, depending on the type of query, and also on the level of precision the user wants the result of the query to have. The startup window (Figure 5) consists of:

- A chooser to select the image classes. Images stored in the database are categorized into user-defined classes. Thus, the system allows the user to select a subset of the database to search over. The root image class is set as the default.
- A salient object class browser that allows the user to choose the objects that he/she wants. All salient objects and their associated attribute values are identified during database population. They are organized into a salient object hierarchy, and the root salient object class is set as the default.

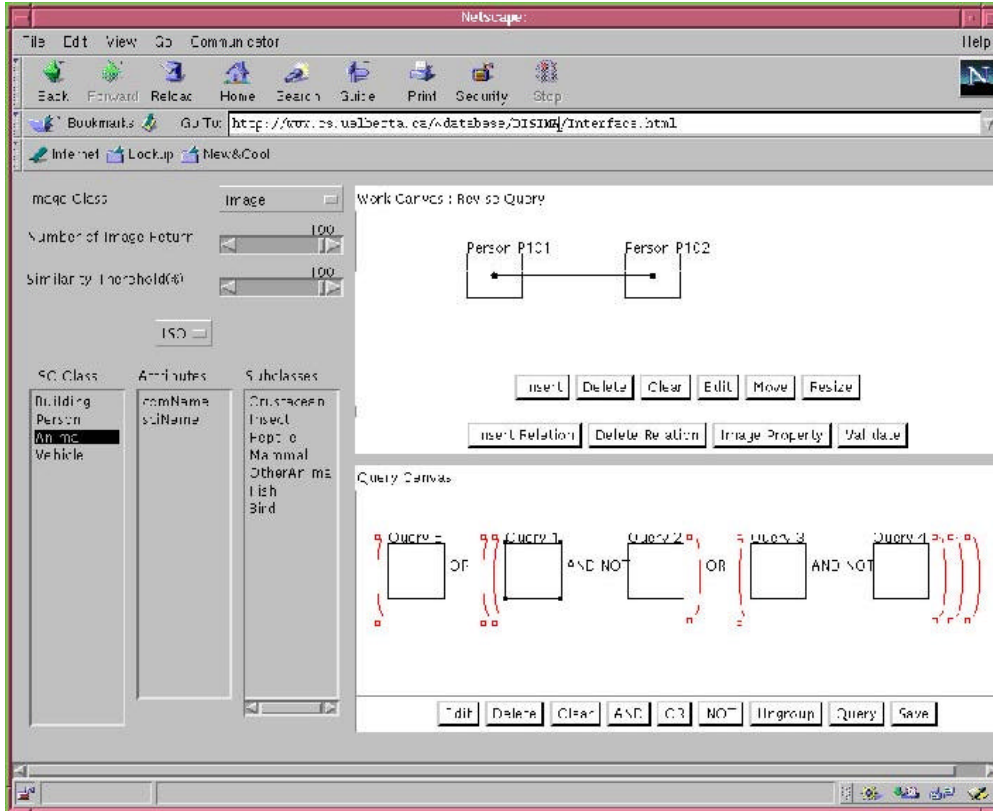


Figure 5. VisualMOQL Interface

- A horizontal slider to specify the maximum number of images that will be returned as the result of the query. This is a quality of service parameter used by the query result presentation interface.
- A horizontal slider to specify the similarity threshold between the query image and the large images stored in the database. It is also used for color comparison. This is also a quality of service parameter for the presentation interface.
- A working canvas where the user constructs simple queries that involve one or more salient objects whose properties can be specified, and, in the case of multiple objects, spatial relationships among them be declared. In addition, image properties can be specified for images. Property specification is accomplished through customized dialogs that are aware of the relevant salient object and image properties. Figure 5 shows a simple query which consist of two Person objects (identified by their MBRs) between which a spatial relationship (identified by the line connecting the MBRs) is defined.
- A query canvas where the user can construct compound queries based on simple queries. In this sense, a simple query can be considered a sub-query of a compound database query. Compound queries are formed from simple queries by connecting them using the logical connectives AND, OR, and NOT. Simple queries are formu-

lated one-at-a-time in the working canvas, and transferred to the query canvas when they are “validated”.

Once the query formulation is completed, it is translated into an MOQL query that is executed by the DISIMA kernel to retrieve thumbnails of the images that satisfy it. Clicking on the desired thumbnail retrieves the actual image.

### INDEXING OF THE IMAGE DATABASE

*Milestone 7: Incorporation of image indexing and content-based access features of the image DBMS (March 1998).*

Efficient processing of queries against a database requires sophisticated indexing techniques. Content-based querying of image databases requires development of indexing techniques that are significantly different than traditional DBMSs. Within the context of the DISIMA project, we have focused on two types of indexes: *spatial indexes* to facilitate queries of the type “Find all the images that show Jean Chrétien to the left of Bill Clinton in front of the White House” and color similarity indexes that facilitate queries of the type “Find all the images that are similar, in color, to this sample image.” We describe our work in each of these areas below.

#### 2-D-S TREES

2-D string is a popular data structure [CS787, CJL89, LH90, CL95, LYC92] for spatial reasoning and spatial similarity computing. The spatial relationships in images, as well as the queries, are represented as two-dimensional (2-D) strings (hence the name). The query execution is reduced to 2-D string subsequence matching.

Three types of string subsequences are defined for 2-D strings:

- Type-0, which finds all images containing specified salient objects with some exclusive spatial conditions. For example, “Find all images containing a tree and a house as long as the house is not to the left of the tree.”
- Type-1 query, which finds all images in which the salient objects in the pattern maintain alignments between each other, but the “distance” between any two objects is insignificant and may be different than the query pattern. For example, given the query “Find all images containing a house to the right of a tree”, the index would retrieve images where the condition holds even if there are other objects in between the house and the tree.
- Type-2 query is the most precise one where the “distance” between objects is important. If the same query as above is given, type-2 search would only retrieve images where the house and the tree are next to each other, with no other salient object between them.

The technique works well for queries that are posed on a single image (i.e., find subimages in a given image), but searches over a large set of images are inefficient, since string matching is performed sequentially, making it inappropriate for large databases. The proposed search algorithms



over 2-D strings assume a small database, and perform a linear search over the 2-D strings. This approach clearly does not scale up to large image databases such as the ones we address in this project.

In this project, we studied techniques that are based on 2-D strings, but have overcome this fundamental drawback. We have proposed 2-D-S trees [17] that build a B+-tree like index on top of the 2-D strings. In other words, we build another index on top of the 2-D index. The results show a dramatic improvement in performance. Table 2 depicts, as an example, the result of one experiment.

Database Size (no. of images)	Type-0		Type-1		Type-2	
	Regular	2-D-S tree	Regular	2-D-S tree	Regular	2-D-S tree
2000	167.40	0.75	104.90	0.08	79.00	0.07
5000	350.20	0.79	279.15	0.12	202.99	0.12
10000	-	1.02	569.16	0.12	364.17	0.11
15000	-	1.10	927.02	0.20	629.19	0.17

Table 2. 2-D-S Tree performance with changing database size (seconds)

#### DYNAMIC BINTREE (DB-TREE):

We developed a second technique along the same lines as the dynamic bintree (db-tree for short). This data structure records the relative location of the objects within the image in a way similar to 2D-strings.

A db-tree is a balanced binary tree with no coordinates. Its leaf nodes contain the image objects, where the internal nodes and the tree structure indicate the spatial relations among image objects. The root node of a db-tree represents the entire space occupied by image objects.

The space is divided recursively into two subspaces, or *bins*, in the  $x$  and  $y$  dimension alternately, until each bin contains at most one object. By then, the entire set of objects is totally ordered in each dimension. The db-tree structure is decided by bin-indices, or the bin-partition of the image objects. In the manual annotation process, we need only to draw some horizontal and vertical lines to separate image objects. The position of the lines does not have to be precise, as long as the bin boundaries separate the objects so that the relative spatial ordering of the objects is preserved.

The bin boundaries will be used to generate the bin indices, which are then used as input for the db-tree creation. There are two types of internal nodes, namely  $x$ -nodes and  $y$ -nodes, each of which have two children.

For an  $x$ -node, all the objects in the left subtree are located to the left of the objects in the right subtree. Similarly, for a  $y$ -node, all the objects in the left subtree are underneath the objects in the right subtree.

The time complexity of our matching algorithm is  $O(n \log m + n)$ , where  $n$  and  $m$  are the number of objects in a query image and a database image.

We have compared the db-tree data structure and the matching algorithm with other schemes, such as 2D-strings. For image retrieval applications such as the one we have considered – involving images with multiple objects – the theoretical analysis and experimental results show that the db-tree approach outperforms 2D-strings in several aspects, especially in the efficiency of inserting new objects. The details of db-tree and its use in image retrieval are given in [12, 13].

#### COLOR SIMILARITY INDEXING

Color histograms are one of the most extensively used visual features in content-based image retrieval. In order to determine if two images are color similar, their color histograms are compared. This is a well-known approach, and there have been many studies along these lines.

What we were interested in within the context of the DISIMA project was to be able to perform efficient searches for color similarity at different levels of precision. In other words, we investigated techniques that would allow us to search for the images in the database a particular portion of which is color similar to a sample (query) image. We considered four levels of precision, which means that we decompose an image into four levels, each with four quadrants:

- 1) At level one, the color histogram of the entire image is compared to the query image color histogram;
- 2) At level two, the color histogram of the four quadrants of the full image (i.e.,  $\frac{1}{4}$  of the full image) is compared to the corresponding quadrants' color histogram of the query image;
- 3) At level three, the color histogram of the sixteen quadrants (i.e.,  $\frac{1}{16}$  of the full image) is compared to the corresponding quadrants' color histogram of the query image;
- 4) At level four, the color histograms corresponding to 64 quadrants (i.e.,  $\frac{1}{64}$  of the full image) are compared to their counterpart quadrant's color histogram in the query image.

We also investigated sub-image color similarity matching. In this case, a region in a query image is compared with a region of the images in the database for color similarity.

To facilitate searches over large image databases, where each image is decomposed into four levels, we designed a hash structure to index color values. Normally one color value has three color components. For example, under the RGB model, one color value has the R, G, and B components, where each component value ranges from 0 to 255. The problem then becomes to hash a 24-bit (8 bits for each color component) color value to an address. For this, we designed an extendible hashing scheme and conducted experiments. This work is fairly recent and the results have not yet been reported in a paper.

#### IMAGE PROCESSING AND OBJECT RECOGNITION

The database that we developed for testing our database contained images about news events. Therefore, the most important salient objects in these images were people. Consequently, to facilitate annotation of the pictures in such a database, and to facilitate content-based image retrieval, we focused on automatic face detection and location. This procedure scans through an image and looks



for human faces. If it finds one, it marks that face with a box around it. A human operator can either annotate the picture by naming each box (each face), or let the program automatically name the face with some measurable characteristics.

Face detection is usually the first step in face recognition. Because of the requirements of automatic entrance control and machine interface, recognition of human faces has become increasingly important. Hundreds of face recognition schemes have been developed, and formal program contests have been organized worldwide. A number of recognition systems have out-performed human recognition ability. However, to locate a human face from an arbitrary image takes about 100 times longer than to recognize that face once the face image is normalized in size and orientation. Most existing face detection algorithms deal only with portrait-like or id-like facial images where the person is facing directly to the camera, and both eyes are visible in the picture.

Detecting a face of arbitrary orientation, especially detecting a side-view face, remains a challenging problem. In the campus event image database, the faces are of many different sizes and orientations, and side-view faces are quite common.

Our method has four stages:

- 1) Examine all pixels of the image and locate the ones with a "face color." A clustering analysis is done on a large training set of face images to decide several face colors. After those representative face colors have been determined, any future pixel color will be compared against them. All pixels failing to match with any face color are eliminated from further processing.
- 2) Morphological filtering is used to process the binary image, with face pixels marked 1 and other pixels marked 0. This filtering operation eliminates isolated pixels and connects the face pixels into regions of reasonable shape. Those regions are potentially faces.
- 3) For each "possible-face" region, a simple procedure decides if it is a front-view face. Checking if there are two eyes of the same relative size can do this. If it is a front-view face, a box is drawn around the face, and the process terminates. If it is not a front-view face, a "side-face detector" procedure is called.
- 4) The side-view face detector first extracts the contour of the region, then an arbitrary-shape Hough transform is applied to the contour. Our preliminary experiments show that 70% of the side-view faces can be successfully detected. A new version of this procedure, using fuzzy measures, is currently under investigation.

#### IMAGE CODING

*Milestone 2: Develop adaptive sampling/decomposition algorithms, test different compression schemes working with decomposition (December 1995).*

A large image database demands compression coding of the images and videos. Powerful compression schemes are crucial in the success of our system. Especially in the case of a distributed image database, reducing the image/video size can save transmission time, as well as disk storage. An image database on a publicly accessible network, such as the Internet, also requires encryption coding

to provide privacy and security of the image data. We carried out an intensive study in this area that was also partially supported by Motorola Canada.

After our preliminary work on a spatial-domain decomposition and predictive coding scheme [LN95], we investigated a large number of image decomposition algorithms, from the simplest spatial-domain methods to powerful wavelet transform-domain methods.

The simplest spatial-domain compression scheme is a quadtree. An image is recursively divided into quarters, and a tree structure is generated for the parent node being the entire square region, and the children nodes being the four quarters. Homogeneous regions will not be divided further. The structure of the tree, and the property of each node, is coded as the compression result. We proposed an improved quadtree algorithm that is simpler and faster than the existing ones, and achieves a higher compression ratio and better visual quality [15].

We also investigated an even simpler compression algorithm: one-dimensional (1D) piecewise approximation. This method does not even consider the two-dimensional (2D) dependency between the image pixels. It treats the pixel values as a 1D signal and uses a piecewise linear approximation to approach the true pixel values. It requires far less computation than 2D methods, while achieving a compression ratio and quality almost identical to JPEG [14].

At the other extreme, we studied the most powerful image compression scheme: wavelet decomposition and vector quantization. We proposed a number of modifications to a newly (at that time) proposed wavelet algorithm SPIHT. Our algorithm gives higher performance than SPIHT on images with a large number of details, that are more difficult to compress in general [19].

The quadtree method and wavelet method, as well as our predictive coding, use a tree structure, which gives us an extra advantage: we can encrypt the data structure only, without encrypting the entire image. Based on this idea, we examined the partial encryption scheme details and trade-offs. The results show that this scheme gives almost the same security level but only requires about 10% of the work (2-3% in the case of wavelet compression). In other words, a much simpler partial encryption works with compression, and achieves data privacy and security for the images only until one million years from now, instead of until one billion times of the age of the universe [20].

---

#### PROTOTYPE DEVELOPMENT AND TECHNOLOGY TRANSFER

---

*Milestone 5: Combine the feature definition procedures and compression procedures together in the prototype (March 1997).*

*Milestone 7: Incorporation of image indexing and content-based access features of the image DBMS (March 1998).*

*Milestone 8: Completion of the implementation of the complete image DBMS (December 1999).*

Concurrent with the research activity summarized in the previous section, we were involved in the development of a prototype system. We developed the DISIMA prototype in three stages, preceded by a proof-of-concept prototyping of the modeling ideas. This proof-of-concept prototype was part of a Ph.D. dissertation, and was reported in [7]. The technology transfer to our partners and

to the research community at large, was largely accomplished by means of these prototypes and the published articles.

The first prototype was developed in the 1996-1997 time frame. This prototype included the first version of VisualMOQL, and a very simple image access mechanism without the full DISIMA kernel. We were able to show how queries would be formed, and had a simple engine to evaluate a restricted set of queries. This version was demonstrated at the *4<sup>th</sup> International Conference on Visual Database Systems (VDB-4)* in L'Aquila, Italy during May 1998. The client interface (i.e., the VisualMOQL code) ran on a Windows machine in L'Aquila and the database engine ran on the Sun SparcStation computers back in Edmonton. The connection between the client and the server was over the Internet and, despite the preliminary nature of the code and the distance involved, the demonstration went flawlessly. A cleaned-up version of this demonstration was also given during the 1998 IBM *CASCON Conference* in Toronto (November 1998).

During 1998-1999, we implemented the complete DISIMA kernel, and incorporated a number of improvements to the VisualMOQL interface. The system is now more stable and can handle a richer array of queries. We gave a demonstration of this version at the recent *IEEE International Conference on Data Engineering (ICDE)* in San Diego (February 2000) [21]. ICDE is one of the top three database conferences and it was important for us to show the system to our colleagues and get their feedback.

The final demonstration will be conducted at the upcoming *ACM International Conference on Management of Data (SIGMOD)*. This version includes the color similarity matching work, as well as distribution over CORBA [22]. SIGMOD is arguably the top database conference, and we aimed at showing the full version of the system at this event to cap the project.

Finally, we have put the DISIMA demonstration system on-line. It is accessible to anyone who wishes to try the capabilities of the system over the Internet. Since the VisualMOQL interface is written as Java applets, using the system does not require any software other than a browser. The URL for the demo is <http://www.cs.ualberta.ca/~database/DISIMA/Interface.html>.

The focus in each version of the system prototype has been the main execution engine. There are a number of areas where we have conducted research, and, in some cases, have even built stand-alone proof-of-concept prototypes, but have not fully implemented in the demonstrator system. These are listed below:

- 1) MOQL is an extension of OQL. However, the current prototype implements a simple query execution engine. The language is not coupled with a full OQL compiler. Therefore, we do not yet have ODMG DDL capability, or the ability to perform query optimization. Furthermore, current implementation of MOQL can select only full images.
- 2) The object manager and the meta-data manager are implemented only to the extent that the underlying repository, ObjectStore, provides support for them. In other words, we have not implemented any new techniques in these areas other than what ObjectStore provides.
- 3) The distributed version of DISIMA is restricted to retrieval from multiple repositories that have the same schema. We have not yet performed any research on the semantic heterogeneity problem in image databases.

- 4) Even though the conceptual model allows multiple representations for an image and multiple representations for each physical salient object, the prototype is restricted to one representation for each. For images, this is the raster representation; for physical salient objects, it is the vector representation.
- 5) The work on 2-D-S indexing has not been incorporated into the current prototype, even though the algorithm has been coded for experimentation.
- 6) Face detection code, and image coding code, are stand-alone and have not been incorporated into the current prototype. The intention is to use the face detection software as part of the image annotation interface. This has not yet been done.

---

#### **TRAINING OF RESEARCH PERSONNEL**

---

This project has attracted many good students and researchers. Overall, we have graduated, or are in the process of graduating one Ph.D. student (Mr. John Z. Li) whose research was solely devoted to DISIMA, and five M.Sc. students (Ms. Irene Lin Cheng, Mr. Bing Xu, Mr. Xun Tan, Ms. Shu Lin, and Mr. Bin Yao). The responsibilities of these students, as well as their graduation years, are indicated below:

- 1) Mr. John Z. Li (Ph.D.) worked on the image modeling aspects and on image query language development. He was responsible for the fundamental model that was implemented in DISIMA. He graduated in 1998 and is now employed by IBM Toronto Laboratories.
- 2) Ms. Irene Lin Cheng (M.Sc.) was responsible for developing the image database kernel of the system. In a sense, she implemented the basic model developed in Mr. Li's dissertation. She graduated in 1999 and is now working as an independent consultant.
- 3) Mr. Bing Xu (M.Sc.) was responsible for developing the visual query interface, VisualMOQL. The interface was implemented in Java and is used in the current prototype. Mr. Xu accepted a job with IBM Canada in 1998 and, therefore, his thesis defence was delayed. He successfully defended his thesis in April 2000.
- 4) Mr. Xun Tan (M.Sc.) was responsible for the image segmentation and object recognition research – in particular, the recognition of faces in the photograph repository that we generated. He has also taken early employment, and, even though the work is largely completed, he has yet to write his thesis and schedule a defence.
- 5) Ms. Shu Lin (M.Sc.) has worked on color indexing, and has developed a hash-based indexing technique for color similarity searches with finer granularity than what is currently available. Her defence is scheduled for May 2000. Ms. Lin has been accepted into the Ph.D. program at the University of Waterloo, but has decided to accept a job offer from IBM Toronto Laboratories in order to gain industrial experience before starting her Ph.D. studies at a later date.
- 6) Mr. Bin Yao (M.Sc.) has worked on the distribution of the DISIMA system over a CORBA distributed object-computing platform. He has completed his work and has written his thesis. His defence is scheduled for May 2000. He has been admitted

to the Ph.D. program at the University of Waterloo, and will start his studies in September 2000 after a summer internship at IBM Toronto Laboratories.

A second Ph.D. student, Mr. Youping Niu, worked on spatial indexing issues involving 2D-strings. After about two years of work (and some results that produced two published papers), he had to take a leave from the Ph.D. program for family reasons. He took a job with Nortel in Ottawa and is now stationed in their Dallas office. His departure was unfortunate (both for him and for the project) as it came at an inopportune time, putting our indexing work on hold. This research would have generated interesting results.

A third Ph.D. student, Mr. Iqbal Goralwalla, was seconded to the project for assistance in certain aspects related to temporal modeling issues. Mr. Goralwalla's research involved temporal database models, within the context of a separate project involving the development of an object database management system. However, his expertise in temporal modeling was needed within DISIMA, which, in return, provided him with a real-life example to test his own proposals. Thus, he worked as part of the DISIMA team for extended periods of time in the early stages. Mr. Goralwalla graduated in 1998 and is now employed at IBM Toronto Laboratories.

In addition to the students listed above, we were fortunate to hire two professionals. Dr. Vincent Oria was hired in May 1996 as a post-doctoral fellow. He remained in that position until January 2000, when he joined the New Jersey Institute of Technology as an Assistant Professor. Dr. Oria was responsible for day-to-day supervision of M.Sc. students' work, as well as for his own research, which involved view mechanisms for multimedia databases and visual query interfaces. Mr. Paul Iglinski has been working as a research associate with the Principle Investigator's group since January 1995. Even though the scope of his responsibilities is quite wide, his involvement with the DISIMA project was critical: he was our resident expert on the ObjectStore system, and he maintained the DISIMA code base and ensured its consistency as students joined and left the team. Finally, he supervised, jointly with Dr. Oria, the daily work of many part-time undergraduate students who were part of the project group at various times.

Dr. Xiaoqing Qu, who was a former Ph.D. student in our department, returned as a Research Associate for one year. Her work focused on the image retrieval data structure development, primarily the dynamic bintree method.

---

## PUBLICATIONS

---

The research project generated a large number of publications. The ones that are already published are listed below in chronological order. To disseminate the results quickly, we have focused on conference publications; more archival journal publications are being prepared, and will be submitted over the next year. Furthermore, publications that will arise from ongoing work (i.e., those that are either under submission or under development) are not listed.

- [1] J.Z. LI, M.T. ÖZSU, AND D. SZAFRON. "Modeling of Video Spatial Relationships in an Object Database Management System," In *Proc. International Workshop on Multimedia DBMS*, Blue Mountain Lake, NY, August 1996, pages 124–132.
- [2] J.Z. LI, M.T. ÖZSU, AND D. SZAFRON. "Spatial Reasoning Rules in Multimedia Management System," In *Proc. International Conference on Multimedia Modeling*, Toulouse, France, November 1996, pages 119–133.

- [3] J.Z. LI, I.A. GORALWALLA, M.T. ÖZSU, AND D. SZAFRON. "Modeling Video Temporal Relationships in an Object Database Management System," In *Proc. SPIE Multimedia Computing and Networking (MMCN97)*, San Jose, California, February 1997, pages 80–91.
- [4] J.Z. LI, M.T. ÖZSU, AND D. SZAFRON. "Modeling of Moving Objects in a Video Database," In *Proc. IEEE Int. Conf. on Multimedia Computing and Systems*, Ottawa, Canada, June 1997, pages 336–343.
- [5] J.Z. LI, M.T. ÖZSU, D. SZAFRON AND V. ORIA. "MOQL: An Object-Oriented Multimedia Query Language," In *Proc. Int. Workshop on Multimedia Information Systems (MIS'97)*, Como, Italy, September 1997, pages 19–28.
- [6] Y. NIU, M.T. ÖZSU, AND X. LI. "2D-h Trees: An Index Scheme for Content-Based Retrieval of Images in Multimedia Systems," In *Proc. IEEE International Conference on Intelligent Systems (ICIPS'97)*, Beijing, China, October 1997, pages 1710–1715.
- [7] J.Z. LI AND M.T. ÖZSU, "STARS: A SpaTial Attributes Retrieval System for Images and Videos", In *Proc. 4th International Conference on Multimedia Modeling (MMM'97)*, Singapore, November 1997, pages 69–84.
- [8] V. ORIA, M.T. ÖZSU, X. LI, L. LIU, AND P. IGLINSKI. "Modeling Image for Content-Based Queries: The DISIMA Approach," In *Proc. Visual 1997 Conference*, San Diego, California, December 1997, pages 339 – 346.
- [9] J. KNIPE, X. LI AND B. HAN. "An Improved Lattice Vector Quantization Based Scheme for Wavelet Compression," *IEEE Transactions on Signal Processing*, 46(1): 239-242, January 1998.
- [10] J.Z. LI AND M.T. ÖZSU. "Point-set Topological Relations Processing in Image Databases," In *Proc. of the 1st International Forum on Multimedia & Image Processing*, Anchorage, Alaska, May 1998, pages 54.1 – 54.6.
- [11] V. ORIA, B. XU AND M. T. ÖZSU. "VisualMOQL: A Visual Query Language for Image Databases," In *Proc. 4th IFIP 2.6 Working Conference on Visual Database Systems (VDB-4)*, L'Aquila, Italy, May 1998, pages 186-191. (Demo description)
- [12] X. LI AND X. QU, "Image Retrieval Using Dynamic Bintree," In *Proc. IEEE Int. Conf. on Intelligent Processing Systems*, Gold Coast, Australia, August 1998, pages 526-530.
- [13] X. LI AND X. QU, "Matching Spatial Relations Using db-Tree for Image Retrieval," In *Proc. Int. Conference in Pattern Recognition*, Brisbane, Australia, August 1998, pages 1230-1234.
- [14] J. MODAYIL, H. CHENG, AND X. LI. "An Improved Piecewise Approximation Algorithm for Image Compression," *Pattern Recognition*, 31(8): 1179-1190, August 1998.
- [15] J. KNIPE AND X. LI, "On the Reconstruction of Quadtree Data," *IEEE Transactions on Image Processing*, 7(12): 1653-1660, December 1998.



- [16] V. ORIA, M.T. ÖZSU, D. SZAFRON, AND P. IGLINSKI. “Defining Views in an Image Database System,” In *Proc. 8th IFIP 2.6 Working Conference on Data Semantics: Semantics Issues in Multimedia (DS-8)*, Rotorua, New Zealand, January 1999, pages 231–250.
- [17] Y. NIU, M.T. ÖZSU, AND X. LI. “2-D-S Tree: An Index Structure for Content-Based Retrieval of Images,” In *Proc. SPIE Int. Conf. on Multimedia Computing and Networking*, San Jose, January 1999, pages 110–121.
- [18] V. ORIA, M.T. ÖZSU, B. XU, L.I. CHENG, AND P. IGLINSKI. “VisualMOQL: The DISIMA Visual Query Language,” In *Proc. IEEE Int. Conf. on Multimedia Computing and Systems (ICMCS’99)*, Florence, Italy, June 1999, pages 536-542.
- [19] V. ORIA, M.T. ÖZSU, L.I. CHENG, P. IGLINSKI AND Y. LEONTIEV. “Modeling Shapes in an Image Database System,” In *Proc. 5th Int. Workshop on Multimedia Information Systems (MIS’99)*, Palm Springs, California, October 1999, pages 34-40.
- [20] H. CHENG AND X. LI. “Partial Encryption of Compressed Images and Videos,” *IEEE Transactions on Signal Processing*, accepted Nov. 8, 1999.
- [21] V. ORIA, M.T. ÖZSU, P. IGLINSKI, B. XU, AND L.I. CHENG. “DISIMA: An Object-Oriented Approach to Developing an Image Database System,” In *Proc. 16th Int. Conf. On Data Engineering (ICDE 2000)*, San Diego, California, February 2000, pages 672-673. (Demo description)
- [22] V. ORIA, M.T. ÖZSU, P. IGLINSKI, S. LIN, AND B. YAO. “DISIMA: A Distributed and Interoperable Image Database System,” In *Proc. ACM SIGMOD Int. Conf. on Management of Data (SIGMOD 2000)*, Dallas, Texas, May 2000, in press. (Demo description)

---

## CONCLUSIONS

---

In our view, the DISIMA project has been highly successful. It has produced nineteen research publications, and three demo description papers; it has trained eight graduate students and one post-doctoral fellow; it has provided training and employment to two other researchers. The project has also produced technology that pushes the limits of the existing image search systems. In a sense, we have reversed the typical modus operandi where database researchers are handed a set of data and are asked to find ways of managing them efficiently. The DISIMA project demonstrates what we can do with image data if image processing techniques improve to provide better image segmentation and object recognition – challenging the image processing community to improve their techniques to fully exploit the database querying capabilities.

The DISIMA project has had a number of socio-economic benefits. First and foremost, the project has contributed to the development of basic image data management technology in ways that are unique and not found in other systems. In the original application, we claimed the following: “With the increasing use of multimedia information systems, the need for systems that enable the management of images becomes more important. Some of the application areas with significant requirements are geographic information systems where maps are the basic entities that are manipulated, law enforcement agencies which deal with mug shots or fingerprint images, and health care facilities which store medical images for diagnostic and treatment purposes. The social benefits of effective and efficient image handling and manipulation tools is quite high for these applications. The fundamental economic impact of this project is in the development of basic technology for managing im-

ages. This is a very fast growing segment of the computer systems market whose full market potential is not yet realized.” From the description of the technical achievements in previous sections, it is clear that the project has significantly contributed to the scientific infrastructure in this area. DISIMA is the only concentrated Canadian research endeavour on image data management. The large number of publications and the venues that they have appeared in attests the quality of the research work.

A second important socio-economic contribution has been the training of a large number of researchers. As listed elsewhere in this report, three Ph.D. students have been involved with the project in various capacities, five M.Sc. students have directly worked on DISIMA, one post-doc spent three-and-a-half years training in image DBMSs, and two research associates have participated in the project. All but two of these people have obtained employment with Canadian companies and institutions.

The development of the prototype system has facilitated interaction with many other research groups and institutions. The prototype has been used as part of courses in a number of universities and we have had contact with other researchers (e.g., in medical imaging) who have expressed interest in using DISIMA within their organizations. Even though it is still very premature to report these, they indicate the quality of the work and the importance of the technology developed in this project.

Finally, it is important to point out that DISIMA has found its way into other projects. The principle investigator is involved in a CITR (Canadian Institute for Telecommunications Research) sponsored project on electronic commerce that involves the development of catalog servers, and DISIMA has been incorporated into the catalog architecture to manage still images.

The building of a demonstration system was critical in the success of the project. It provided an infrastructure that integrated the research work, and forced an engineering discipline on the entire endeavor. The various prototypes also served as checkpoints for us to determine how close we were to achieving our goals. Finally, the prototypes were critical for technology transfer to our industrial partners and to the community at large.

In closing this project, we should note that there are many other things that we would like to investigate, perhaps within the context of a follow-up strategic grant. Some of these include better integration of image processing and database techniques; more sophisticated query execution and optimization algorithms; improving the interoperability framework to include access to other image repositories over the Internet; and finally, and perhaps most importantly, extension of the system to include video data. We have already done some work in this regard, but only to verify that DISIMA can be extended in this way.

---

#### OTHER REFERENCES

---

- [All83] J.F. Allen. “Maintaining knowledge about temporal intervals,” *Comm. ACM*, 26(11): 832-843, 1983.
- [AS94] J. T. Anderson and M. Stonebraker. Sequoia 2000 metadata schema for satellite images. *ACM SIGMOD Record*, 23(4):42–48, December 1994.
- [ATS96] G. Ahanger, D. Benson, and T.D.C. Little. “Video Query Formulation,” In *Proc. IS&T/SPIE Symp. On Storage and Retrieval for Images and Video Databases II*, San Jose, CA, February 1995, pages 280-291.



- [Cat94] R. Cattell. *The Object Database Standard: ODMG-93* (Release 1.1), Morgan Kaufmann, San Francisco, CA, 1994.
- [CIT+93] A.F. Cardenas, I.T. Jeong, R.K. Taira, R. Barker, and C.M. Breant. "The Knowledge-based Object-Oriented PICQUERY+ Language," *IEEE Trans. Knowledge and Data Engineering*, 5(4): 644-657, August 1983.
- [DG96] N. Dimitrova and F. Golshani. "EVA: A Query Language for Multimedia Information Systems," In *Proc. Workshop on Multimedia Information Systems*, Tempe, AZ, February 1992.
- [HK95] N. Hirzalla and A. Karmouch. "A Multimedia Query Specification Language," In *Proc. Int. Workshop on Multimedia Database Management Systems*, Blue Mountain Lake, New York, August 1995, pages 73-81.
- [KC96] T.C.T. Kuo and A.L.P. Chen. "A Content-Based Query Language for Video Databases," In *Proc. IEEE Int. Conf. on Multimedia Computing and Systems*, Hiroshima, Japan, June 1996, pages 456-461.
- [LN95] X. Li and V. Nehru. "Experiments in Adaptive Predictive Coding," In *Proc. IEEE Int. Conference on Systems, Man and Cybernetics*, Vancouver, BC, October 1995, pages 3525-3530.
- [MS96] S. Marcus and V.S. Subrahmanian. "Foundations of Multimedia Database Systems," *Journal of ACM*, 43(3): 474-523, 1996.
- [OMG93] Object Management Group. *The Common Object Request Broker: Architecture and Specification*. OMG Document 93.12.43, December 1993.
- [LT92] R. Laurini and D. Thompson. *Fundamentals of Spatial Information Systems*, Academic Press, 1992.