

An Image Retrieval Model Based on Classical Logic

Carlo Meghini

Consiglio Nazionale delle Ricerche
Istituto di Elaborazione dell'Informazione
Via S. Maria, 46 - I-56126 Pisa, Italy
meghini@iei.pi.cnr.it

Abstract

Images are a communication medium, hence objects of a linguistic nature having a form and a content. The form of an image is the image appearance and is understood as depicting a scene, the image content. The relationship between the form of an image and its content is established through a process of interpretation, capturing the meaning of the image form. Any information need on images can, and indeed has to, be seen as addressing either the image form, or its content, or the relationship between them. Consequently, any general, domain independent image retrieval facility should be based on a model supporting all these aspects of images. An image retrieval model, based on classical logic, is proposed which fulfills this basic requirement.

1 Introduction

Information systems offering images serve nowadays an increasingly wide community of users. In order to cope with the requirements of their users, these systems must provide a sophisticated retrieval capability, which, while abstracting implementation details, must accommodate the many forms of image retrieval independently studied in various areas of computer science, ranging from pattern matching to semantic information processing.

Certain users of an image base (IB for short) would like to retrieve images by posing images themselves as queries and having the system performing a match, capturing a similarity criterion. Other users would want to query the IB at a more abstract level, by specifying conceptual properties of the individuals that occur in them. At an intermediate level, retrieval could be performed on spatial relationships between individuals depicted in images.

In order to incorporate in a clean and principled way all these forms of retrieval, an image retrieval model must support a universal representation of images. We claim that this representation is to be based on a linguistic understanding of images, providing the distinction between the *form* and the *content* of images and a relation between these, reflecting the notion of semantics.

We propose a model for image retrieval which views images as 3-level objects, comprised of the form, content, and mapping level. The form level is a model of the image seen as a combination of colored regions. The content level is a propositional representation of the scene depicted by the image, couched in terms of a semantic data model offering object-orientation and abstraction mechanisms. The mapping level is an association between form and content which can be used in both directions.

Queries are requests of the form “Retrieve all images such that α ”, where α is a condition on images which can address any combination of the image components and the spatial relationships between image objects. An image is returned in response to a query if it *logically satisfies* the query.

2 Relation to previous work

Research on image representation and retrieval has been very fruitful in the last decade. The assumption that images can be viewed from a linguistic stance goes back to at least 30 years ago, and has been the basis of research investigating the automatic generation of images (for a survey, see [Rosenfeld and Siromoney, 1993]). Chomsky’s generative grammars and related concepts have been used in this field to capture the basic production mechanisms of image languages. In artificial intelligence, as well as in psychology and cognitive science, mental imagery has been investigated in order to understand and reproduce human problem solving methods which are deemed to be based on image inspection and manipulation. The models proposed in all these fields do not directly relate to the problem of image retrieval, but, to different extents, can be considered as a useful source of inspiration.

The retrieval of images is central to iconic and pictorial databases. An iconic image is a two dimensional array showing the objects of interest to the application in symbolic (or iconic) form. As such, an image of this kind is half-way between the form and the content level, since it is about conceptual entities but at the same time it preserves the spatial relationships among these entities, as they appear in the image syntactic structure. Iconic images support retrieval based on spatial relationships. A typical query to an iconic database is “find all pictures having a tree to the left of a house”, and is expressed by means of an iconic image. As it will be shown, iconic queries are a proper subsets of the queries expressible in our model.

A pictorial database provides images and image components as built-in data types, thus allowing the manipulation and retrieval of image data objects. This reflects a fundamental difference between a data model and an information retrieval model: the former is the result of an abstraction performed with the application requirements in mind; the latter supports generic content-based retrieval on “raw” documents, images in our case. As a result, a pictorial data model typically copes with the efficient implementation of pictorial domains, of the manipulation functions defined for them, and of their relation operators. An information retrieval model has to address more general issues, such as the distinction between form and content representation, the provision of a powerful language for describing the various levels of images, and that of a retrieval function capturing the notion of relevance of a document to a query.

The interested reader can find a more detailed discussion in [Meghini, 1994b]. To the best of our knowledge, none of the image retrieval models proposed so far possesses the representation capabilities and the generality of our model. There are models, notably [Faloutsos *et al.*, 1994], that allow the users to pose visual queries and models that address the content representation of images. But not models offering both these functionalities. Furthermore, models of the former category often fail to state in semantical terms what is the relation between a query and its answer set, and how this relation stands to intuition. On the other hand, models providing semantical querying hardly go beyond text retrieval techniques and are mostly bound to specific application fields.

3 Preliminaries

Before entering into the technical development, a brief discussion of the model’s basic assumptions may help in clarifying the following presentation.

From a terminological point of view, there seem to be two meanings of the word “image”: for some an image is a recording, such as a photograph, whose content is a bi-dimensional composition of colors. In this vein, “content-based retrieval” is understood as a match of visual features such as shape and color. The other meaning

of the word views an image as a bi-dimensional composition of colors, which can be reproduced in various ways (for instance via a photograph), and whose content is the scene depicted by the image. Our model subscribes to this latter meaning, importing it in its definitions.

The typical view of an image is pixel-based: an image is a function associating a color (however represented) to the points of a discrete, connected, bi-dimensional set. The form level of our model will essentially conform to this view, by postulating an ontology consisting of colors and of connected, bi-dimensional sets, or regions, of a discrete space. In order to convey the informal meaning of a region, we suggest that it can be seen as a collection of “contiguous” points having the same color in the image; but it should be clear that the notion of point is introduced only at an informal level and for purely rhetorical reasons: it plays no role in the model, as it is not included in the model’s ontology. Colors are assumed to be primitive entities, of which everything is abstracted. These choices are dictated by the desire of defining a model which is closer to the user than to the machine. Specific implementations of the model will select the appropriate physical representation of regions and colors.

The representation of image contents is more controversial. As already pointed out, the content of an image is for us a scene of the real world, hence any symbolic construction which can be understood as describing a slice of reality can legitimately be considered as a content representation. The sentences of a logical language are the most natural candidates to play the role of image content representations. We will instead adopt a model-theoretic stance, viewing both the form and the content of an image as “analogs” of their corresponding realities. The initiated will recognize in this only an apparent departure from the sentential approach.

Unlike images, queries must be expressions of a formal language, by which users describe their information needs. The choice of a representation for images induces that for queries, as queries “talk” about images.

Finally, an image retrieval model must provide a correspondence function, that is a function specifying, for each query, what images are to be returned in response. This problem is often posed in terms of *aboutness*: what documents are about a given query q ? Following a common interpretation of aboutness from the theory of meaning¹ we will let it be truth, hence a document d is about a query q just in case q is true in d , in symbols: $d \models q$.

4 Image forms

The notion of image form realizes the form level of the image retrieval model being presented.

In order to obtain a representation valid for all images,

¹see for instance [Davidson, 1985], pages 129-137.

regardless of their position in space and contour shape, we fix a 2-dimensional discrete Cartesian space and, for a given image, align the bottom left corner of the smallest rectangle of the fixed space including that image with the origin of the space. In this way, any image can be represented as a triple (A, π, F) , where:

- A , the *region* of the image, is a set of pairs of natural numbers having the property of being connected, that is it has no isolated sub-regions;
- $\pi = \{S_1, S_2, \dots, S_n\}$, the *spots* of the image, is a partition of A consisting of connected sets;
- F is a total function from π to a given set of colors L , such that for all i , $F(S_i)$ is the color of spot S_i ; and
- no other triple (A, π', F') exists such that for all $X \in \pi$ there exists $X' \in \pi'$ such that $X \subseteq X'$ and $F(X) = F(X')$.

(A, π, F) is said to be an *image form*.

An image form does not impose any commitment on the way images are stored in a computer, but only provides an abstract, implementation independent way of looking at images. It will serve as a basis for the evaluation of visual queries.

5 Modelling image contents

The content of an image is understood by humans through a process of interpretation, which produces a mental reconstruction of the scene depicted by the image. This reconstruction, which will be called *content reconstruction*, may vary from interpreter to interpreter, and depends on the context in which the interpretation is carried out, including its use. We will confine ourselves to single reconstructions, although the extension to multiple reconstructions does not present conceptual difficulties.

As already argued, an image model must provide a tool for representing content reconstructions allowing to naturally represent slices of reality, while at the same time let these representations be efficiently queried. This is not the case with current image retrieval systems, which employ, if any, one of two techniques to represent the content of an image:

- the *caption* method, by which an image is associated with a piece of text describing the scene depicted by the image. Content-based retrieval is then carried out by applying text retrieval techniques to the image description.
- The *keyword* method, by which an image is associated with a set of terms drawn from a prefixed language.

Both these content representation schemes do not allow to capture in a satisfactory way the content of images. As a result, the performance of image retrieval allowed by them is poor.

We argue that a semantic data model is a suitable tool for the representation of image contents (for a survey on semantic data models, see [Hull and King, 1987]), as it can be seen as a close relative of predicate calculus emphasizing organization and computational amenity. The semantic data model that we employ for content reconstructions is inspired to Taxis [Mylopoulos *et al.*, 1980; Mylopoulos and Borgida, 1980]; it views reality as consisting of interrelated entities, and provides objects and properties to represent these entities and their interrelationships in a one-to-one fashion. For instance, let us consider an image whose contents can be described in natural language as follows:

Francesco is hugging his sister Giulia at their uncle's house in the summer 1993.

In order to express this sentence, the model would use two objects, let them be named **francesco** and **giulia**, to represent, respectively, the entities Francesco and Giulia, and two properties, namely **Sister** and **Brother**, to represent the relation between them. This can be stated in the following way: (**francesco Sister giulia**) and (**giulia Brother francesco**), where each triple is a *factual property*, and in particular a *multivalued* one, given the nature of the represented relations. By carrying this process on to all the entities and relationships of the world to be represented, a nexus of objects and links is generated, which is organized by means of three *abstraction* mechanisms.

First, classes are introduced as collections of similar objects, thus defining the *classification* mechanism. Each object belongs, or is an *instance of*, at least one class. In our example, two classes are introduced: **BOY** and **GIRL**, whose instances include, respectively, **francesco** and **giulia**, in symbols (**francesco** \rightarrow **BOY**) and (**giulia** \rightarrow **GIRL**). Each pair is called an InstanceOf link.

Second, a class defines a set of properties, which can be single- or multi-valued, for each of which the class members can specify an appropriate number of values. This gives the *aggregation* mechanism. In our example, we can make **Brother** and **Sister** multi-valued properties of the classes just introduced, ranging on the proper classes: (**BOY Brother BOY**), (**BOY Sister GIRL**), (**GIRL Brother BOY**), and (**GIRL Sister GIRL**). In words, boys and girls have boys as brothers and girls as sisters. Each of the above triples is called a *multi-valued definitional property*; the first element of the triple gives the class defining the property, the second gives the property name, and the third gives the property range, that is the class whose instances can be specified as values of the property. Factual and def-

initial properties are of course related, as it will be shown below.

Third, classes are taxonomically organized by means of the *IsA* relationship. This gives the *specialization* mechanism, which in our example can be used by having the class **PERSON** as a more general class (or super-class) of **BOY** and **GIRL**. *IsA* links will be denoted: (**BOY** \Rightarrow **PERSON**) (**GIRL** \Rightarrow **PERSON**). The *IsA* relationship is a partial order which captures the notion of concept inclusion: if a class *IsA* another class, then an object instance of the former is also an instance of the latter. This mechanism is usually called *instance inheritance*, and is illustrated in our example by the membership of **francesco** and **giulia** in the class **PERSON**: (**francesco** \rightarrow **PERSON**) and (**giulia** \rightarrow **PERSON**), coming as a consequence of their membership in the classes **BOY** and **GIRL** and of the *IsA* link from these classes to **PERSON**. More generally, each property defined by a class applies to the instances of its less general classes (or specializations), and this introduces in the model the so called *structural inheritance*. For instance, the name of a person can be defined as a single-valued definitional property of the corresponding class, ranging on the class of character strings (a built-in class of the model): (**PERSON Name STRING**). As already explained, this definition allows all persons to specify one value for **Name**, hence all boys and girls, which are special persons. We thus obtain by structural inheritance the ability of giving a name to **francesco** and **giulia**, which is done through the following single-valued factual properties: (**francesco Name "Francesco"**) and (**giulia Name "Giulia"**).

Formally, a content reconstruction (CR for short) is an 8-tuple $(O, C, SDP, MDP, SFP, MFP, \rightarrow, \Rightarrow)$ where, letting *ID* be a set of identifiers:

- O is a set of objects;
- C is a set of classes;
- $SDP \subseteq (C \times ID \times C)$ are the single-valued definitional properties;
- $MDP \subseteq (C \times ID \times C)$ are the multi-valued definitional properties;
- $SFP \subseteq (O \times ID \times O)$ are the single-factual properties;
- $MFP \subseteq (O \times ID \times O)$ are the multi-factual properties;
- $\rightarrow \subseteq (O \times C)$ is the InstanceOf relation, relating an object to the classes where it belongs;
- $\Rightarrow \subseteq (C \times C)$ is the *IsA* relation, relating a class to its superclasses.

A CR is consistent if it satisfies a number of constraints, aiming at capturing the intuitive meaning and role of its components. These constraints are:

1. Each object is an instance of at least one class, that is: for every o in O there exists a c_i in C such that $(o \rightarrow c_i)$.
2. SDP and MDP must be consistent, that is a property of a class cannot be single- and multi-valued at the same time: $\pi_{1,2}(SDP) \cap \pi_{1,2}(MDP) = \emptyset$, where, seeing R as a table, $\pi_{1,2}(R)$ denotes its projection on its first two columns.
3. SDP , MDP and SFP are functional, that is: $(c \ i \ d_1), (c \ i \ d_2) \in (SDP \cup MDP)$ implies $d_1 = d_2$, and $(o \ i \ o_1), (o \ i \ o_2) \in SFP$ implies $o_1 = o_2$.
4. Each factual property must be induced by a definitional property, that is: $(o_1 \ i \ o_2) \in SFP$ (resp. MFP) implies $(c_1 \ i \ c_2) \in SDP$ (resp. MDP) and $(o_i \rightarrow c_i)$ for $i = 1, 2$. For instance, (**francesco Sister giulia**) is allowed in MFP by the fact that (**BOY Sister GIRL**) is in MDP and that **francesco** and **giulia** are instances of **BOY** and **GIRL**, respectively. Moreover, each factual property must be consistent with *all* definitional properties that concern it (there may be many due to structural inheritance), that is: $(o_1 \ i \ o_2) \in SFP$ (resp. MFP), $(o_1 \rightarrow c_1)$ and $(c_1 \ i \ c_2) \in SDP$ (resp. MDP) imply $(o_2 \rightarrow c_2)$.
5. \Rightarrow is a partial order with a minimum \perp and a maximum \top . This implies that \perp and \top be in C .
6. Property definitions are inherited by specializations, that is: $(c_1 \ i \ d_1) \in SDP$ (resp. MDP) and $(c_2 \Rightarrow c_1)$ imply $(c_2 \ i \ d_2) \in SDP$ (resp. MDP) and $(d_2 \Rightarrow d_1)$. As already seen, (**BOY Name STRING**) and (**GIRL Name STRING**) must be in SDP as a consequence of the facts that (**PERSON Name STRING**) is in SDP and that (**BOY** \Rightarrow **PERSON**) and (**GIRL** \Rightarrow **PERSON**).
7. Instances are inherited by superclasses, that is: $(o \rightarrow c_1)$ and $(c_1 \Rightarrow c_2)$ imply $(o \rightarrow c_2)$.

From now on, we will tacitly assume only consistent CR's. When no ambiguity will arise, we will collect definitional and factual properties into the sets:

$$DP = SDP \cup MDP$$

$$FP = SFP \cup MFP.$$

6 Images

One of the basic steps of image interpretation by humans is the recognition of the entities that occur in an image, that is the association of the pictorial representation of these entities with the entities themselves, or rather with some mental representation of them. This association is expressed in our model by the *image mapping*, a partial function from disjoint sets of spots in an image form to objects in the CR of that image.

An *image* is a triple $(\mathcal{M}, \mathcal{D}, M)$, where $\mathcal{M} = (A, \pi, F)$ is an image form, $\mathcal{D} = (O, C, DP, FP, \rightarrow, \Rightarrow)$ is a CR, and M is a partial function:

$$M : 2^\pi \rightarrow O$$

such that for all $\Pi_i, \Pi_j \in \text{dom}(M)$, $M(\Pi_i) = M(\Pi_j)$ implies $\Pi_i = \Pi_j$, i.e. M is an injection. We will call M the *image mapping*.

The injectivity of the image mapping captures a maximality criterion, aiming at ruling out mappings which associate the same content object to several sets of spots. No other constraint is set on M to gain generality. In particular, a spot in an image can be assigned to more than one object in the image's content reconstruction (this allows to represent object decomposition), and the domain of the image mapping is not required to cover π , as not necessarily any spot of color in an image depicts an entity or a part of it.

7 An image query language

The query facility for the model so far introduced is a many-sorted first-order logic, whose language \mathcal{L}_I is built on the following sorts:

1. σ_r , the sort of *regions*; the symbols from this sort are: countably many constant symbols, denoted by capital letters drawn from the beginning of the alphabet, such as A, B , (metasymbols r_1, r_2, \dots); countably many variables, small letters from the end of the alphabet with a distinguishing subscript r , such as x_r, y_r ; the existential quantifier \exists_r .
2. σ_c , the sort of *colors*; the symbols from this sort are: countably many constant symbols, denoted by their English names (metasymbols c_1, c_2, \dots); countably many variables: x_c, y_c , and so on; the existential quantifier \exists_c .
3. σ_o , the sort of *objects*; the constant symbols from this sort will be denoted by lower case names, such as: *francesco*; object variables will be small letters from the end of the alphabet with the usual distinguishing subscript, such as x_o, y_o ; as existential quantifier for objects, \exists_o will be used.
4. σ_p , the sort of *classes*; the constant symbols from this sort will be denoted by upper case names, such as *PERSON*, with the exception of the two symbols \top and \perp ; class variables and existential quantifier will be the usual symbols with the distinguishing subscript p .
5. σ_i , the sort of *identifiers*; the constant symbols from this sort will be denoted by names starting with a capital letter, such as *Brother*; identifier variables and existential quantifier will be the usual symbols with the distinguishing subscript i .

In order to specify a semantics for the sentences of \mathcal{L}_I , we define the notion of *denotation function*, as a one-to-one mapping associating:

- the constant symbols of sort σ_r and the finite subsets of pairs of natural numbers (notice that the latter are countably many, so that the mapping can be effectively established);
- the constant symbols of sort σ_c and the set of colors L ;
- the constant symbols of sort σ_o, σ_p , and σ_i onto the sets O, C , and ID , respectively. For simplicity, this mapping is assumed to be the identity function.

Out of the denotation functions we factor one function, d , and call it the *image denotation function*. An *image structure* is a pair (\mathcal{I}, d) , where \mathcal{I} is an image. Notice that the function d does not change from structure to structure.

\mathcal{L}_I provides four kinds of predicate symbols, introduced in the following sections.

7.1 Visual predicate symbol

A visual query is an image itself stated against the form component of an image, and is typically expressed through an appropriate visual tool. From the modelling point of view, a visual query is an expression of the query language built out of the dyadic symbol I of sort (σ_r, σ_c) , which names the association between regions and colors necessary to describe image spots. For instance, the atomic ground sentence $I(A, \text{blue})$ means that region A is *blue*.

An image structure is said to *satisfy* the atomic ground sentence $I(r_1, c_1)$,

$$(\mathcal{I}, d) \models I(r_1, c_1) \text{ if and only if } F(d(r_1)) = d(c_1).$$

In words, an image structure satisfies a ground atomic instance of I if, in the image form of the structure, the region designated by r_1 (i.e. $d(r_1)$) is of the color designated by c_1 ($d(c_1)$).

This defines an *exact* visual match. It is well known that non-exact queries are also very useful in image retrieval, as they allow to retrieve images that match queries only to a certain extent.

The non-exact image matching problem is a special case of the *registration* problem, which consists in finding a mapping between two given images, possibly with an associated degree of precision according to a specified similarity metric (for a survey on image registration, see [Brown, 1992]). This problem arises in a number of areas in computer vision, pattern recognition, medical image analysis and others.

There is a large variety of techniques which have been proposed to solve the registration problem, yet none of these stands out as *the* image matching function able to satisfy any user. Indeed, given the complexity of image

semantics and the variety of applications requiring some form of image matching, the search for a universal image matching function seems to be much of an ill-defined problem.

The solution we propose to endow our model with a non-exact query facility is to let the user select the image registration techniques that best capture the application’s requirements, and import these techniques in the model. Formally, let Φ_1, \dots, Φ_n , $n \geq 0$, be n *image predicates*, that is total functions each receiving in input an image form, a region, and a color and returning an element of the set $\{0, 1\}$. On the basis of image predicates, $(n + 1)$ *partial satisfaction* relations on the ground atomic instances of I can be defined as follows:

$$\begin{aligned} (\mathcal{I}, d) \models_0 I(r_1, c_1) &\text{ iff } F(d(r_1)) = d(c_1) \\ (\mathcal{I}, d) \models_i I(r_1, c_1) &\text{ iff } \Phi_i(\mathcal{M}, d(r_1), d(c_1)) = 1, \end{aligned}$$

for all $1 \leq i \leq n$. The relation \models_0 captures the exact query mechanism, the other relations \models_i capture an uncertain match, as established by the image predicates. It should be noted that an image predicate need not be as concise and elegant as a sentence of a formal language. It may be a complex system of equations or a computer programme, as long as it specifies a total computable function which decides whether or not a given image form matches a given spot.

A satisfaction relation for ground atomic visual queries capturing also non-exact matchings can now be stated as follows:

$$1. (\mathcal{I}, d) \models I(r, l) \text{ iff for some } 0 \leq i \leq n, (\mathcal{I}, d) \models_i I(r, l).$$

7.2 Spatial predicate symbols

Spatial reasoning plays a primary role in many image retrieval applications. Typically, users may want to talk in their queries about objects being “left to”, or “north-east to” or “surrounded by” other objects. In order to express these queries, we introduce spatial predicate symbols in \mathcal{L}_I .

Considering images as rectangles, the complete set of spatial relationships involved in image retrieval can be obtained as a binary combination of the complete set of spatial relationships for one-dimensional intervals. This set has been derived in [Allen, 1983] upon investigating a form of reasoning about time; it consists of 13 relationships, representable with 7 different symbols, named: **before**, **equal**, **meets**, **overlaps**, **during**, **starts** and **finishes**. Figure 1 illustrates pictorially the meaning of each symbol.

As we have to deal with two dimensions, we introduce in \mathcal{L}_I two predicate symbols for each one shown in Figure 1, using a prefix to indicate the dimension of the predicate. Each symbol is of sort (σ_r, σ_r) . For instance, $X_before(r_1, r_2)$ means that the projection on the x axis of region r_1 is before that of region r_2 , in the sense illustrated pictorially in Figure 1.

Symbol	Meaning
X before Y	XXX YYY
X equal Y	XXX YYY
X meets Y	XXXYYY
X overlaps Y	XXX YYY
X during Y	XXX YYYYYY
X starts Y	XXX YYYYYY
X finishes Y	XXX YYYYYY

Figure 1: Interval Relationships.

More formally, the semantics of ground atomic instances of the 14 spatial predicate symbols is given in terms of image structures with four additional functions: two projection functions, Π_x and Π_y , which return, respectively, the x and y projection of a given region; and two relations on intervals: precedence ($<$) and inclusion (\sqsubseteq). Notice that the projection of an image region is always an interval, as an image region is defined to be a connected subset of ω^2 . Given two region symbols r_1 and r_2 , the satisfaction relation on ground atomic instances of spatial predicates is defined as follows (for brevity, only some predicate symbols on the horizontal dimension are considered):

2. $(\mathcal{I}, d) \models X_before(r_1, r_2)$ iff $\Pi_x(d(r_1)) < \Pi_x(d(r_2))$.
3. $(\mathcal{I}, d) \models X_during(r_1, r_2)$ iff $\Pi_x(d(r_1)) \sqsubseteq \Pi_x(d(r_2))$.
4. $(\mathcal{I}, d) \models X_equal(r_1, r_2)$ iff $\Pi_x(d(r_1)) \sqsubseteq \Pi_x(d(r_2))$ and $\Pi_x(d(r_2)) \sqsubseteq \Pi_x(d(r_1))$.
5. $(\mathcal{I}, d) \models X_overlaps(r_1, r_2)$ iff for some interval h , $h \sqsubseteq \Pi_x(d(r_1))$ and $h \sqsubseteq \Pi_x(d(r_2))$.
6. $(\mathcal{I}, d) \models X_meets(r_1, r_2)$ iff $\Pi_x(d(r_1)) < \Pi_x(d(r_2))$ and for no interval h , $\Pi_x(d(r_1)) < h < \Pi_x(d(r_2))$.

We have thus endowed our query language with the machinery to reason about certain spatial relationships between regions. We will see in section 8.6 how this expressive power can be used to reason about spatial relationships between objects.

7.3 Content predicate symbols

The predicate symbols for content queries are:

- *MDP* and *SDP*, of sort $(\sigma_p, \sigma_i, \sigma_p)$;
- *MFP* and *SFP*, of sort $(\sigma_o, \sigma_i, \sigma_o)$;
- *InstanceOf*, of sort (σ_o, σ_p) ;
- *IsA*, of sort (σ_p, σ_p) .

An image structure satisfies a ground instance of the above predicate symbols:

7. $(\mathcal{I}, d) \models MDP(c_1, i, c_2)$ iff $(c_1, i, c_2) \in MDP$
8. $(\mathcal{I}, d) \models SDP(c_1, i, c_2)$ iff $(c_1, i, c_2) \in SDP$
9. $(\mathcal{I}, d) \models MFP(o_1, i, o_2)$ iff $(o_1, i, o_2) \in MFP$
10. $(\mathcal{I}, d) \models SFP(o_1, i, o_2)$ iff $(o_1, i, o_2) \in SFP$
11. $(\mathcal{I}, d) \models InstanceOf(o, c)$ iff $(o, c) \in \rightarrow$
12. $(\mathcal{I}, d) \models IsA(c_1, c_2)$ iff $(c_1, c_2) \in \Rightarrow$

This semantics is simply an extension to our semantic data model of the model theoretic view of databases. We are assuming that a CR is a strict analog of the reality being modelled, thus a closed world in which a sentence holds true (i.e. the CR satisfies the sentence) or false.

7.4 Mapping predicate symbol

In order to query the mapping component of an image, \mathcal{L}_I provides the dyadic predicate symbol *Map*, of sort (σ_r, σ_o) , which can be used to associate image regions with content objects. For instance, $Map(r, o)$ means that the region r is mapped by the mapping component M of the image being considered onto the content object o . Formally, given an image \mathcal{I} ,

13. $(\mathcal{I}, d) \models Map(r, o)$ iff for some set of regions X , $d(r) \subseteq \bigcup X$ and $M(X) = o$.

8 Querying an image base

Having introduced the alphabet of \mathcal{L}_I and its semantics, a query facility for our model can now be completely specified. This is done in the next section, in two steps. First, the syntax of \mathcal{L}_I is completed, allowing us to define what formulas of the language count as image queries. Then, the semantics of the full language is given, allowing us to define the image answer function. In the following sections, the expressive power of the introduced query facility is examined.

8.1 A query facility

As \mathcal{L}_I has no function symbol, its *terms* are just constant symbols or variables, whose sorts give the sorts of the corresponding terms. The *atomic formulas* of \mathcal{L}_I are the atomic ground instances of the predicate symbols introduced in the previous section. The *well-formed formulas* of \mathcal{L}_I are the smallest set containing the atomic formulas and the formulas: $\neg\alpha$, $(\alpha \vee \beta)$, $(\exists_r x)\alpha$, $(\exists_c x)\alpha$, $(\exists_o x)\alpha$,

$(\exists_p x)\alpha$, and $(\exists_i x)\alpha$, where α and β are well-formed formulas. Notice the usage of one existential quantifier for each sort of the language, ranging on the corresponding sort. As customary, the well-formed formulas made up from all the interesting connectives and one universal quantifier for each sort can be assumed in the language as abbreviations of primitive expressions. A *sentence* is a well-formed formula with no free variables. An *image query* is any sentence of \mathcal{L}_I .

For simplicity, we will provide a semantics only for the sentences of \mathcal{L}_I , which model image queries. The semantics of the atomic sentences has been already given in the previous section. The following rules extend that semantics to all the sentences of the language. In them, (\mathcal{I}, d) is an image structure, α and β well-formed formulas, γ a well-formed formula in which the variable x occurs in instances of I or of spatial predicate symbols, and δ a well-formed formula in which the variable x occurs only in instances of *Map*. Finally, α_c^x is the same formula as α except that all the occurrences of the variable x are replaced by occurrences of the constant symbol c .

14. $(\mathcal{I}, d) \models \neg\alpha$ iff $(\mathcal{I}, d) \not\models \alpha$
15. $(\mathcal{I}, d) \models (\alpha \vee \beta)$ iff either $(\mathcal{I}, d) \models \alpha$ or $(\mathcal{I}, d) \models \beta$
16. $(\mathcal{I}, d) \models (\exists_r x)\gamma$ iff for some constant symbol c of sort σ_r , $d(c) \in \pi$ and $(\mathcal{I}, d) \models \gamma_c^x$
17. $(\mathcal{I}, d) \models (\exists_r x)\delta$ iff for some constant symbol c of sort σ_r , $(\mathcal{I}, d) \models \delta_c^x$
18. $(\mathcal{I}, d) \models (\exists_c x)\alpha$ iff for some constant symbol c of sort σ_c , $(\mathcal{I}, d) \models \alpha_c^x$
19. $(\mathcal{I}, d) \models (\exists_o x)\alpha$ iff for some constant symbol c of sort σ_o , $(\mathcal{I}, d) \models \alpha_c^x$
20. $(\mathcal{I}, d) \models (\exists_p x)\alpha$ iff for some constant symbol c of sort σ_p , $(\mathcal{I}, d) \models \alpha_c^x$
21. $(\mathcal{I}, d) \models (\exists_i x)\alpha$ iff for some constant symbol c of sort σ_i , $(\mathcal{I}, d) \models \alpha_c^x$

These rules extend the notion of satisfaction to the non atomic sentences of \mathcal{L}_I , taking into account the sorted nature of the language. Notice that in 16 the membership of $d(c)$ in π is required to ensure that when the variable x denotes a region of an image form, the quantifier \exists_r ranges on the spots of that image form rather than on the finite sets of pairs of natural numbers. In this way, the sentence $(\forall_r x)I(x, green)$ is satisfied by all images which are entirely green, as desired, whereas the sentence $(\exists_r x)X_before(x, r)$ is satisfied by all images having a region at the left of r . Without this provision, the former sentence would be satisfied by no image, as images forms are finite objects, whereas the latter would be satisfied by no images if r is a leftmost region, and by all images otherwise, regardless of their shape. The

same restriction does not apply to *Map* instances, because the image mapping function takes as argument sets of regions. Thus the sentence $Map(r, o)$ may be satisfied by an image even though r is not a region of its image form; the semantics requires only that r be included in the regions mapped by M onto o .

Given a collection of images IB and an image query q , the *answer of q in IB* , $a(q, IB)$, is given by:

$$a(q, IB) = \{ \mathcal{I} \in IB \mid (\mathcal{I}, d) \models q \}.$$

This query facility subscribes to the logical view of information retrieval recommended in [van Rijsbergen, 1986], as it models retrieval as logical inference. At the same time, it conforms to the model-theoretic approach to databases, as it views an image as a strict analog of the reality being modelled, thus a closed world in which every sentence of \mathcal{L}_I holds true (i.e. the image satisfies the sentence) or false. We thus have an image model which is based on solid philosophical and mathematical grounds, and offers agreeable computational properties. Firstly, answers are decidable. Secondly, the decision algorithm is conceptually simple. Thirdly, the cost of that algorithm can be reduced at will by limiting the expressive power of the query language, in a way that is well-known.

The expressive power of our image query facility can be investigated by considering that a query can span over four information dimensions: the visual, spatial, content, and mapping dimensions, which partition the predicate symbols of the query language. From this point of view, the language offers 15 types of queries, one for each non-null assignment to four binary variables corresponding to these dimensions. Preserving the above ordering, queries of type 1 are those having only the mapping dimension (assignment 0001), while queries of type 6 (0110) have both the spatial and content dimensions. In the following sections we will review the types of queries on images offered by the model, focusing on the most important of them.

8.2 Visual queries

A visual query is a query in which only the predicate symbol I occurs and, as already observed, is expected to be expressed through an appropriate visual tool. Assuming that such a tool provides the machinery to deal with first-order syntax, logical connectives and quantifiers can be used to articulate complex visual queries, such as $(I(A, blue) \rightarrow I(B, green))$ returning the images whose A region is not *blue* or whose B region is *green*, or:

$$(\exists_r x)I(x, blue) \vee (\exists_c x)(\forall_r y)I(y, x)$$

returning the images which either have a *blue* spot or whose regions are all of the same color.

In general, an image satisfies the queries that are “less specified” or “more vague” than its image form. For instance, let us consider an image form \mathcal{M} having 3 regions, named for simplicity A , B , and C , whose colors

are, respectively, *white*, *blue*, and *green*. A query less specified than \mathcal{M} is one describing only a subset of the image’s spots, such as $(I(A, white) \wedge I(C, green))$ or denoting *also* other images, such as $(I(B, blue) \vee \delta)$ where δ is any visual query. It is not difficult to see that both these sentences are satisfied by \mathcal{M} . In the former case the satisfaction can even be visualized, since the sentence has a straightforward pictorial representation. It is important to realize that \mathcal{M} satisfies the above two sentences regardless of the nature of the involved objects. The reason is that these inferences are valid in the first-order predicate calculus, and, by interpreting in a certain way the sentences that occur in them, we have simply rephrased the classical notion of satisfaction in terms of images. The interested reader may find more about the logic underlying visual queries in [Meghini, 1994a].

8.3 Spatial queries

This kind of queries contain only instances of spatial predicate symbols and can be used to retrieve images on the bases of spatial relationships between the images’ spots.

Ground spatial queries state properties of specified regions, and therefore are not particularly interesting. They either return no image, in case the stated property does not hold, or otherwise they return all the images in the image base. For instance, the sentence:

$$X_before(r_1, r_2) \wedge \neg Y_before(r_1, r_2) \wedge \neg Y_before(r_2, r_1)$$

asserts that region r_1 precedes region r_2 on the x axis and they share one interval on the y axis, and is clearly either true or false regardless of any particular image. By quantifying on either region, a more interesting query results, whose answer depends on the images stored in the image base. For instance, using an existentially quantified region variable in place of r_1 , we have a query which retrieves the images in whose image forms there is a region standing to r_2 in the above mentioned relationship.

In order to query spatial relationships between objects, the spatial predicate symbols are to be used in conjunction with the mapping symbol. Queries of this kind are said to be strongly mixed, as they involve more than one dimension of the image representation, and will be discussed in section 8.6.

8.4 Content queries

A content query includes only instances of the content predicate symbols, allowing to specify conditions which the retrieved images are to be *about*. The expressiveness of the content query language allows to state conditions on objects, classes, properties, and IsA and InstanceOf links in a simple and uniform way.

For instance, the images that are about someone named “Francesco” can be retrieved by the query:

$$(\exists_o x)(\exists_i y)SFP(x, y, \text{“Francesco”})$$

which returns the images in whose content part some object x has value ‘‘Francesco’’ for some property y , which is presumably a naming property. Similarly, the images which are about a musician, brother of Giulia can be requested via the following query:

$$(\exists_o x)(InstanceOf(x, MUSICIAN) \wedge MFP(x, Brother, giulia)).$$

In order to retrieve the images with a non-empty content component, the following query can be used:

$$(\exists_o x)(\exists_p y)InstanceOf(x, y).$$

8.5 Mapping queries

A mapping query contains only instances of the *Map* predicate symbol and can be used to retrieve images whose regions are associated to content objects. When both regions and content objects are denoted by constant symbols, the query returns, as established by the semantics of *Map*, the images whose mapping component associates the mentioned regions (or supersets of them) with the mentioned objects. By quantifying on the regions, a query asks images which have map information on the specified content objects; for instance the query $(\exists_r x)Map(x, o)$ returns the images whose mapping function has o in its range, which means that o is shown in the image form and that the mapping function ‘‘points it out’’.

A form of shape-based retrieval can be performed via mapping queries which quantify on objects. For instance the query:

$$(\exists_o x)(Map(r_1, x) \vee Map(r_2, x))$$

retrieves the images in which either r_1 or r_2 (or a superset of them) are mapped into some unspecified object. Clearly, there is some noise due to the superset matching; in addition, the position of the shape is fixed, since a region is a fixed portion of the assumed 2-dimensional space. The latter problem can be solved by introducing in the query language suitable operators for scaling, rotating and translating regions, while the former is not expected to be significantly serious.

8.6 Mixed queries

Mixed queries exhibit instances of predicate symbols of at least two kinds. They can be subdivided into two categories:

- *weakly* mixed queries, in which no term is shared by instances of predicate symbols of different kinds; and
- *strongly* mixed queries, where there is at least one common variable or constant symbol between instances of predicate symbols of different kinds.

Queries belonging to the former category exploit the power of the image representation only to a limited extent, as they are Boolean combinations of unrelated

mono-dimensional queries. As a consequence, weakly mixed queries cannot address spatial relationships between objects, because these cannot be asserted of objects but simply of regions. Assuming that δ is a visual query, an example of a weakly mixed query is:

$$\delta \wedge (\exists_p x)InstanceOf(francesco, x)$$

which returns the images whose image form matches δ and whose content component includes the object *francesco*. No connection between the two conditions is expressed by the query. As Boolean combinations of types of queries already examined, weakly mixed queries will not be discussed in more detail. Notice that queries of type 6, having spatial and content predicate symbols, and queries of type 10 (visual and content) can only be weakly mixed, as the predicate symbols occurring in them have no sort in common.

Strongly mixed queries permit the expression of conditions which address more than one level of the image representation. A systematic discussion of these queries can be found in [Meghini, 1994b]. In the rest of this section, examples of various types of strongly mixed queries will be presented to give an idea of the power of the query language.

A query of type 3 (content + mapping):

$$(\exists_o x)(Map(r, x) \wedge InstanceOf(x, TREE))$$

returning the images in which a group of regions, which include r , is mapped onto an instance of the class *TREE* by the mapping component. More succinctly, this query can be phrased as ‘‘A tree of shape (the contour of) r ’’. By quantifying on the region, the query is obtained:

$$(\exists_o x)(\exists_r y)(Map(y, x) \wedge InstanceOf(x, TREE))$$

requesting images in which some region, regardless of its shape, is mapped onto a tree (i.e. ‘‘A tree’’). There is an important difference to be noted between this query and the query:

$$(\exists_o x)InstanceOf(x, TREE).$$

The former returns the images showing a tree, while the latter those which are about a tree, which means that the image content reconstruction includes a tree, but not necessarily the image form.

Queries of type 11 are queries of type 3 with an additional visual component. As an example, the query:

$$(\exists_r x)(\exists_o y)(I(x, yellow) \wedge Map(x, y) \wedge InstanceOf(y, TREE))$$

returns the images in which a region of color *yellow* is mapped onto a tree (i.e. ‘‘A yellow tree’’).

Queries of type 5 (spatial and mapping) can be used to state spatial conditions between content objects or regions and other content objects. The following query retrieves the images in which *giulia* is left to *francesco*:

$$(\forall_r x)(Map(x, giulia) \rightarrow ((\forall_r y)Map(y, francesco) \rightarrow X_before(x, y)) \wedge ((\exists_r y)Map(y, francesco) \wedge Y_includes(x, y)))$$

where $Y_includes(a, b)$ is an abbreviation for:

$$Y_starts(a, b) \vee Y_during(a, b) \vee \\ Y_finishes(a, b) \vee Y_equal(a, b).$$

The query requires all *giulia*'s regions to be (a) left of all *francesco*'s regions on the x axis and (b) entirely contained in at least one of *francesco*'s regions on the y axis.

Queries of type 7 permit to perform “iconic” image retrieval, consisting in requesting images on the basis of the spatial relationships between their content objects. A typical iconic query is [Chang *et al.*, 1987] “find all pictures having a tree to the left of a house”, expressible in our query language as:

$$(\exists oxy)(InstanceOf(x, TREE) \wedge InstanceOf(y, HOUSE) \wedge \\ (\forall_r u)(Map(u, x) \rightarrow \\ ((\forall_r v)Map(v, y) \rightarrow X_before(u, v)) \wedge \\ ((\exists_r w)Map(w, y) \wedge Y_includes(u, w))))).$$

Finally, queries of type 15 include all four components, permitting to express requests such as: “Giulia is wearing a pink sweater and there is a person at her right”, “A yellow tree on a grass”, and “A house with green windows left to a tree”.

9 Conclusions

We have presented a model for image bases offering a 3-level representation of images and a query facility for exploiting the content of an image base in a complete way. Our work must be considered as having a foundational spirit, as both the proposed representation schemes and their query languages need extension in order to arrive at a rich model, able to cope with the applications complexity. Besides its applicability to image bases, the model provides a basic philosophy to deal with the representation and retrieval of objects with a content. As such, the model makes a step towards a general model for multimedia information systems, providing a paradigm and representational and querying primitives.

10 Acknowledgements

We thank the components of the MIRO Working Group (ESPRIT Basic Research Action n. 6576) for providing an ideal environment of growth of ideas. The work reported in this paper has been partially funded by the EEC under ESPRIT Basic Research Action FERMI, n. 8134.

References

- [Allen, 1983] J. F. Allen. Maintaining knowledge about temporal intervals. *Communications of the ACM*, 26:832–843, 1983.
- [Brown, 1992] Lisa G. Brown. A survey of image registration techniques. *ACM Computing Surveys*, 24(4):325–376, December 1992.
- [Chang *et al.*, 1987] Shi-Kuo Chang, Qing-Yun Shi, and Cheng-Wen Yan. Iconic indexing by 2-D strings. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 9(3):413–427, May 1987.
- [Davidson, 1985] Donald Davidson. *Essays on Actions and Events*. Clarendon Press, Oxford, England, 1985.
- [Faloutsos *et al.*, 1994] C. Faloutsos, R. Barber, R. Flickner, J. Hafner, W. Niblack, D. Petkovic, and W. Equitz. Efficient and effective querying by image content. *Journal of Intelligent Information Systems*, 3:231–262, 1994.
- [Hull and King, 1987] Richard Hull and Roger King. Semantic database modeling: Survey, applications and research issues. *ACM Computing Surveys*, 19(3):201–259, 1987.
- [Meghini, 1994a] Carlo Meghini. Logical image modelling and retrieval. Technical Report B4-05, Consiglio Nazionale delle Ricerche, Istituto di Elaborazione della Informazione, April 1994.
- [Meghini, 1994b] Carlo Meghini. A model for image bases and its query facility. Technical Report B4-46, Consiglio Nazionale delle Ricerche, Istituto di Elaborazione della Informazione, November 1994.
- [Mylopoulos and Borgida, 1980] John Mylopoulos and Alex Borgida. Some features of the Taxis data model. In *Proceedings of the 6th International Conference on Very Large Data Bases*, pages 399–410, Montreal, 1980.
- [Mylopoulos *et al.*, 1980] John Mylopoulos, Harry K.T. Wong, and Philip A. Bernstein. A language facility for designing database-intensive applications. *ACM Transactions on Database Systems*, 5(2):185–207, June 1980.
- [Rosenfeld and Siromoney, 1993] Azriel Rosenfeld and Rani Siromoney. Picture languages—a survey. *Languages of design*, 1(3):229–245, Aug. 1993.
- [van Rijsbergen, 1986] Corneliis J. van Rijsbergen. A new theoretical framework for information retrieval. In *Proceedings of SIGIR-86, 9th ACM Conference on Research and Development in Information Retrieval*, pages 194–200, Pisa, I, 1986.