Consiglio Nazionale delle Ricerche

ISTITUTO DI ELABORAZIONE DELLA INFORMAZIONE

PISA

A Model for Image Bases and Its Query Facility

Carlo Meghini

Rapporto Tecnico B4-46
Novembre 1994

A Model for Image Bases and its Query Facility

Carlo Meghini

Consiglio Nazionale delle Ricerche

Istituto di Elaborazione dell'Informazione

Via S. Maria, 46 - 1-56126 Pisa, Italy

meghini@iei.pi.cnr.it

Abstract

A model for image bases is presented, along with a query facility for exploiting image bases contents. The models views an image as a 3-level object, comprised of a form, a content and a mapping component. The form level provides an objective model for images as syntactical objects, against which visual queries (Le. queries being images themselves) can be stated. The content level provides a representation of the scene depicted by an image, in a semantic data modelling style. The mapping level serves to connect the form and content levels, establishing a relationship between visual appearances of individuals and their conceptual descriptions. A query is a sentence of a logical language that can address any subset of the three image components and also the spatial relationships between objects of the form level (i.e. image regions) or of the content level (Le. content entities). An image is retrieved in response to a query if it logically satisfies the query.

Categories and subject descriptors: H.1.0, H.3.4 Retrieval Models, H.2.3 Query Languages.

General terms: Theory, Languages.

Additional keywords and phrases: Image Modelling, Image Retrieval.

This work has been partially funded by the EEC under Basic Research Action FERMI, n. 8134.

1 Introduction

Due to the impressive achievements of hardware, network and HCI technologies in the last decade, large repositories of information centered around image objects, called *image bases* (IBs), are being constructed, maintained and accessed by specialized users. It is foreseen that in the medium term IBs will serve the information needs of an increasing community of users with decreasing computer skills.

Images in an IB are stored and accessed according to an *image model,* which plays to an IB a role analogous to that of a data model to a database: it provides a representation for images and for queries on images, and a relation between these two, establishing which images are to be returned in response to each query. The nature of images and that of an information system model dictate a number of requirements to be satisfied by an image model.

First, the model must abstract implementation details, and, at the same time, be implementable with the available technology. The very notion of a data model was conceived with both physical data independence and productivity in mind, and this requirement continues to be the basic concern of any information system model.

Second, the model should provide a unique framework for accommodating the many forms of image retrieval that have been studied in various areas of computer science, ranging from pattern matching to semantic information processing. Typically, certain users of an IB would like to retrieve images by posing images themselves as queries and having the system performing a match, perhaps 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, the model must support a universal representation of images, providing the distinction between *form* and *content* and a mechanism to relate the two. In this way, the model can accommodate a wide range of retrieval functionalities and, more importantly, is open to be extended with new techniques that may emerge from research.

Finally, the model should be stated in formal terms, so that it can be understood by users and implementors and analyzed by system designers, for instance to be compared with other models.

For a more effective comparison with our model, we will postpone the discussion on other image models proposed in the literature until section 7. For the present time, we can summarize the conclusions of this discussion by saying that the image models proposed so far fail to meet one or another of these requirements. From one hand, the models that allow the users to pose visual queries often fail to state in semantical terms what is the relationship between a query and its answer, and how this relationship stands to intuition. From the other hand, the models that address the content representation of images in a way that goes beyond text retrieval techniques are mostly bound to specific application fields, thus lacking the generality which is necessary to meet the first of the above requirements. To the best of our knowledge, no model exists that possesses

the representation capabilities and the generality recommended by the second requirement.

We propose a model for image bases which satisfy all the above stated requirements. In brief, our model views an IB as consisting of two main components:

- a *concept schema,* analogous to a database schema, containing the definition of the conceptual entities occurring in the content representations of the images; and
- an *image population,* analogous to the objects of a database, consisting of the images stored in the IB.

Images are viewed 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 spots. The content level is an extension of the concept schema, and, along with this schema, is couched in terms of a semantic data model, offering object-orientation and abstraction mechanisms to describe a slice of reality. 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.

We will present the model in a bottom-up fashion, starting from image models (Section 2), and then proceeding to image contents (Section 3). This will allow us to introduce the model while providing the rationale for its machinery (for convenience, a summary of the model is provided in the Appendix). Images and image bases are introduced in Section 4, while section 5 describes the basic elements of an image query language. The query facility of the model is given is Section 6, where image queries and answers to them are defined. The expressive power of the image query language is also investigated, by presenting the various categories of queries allowed by the language. Finally, Section 7 relates the present work to image modelling in a wide range of different areas, with emphasis on iconic and pictorial databases.

2 Image models

The distinction between the form of an image and its content is not always definite in our mind. It is commonly believed that visual perception and interpretation are deeply intertwined, thus making it difficult to discriminate between an "objective" and a "subjective" level of an image. Fortunately, the epistemological apparatus of a computer is much simpler, or at least better understood, and allows us to identify an objective, or form level in an image base.

Whether physically stored in a raster, or pixel, or vector mode, an image can be conveniently abstracted into a composition of colored areas. More precisely, an image can be viewed as a partition of a 2-dimensional, connected region into sub-regions, called *spots,* such that: (a) each

spot consists of discrete points having the same color, and (b) no two contiguous spots have the same color.

In order to be able to relate all images, regardless of their relative position in space and contour shape, we set the origin of a discrete, 2-dimensional Cartesian space on the bottom left corner of the smallest rectangle including an image so that any image can then be represented as an *image model*, that is 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;
- $\pi = \{S_1, S_2, \ldots, S_n\}$, the *spots* of the image, is a partition of A consisting of connected sets; and
- 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.

In order to capture the condition (b) above, we further require that, for each image model (A, π, F) and pair of contiguous spots S_i and S_j in π, $F(S_i) \neq F(S_j)$.

An image model 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.

3 Modelling image contents

3.1 Baseline

The past experience with information retrieval systems has revealed that the retrieval of informal information objects, such as pieces of text or images, can be effective only if a possibly rich representation of the *content* of such objects is available [27]. This has been more recently argued for multimedia information systems, upon proposing a conceptual view of multimedia documents [17].

The content of an image is a scene and 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 of the model to multiple reconstructions does not present conceptual difficulties.

Modelling the interpretation of images is beyond the scope of an image model, which must instead provide a tool for representing the result of that process, i.e. content reconstructions. This tool must allow 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.

To see why, 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.

Clearly, this description shows knowledge of "hidden" details, that is information which could not be derivable by *any* interpreter of the same image. We would like to stress that this is often the case with indexers of documents: they are knowledgeable in the field which the documents are about, and it is important to provide them with a language that allows to express such knowledge and use it in retrieving information. Current retrieval systems typically employ, if any, one of two techniques to represent the content of an image:

- The *caption* method, which consists in associating an image with a piece of text describing the scene depicted by the image. In our case, a suitable candidate could be the above description. Content-based retrieval is then carried out by applying text retrieval techniques to the image description, perhaps with the aid of some information structure, such as a thesaurus.
- The *keyword* method, which consists in associating an image with a set of terms drawn from a prefixed language. In our case, a possible description could be "Francesco Giulia summer-1993", but usually things go worse, because an index language very seldom provides names for individuals.

It is not difficult to see that these content representations allow to retrieve their associated image only in case of very precise queries, that is queries containing a word occurring in the representation, such as "Giulia". But it would already be problematical for either of these representations to satisfy a query mentioning "Francesco's sister": in the keyword representation the system would have to make the guess "Giulia is Francesco's sister", whereas in the former case the system would have to understand a natural language sentence, a notoriously difficult task. Queries asking for "Francesco at his uncle house" or for "Giulia's brother" would in no case be satisfied by either representation, despite the fact that the involved information was known by the image indexer.

In fact, these two techniques reflect two classical approaches to text retrieval: full-text and keyword-based retrieval. The experimental evaluation of systems employing either of these techniques has almost invariably (and not surprisingly) produced disappointing results, which have been justified on different grounds. But there is a wide-spread agreement that a richer, formal representation of documents contents is, if not the only, at least a very promising way out of the problem.

We argue that a semantic data model is a suitable tool for the representation of image contents in order to perform content-based image retrieval (for a survey on semantic data models, see [9]). Such a model can be seen as a close relative of predicate calculus [20] emphasizing organization and computational amenity.

3.2 Semantic modelling of image contents

The semantic data model that we are proposing for content reconstructions offers the basic features common to almost all the models proposed in the literature, although the terminology and the formal apparatus we use are inspired to Taxis [19]. This model 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. Put at work on the example above, the model would have 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 definitional properties are of course related, as it will be shown below.

Third, classes are taxonomically organized by means of the *IsA* relationship. This give the *specialization* mechanism, which in our example can be used by having the class `PERSON` as a more general class (or superclass) 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")`. The next section will provide a precise definition of the model.

3.3 A model for content representations

A *database state* 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 database state is consistent if it satisfies a number of constraints, aiming at capturing the intuitive meaning and role of its components. These constraints are [18]:

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 $\pi_{1,2}(R)$ is the projection of R 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 $d_1 = d_2$,
   $(o\ i\ o_1), (o\ i\ o_2) \in SFP$ 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 $\bot$ and a maximum $\top$. This implies that $\bot$ and $\top$ be in C.

6. Property definition 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 database states. When no ambiguity will arise, we will collect definitional and factual properties into the sets $DP = (SDP \cup MDP)$ $FP = (SFP \cup MFP)$ naming *definitional part* the C, DP, and $\Rightarrow$ components of a database state and *factual part* the remaining components.

4 Images and Image bases

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 our 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 model to objects in the database state which gives the content reconstruction of that image.

An *image* is a triple $(\mathcal{M}, \mathcal{D}, M)$, where $\mathcal{M}$ is an image model (A, π, F), $\mathcal{D}$ is a database state $(O, C, DP, FP, \rightarrow, \Rightarrow)$, and M is the *image mapping,* that is a partial function:

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

such that for all π_i, π_j in $dom(M)$, $\pi_i \neq \pi_j$ implies $\pi_i \cap \pi_j = \emptyset$.

This constraint captures the intuitive fact that a spot in an image can be assigned only to one object in the image's content reconstruction. This is just another way of saying that two entities cannot be recognized in the same color spot. At the same time, 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.

Image mapping functions are introduced in [21], where first-order languages are proposed to represent map images and their content representations, scenes. In this context, an image mapping function associates a map object to a scene object, and is subject to a severe constraint: every map object is mapped into (exactly) one scene object. Given the more general nature of our model, we dispense with the totality of image mapping functions.

The usefulness of the image mapping will be fully appreciated upon querying an image base. For the time being, we wish to point out that an image mapping permits a two-ways association between the form and content of an image. From form to content, it associates a set of spots to an object of the content description, thus revealing the conceptual view of the objects that appears in an image. From the other way around, it serves to locate an abstract object in an image, thus revealing an object's visual appearance.

An image base is a set of images, with an additional database state, called *background* and denoted $\sigma_0 = (O_0, C_0, DP_0, FP_0, \rightarrow_0, \Rightarrow_0)$, containing information that is relevant to the application but cannot be related to any specific image. However, simply assembling these ingredients will not produce an image base, which we conceive as a *conceptually coherent body of information.* In order to capture this intuitive requirement, we introduce the notion of coherence of database states. A non-empty set of database states:

$$\{(O_1, C_1, DP_1, FP_1, \rightarrow_1, \Rightarrow_1), \ldots, (O_n, C_n, DP_n, FP_n, \rightarrow_n, \Rightarrow_n)\}$$

is said to be *coherent* with a background σ_0 if and only if:

$((O_i \cup O_0), C^u, DP^u, (FP_i \cup FP_0), (\rightarrow_i \cup \rightarrow_0), \Rightarrow^u)$ is a database state, for all $1 \leq i \leq n$,

where:

- $C^u = \bigcup_{i=0,n} C_i$
- $DP^u = \bigcup_{i=0,n} DP_i$
- $\Rightarrow^u = \bigcup_{i=0,n} \Rightarrow_i$,

Coherence rules out the following situations:

- two inconsistent IsA links, for instance for some $0 \leq i, j \leq n$, $(c \Rightarrow d) \in \Rightarrow_i$ and $(d \Rightarrow c) \in \Rightarrow_j$; both these links would be in $\Rightarrow^u$ against constraint 5;

- two inconsistent definitional properties, i.e. for some $0 \leq i, j \leq n$, $(c\ i\ d_1) \in DP_i$, $(c\ i\ d_2) \in DP_j$ and $d_1 \neq d_2$; both these properties would end up in DP^u, which would then violate constraint 3;

- two inconsistent single-valued factual properties, i.e. for some $0 \leq i \leq n$, $(o\ i\ o_1) \in SFP_0$ and $(o\ i\ o_2) \in SFP_i$; both these properties would end up in $(FP_i \cup FP_0)$ which would then violate constraint 3.

It should be noted that the above constraint prescribes a much weaker consistency for factual parts than for definitional parts, as it does not prevent two factual parts to have inconsistent factual properties. The rationale of this choice is that the contents of two images are prone to be inconsistent at the factual level whenever they share an entity which does not look the same in both images. For example, the factual part of an image showing Francesco as a programmer, would have the following factual property: `(francesco Job programmer)` whereas that of an image showing Francesco a few years later may have: `(francesco Job manager)`. On the contrary, the kind of factual information one is expected to have in the background is less contingent, like for example: `(francesco Sister giulia)`.

For convenience, the conceptual and background information of an image base is collected into the *content schema,* that is the 6-tuple $(O_0, C^u, DP^u, FP_0, \rightarrow_0, \Rightarrow^u)$. Notice that, as a consequence of the above constraint, the content schema is a database schema.

Given a database state σ_0 and a non-empty set Σ of images whose content reconstructions are coherent with σ_0, the *image base* associated to σ_0 and Σ is the pair (σ, Σ), where σ is the content schema of the images in Σ and σ_0.

The structure of an image base is pictorially presented in Figure 1. At the lowest level, image models abstractly represent the images of the image base, each one associated to its content reconstruction and possibly linked to it via the mapping function. The content parts are connected to the concept schema by the InstanceOf links of their objects.

5 An image query language

The basic tool to exploit the content of an information system is a *query facility,* that is a language for posing questions to the system and a function establishing the answer to each question. There are two basic requirements that a query facility for an image base must satisfy. The first one is of a general nature: the language's expressions must be interpretable as questions on which the proper answer function can be defined. The notion that best captures the intuitive relationship between a question and an answer for it is that of logical implication, or its model theoretic *alter ego,* satisfaction. Our image query facility will therefore be a logic, and in particular a many-sorted first-order logic, for reasons of adequacy that, if not already, will be clear in a moment.

The second requirement is that the query facility must allow the extraction of all the information contained in the system. The information objects of interest to us are images, as defined along

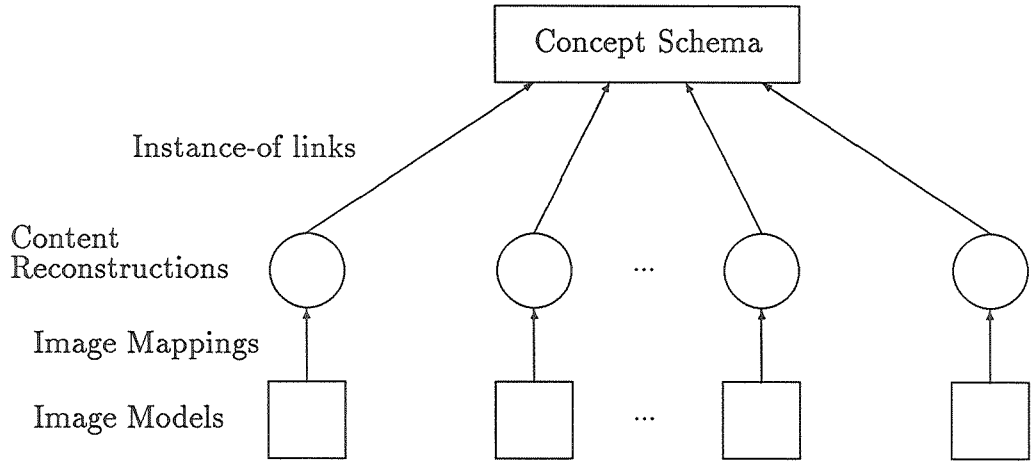

Figure 1: An Image Base.

the three dimensions of form, content, and mapping. Consequently, we want our query language to be able to address all these dimensions and to maximally exploit each of them.

In the rest of this section we will introduce the basic predicate symbols of an image query language, $\mathcal{L}_I$ satisfying these requirements. The semantics of each predicate symbol in $\mathcal{L}_I$ will be given in terms of the images that satisfy the symbol's ground atomic instances, so laying the bases for the definition of the answer function of an image query facility for our model.

5.1 Visual predicate symbol

In order to represent visual queries, i.e. queries that have the form of images, two sorts are introduced:

1. σ_r, the sort of *regions;* the symbols from this sorts are:
   - countably many constant symbols, denoted by capital letters drawn from the beginning of the alphabet (metasymbols r, r_1, r_2);
   - countably many variables;
   - the existential quantifier $\exists_r$.
2. σ_c, the sort of *colors;* the symbols from this sorts are:
   - countably many constant symbols, denoted by their English names (metasymbol l);
   - countably many variables;
   - the existential quantifier $\exists_c$.

As usual, variables will be small letters from the end of the alphabet with no additional sort information, deducible from the associated quantifiers.

The predicate symbol for querying image forms is 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,blue)* means that region A is *blue*. In order to make this precise, a formal semantics is now given.

A *denotation function* is 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.

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 $(\mathcal{M}, \mathcal{D}, M)$. Notice that the function d is the same for all the images structures, it does not change from structure to structure.

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

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

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

When used as a query facility, this satisfaction relation would return, in response to the sentence $I(r, l)$, the images which visually include the sentence, thus yielding an *exact* visual query facility. 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 [2]). This problem arises in a number of areas in computer vision, pattern recognition, medical image analysis and others.

In the mathematical formulation of the image registration problem, an image is seen as a 2-dimensional discrete function $H(x, y)$ associating a density (or other color measurement) to the pairs of a fixed space; the sought mapping between two images is expressed as:

$$H_1(x, y) = g(H_2(f(x, y)))$$

where f is a 2-dimensional spatial-coordinate transformation and g is a 1-dimensional intensity transformation.

There is a large variety of techniques which have been proposed to solve the registration problem, each one with its own pros, cons and range of applicability; yet none of these methods 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. Given the logical formulation of the model, this can be done in a clean mathematical way, as follows.

An *image predicate* is a total function receiving in input an image model, a region, and a color and returning an element of the set $\{0,1\}$. Given $n \geq 0$ image predicates $\Phi_1, \ldots, \Phi_n$, the *partial satisfaction* relations on the ground atomic instances of I is defined as follows:

$$\begin{cases} (\mathcal{I},d) \models_0 I(r,l) & \text{iff } F(d(r)) = d(l) \\ (\mathcal{I},d) \models_i I(r,l) & \text{iff } \Phi_i(\mathcal{M},d(r),d(l)) = 1, \text{ for all } 1 \leq i \leq n \end{cases}$$

where $\mathcal{I} = (\mathcal{M},\mathcal{D},M)$. The relation $\models_0$ captures the exact query mechanism by requiring the equality of the query with (a portion of) the image. All 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 model matches a given spot.

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

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

5.2 Spatial predicate symbols

As discussed in detail in section 7, 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 $\mathcal{L}_I$.

As pictures are 2-dimensional objects, 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 [1] 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 2 illustrates 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 2, 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 2.

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

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 2: Interval Relationships.

and Π_y, which return, respectively, the x and y projection of a given region; and two interval function, β and ϵ, which return, respectively, the begin and the end point of the given interval. 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 the predicate symbols on the horizontal dimension are considered):

2. $(\mathcal{I},d) \models X\_before(r_1, r_2)$ iff $\epsilon(\Pi_x(d(r_1))) < \beta(\Pi_x(d(r_2)))$

3. $(\mathcal{I},d) \models X\_equal(r_1, r_2)$ iff $\beta(\Pi_x(d(r_1))) = \beta(\Pi_x(d(r_2)))$ and $\epsilon(\Pi_x(d(r_1))) = \epsilon(\Pi_x(d(r_2)))$

4. $(\mathcal{I},d) \models X\_meets(r_1, r_2)$ iff $\epsilon(\Pi_x(d(r_1))) = \beta(\Pi_x(d(r_2)))$

5. $(\mathcal{I},d) \models X\_overlaps(r_1, r_2)$ iff $\beta(\Pi_x(d(r_1))) < \beta(\Pi_x(d(r_2)))$ and $\epsilon(\Pi_x(d(r_1))) < \epsilon(\Pi_x(d(r_2)))$

6. $(\mathcal{I},d) \models X\_during(r_1, r_2)$ iff $\beta(\Pi_x(d(r_1))) > \beta(\Pi_x(d(r_2)))$ and $\epsilon(\Pi_x(d(r_1))) < \epsilon(\Pi_x(d(r_2)))$

7. $(\mathcal{I},d) \models X\_starts(r_1, r_2)$ iff $\beta(\Pi_x(d(r_1))) = \beta(\Pi_x(d(r_2)))$ and $\epsilon(\Pi_x(d(r_1))) < \epsilon(\Pi_x(d(r_2)))$

8. $(\mathcal{I},d) \models X\_finishes(r_1, r_2)$ iff $\beta(\Pi_x(d(r_1))) > \beta(\Pi_x(d(r_2)))$ and $\epsilon(\Pi_x(d(r_1))) = \epsilon(\Pi_x(d(r_2)))$.

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

5.3 Content predicate symbols

In order to express queries on the contents of images, three more sorts are introduced in $\mathcal{L}_I$:

- σ_o, the sort of *objects;* the constant symbols from this sort will be denoted by lower case names, such as: *francesco* and *giulia* (metasymbols o, o_1, o_2); the existential quantifier for objects will be $\exists_o$;
- 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 $\bot$ (metasymbols c, c_1, c_2, d_1, d_2); the class existential quantifier will be $\exists_p$;
- 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* (metasymbol i); the identifier existential quantifier will be $\exists_i$.

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).

A formal semantics for content queries is now introduced, based on database states. To this end, the image denotation function d is extended as follows. Given a database state, d maps one-to-one the constant symbols of the sorts 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.

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

9. $(\mathcal{I},d) \models MDP(c_1, i, c_2)$ iff $(c_1, i, c_2) \in MDP$
10. $(\mathcal{I},d) \models SDP(c_1, i, c_2)$ iff $(c_1, i, c_2) \in SDP$
11. $(\mathcal{I},d) \models MFP(o_1, i, o_2)$ iff $(o_1, i, o_2) \in MFP$
12. $(\mathcal{I},d) \models SFP(o_1, i, o_2)$ iff $(o_1, i, o_2) \in SFP$
13. $(\mathcal{I},d) \models InstanceOf(o, c)$ iff $(o, c) \in \rightarrow$
14. $(\mathcal{I},d) \models IsA(c_1, c_2)$ 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 database state is a strict analog of the reality being modelled, thus a closed world in which a sentence holds true (i.e. the database state satisfies the sentence) or false.

Table 1: The predicate symbols of $\mathcal{L}_I$.

Visual predicate symbol		Spatial predicate symbols	
$I(\sigma_r, \sigma_c)$		$X\_before(\sigma_r, \sigma_r)$	$Y\_before(\sigma_r, \sigma_r)$
Content predicate symbols		$X\_equal(\sigma_r, \sigma_r)$	$Y\_equal(\sigma_r, \sigma_r)$
$MDP(\sigma_p, \sigma_i, \sigma_p)$	$SDP(\sigma_p, \sigma_i, \sigma_p)$	$X\_meets(\sigma_r, \sigma_r)$	$Y\_meets(\sigma_r, \sigma_r)$
$MFP(\sigma_o, \sigma_i, \sigma_o)$	$SFP(\sigma_o, \sigma_i, \sigma_o)$	$X\_overlaps(\sigma_r, \sigma_r)$	$Y\_overlaps(\sigma_r, \sigma_r)$
$InstanceOf(\sigma_o, \sigma_p)$	$IsA(\sigma_p, \sigma_p)$	$X\_during(\sigma_r, \sigma_r)$	$Y\_during(\sigma_r, \sigma_r)$
Mapping predicate symbol		$X\_starts(\sigma_r, \sigma_r)$	$Y\_starts(\sigma_r, \sigma_r)$
$Map(\sigma_r, \sigma_o)$		$X\_finishes(\sigma_r, \sigma_r)$	$Y\_finishes(\sigma_r, \sigma_r)$

5.4 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 structure $\mathcal{I}$,

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

6 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.

6.1 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, summarized with their sorts in Table 6. 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.

16. $(\mathcal{I},d) \models \neg\alpha$ iff $(\mathcal{I},d) \not\models \alpha$

17. $(\mathcal{I},d) \models (\alpha \vee \beta)$ iff either $(\mathcal{I},d) \models \alpha$ or $(\mathcal{I},d) \models \beta$

18. $(\mathcal{I},d) \models (\exists_r x)\gamma$ iff for some constant symbol c of sort σ_r, $d(c) \in \pi$ and $(\mathcal{I},d) \models \gamma_c^x$

19. $(\mathcal{I},d) \models (\exists_r x)\delta$ iff for some constant symbol c of sort σ_r, $(\mathcal{I},d) \models \delta_c^x$

20. $(\mathcal{I},d) \models (\exists_c x)\alpha$ iff for some constant symbol c of sort σ_c, $(\mathcal{I},d) \models \alpha_c^x$

21. $(\mathcal{I},d) \models (\exists_o x)\alpha$ iff for some constant symbol c of sort σ_o, $(\mathcal{I},d) \models \alpha_c^x$

22. $(\mathcal{I},d) \models (\exists_p x)\alpha$ iff for some constant symbol c of sort σ_p, $(\mathcal{I},d) \models \alpha_c^x$

23. $(\mathcal{I},d) \models (\exists_i x)\alpha$ 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 18 the membership of $d(c)$ in π is required to ensure that when the variable x denotes a region of an image model, the quantifier $\exists_r$ ranges on the spots of that image model 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 models are finite objects, whereas the latter would be satisfied by no images if the end-point of the x-projection of r is 0 (i.e. 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 model; the semantics requires only that r be included in the regions mapped by M onto o.

Given an image base (σ, Σ) and an image $\mathcal{I} = (\mathcal{M},\mathcal{D},M)$ in Σ, with $\mathcal{D} = (O_i, C_i, DP_i, FP_i, \rightarrow_i, \Rightarrow_i)$, the *extension* of $\mathcal{I}$ is the image $\mathcal{I}^e = (\mathcal{M},\mathcal{D}^e, M)$ where $\mathcal{D}^e = ((O_i \cup O_0), (C_i \cup C_0), (DP_i \cup$

$DP_0), (FP_i \cup FP_0), (\rightarrow_i \cup \rightarrow_0), (\Rightarrow_i \cup \Rightarrow_0))$. In essence, the extension enriches an image content component with the background information. Given an image base IB and an image query q, the *answer of q in IB, a(q,IB),* is given by:

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

Notice the usage of the background information in the retrieval of images.

This query facility subscribes to the logical view of information retrieval recommended in [26], as it models retrieval as logical inference. At the same time, it conforms to the model-theoretic approach to databases presented in [20], 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. First of all, 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.

6.2 Visual queries

A visual query is a query in which only the predicate symbol I occurs. A query of this kind is typically expressed through an appropriate visual tool, but from our analysis' point of view it is an expression of the query language, like a query of any other kind.

The answer of an atomic ground visual query, such as $I(A, blue)$ consists of the images whose image models have the region named A of color *blue*. How exact this match is, depends on the non-exact matching functions included in the model; in order to make the following discussion more concrete, we will ignore this aspect. Assuming that the visual querying tool provides the machinery to deal with first-order syntax, logical connectives and quantifiers can be used to articulate more complex 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,* while:

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

returns 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 model. For instance, let us consider an image model $\mathcal{M}$ 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: A, B, and C could be names of stars, *white, blue,* and *green* could be natural numbers and I could name the distance of a star from the Earth and the satisfaction relations mentioned above would hold anyway. 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. As a further verification of this fact, we can observe that the "trivial" cases of the logical satisfaction relation hold in our query facility:

- an image model $\mathcal{M}$ satisfies the simplest sentence that represents it, given by $\bigwedge_i I(A_i, c_i)$ where A_i are names for the regions of $\mathcal{M}$ and for all i $F(A_i) = c_i$; this ensures that the membership of an image in an image base can be tested via the query facility;
- a contradictory sentence, such as $(I(A, green) \wedge \neg I(A, green))$ is satisfied by no image model, thus it would return no images when posed as a query; this ensures that ill-defined queries are harmless;
- a tautological sentence, such as $(I(A, blue) \rightarrow I(A, blue))$ is satisfied by any image model with the same underlying partition.

The interested reader may find more about the logic underlying visual queries in [16].

6.3 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 point 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 models 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 mixed, as they involve more than one dimension of the image representation, and will be discussed later.

6.4 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 notion of aboutness is central to content-based retrieval and has been formalized in many different ways by different categories of information retrieval models. The query facility of our model returns, in response to a query, the images in which a query is true, thus interpreting aboutness as truth. In so doing the model subscribes to a classical view, as assessed, for instance, in [6].

The expressiveness of the content query language is similar to that of classical object-oriented query languages, allowing to state conditions on objects, property values, and InstanceOf links in a simple and uniform way. In addition, our language permits to express queries whose variables range on property or class names.

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

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

which returns the images in whose content part some object x has value ``Francesco'' for some property y, which is presumably a kind of 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:

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

There is a basic difference between our query facility and that of databases. The queries of a database typically return objects, whereas our query facility is designed to return images, i.e. documents including, among other things, whole database states. Technically, this difference is due to the logical nature of queries. Those offered by a retrieval model are sentences, whereas database queries are open formulas whose free variables give the structure of the query answer. For instance, the images whose content reconstructions contain instances of the *PERSON* class who have *giulia* as sister can be retrieved by posing the query:

$$(\exists_o x)(InstanceOf(x, PERSON) \wedge MFP(x, Sister, giulia)).$$

By removing the quantifier from this sentence, we obtain a typical database query asking for the brothers of Giulia. For this reason, our model is closer to an information retrieval model [25] than to a traditional data model.

6.5 Mapping queries

Pure mapping queries contain 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 model and that the mapping function "points it out".

A form of shape-based retrieval can be performed via mapping queries which quantify on objects while specifying regions; 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.

6.6 Mixed queries

Mixed queries exhibit instances of predicate symbols of at least two of the kinds shown in Table 6. 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 1-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 picture representing 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 model 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.

In queries of type 3, content and mapping predicate symbols can share object terms, so they allow to request images whose regions satisfy certain conceptual properties. For instance, the query:

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

can be used to retrieve 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": the first conjunct permits to constraint the shape of the object, while the second one constraints its class membership. By quantifying on the region, we obtain the query:

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

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.

When the mapping and the content components of this kind of queries share object constant symbols, the content components are not used to identify objects on the basis of their relationships, but to express stricter conditions on the specified objects. For instance, the query:

$$(Map(r, o) \wedge InstanceOf(o, SALESMAN))$$

retrieves the images in which o has a shape included within the contours of r, and is known to be a salesman.

Queries of type 11 are queries of type 3 with an additional visual component. When tighted to the other components of the query by a region term, such visual component serves to specify that the mapping applies to *single* regions, and possibly to state the colors of those regions. To exemplify, 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"). The color specification can be omitted by using an appropriate color variable in place of the symbol *yellow*. In this case, the query would return the images whose models have one region mapped onto a tree. The use of a region constant symbol r in place of the variable x would constraint the shape of the tree, leaving to the content component the possibility of specifying conceptual properties to be satisfied by that tree, such as its location.

Queries of type 9 (visual and mapping) are a restricted form of queries of type 11, permitting to retrieve images on content objects with a color specification. The request to retrieve images in which "Giulia is wearing something pink" can be formulated as the following query of type 9:

$$(\exists_r x)(I(x,pink) \wedge Map(x,giulia)).$$

Queries of type 5 (spatial and mapping) can be used to state spatial conditions between content objects or regions and other content objects. As an example, the query:

$$(\forall_r x)(Map(x,francesco) \rightarrow (X\_before(x,r) \wedge Y\_during(x,r)))$$

can be used to retrieve the images in which all *francesco*'s regions (that is, all the regions in which he is mapped onto) precede r on the x axis and are properly included in r on the y axis. The following query retrieves the images in which *giulia* is left to *francesco:*

$$\begin{aligned}(\forall_r x)(Map(x,giulia) \rightarrow ((\forall_r y)Map(y,francesco) \rightarrow X\_before(x,y)) \wedge \\ ((\exists_r y)Map(y,francesco) \wedge Y\_includes(x,y)))\end{aligned}$$

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.

Analogously to the previous case, the addition of a visual component to this kind of queries (so obtaining queries of type 13) restricts region terms to denote single image regions, on which a color condition can possibly be stated.

Queries of type 7 include all kinds of predicate symbols except the visual one. They extend queries of type 5 by allowing a richer articulation of content conditions, in which not only content objects can be addressed, but also conceptual relationships involving these objects. For this reason, queries of this kind 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 [5] "find all pictures having a tree to the left of a house", expressible in our query language as:

$$\begin{aligned}(\exists_o xy)(InstanceOf(x,TREE) \wedge InstanceOf(y,HOUSE) \wedge \\ (\forall_r u)(Map(u,x) \rightarrow ((\forall_r v)Map(v,y) \rightarrow X\_before(u,v)) \wedge \\ ((\exists_r w)Map(w,y) \wedge Y\_includes(u,w)))).\end{aligned}$$

The content component gives the possibility of specifying, among other things, the class which the content objects mentioned in the query belongs to, thus permitting retrieval on generic objects. A comparison of our retrieval capability with that of iconic databases is carried out in section 7.2.

Queries of type 12 include visual and spatial queries which share region terms. If the shared term is a constant symbol, then the spatial properties of that region do not depend on any particular image, and the resulting query is equivalent to its visual (sub-)query. When the shared term are region variables, the visual component of the query, as already pointed out, serves to have those variables ranging on single image regions with an additional color specification, while the spatial component states conditions on them. For instance the query:

$$(\exists_r x)(I(x,green) \wedge X\_before(x,r))$$

retrieves the images having a green spot at the left of region r.

Queries of type 14 are queries of type 12 with an additional content component. As this content component cannot share any term with the visual and spatial components, this type of queries are in fact Boolean combinations of queries of type 12 and queries of type 2. For instance:

$$(\exists_r x)(I(x,green) \wedge X\_before(x,r) \wedge (\exists_o y)InstanceOf(y,APPLE))$$

return the subset of the images returned by the previous query which are about an apple.

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"
"A house with green windows left to a tree".

7 Relation to other work

At this point we are in the position of relating the introduced model with other proposals sharing, to a reasonable extent, the same goal as ours. Considering that the number of such proposals is too large to permit a comprehensive discussion, we will have a two-stepped approach. First, in the next section, we will take a broad perspective, and look at some well known image models, coming from a fairly wide range of different fields. Then, in the successive two sections, we will narrow down the scope of the discussion to image databases, examining two important classes: iconic and pictorial databases.

7.1 Image models

The assumption that images can be regarded as sentences in a natural language goes back to at least 30 years ago, and has been the basis of the activity of researchers investigating the automatic generation of images (for a survey, see [22]). 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, image models are investigated in order to understand and reproduce human problem solving methods which are deemed to be based on image inspection and manipulation. These operations can be performed on scene reproductions as in a vision system, or on mental reconstructions as in imagery [7]. In fact, it is argued that imagery and vision have parallel purposes and essentially differ in the image source (human memory in the former case, the external world in the latter) [11].

Mental imagery finds its theoretical foundations in psychology and cognitive science, where the primary role of images in certain human inference processes (such as reasoning [15] and discourse understanding [10]) has been postulated and empirically tested.

The role of images in our mental processes is still controversial, and not all the defenders of this role have been precise about image representation (this is not necessarily a limit of their proposals). Those who have gone as far as proposing an image representation scheme have resorted to various formalisms, ranging from generative array grammars [22] to array theories [7], depending on the purpose of the formalization and the formalizer's style. Despite the fact that none (with the exception of [21], but only for content and mapping), to the best of our knowledge, has used a mathematical logic as an image model, we can observe that the underlying ontology of our logic, consisting of spots and colors, is consistent with that of these other image models. The logical form of our model is due, as argued in the introduction, to the view of images as information bearers that underlies any image information system. However, we believe that the formal development we have presented can be used to capture other image inferences, beside those needed for retrieval, and thus it is relevant to any attempt aiming at capturing aspects of reasoning on images.

7.2 Iconic databases

An *iconic database* contains abstractions of images, consisting in two dimensional arrays homologous to the abstracted images, showing the objects of interest to the application in symbolic (or iconic) form. Such abstractions are half-way between the form and the content level of our model, since they are about conceptual entities (i.e. objects, as opposed to regions and colors) but at the same time preserve the spatial relationships among these objects as they appear in the image. Such representations have been called *iconic images*. Figure 3 shows iconic images containing four the objects a, b, c, and d.

Iconic images support the retrieval based on the spatial relationships of the entities shown in the corresponding, non-abstracted images, as they abstract everything from these images but the spatial relationships between objects. A typical query to an iconic database is "find all pictures having a tree to the left of a house" [5], and is expressed by means of an iconic image. For instance, the iconic image shown in Figure 3.b can be a query on the image presented in Figure 3.a.

This form of image retrieval has received considerable attention in the last years. However, research in this field has mostly been focused on efficiency, disregarding, until lately, the logical level of the proposed retrieval models, that is what spatial relationships were really represented and used

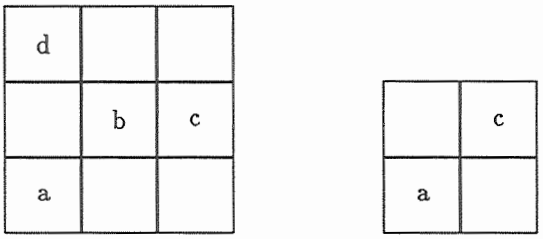

Figure 3: An iconic image and an iconic query.

for image retrieval. The reader interested in this aspect has to decode it from substring matching operations, carried out on a special kind of strings, named 2-D strings, and on several variations thereof. Putting the stress on implementation matters has delayed the full understanding of the logical nature of the retrieval realized by these methods.

In the seminal paper [5], 2-D strings are introduced as a way of encoding 5 spatial relationships between pairs of rectangles, the minimum bounding rectangles of the objects appearing in an iconic image. These relationships are: "left," "right," "below," "above," and "same location," and are represented by using 4 symbols. Three matching functions between 2-D strings are defined which capture, to different extents, the semantics of the 5 spatial relationships represented by the model. These string matching functions are extended in [14], in order to capture a "similarity" criterion between iconic images. The criterion is that of maximum likelyhood of two 2-D strings, and, being defined on the implementational data structure rather than on the semantics of the representation, it is difficult to say what it amounts to in terms of spatial reasoning.

In [4], a rule-based database system is presented which provides the same 5 primitive spatial relations as [5], but allows to combine them in a Boolean way, so obtaining a system reasoning on 9 relationships ("north", "south", "west", "east", "north-west", "north-east", "south-west", "south-east", and "same location"). This combinational power permits also to define other interesting spatial relationships, such as "surrounded-by", "partly-surrounded-by", and "near". A similar representation is proposed in [3], where the 9 spatial relationships above are assumed as primitives and 2-D strings are abandoned in favour of less cryptic triples (O_1, O_2, r), where the O_i are objects and r is an integer representing a spatial relationship above. Retrieval on the basis of these relationships is performed on a special data structure, called picture indexing table, where the triples, one for each pair of objects, are hashed.

The bottom line of this approach is reached in [12], where all the 13 possible spatial relationships between two intervals in a one-dimensional space are identified and captured by 7 symbols. This result had been already obtained some time before in [1], which proposes one-dimensional intervals to model time, and provides a logical formalization. Incidentally, this formalization gives a much clearer account of the reasoning on spatial relationships underlying iconic image retrieval than that found in [13], which purports at capturing formal aspects of spatial reasoning but is couched

in terms of 2D C-strings.

Our model includes the representational primitives proposed in [1] and in [12], thus permitting the specification of any spatial relationship between rectangles upon querying the image base. Furthermore, as discussed in section 6.6, iconic queries are a proper subsets of queries of type 7. Evidently, the specification of one such queries is much more elaborated than that of an iconic query, as it can be verified by comparing Figure 3.b with the corresponding query. This is due to the fact that the scope of iconic retrieval is much more limited than the general purpose image retrieval offered by our model. However, once a general model is defined which establishes the theoretical bases of a class of information systems, the simplicity of more limited and informal models can be re-gained by acquiring these models as parts of the general model. In our case, iconic queries can be included *tout cour* in our query facility, with all their graphical and computational machinery, either as queries on their own right, or as sub-queries of more articulated queries. The evaluation of an iconic query, then, can be performed via 2D strings, leaving to the system the task of combining its result with that of the rest of the query, if any. As an alternative, a program can be devised that translates an iconic query into an equivalent formula of our query language, to be evaluated in the standard way.

7.3 Pictorial databases

A *pictorial database* is a database whose built-in data types include images and image components. The basic functionality offered by a pictorial database is thus the obvious extension of that provided by traditional databases: manipulation and retrieval of data objects including images. Not surprisingly, then, the data models of pictorial databases have emerged in an evolutionary way as extensions of successful traditional data models.

For instance, the PSQL query language [23] enhances SQL by allowing user-defined abstract data types for defining pictorial domains on top of 3 primitives domains offered by the model: points, line segments, and regions. These built-in domains come equipped with a number of operators, capturing spatial and topological relations between their members. PSQL queries are SQL queries with an additional clause for introducing the images which the query is on. Conditions on images (or *superimpositions*) can thus be specified, in terms of the operators provided by the model.

Similarly, the PICQUERY query language [24] extends the query-by-example approach to deal with image manipulation and retrieval, yielding a new tabular query language for accessing a pictorial DBMS, PICDMS, in a generic, application independent way. PICQUERY provides a large variety of operations, ranging from image manipulation (such as shifting, rotation, zooming, superimposing) to pattern recognition (edge detection, contour drawing, similarity retrieval), spatial and geometric computing (on points, segments, and regions).

We have already remarked in section 5.3 one important difference between our model and a data model: while our model is essentially an information retrieval model, concerned with the

retrieval of image objects, carrying form, content and mapping of an image, a data model allows to retrieve elements of a data structure, like objects or property values, rather than whole content representations. But there is a more fundamental difference between a data model and an information retrieval model: the former is the result of an abstraction, usually called conceptual modelling, performed on the specific requirements of the application, the latter purports at supporting generic content-based retrieval on "raw" documents, images in our case. In other words, a database is defined having in mind the requests that the users will make to it, whereas very little knowledge, if any at all, is assumed on the information need of the users of a document base.

As a result, a pictorial data model typically copes with the efficient, physically independent implementation of pictorial domains, of the manipulation functions defined on them, of their comparison operators and so on. 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. Consequently, the comparison between the retrieval capability of a pictorial data model and that of our model can be pursued only to a limited extent.

Work much in the spirit of our model is presented in [8], which introduces image retrieval based on a class of visual queries, namely hand-drawn sketches. The choice of this particular kind of queries is driven by the nature of the considered image base, which is a collection of paintings; but the claim that *any* image retrieval system should provide a facility for visual querying goes beyond the application at hand, and is supported by our model as well. Retrieval in [8] is realized on the basis of local correlation between the user drawings and pictorial indexes of images. This form of image matching can be imported in our model as an image predicate, yielding 1 when the local correlation exceeds a predefined threshold, and 0 otherwise.

8 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. However, we claim that these extensions are details that can be easily accommodated in the presented model, similarly to the extensions which led from the tuple calculus to SQL.

Besided its immediate applicability to image bases, our model provides a basic philosophy to deal with the representation and retrieval of (a class of) "intensional" objects, i.e. objects with a content. The more general class of multimedia objects is rapidly emerging on the information system scene, conquering a number of applications and creating new opportunities for the enterprise of information systems. Among the many demands of multimedia information systems, that for adequate data models is perhaps the most crucial. Our model makes a step towards a

general model for multimedia information systems, providing a paradigm and representational and querying primitives.

9 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.

References

[1] J. F. Allen. Mantaining knowledge about temporal intervals. *Communications of the ACM*, 26:832–843, 1983.

[2] Lisa G. Brown. A survey of image registration techniques. *ACM Computing Surveys*, 24(4):325–376, December 1992.

[3] C. C. Chang and S. Y. Lee. Retrieval of similar pictures on pictorial databases. *Pattern Recognition*, 24(7):675–680, 1991.

[4] S. K. Chang, C. W. Yan, D. C. Dimitroff, and T. Arndt. An intelligent image database system. *IEEE Transactions on Software Engineering*, 14(5):681–688, May 1988.

[5] 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.

[6] Donald Davidson. The logical form of action sentences. In *Essays on Actions and Events*, pages 105–148. Clarendon Press, Oxford, England, 1985.

[7] Janice I. Glasgow. The imagery debate revisited: a computational perspective. *Computational Intelligence*, 9(4):309–333, Nov. 1993.

[8] Kyoji Hirata and Toshikazu Kato. Query by visual example. In *Proceedings of the 3rd International Conference on Extending Database Technology EDBT'92*, pages 56–71, Vienna, 1992.

[9] Richard Hull and Roger King. Semantic database modeling: Survey, applications and research issues. *ACM Computing Surveys*, 19(3):201–259, 1987.

[10] Philip N. Johnson-Laird. *Mental Models.* Cambridge University Press, Cambridge, MA, 1983.

[11] Stephen M. Kosslyn. Seeing and imagining in the cerebral hemispheres: a computational approach. *Psychological Review*, 94(2):148–175, 1987.

[12] S. Y. Lee and Fang-Jung Hsu. 2D C-string: a new spatial knowledge representation for image database systems. *Pattern Recognition*, 23(10):1077–1088, 1990.

[13] S. Y. Lee and Fang-Jung Hsu. Picture algebra for spatial reasoning of iconic images represented in 2D C-string. *Pattern Recognition Letters*, 12:425–435, 1991.

[14] Suh-Yin Lee, Man-Kwan Shan, and Wei-Pang Yang. Similarity retrieval of iconic image database. *Pattern Recognition*, 22(6):675–682, 1989.

[15] Robert K. Lindsay. Images and inference. *Cognition*, 29:229–250, 1988.

[16] Carlo Meghini. Logical image modelling and retrieval. Technical Report B4-05, Consiglio Nazionale delle Ricerche, Istituto di Elaborazione della Informazione, April 1994.

[17] Carlo Meghini, Fausto Rabitti, and Costantino Thanos. Conceptual modelling of multimedia documents. *IEEE Computer. Special Issue on Multimedia Systems*, 24(10):23–30, Oct. 1991.

[18] 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.

[19] 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.

[20] Raymond Reiter. Towards a logical reconstruction of relational database theory. In Michael L. Brodie, John Mylopoulos, and Joachim W. Schmidt, editors, *On Conceptual Modelling*, pages 191–233. Springer Verlag, New York, NY, 1984.

[21] Raymond Reiter and Alan K. Mackworth. A logical framework for depiction and image interpretation. *Artificial Intelligence*, 41:125–155, 1989.

[22] Azriel Rosenfeld and Rani Siromoney. Picture languages–a survey. *Languages of design*, 1(3):229–245, Aug. 1993.

[23] Nick Rossopoulos, Christos Faloutsos, and Timos Sellis. An efficient pictorial database system for PSQL. *IEEE Transactions on Software Engineering*, 14(5):639–650, May 1988.

[24] Joseph Thomas and Alfonso F. Cardenas. PICQUERY: a high level query language for pictorial database management. *IEEE Transactions on Software Engineering*, 14(5):630–638, May 1988.

[25] Corneliis J. van Rijsbergen. *Information Retrieval.* Butterworths, London, GB, second edition, 1979.

[26] 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.

[27] Corneliis J. van Rijsbergen. A non-classical logic for information retrieval. *The Computer Journal*, 29:481–485, 1986.

A Summary of the model

A.1 Image models

Given a set of colors L, an *image model* is a triple (A, π, F) where:

- A is a connected set of pairs of natural numbers;
- π is a partition of A consisting of connected sets;
- $F : \pi \rightarrow L$, such that for all contiguous sets S_i and S_j in pi, $F(S_i) \neq F(S_j)$.

A.2 Database states

A *database state* 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.

such that:

1. for every o in O there exists a c in C such that $(o \rightarrow c)$
2. $\pi_{1,2}(SDP) \cap \pi_{1,2}(MDP) = \emptyset$
3. $(c\ i\ d_1), (c\ i\ d_2) \in (SDP \cup MDP)$ implies $d_1 = d_2$
4. $(o\ i\ o_1), (o\ i\ o_2) \in SFP$ implies $o_1 = o_2$
5. $(o_1\ i\ o_2) \in SFP$ implies $(c_1\ i\ c_2) \in SDP$ and $(o_i \rightarrow c_i)$ for $i = 1, 2$.
6. $(o_1\ i\ o_2) \in MFP$ implies $(c_1\ i\ c_2) \in MDP$ and $(o_i \rightarrow c_i)$ for $i = 1, 2$
7. $(o_1\ i\ o_2) \in SFP$, $(o_1 \rightarrow c_1)$ and $(c_1\ i\ c_2) \in SDP$ imply $(o_2 \rightarrow c_2)$
8. $(o_1\ i\ o_2) \in MFP$, $(o_1 \rightarrow c_1)$ and $(c_1\ i\ c_2) \in MDP$ imply $(o_2 \rightarrow c_2)$

9. $\Rightarrow$ is a partial order with a minimum $\perp$ and a maximum $\top$

10. $(c_1\ i\ d_1) \in SDP$ and $(c_2 \Rightarrow c_1)$ imply $(c_2\ i\ d_2) \in SDP$ and $(d_2 \Rightarrow d_1)$

11. $(c_1\ i\ d_1) \in MDP$ and $(c_2 \Rightarrow c_1)$ imply $(c_2\ i\ d_2) \in MDP$ and $(d_2 \Rightarrow d_1)$.

12. $(o \rightarrow c_1)$ and $(c_1 \Rightarrow c_2)$ imply $(o \rightarrow c_2)$

Definitional and factual properties are collected into the sets:

$$DP = SDP \cup MDP$$
$$FP = SFP \cup MFP$$

A.3 Images

An *image* is a triple $(\mathcal{M}, \mathcal{D}, M)$, where $\mathcal{M}$ is an image model (A, π, F), $\mathcal{D}$ is a database state $(O, C, DP, FP, \rightarrow, \Rightarrow)$, and M is the *image mapping*, that is a partial function:

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

such that for all π_i, π_j in $dom(M)$, $\pi_i \neq \pi_j$ implies $\pi_i \cap \pi_j = \emptyset$.

A.4 Image bases

A non-empty set of database states:

$$\{(O_1, C_1, DP_1, FP_1, \rightarrow_1, \Rightarrow_1), \ldots, (O_n, C_n, DP_n, FP_n, \rightarrow_n, \Rightarrow_n)\}$$

is said to be coherent with the state $\sigma_0 = (O_0, C_0, DP_0, FP_0, \rightarrow_0, \Rightarrow_0)$, if and only if:

$((O_i \cup O_0), C^u, DP^u, (FP_i \cup FP_0), (\rightarrow_i \cup \rightarrow_0), \Rightarrow^u)$ is a database state, for all $1 \leq i \leq n$,

where:

- $C^u = \bigcup_{i=0,n} C_i$
- $DP^u = \bigcup_{i=0,n} DP_i$
- $\Rightarrow^u = \bigcup_{i=0,n} \Rightarrow_i$,

The 6-tuple $(O_0, C^u, DP^u, FP_0, \rightarrow_0, \Rightarrow^u)$ is called content schema.
Given a database state σ_0 and a non-empty set Σ of images whose content reconstructions are coherent with σ_0, the *image base* associated to σ_0 and Σ is the pair (σ, Σ), where σ is the content schema of the images in Σ and σ_0.

A.5 The image query language $\mathcal{L}_I$

The image query language, $\mathcal{L}_I$, is a many-sorted first order language. The sorts of $\mathcal{L}_I$ are:

- σ_r, the sort of regions
- σ_c, the sort of colors
- σ_o, the sort of objects
- σ_p, the sort of classes
- σ_i, the sort of identifiers

The alphabet of $\mathcal{L}_I$ consists, for each sort, of countably many constant symbols, countably many variables and one existential quantifier; in addition, the predicate symbols of $\mathcal{L}_I$ are those shown in Table 6.
An image query is any sentence of $\mathcal{L}_I$.

A.6 The semantics of $\mathcal{L}_I$

The image denotation function d is a one-to-one function which maps:

- the constant symbols of sort σ_r onto the finite subsets of pairs of natural numbers
- the constant symbols of sort σ_c onto the set of colors L
- the constant symbols of the sorts σ_o, σ_p, and σ_i onto themselves.

An image structure is a pair $(\mathcal{I},d)$, where $\mathcal{I}$ is an image.
An *image predicate* is a total function receiving in input an image model, a region, and a color and returning an element of the set $\{0,1\}$. Given $n \geq 0$ image predicates $\Phi_1, \ldots, \Phi_n$ and an image $\mathcal{I} = (\mathcal{M},\mathcal{D},M)$,

$$\begin{cases} (\mathcal{I},d) \models_0 I(r,l) & \text{iff } F(d(r)) = d(l) \\ (\mathcal{I},d) \models_i I(r,l) & \text{iff } \Phi_i(\mathcal{M},d(r),d(l)) = 1, \text{ for all } 1 \leq i \leq n \end{cases}$$

Let $(\mathcal{I},d)$ be 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, δ a well-formed formula in which the variable x occurs only in instances of *Map*, and α_c^x the same formula as α except that all the occurrences of the variable x are replaced by occurrences of the constant symbol c.
An image structure $(\mathcal{I},d)$ is said to be satisfied by an image query q, $(\mathcal{I},d) \models q$, if and only if:

1. $(\mathcal{I},d) \models I(r,l)$ iff for some $0 \leq i \leq n$, $(\mathcal{I},d) \models_i I(r,l)$
2. $(\mathcal{I},d) \models X\_before(r_1,r_2)$ iff $\epsilon(\Pi_x(d(r_1))) < \beta(\Pi_x(d(r_2)))$
3. $(\mathcal{I},d) \models X\_equal(r_1,r_2)$ iff $\beta(\Pi_x(d(r_1))) = \beta(\Pi_x(d(r_2)))$ and $\epsilon(\Pi_x(d(r_1))) = \epsilon(\Pi_x(d(r_2)))$

4. $(\mathcal{I},d) \models X\_meets(r_1, r_2)$ iff $\epsilon(\Pi_x(d(r_1))) = \beta(\Pi_x(d(r_2)))$

5. $(\mathcal{I},d) \models X\_overlaps(r_1, r_2)$ iff $\beta(\Pi_x(d(r_1))) < \beta(\Pi_x(d(r_2)))$ and $\epsilon(\Pi_x(d(r_1))) < \epsilon(\Pi_x(d(r_2)))$

6. $(\mathcal{I},d) \models X\_during(r_1, r_2)$ iff $\beta(\Pi_x(d(r_1))) > \beta(\Pi_x(d(r_2)))$ and $\epsilon(\Pi_x(d(r_1))) < \epsilon(\Pi_x(d(r_2)))$

7. $(\mathcal{I},d) \models X\_starts(r_1, r_2)$ iff $\beta(\Pi_x(d(r_1))) = \beta(\Pi_x(d(r_2)))$ and $\epsilon(\Pi_x(d(r_1))) < \epsilon(\Pi_x(d(r_2)))$

8. $(\mathcal{I},d) \models X\_finishes(r_1, r_2)$ iff $\beta(\Pi_x(d(r_1))) > \beta(\Pi_x(d(r_2)))$ and $\epsilon(\Pi_x(d(r_1))) = \epsilon(\Pi_x(d(r_2)))$

9. $(\mathcal{I},d) \models Y\_before(r_1, r_2)$ iff $\epsilon(\Pi_y(d(r_1))) < \beta(\Pi_y(d(r_2)))$

10. $(\mathcal{I},d) \models Y\_equal(r_1, r_2)$ iff $\beta(\Pi_y(d(r_1))) = \beta(\Pi_y(d(r_2)))$ and $\epsilon(\Pi_y(d(r_1))) = \epsilon(\Pi_y(d(r_2)))$

11. $(\mathcal{I},d) \models Y\_meets(r_1, r_2)$ iff $\epsilon(\Pi_y(d(r_1))) = \beta(\Pi_y(d(r_2)))$

12. $(\mathcal{I},d) \models Y\_overlaps(r_1, r_2)$ iff $\beta(\Pi_y(d(r_1))) < \beta(\Pi_y(d(r_2)))$ and $\epsilon(\Pi_y(d(r_1))) < \epsilon(\Pi_y(d(r_2)))$

13. $(\mathcal{I},d) \models Y\_during(r_1, r_2)$ iff $\beta(\Pi_y(d(r_1))) > \beta(\Pi_y(d(r_2)))$ and $\epsilon(\Pi_y(d(r_1))) < \epsilon(\Pi_y(d(r_2)))$

14. $(\mathcal{I},d) \models Y\_starts(r_1, r_2)$ iff $\beta(\Pi_y(d(r_1))) = \beta(\Pi_y(d(r_2)))$ and $\epsilon(\Pi_y(d(r_1))) < \epsilon(\Pi_y(d(r_2)))$

15. $(\mathcal{I},d) \models Y\_finishes(r_1, r_2)$ iff $\beta(\Pi_y(d(r_1))) > \beta(\Pi_y(d(r_2)))$ and $\epsilon(\Pi_y(d(r_1))) = \epsilon(\Pi_y(d(r_2)))$

16. $(\mathcal{I},d) \models MDP(c_1, i, c_2)$ iff $(c_1, i, c_2) \in MDP$

17. $(\mathcal{I},d) \models SDP(c_1, i, c_2)$ iff $(c_1, i, c_2) \in SDP$

18. $(\mathcal{I},d) \models MFP(o_1, i, o_2)$ iff $(o_1, i, o_2) \in MFP$

19. $(\mathcal{I},d) \models SFP(o_1, i, o_2)$ iff $(o_1, i, o_2) \in SFP$

20. $(\mathcal{I},d) \models InstanceOf(o, c)$ iff $(o, c) \in \longrightarrow$

21. $(\mathcal{I},d) \models IsA(c_1, c_2)$ iff $(c_1, c_2) \in \Rightarrow$

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

23. $(\mathcal{I},d) \models \neg\alpha$ iff $(\mathcal{I},d) \not\models \alpha$

24. $(\mathcal{I},d) \models (\alpha \vee \beta)$ iff either $(\mathcal{I},d) \models \alpha$ or $(\mathcal{I},d) \models \beta$

25. $(\mathcal{I},d) \models (\exists_r x)\gamma$ iff for some constant symbol c of sort σ_r, $d(c) \in \pi$ and $(\mathcal{I},d) \models \gamma_c^x$

26. $(\mathcal{I},d) \models (\exists_r x)\delta$ iff for some constant symbol c of sort σ_r, $(\mathcal{I},d) \models \delta_c^x$

27. $(\mathcal{I},d) \models (\exists_c x)\alpha$ iff for some constant symbol c of sort σ_c, $(\mathcal{I},d) \models \alpha_c^x$

28. $(\mathcal{I},d) \models (\exists_o x)\alpha$ iff for some constant symbol c of sort σ_o, $(\mathcal{I},d) \models \alpha_c^x$

29. $(\mathcal{I},d) \models (\exists_p x)\alpha$ iff for some constant symbol c of sort σ_p, $(\mathcal{I},d) \models \alpha_c^x$

30. $(\mathcal{I},d) \models (\exists_i x)\alpha$ iff for some constant symbol c of sort σ_i, $(\mathcal{I},d) \models \alpha_c^x$

A.7 The query answer

Given an image base $IB = (\sigma, \Sigma)$ and an image $\mathcal{I} = (\mathcal{M},\mathcal{D},M)$ in Σ, where:

$$\mathcal{D} = (O_i, C_i, DP_i, FP_i, \rightarrow_i, \Rightarrow_i),$$

the *extension* of $\mathcal{I}$ is the image $\mathcal{I}^e = (\mathcal{M},\mathcal{D}^e, M)$ where:

$$\mathcal{D}^e = ((O_i \cup O_0), (C_i \cup C_0), (DP_i \cup DP_0), (FP_i \cup FP_0), (\rightarrow_i \cup \rightarrow_0), (\Rightarrow_i \cup \Rightarrow_0)).$$

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

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