A System for the Fast Prototyping of Multidimensional Image Retrieval

Carlo Meghini, Fabrizio Sebastiani and Umberto Straccia
Istituto di Elaborazione dell'Informazione - C.N.R.
Via S. Maria 46, I-56126 Pisa, Italy
{lastname}@iei.pi.cnr.it

Abstract

Multidimensional image retrieval (MIR) views an image as a multidimensional object, where each dimension is a channel for retrieval. MIR has the potential of putting at work together the many methods and techniques for image retrieval proposed in several different fields of computer science. We have developed a model of MIR, based on a fuzzy description logic, that identifies two main dimensions in an image (form and content) and views MIR as a special form of uncertain implication. In this paper, we present ARIANNA, *a system that implements the model.* ARIANNA *allows to quickly develop a prototype of a MIR application, and use it to test the adequacy of the application to the user's functional requirements.*

1. Introduction

Research on image retrieval (IR) has witnessed a booming interest during the last decade. The most striking feature of this research is its simultaneous but independent materialization within several fields of computer science. This fact reveals that there are many different aspects involved in IR, each requiring a specific background and methodology to be successfully tackled; but also that there may be complementary approaches to the same problems, not only within the same discipline (such as two different index structures for image data), but also cutting across different disciplines (such as similarity- versus semantic-based image retrieval). Such a richness of paradigms, methods and systems, on the long run, may result in a fragmentation prone to slow down progress. In order to overcome this problem, we have developed a model of IR [5], named the Terminological Image Retrieval Model (TIM) after the tool we have used to specify it, that places in a unified and coherent perspective the many efforts and results that are being produced under the IR label. The model has been successively extended to deal with multimedia [4]. The basic feature of TIM is to allow the capturing of all kinds of retrieval on images that have been deemed as useful, and therefore investigated in the various areas mentioned above. These kinds of retrieval can be broadly classified on the basis of the aspect of images that each of them addresses. Thus we have a first, broad categorization of IR in: syntactic similarity *vs* semantic IR. The former, in turn, can be categorized depending on the addressed image feature: thus we have color-, shape-, texture-based similarity, and there may be more. Finally, our TIM is parametric with respect to the functions that are used to asses similarity of image features, as it is well-known that those functions vary depending on the subject matter and the goals underlying the MIR application being developed.

The present paper describes an implementation of TIM aimed at supporting the construction of fast prototypes of MIR applications. For space reasons, the presentation just touches upon the most important aspects of ARIANNA. The readers interested in knowing more about the model or the system are referred to a long version of this paper [6].

2. The terminological image retrieval model

As any retrieval model, TIM consists of a language for representing images and queries, namely fuzzy description logic [8], and a retrieval function (implication relation) establishing the retrieval status value (RSV) of each image for each query. RSV estimates the degree of relevance of an image to a query.

2.1. Image representation

In TIM, each image has a form and a content dimension. The former, named "image layout", consists of the pixels that make up the image. No language facility is provided to represent a layout, as it can be dealt with in an entirely automatic way. Content, on the other hand, requires interpretation to be disclosed and is therefore of a subjective nature. TIM provides the means to represent explicitly image contents in terms of the assertions of the fuzzy description logic $\mathcal{ALC}$[8]. The syntactic components of

0-7695-0253-9/99 $10.00 © 1999 IEEE

$\mathcal{ALC}$ are: individuals (names for objects), concepts (unary predicate symbols) and roles (binary predicate symbols). Concepts can be either primitive (just names) or complex, *i.e.* expressions involving primitive concepts and constructors, which, not surprisingly, closely mimic Boolean operators. For instance, the complex concept Musician⊓∀Plays.¬ElectricInstrument is obtained by combining the primitive concepts Musician and ElectricInstrument and the primitive role Plays by the conjunction (⊓), universal quantification (∀) and negation (¬) constructors. Under the intended interpretation, such concept denotes the musicians who do not play any electric instrument. The syntax rule for $\mathcal{ALC}$-concepts (*cpt*) are given below:

$$\begin{array}{lll} \langle cpt \rangle ::= & \langle prim\text{-}cpt \rangle \mid \neg\langle cpt \rangle \mid \langle cpt \rangle \sqcap \langle cpt \rangle \mid \langle cpt \rangle \sqcup \langle cpt \rangle \mid \\ & \forall\langle role \rangle.\langle cpt \rangle \mid \exists\langle role \rangle.\langle cpt \rangle \end{array}$$

A fuzzy assertion γ is an expression of the form $\langle \alpha, n \rangle$, where $n \in [0,1]$ and α is one of the following: (1) $C(a)$, asserting that the individual a is an instance of the concept C; for example, (Musician ⊓ Teacher)(tim) makes the individual constant tim a Musician and a Teacher; (2) $R(a_1, a_2)$ (e.g. Friend(tim,tom)); (3) $C_1 \sqsubseteq C_2$, asserting that C_1 is more specific than C_2 (e.g. Pianist ⊑ (Artist ⊓ ∃Plays.Piano)). A set of assertions will be called a *knowledge base*. From a semantics point of view, please refer to [8], where the notions of interpretation, satisfiability, model and logical consequence are defined. Just let us mention that a fuzzy assertion like ⟨Musician(tim), .8⟩ has intented meaning, "the degree of being tim a Musician is ≥ .8". With $\Sigma \models \gamma$ we denote the fact that from the KB Σ we may infer the fuzzy assertion γ. We define the *maximal degree of truth* of α with respect to Σ (written $Maxdeg(\Sigma, \alpha)$) to be $\max\{n > 0 : \Sigma \models \langle \alpha, n \rangle\}$ ($\max \emptyset = 0$).

Let us now see how our logic is to be employed in order to represent image contents. Let l be an image layout, identified by individual l. In TIM, l may have an arbitrary number of associated *content descriptions*. Each such content description is the union of four component subsets of fuzzy assertions: (1) The *layout identification*, ⟨Self(l), 1⟩ whose role is to associate a content description with the layout it refers to. σ(l) denotes the set of the content descriptions associated to the layout l. (2) The *object anchoring*, a set of ⟨Rep(r, o), n⟩ fuzzy assertions where r is an individual identifying a region r of l and o is an individual identifying the object represented in r. (3) The *situation anchoring*, a set of ⟨About(l, o), n⟩ fuzzy assertions where l and o are as above. By using these assertions, it can be stated what the situation described by the layout is "globally" about. (4) The *situation description*, a set of fuzzy simple assertions (different from (1), (2) and (3)), describing important facts stated in the layout about the individuals identified by assertions of the previous two kinds.

2.2. Query representation

Queries referring to the form dimension of images are called *visual* queries. TIM addresses the following visual query types: (a) *concrete visual queries:* these consist of full-fledged images that are submitted to the system as a way to indicate a request to retrieve "similar" images w. r. t. *e.g.* color, texture features, etc.; (b) *abstract visual queries:* these are artificially constructed image elements (hence, "abstractions" of image layouts) that address specific aspects of image similarity; they can be categorized into: (b.1) *color queries:* specifications of color patches (retrieve images with similar color patch); (b.2) *shape queries:* specifications of one or more shapes (closed simple curves in the 2D space) (retrieve images with similar shapes); (b.3) any combination of the above. In order to express these queries, TIM offers a language that is considerably richer than the language for representing images, as it includes the symbols (termed SPSs) for expressing all the above query kinds. Essentially, SPSs are roles that make up an image mereology. For TIM, the atomic image element is the atomic region (a connected region, homogeneous from the color viewpoint). Connected aggregates of atomic regions constitute regions. Colors and shapes are associated to regions. The following SPSs are, thus, considered: let i be image layout and image region and r of its image regions. (i) HAIR(i,r) relates i to an atomic r; (ii) HIR(i,r) relates i to r; (iii) HS(r,s) relates r to its shape s; (iv) HC(r,c) relates r to its color c. In order to properly handles SPSs, we restrict the models of TIM image databases to those that capture the intended meaning of these symbols, expressed through conditions that will not be given here for space reasons.

The syntax of image queries is given below.

$$\begin{array}{rcl} \langle image\text{-}qry \rangle & ::= & \langle image\text{-}cpt \rangle \mid \langle image\text{-}qry \rangle \sqcap \langle image\text{-}qry \rangle \mid \\ & & \langle image\text{-}qry \rangle \sqcup \langle image\text{-}qry \rangle \\ \langle image\text{-}cpt \rangle & ::= & \exists \mathsf{About}.\langle cpt \rangle \mid \exists \mathsf{SI}.\{\langle layout\text{-}name \rangle\} \mid \\ & & \exists \mathsf{HAIR}.(\exists \mathsf{HC}.\langle color\text{-}cpt \rangle) \mid \exists \mathsf{HIR}.(\exists \mathsf{Rep}.\langle cpt \rangle) \mid \\ & & \exists \mathsf{HIR}.((\exists \mathsf{Rep}.\langle cpt \rangle) \sqcap \langle region\text{-}cpt \rangle) \\ \langle region\text{-}cpt \rangle & ::= & \exists \mathsf{HC}.\langle color\text{-}cpt \rangle \mid \exists \mathsf{HS}.\langle shape\text{-}cpt \rangle \mid \\ & & ((\exists \mathsf{HC}.\langle color\text{-}cpt \rangle) \sqcap (\exists \mathsf{HS}.\langle shape\text{-}cpt \rangle)) \\ \langle color\text{-}cpt \rangle & ::= & \{\langle color\text{-}name \rangle\} \mid \exists \mathsf{SC}.\{\langle color\text{-}name \rangle\} \\ \langle shape\text{-}cpt \rangle & ::= & \{\langle shape\text{-}name \rangle\} \mid \exists \mathsf{SS}.\{\langle shape\text{-}name \rangle\} \end{array}$$

The first thing to observe is the presence, in the clauses defining *image-*, *color-* and *shape-cpt*, of a new concept constructor of the form $\{a\}$ where a is an individual, which may be a *layout-*, a *color-* or a *shape-name,* respectively. This constructor is called *singleton*, and represents a concept having only the individual a as instance. Image queries are thus concepts of the DL $\mathcal{ALCO}$, which extends $\mathcal{ALC}$ with the singleton constructor. A query is a combination (through ⊓ and ⊔) of *image-cpt*, having four possible forms. (i) *image-cpt* may be a query on some content object, explicitly asserted to be related to the sought images through an

$(C \sqcap D)(\mathsf{i})$ $(C \sqcup D)(\mathsf{i})$	$\Phi(C(\mathsf{i})) \cup \Phi(D(\mathsf{i}))$
$(\exists\mathsf{SI}.\{\mathsf{qi}\})(\mathsf{i})$	$\{\langle \mathsf{SI}(\mathsf{i},\mathsf{qi}), \sigma_i(\mathsf{i}^{\mathcal{I}},\mathsf{qi}^{\mathcal{I}})\rangle\}$ if $\sigma_i(\mathsf{i}^{\mathcal{I}},\mathsf{qi}^{\mathcal{I}})\rangle\} > 0$, $\emptyset$ otherwise
$(\exists\mathsf{About}.C)(\mathsf{i})$	$\emptyset$
$(\exists\mathsf{HAIR}.\exists\mathsf{HC}.\{\mathsf{c}\})(\mathsf{i})$	$\{\langle x,1\rangle\}$ if $\mathsf{i}^{\mathcal{I}}$ has an atomic region of color $\mathsf{c}^{\mathcal{I}}$, $\emptyset$ otherwise
$(\exists\mathsf{HAIR}.\exists\mathsf{HC}.\exists\mathsf{SC}.\{\mathsf{c}\})(\mathsf{i})$	$\{\langle x,n\rangle\}$ if n, the best approximation of $\mathsf{c}^{\mathcal{I}}$ in $\mathsf{i}^{\mathcal{I}}$, is positive; $\emptyset$ otherwise
$(\exists\mathsf{HIR}.C)(\mathsf{i})$	$\{\langle \mathsf{HIR}(\mathsf{i},\mathsf{r}_1),1\rangle,\ldots,\langle \mathsf{HIR}(\mathsf{i},\mathsf{r}_k),1\rangle\} \cup \Phi(C(\mathsf{r}_1)) \cup \ldots \cup \Phi(C(\mathsf{r}_k))$, $(k \geq 0)$, for all regions $\mathsf{r}_j{}^{\mathcal{I}}$ in $\mathsf{i}^{\mathcal{I}}$ such that $\mathsf{Rep}(\mathsf{r}_j,\mathsf{o}) \in \delta$ for some $\delta \in \sigma(\mathsf{i})$
$(\exists\mathsf{Rep}.C)(\mathsf{r})$	$\emptyset$
$((\exists\mathsf{Rep}.C) \sqcap D)(\mathsf{r})$	$\Phi(D(\mathsf{r}))$
$(\exists\mathsf{HC}.\{\mathsf{c}\})(\mathsf{r})$	$\{\langle \mathsf{HC}(\mathsf{r},\mathsf{c}),n\rangle\}$ if n, the percentage of $\mathsf{c}^{\mathcal{I}}$ in $\mathsf{r}^{\mathcal{I}}$, is positive; $\emptyset$ otherwise
$(\exists\mathsf{HC}.\exists\mathsf{SC}.\{\mathsf{c}\})(\mathsf{r})$	$\{\langle x,n\rangle\}$ if $n = max_{c\in C} v_c$ is positive; $\emptyset$ otherwise
$(\exists\mathsf{HS}.\{\mathsf{s}\})(\mathsf{r})$	$\{\langle \mathsf{HS}(\mathsf{r},\mathsf{s}),1\rangle\}$ if $\mathsf{s}^{\mathcal{I}}$ is the shape of $\mathsf{r}^{\mathcal{I}}$, $\emptyset$ otherwise
$(\exists\mathsf{HS}.\exists\mathsf{SS}.\{\mathsf{s}\})(\mathsf{r})$	$\{\langle x,n\rangle\}$ if n, the similarity between the shape of $\mathsf{r}^{\mathcal{I}}$ and $\mathsf{s}^{\mathcal{I}}$ is positive; $\emptyset$ otherwise

Figure 1. The decomposition function

About role assertion (termed "situation anchoring"). In the query, the object is required to be an instance of *cpt*, *i.e.* an $\mathcal{ALCO}$ concept built with the symbols used for situation descriptions. For instance, under the obvious lexicon, the images about an Italian musician are retrieved via the query $\exists\mathsf{About}.(\mathsf{Musician} \sqcap \exists\mathsf{Born}.\mathsf{Italy})$. (ii) *image-cpt* may be a concrete visual query, *i.e.* a prototype image layout l is provided in the query by specifying the singleton l in the scope of the existential quantification on the SPS SI. By so doing, the similarity with l is captured in the query. (iii) *image-cpt* may be a color abstract visual query (using existential quantification on the HC SPS) followed by a *color-cpt*; the latter is a singleton with the name of the color, optionally preceded by a color similarity predicate. (iv) *image-cpt* may be a query on an image region. This kind of queries come in two forms. (a) A form addressing the content dimension, and just consists of a Rep clause. In order to qualify for this kind of queries, an image must have an associated content description containing a Rep role assertion (object anchoring), relating a region of the image to an individual constant that is an instance of the *cpt* that follows. (b) A form extending the first with an additional condition (*region-cpt*) on the color or the shape (or both) of the involved region. A shape condition is expressed via a *shape-cpt*, which is strictly analogous to *color-cpt*. As an instance of an image query, consider the query asking for the images showing a cylindric reddish hat: $\exists\mathsf{HIR}.((\exists\mathsf{Rep}.\mathsf{Hat}) \sqcap (\exists\mathsf{HC}.\exists\mathsf{SC}.\{\mathsf{red}\}) \sqcap (\exists\mathsf{HS}.\{\mathsf{cylinder}\}))$. It presents an interesting case of mixed form- and content-based retrieval. The Rep clause refers to the semantics of the image, namely to what an object is. An image is retrieved only if it displays something that has been explicitly asserted to be a hat. The HC clause refers to image form, and requires, in the retrieved images, the presence of a patch of color similar to red. The HS clause poses a condition on the contour of an atomic image region. The conjunction of these three clauses constraints the condition that they each of them expresses to be true of the same region, thus capturing the query spelled out above.

2.3. The retrieval function

A TIM image database has three main components: a collection of image layouts; a collection of their content descriptions; and a knowledge base, providing definitions of the concepts employed. An *image database* is a triple $DB = \langle D, \Sigma_C, \Sigma_D\rangle$ where D is a set of layouts, Σ_C is the set of content descriptions of the layouts in D, and Σ_D is a set of fuzzy assertions. In response to a query Q addressed to $DB = \langle D, \Sigma_C, \Sigma_D\rangle$, each image is attributed a <u>*retrieval status value*</u> m given by: $m = max_{\delta\in\sigma(\mathsf{i})}\ \{\overline{Maxdeg}(\Sigma_D \cup \delta, Q(\mathsf{i}))\}$. where $\overline{Maxdeg}$ is the restriction of $Maxdeg$ that assigns to SPSs the intended semantics. As an example, let us consider a DB containing just the image named i, Σ_C consists of the content description $\{\langle\mathsf{Self}(\mathsf{i}),1\rangle, \langle\mathsf{About}(\mathsf{i},\mathsf{o}),.8\rangle, \langle\mathsf{DonGiovanni}(\mathsf{o}),1\rangle\}$, and Σ_D consists of $\langle\mathsf{DonGiovanni} \sqsubseteq \mathsf{EuropeanOpera},1\rangle$, $\langle\mathsf{EuropeanOpera} \sqsubseteq \mathsf{Opera} \sqcap (\exists\mathsf{ConductedBy}.\mathsf{European}),.9\rangle$. Suppose we are interested in images about operas conducted by a European director. To this end, we can use the query $\exists\mathsf{About}.(\mathsf{Opera} \sqcap \exists\mathsf{ConductedBy}.\mathsf{European})$. It can be verified that the RSV attributed to i is .8.

3. The decomposition strategy

The logical kernel of the model implementation is the Query Evaluation Procedure (QEP), which, given a query Q and a *DB*, returns a ranking of the images in *DB*. The basic tasks of the QEP are (1) the identification of the structure of the sub-queries Q_i of Q; (2) the evaluation of Q_i; and (3) the calculation of the RSV of the image on the basis of the Q_i. QEP relies on special processors for the evaluation of the Q_i, and on a Theorem Prover (TP) for our logic. For space reasons, we cannot discuss the TP here. The interested reader is referred to [8][1]. The QEP follows a method, called the *decomposition strategy*, which guarantees the correctness of

[1] To download the TP refer to the author's www home page.

the result, given the correctness of the sub-query evaluations. The rest of the section is devoted to the illustration of the decomposition strategy.

3.1. Foundations

Let Q be a query and i a candidate image in DB. By definition, the RSV of i to Q is the value m associated to $Q(\mathsf{i})$. In order to derive m, the function Φ is applied to $Q(\mathsf{i})$ obtaining a *decomposition* of query Q, *i.e.* a set of fuzzy assertions $\Phi(Q(\mathsf{i}))$ which satisfies the following property: For each $\delta \in \sigma(\mathsf{i})$: $\overline{Maxdeg}(\Sigma_D \cup \delta, Q(\mathsf{i})) = Maxdeg(\Sigma_D \cup \Phi(Q(\mathsf{d})) \cup \delta, Q(\mathsf{i}))$. From this, the decomposition principle that is at the basis of the implementation of the model follows. *Query Decomposition Principle*: "The RSV of the image i to the query Q is given by the maximal degree of truth of $Q(\mathsf{i})$ with respect to the knowledge base consisting of $\Phi(Q(\mathsf{d}))$ and Σ_D, maximized over the set of content descriptions $\sigma(\mathsf{i})$". A formal proof of the correctness and completeness of the query decomposition principle can be found in [6]. Operatively, it permits to perform query evaluation in the general context of our logic, once the sub-queries involving the special symbols SPSs are identified and evaluated by the function Φ and the corresponding fuzzy assertions are collected in $\Phi(Q(\mathsf{d}))$.

Now let us see how such a Φ is defined. For clarity, Φ will be introduced by following the query language syntactical structure.

3.2. Decomposition-evaluation of sub-queries

The decomposition and evaluation of image queries is presented in Figure 1. Recall that the object denoted by the individual o is written as $\mathsf{o}^{\mathcal{I}}$. (i) Concrete visual queries, having the form $(\exists \mathsf{SI}.\{\mathsf{qi}\})(\mathsf{i})$ are evaluated by generating the fuzzy assertion stating the similarity between the given image layout i and the query layout qi, with degree of truth equal to the degree of similarity between these layouts, as established by the global similarity function σ_i. Note that in case the latter value is zero, no assertion is generated in order not to block the inference on the rest of the query. The same behavior is adopted whenever a similarity function is involved. (ii) Queries on situation anchoring, formulated in terms of the About SPS, are just ignored by Φ, as the knowledge for their evaluation is already part of the image database, namely in Σ_C and Σ_D. (iii) Similarly for queries on object anchoring, formulated in terms of the Rep SPS. These queries may stand alone (*i.e.* be of the form $(\exists \mathsf{Rep}.C)(\mathsf{r})$) or be conjoined to a *region concept* (*i.e.* $((\exists \mathsf{Rep}.C) \sqcap D)(\mathsf{r})$); in both cases, the content sub-query gives no contribution to $\Phi(Q(\mathsf{d}))$. (iv) Abstract visual queries come in two sorts, depending on the kind of image region addressed. (a) The first address exclusively atomic regions. It consists of color queries and aim at retrieving images having a patch of a specified color. They have the form $(\exists \mathsf{HAIR}.\exists \mathsf{HC}.\{c\})(\mathsf{i})$, where c is the name of the color that an atomic region of the image layout named i must have. If this is indeed the case, Φ evaluates the query by generating the fuzzy assertion made by attaching to the query assertion with degree of truth 1. If not, the empty set is generated. Optionally, a similarity condition on the specified color may be stated, like $(\exists \mathsf{HAIR}.\exists \mathsf{HC}.\exists \mathsf{SC}.\{\mathsf{c}\})(\mathsf{i})$. The specification of the color similarity condition radically changes the query evaluation, which yields, as degree of truth, the degree of similarity between the given color and the color of the atomic regions of i that comes closest to it. If i has an atomic region of color c, then the degree of truth is 1^2; otherwise, the evaluation produces the "best match" among i's colors and c. As a desirable consequence, the latter type of color queries generalizes the former. (b) The second sort address both atomic and non-atomic regions and takes the form $(\exists \mathsf{HIR}.C)(\mathsf{i})$. As for the other mereological symbols, Φ treats these queries by generating an assertion of the form $\langle \mathsf{HIR}(\mathsf{i},\mathsf{r}), 1\rangle$ for each region r of i which is the subject of an object anchoring assertion, while recursively applying itself to the assertion $C(\mathsf{r})$. The reason for this is that C is bound to include a Rep clause, which, of course, restricts the candidate regions to all and only those referenced by object anchorings. As discussed above, C may optionally contain a *region concept*, which may be a color query, a shape query or a conjunction of the two. The last case is handled, as customary, by separately evaluating the conjuncts, and is not reported in Figure 1 for brevity. Let us quickly review the first two cases: (i) $(\exists \mathsf{HC}.\{\mathsf{c}\})(\mathsf{r})$: is evaluated by generating the corresponding fuzzy assertion, having as degree of truth the percentage of color c in the region r. (ii) $(\exists \mathsf{HC}.\exists \mathsf{SC}.\{\mathsf{c}\})(\mathsf{r})$: the truth degree assigned is, for each color c in r, the maximum over the values v_c, where v_c is the minimum between the similarity between c and $\mathsf{c}^{\mathcal{I}}$, and the percentage of c in r. It is easy to verify that this is a generalization over the previous case. (iii) $(\exists \mathsf{HS}.\{\mathsf{s}\})(\mathsf{r})$: if the shape of r equals s, the evaluation of this query yields the corresponding assertion with degree 1; otherwise, no assertion is generated. (iv) $(\exists \mathsf{HS}.\exists \mathsf{SS}.\{\mathsf{s}\})(\mathsf{r})$: same as before, except that in this case the similarity between r's shape and s is assigned as degree of truth to the corresponding assertion.

4. A prototypical implementation

The prototype system that we have developed, named ARIANNA, consists of two main modules: the *indexing* module (hereafter IM for short) supporting the acquisition of images and the creation of the various represen-

[2] At least as long as $\sigma_c(\mathsf{c}^{\mathcal{I}}, \mathsf{c}^{\mathcal{I}}) = 1$, which would seem a quite reasonable assumption on similarity functions, even though it has not been so stated for generality.

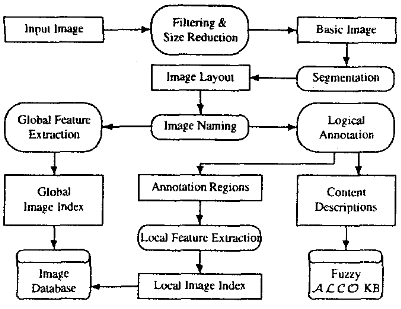

Figure 2. Image acquisition in ARIANNA

Figure 3. Input and relative Basic Image

tations needed for performing retrieval, and the *query* module (QM), performing query evaluation.

4.1. The indexing module

Figure 2 illustrates the sequence of the various operations inherent image acquisition (rectangular boxes represent data, while ovals represent modules). **Filtering and Size Reduction:** acquisition begins from an *Input Image I* (GIF or JPEG). The input image is reduced, if necessary, to the system size, (currently fixed to 128×128 pixels). After size reduction, the RGB color space is abandoned for the HSB space, and noise reduction is performed on the image, by applying a color median filter. The result is the *Basic Image* I_B. Figure 3 presents a sample input image (left) and the corresponding basic image. **Segmentation:** the Segmentation Module derives the Image Layout I_L from I_B. I_L is used solely to support the user in specifying the image regions that are to be annotated via Rep assertions. The derivation of the I_L implies two operations: segmentation and color quantization. These operations are strictly related and are both performed by the Segmentation Module. Given the generality of the tool being presented, we have adopted a flexible solution which produces 7 different segmentations, each provided at several levels of quantization of the color space. The image indexer can use the image partition of anyone of these segmentations, or of any combination of them, in order to select the image regions to be annotated. The channels on which the Basic Image is segmented are: color, saturation, color and saturation, brightness. For each channel, 3 levels of quantizations are used, namely 3, 7, and 15 levels. In order to obtain these segmentations, textbook techniques based on region growing have

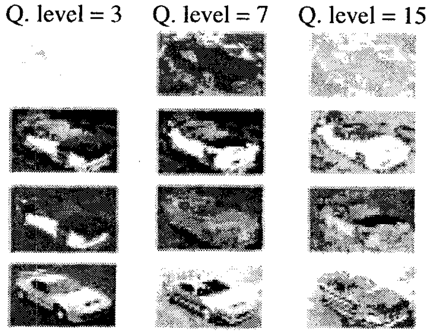

Figure 4. Various segmentations

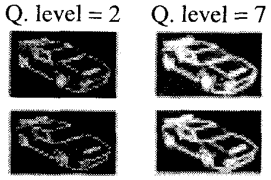

Figure 5. Edge segmentations

been employed; these techniques, as well as the others used for image segmentation, are not presented as not new nor central to the present context. The color, saturation, color and saturation, brightness segmentations of the sample image of Figure 3 are showed in Figure 4, in this order, from the top down (in this figure, colors are used to highlight regions and do not have direct correspondence with those in the image). In addition, two segmentations based on edges are generated, each with two levels of quantization: 2 and 7 colors. Edge detection techniques have been employed to obtain these segmentations (see Figure 5). Finally, a segmentation on texture is derived, at two levels of quantization (see Figure 6). The reason for having these segmentations and not others are, of course, mostly empirical: we presume that the combination of these segmentations covers a significant range of "difficult" images. Different presumptions would maybe lead to a different choice, but this is not important for the model. Figure 7 shows how the screen looks like after the segmentation operation has been performed on the sample image. A 3×3 grid is used to display the various images (the central image is the input image) and is surrounded by the 7 differ-

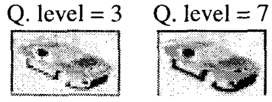

Figure 6. Texture segmentations

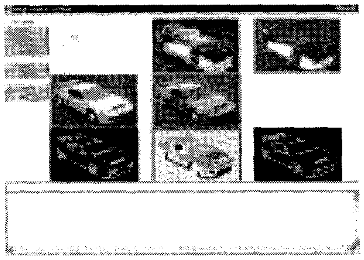

Figure 7. Screen after segmentation

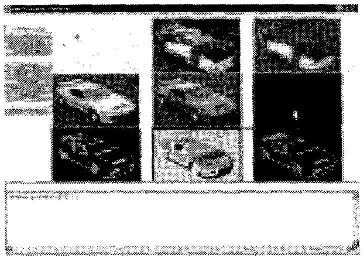

Figure 8. Screen during region selection

ent segmentations introduced above. Only one level is shown for each segmentation, and the user can move through the different levels by clicking on the corresponding cell of the grid. The empty cell is reserved to region selection for annotation, as we will see in a moment. **Image Naming:** prior to any indexing operation, I_L must be identified by means of an individual, and this is the objective of the Image Naming operation: the user is asked to give an unique name for the image being acquired. From that point on, the name becomes the unique image identifier and two operations are possible: *Global Feature Extraction* and *Logical Annotation*. **Global Feature Extraction:** it aims at deriving the representation of the image needed to answer user queries. The so obtained representation, named *Global Image Index* to stress its being relative to the whole image, is stored into an archive which is part of the *Image Database DB*. According to the definition of the Φ function given in Figure 1, the following features are extracted from the Basic Image named i: (i) The first three moments of the image (true) color histogram for each channel c of the HBS color space, that is: $m_{1c}^{\mathsf{i}^{\mathcal{I}}} = \frac{1}{N}\sum_{j=1}^{N} p_{c,j}^{\mathsf{i}^{\mathcal{I}}}$, $m_{2c}^{\mathsf{i}^{\mathcal{I}}} = \sqrt{\frac{1}{N}\sum_{j=1}^{N}(p_{c,j}^{\mathsf{i}^{\mathcal{I}}} - m_{1c}^{\mathsf{i}^{\mathcal{I}}})^2}$ and $m_{3c}^{\mathsf{i}^{\mathcal{I}}} = \sqrt[3]{\frac{1}{N}\sum_{j=1}^{N}(p_{c,j}^{\mathsf{i}^{\mathcal{I}}} - m_{1c}^{\mathsf{i}^{\mathcal{I}}})^3}$, where $p_{c,j}^{\mathsf{i}^{\mathcal{I}}}$ is the value of the channel c in the point j of the image layout $\mathsf{i}^{\mathcal{I}}$. These features are extracted in order to compute a special kind of image similarity σ_i, namely the one based on global color [7]. Therefore, they will be employed upon evaluating queries of the form (∃SI.{qi})(i), in a way to be explained upon introducing the query module. (ii) The list of colors occurring in the image; this is used in order to process color queries on atomic regions, *i.e.* (∃HAIR.∃HC.{c})(i). (iii) In order to evaluate queries on global color similarity (*i.e.* (∃HAIR.∃HC.∃SC.{c})(i)), the vector V is extracted: V has as many positions as the elements of the color set from which the user draws in specifying similar color queries on atomic regions (15×3×3=135, in our case); the V position associated to the color $\mathsf{c}^{\mathcal{I}}$ gives the degree of similarity between $\mathsf{c}^{\mathcal{I}}$ and the color in the image that best approximates it, as required by Φ. The distance measure used as color similarity function σ_c is: $\sigma_c(\mathsf{i}^{\mathcal{I}},\mathsf{j}^{\mathcal{I}}) = \sum_c \sum_{k=1}^{3} |m_{kc}^{\mathsf{i}^{\mathcal{I}}} - m_{kc}^{\mathsf{j}^{\mathcal{I}}}|$, where c ranges, as above, on the 3 channels of the color space. **Logical Annotation:** it permits the specification of one or more *Content Descriptions* for named images. Upon requesting it, IM automatically creates the layout identification assertion (*i.e.* ⟨Self(i), 1⟩), and supports the creation of the other kinds of assertions. In particular: (i) Situation anchoring is supported by asking the user for the name of the object to be linked to the present image via an About assertion. (ii) Object anchoring is supported analogously, with an additional help to the user in selecting a region image. Region naming is done "on demand", *i.e.* whenever a new Rep assertion is to be specified, IM automatically creates a name for the involved *Annotation Region* and proposes it to the indexer, who is free to use or change it. Figure 8 shows the IM screen during the selection of a region to be annotated via a Rep assertion. The region is constructed in the cell that is at the right of the cell showing the input image. The user just clicks on any region of any segmentation and, as a result, the region containing the click point is displayed. In the lower part of the screen, the identification assertion, automatically created by the system, is displayed in the format the TP expects. The specification of situation description assertions closes the annotation of an image. Each content description is then passed to the TP, which files it in the *Fuzzy $\mathcal{ALCO}$ Knowledge Base*. **Local Feature Extraction Module:** in order to support abstract visual queries, a feature extraction task is performed on annotation regions. The extracted features make up a *Local Image Index* (local to annotation regions, of course), which is filed in an apposite archive of the *DB*. The structure of this index, relatively to the annotation region r, is as follows (see Figure 1): (i) the name of r. (ii) r's color histogram, used to process color queries on it (*i.e.* (∃HC.{c})(r)). Most of the entries in this histogram will be 0, since r is the union of a few atomic regions. Consequently, the histogram is not expected to be large. (iii) The vector T, having as many positions as the vector V above, and used to evaluate similar color queries on annotation regions; the T position associated to the color $\mathsf{c}^{\mathcal{I}}$, gives $n = max_{c \in C} v_c$, as specified in Figure 1. (iv) r's shape represented by the 8-contour and by 7 invariant moments [3]. The former representation is used to process "precise" shape queries (*i.e.* $(\exists\mathsf{HS}.\{\mathsf{s}\})(\mathsf{r})$) while the latter is used when the optional similarity condition is given (*i.e.* $(\exists\mathsf{HS}.\exists\mathsf{SS}.\{\mathsf{s}\})(\mathsf{r})$).

4.2. The query module

The query module QM provides two basic services: query specification (a) and query evaluation (b). (a) supports the construction of image queries Q, performed by the user via the interface (see Figure 9). (b) is performed by the Image Query Evaluation Procedure. Given Q, the Image Query Assertion Builder iteratively produces a query assertion $\alpha = Q(\mathsf{i})$ for each acquired image i. α is passed on to the Image Query Decomposition & Evaluation (IQDE for short) function which implements the evaluation by decomposition process described in detail in Section 3. In deriving $\Phi(\alpha)$ the IQDE accesses the DB in order to fetch the Global Image Index of i and the Local Image Index of each of i's annotation regions. The way these representations are used by the IQDE is mostly straightforward, once one bears in mind the definition of Φ for image queries and the structure of the representations themselves. For instance, upon evaluating the query assertion $(\exists\mathsf{HAIR}.\exists\mathsf{HC}.\{\mathsf{c}\})(\mathsf{i})$, the IQDE checks whether c is in the list of colors occurring in i, which is part of the global index; if the check is positive, then the assertion $\langle(\exists\mathsf{HAIR}.\exists\mathsf{HC}.\{\mathsf{c}\})(\mathsf{i}), 1\rangle$ is generated. Analogously, in order to evaluate the query $(\exists\mathsf{HIR}.C)(\mathsf{i})$, the IQDE generates an assertion $\langle\mathsf{HIR}(\mathsf{i}, \mathsf{r}_1), 1\rangle$ for each annotation region r, then applies itself to the evaluation of $C(\mathsf{r})$. As a final example, $(\exists\mathsf{HC}.\exists\mathsf{SC}.\{\mathsf{c}\})(\mathsf{r})$ is evaluated by generating a fuzzy assertion whose degree of truth is the value found in the appropriate position of the T vector, which is part of r's local index. The last step of the evaluation procedure is the invocation of the TP, to which α is sent with the purpose of computing its m value against: (a) Σ_D, (b) i's content descriptions (both these are part of the Knowledge Base maintained by the TP); and (c) the just computed $\Phi(\alpha)$.

5. Conclusions

We have presented a system for building prototypical image retrieval applications that reconciles in a unique, well-founded framework the many functionalities that are usually found under the image retrieval label. The most important contribution of the system is the full and proper use of semantics and knowledge, while offering, at the same time, the similarity-based kind of retrieval that is undoubtedly the most significant contribution of the research carried out in these two areas during the last decade. At present, to the best of our knowledge, no other model offering the same functionalities as the one presented here, exists. Since the representations handled by the model have a clean semantics, further extensions to the model are possible. For instance, image retrieval by spatial similarity can be added: at the form level, effective spatial similarity algorithms (e.g. [2]) can be embedded in the model via procedural attachment, while significant spatial relationships can be included in content descriptions by drawing from the many formalisms developed within the qualitative spatial reasoning research community [1]. Analogously, the model can be enhanced with the treatment of texture-based similarity image retrieval.

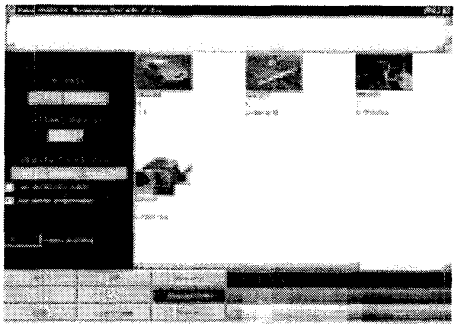

Figure 9. Displaying of the result of a query

References

[1] A. Cohn. Calculi for qualitative spatial reasoning. In *Proc. of AISMC-93*, LNCS, 1996.

[2] V. Gudivada and V. Raghavan. Design and evaluation of algorithms for image retrieval by spatial similarity. *ACM Trans. on Information Systems*, 13(2):115–144, 1995.

[3] B. Mehtre, M. Kankanhalli, and W. Lee. Shape measures for content based image retrieval: a comparison. *Information Processing and Management*, 33(3):319–336, 1997.

[4] C. Meghini, F. Sebastiani, and U. Straccia. Modelling the retrieval of structured documents containing texts and images. In *Proc. of ECDL-97*, LNCS n. 1324.

[5] C. Meghini, F. Sebastiani, and U. Straccia. The terminological image retrieval model. In *Proc. of ICIAP'97*, LNCS n. 1311, pages 156–163.

[6] C. Meghini, F. Sebastiani, and U. Straccia. A model of multimedia information retrieval. Tech. Rep. B4-21-09-98, CNR-IEI, Sept. 1998.

[7] M. Stricker and M. Orengo. Similarity of color images. In *Proc. of SPIE Conf. on Storage and Retrieval for Still Images and Video Databases*, pages 381–392, 1995.

[8] U. Straccia. A fuzzy description logic. In *Proc. of 15th Conf. of the American Association for Artificial Intelligence*, pages 594–599, 1998.