A Model of Multimedia Information Retrieval

Carlo Meghini, Fabrizio Sebastiani and Umberto Straccia

Consiglio Nazionale delle Ricerche

Istituto di Elaborazione dell'Informazione

Ghezzano, Via Alfieri, 1 - 56010 Pisa, Italy

E-mail: {meghini,fabrizio,straccia}@iei.pi.cnr.it

Abstract

Research on multimedia information retrieval (MIR) has witnessed a booming interest during the last five years or so. A prominent feature of this research trend is its simultaneous but independent materialization within several fields of computer science. The resulting richness of paradigms, methods and systems that has occurred as a result may, on the long run, result in a fragmentation of efforts prone to slow down progress.

The primary goal of this study is to promote an integration of methods and techniques for MIR by contributing a conceptual model which places in a unified and coherent perspective the many efforts and results that are being produced under the label of MIR. The model offers a retrieval capability that spans two *media,* text and images, but also several dimensions: form, content and structure. In this way, it reconciles similarity-based methods with semantic-based retrieval, providing the guidelines for the design of systems that are able to provide a generalized multimedia retrieval service, where the existing forms of retrieval not only coexist in harmony, but can be combined in any desired manner. The model is formulated in terms of a fuzzy description logic, which plays a twofold role: (1) it directly models semantic retrieval, and (2) it offers an ideal framework for the integration of the multimedia and multidimensional aspects of retrieval mentioned above. The model also provides relevance feedback for both text and image retrieval, integrating known techniques for taking into account user judgments.

The implementability of the model is finally addressed, by presenting a decomposition technique that reduces query evaluation to the processing of simpler requests, solvable with widely known methods for text and image retrieval, and semantic processing. A prototype for multidimensional image retrieval is presented in order to show this decomposition technique at work in a significant case.

Contents

1 Introduction

The central concern of multimedia information retrieval (MIR, for short) is easily stated: given a collection of multimedia documents, find those that are relevant to a user information need. The meaning of the terms involved in this definition is nowadays widely known. Most of the people using computers have come across, one way or another, a multimedia document, namely a complex information object, with components of different kinds, such as text, images, video and sound, all in digital form. An ever growing amount of people are everyday accessing collections of such documents in order to satisfy the most disparate information needs. Indeed, the number and types of multimedia information providers is steadily increasing: currently, it includes publishing houses, cultural, scientific and academic institutions, cinema and TV studios, as well as commercial enterprises offering their catalogs on-line. The potential users of the provided information range from highly skilled specialists to laymen. All these users share the same expectation: to be able to retrieve the documents that are relevant to their information needs.

While commercial products offering generalized MIR are still to come, research on MIR has witnessed a booming interest during the last 5 years or so. The most striking feature of this research is its simultaneous but independent materialization within several fields of computer science. To mention only the main streams, there has been work on MIR carried out within the multimedia, the information retrieval, the digital signal processing, and the database communities, while significant signs of interests may be observed in other sectors, notably artificial intelligence. It also reveals that there are many different aspects involved in MIR, each requiring a specific background and methodology to be successfully tackled, and also that there may be complementary approaches to the same problems, not only within the same discipline (such as two different index structures for multimedia data), but also cutting across different disciplines (such as similarity- versus semantic-based image retrieval). Such a richness of paradigms, methods and systems is somewhat inherent in the early stages of development of a new discipline, when empirical studies are needed to understand the nature of a phenomenon and try out different ways of capturing it. However, on the long run, this very same richness may ultimately result in a fragmentation prone to slow down progress.

We believe that MIR has reached the stage in which unification of the existing approaches is called for, given that the basic concepts underlying the discipline are understood sufficiently well. The primary goal of this study is to promote such unification by contributing a conceptual model of MIR, which places in a unified and coherent perspective the many efforts and results that are being produced under the MIR label in the above mentioned fields. We aim at establishing MIR as a whole endeavor, endowed with its own goals and methods, and articulated in several sub-fields. Progress in each of the sub-fields is then to be seen as a contribution to the development of an organism, rather than as an addition to a disarticulated corpus.

The basic feature of our model is to capture all kinds of retrieval on multimedia documents that have been deemed as useful, and therefore investigated in the various areas of computer science mentioned above. These kinds of retrieval can be broadly classified on the basis of the aspect of documents that each of them addresses. Thus we have retrieval based on: syntactic similarity, semantics, structure and profile. This categorization is indicative of the method that we have followed in deriving our view of MIR. Rather than proceeding from the bottom up by connecting together existing models, our approach is top-down: it models the various forms of retrieval as the projections of a basic operation on the different dimensions which make up the structure of the information space.

For a greater perspicuity, our conceptual model will be specified both at the informal and at the formal level, by letting each introduced notion be followed by a corresponding definition in a suitable formalism. The formalism we have chosen is mathematical logic. The rationale of this choice lies primarily in the fact that mathematical logic, among other things, is often used as a formal counterpart of natural language, thus a most suitable tool to make concepts precise. In addition, logic has proven most successful in capturing the essential nature of information systems, and a MIR system, as it is to be expected, is no exception. The particular logic we adopt is Description Logic (DL). DLs are a family of contractions of the predicate calculus that, in the recent past, have been subject to thorough investigations both from the logical and the computational viewpoint. In choosing a specific DL, we have been guided by the need, typical of information modelling, of finding a formalism that represents a good compromise between expressive power and the computational complexity of the reasoning problems involved in MIR.

Besides the foundational goal pointed out above, the aim of the present study is to bring concrete benefits to various actors of the MIR stage. In particular:

- To the designers of MIR systems, the model provides guidelines for the design of systems that are able to provide a generalized retrieval service, in which the existing forms of retrieval not only co-exist in harmony, but can be combined in any desired manner. This feature places our model beyond the current state of the art. Following these guidelines, and using an appropriate tool that implements the model, a designer can quickly build a prototype of the system to be developed, and use such prototype to test its adequacy to the user's functional requirements. We have prototyped a significant portion of such a tool, as illustrated later in this paper.

- To users of MIR systems, the model is a tool for an effective communication with the application designers, enabling them to state precisely what they expect from the system. To this end, the model provides a terminology that is at the same time rigorous and shared with the system designers. Once a prototype of the application has been realized, the dialog between users and designers moves to an operational level, where the prototype is evaluated and, if necessary, refined.

- To researchers who offer various contributions to MIR, the model gives the possibility to see these contributions as part of a larger endeavor. Hopefully, this will increase awareness of the limitations of current approaches and will stimulate improvement by integration of complementary approaches. As a further benefit to researchers, the formal specification of the model, by viewing MIR as a special form of implication, may be used as a basis for formal investigations on specific aspects of MIR, including extensions to the present model.

The paper is structured as follows. The conceptual framework underlying our model, and a corresponding terminology, is laid down in Section 2, and subsequently used to review a significant part of related work, in Section 3. The rest of the paper presents the technical development of the model, starting with Section 4, which concisely introduces the description logic that will constitute our main tool throughout the paper. Sections 5 to 7 deal with the aspects of documents that our model addresses; for each of them we first discuss issues related to modelling and then switch to the semantics of the related query facilities. Section 8 presents a unified, hierarchically structured query language which brings together all the issues discussed in Sections 5 to 7. In Section 9 we deal with retrieval and show how the degree of relevance of a document to a query may be seen in terms of the fuzzy DL that underlies both the representation and query languages. Section 10 discusses the implementation of the model, and presents a general technique for query evaluation. Relevance feedback is tackled in the following Section, where it is shown how this important stage of retrieval is incorporated into the previously presented model. Section 12 briefly sketches the main traits of the tool that we have developed to support the development of prototypical MIR systems. Section 13 concludes.

2 A view of MIR

After settling for such ambitious goals, we will limit the scope of our work to the treatment of two media: text and images. These media are by far the most investigated and therefore best understood ones, therefore they suit the foundational work we are presenting here. Throughout the paper, it will be argued that the principles that inspire our approach can also be applied to other media, such as audio and video. However, we have preferred to deal only with text and images in order to be able to discuss the relevant issues with the necessary depth, while relieving the reader from the burden of getting acquainted with a large formal lexicon.

2.1 Information retrieval

The notion of information retrieval (IR, for short) has much evolved from the late 50's, when it attracted significant scientific interest in the context of textual document retrieval. Early characterizations of IR simply relied on an "objective" notion of topic-relatedness (of a document to a query). Later, the essentially subjective concept of relevance gained ground, and eventually became the cornerstone of IR [77]. Nowadays, everybody in the IR community would agree that IR is synonymous with "determination of relevance". Unfortunately, relevance is itself a vaguely defined concept. In quite a different context, philosophers of language [5] and cognitive scientists [89] have proposed alternative characterizations of this notion, all making the subject a controversial one. Not surprisingly then, the debate on what "relevance" may mean in the context of IR is still open (a recent account of such a debate can be found in [62]).

In the meantime, the area of multimedia documents came into existence and demanded an IR functionality that no classical method was able to answer, due to the *medium mismatch problem* (in the image database field, this is often called the *medium clash problem*). This problem refers to the fact that, when documents and queries are expressed in different media, matching is difficult, as there is an inherent intermediate mapping process that needs to reformulate the concepts expressed in the medium used for queries (*e.g.* text) in terms of the other medium (*e.g.* images). In response to this demand, a wide range of methods for achieving IR on multimedia documents has been produced, often based on techniques largely foreign to the IR field. Our basic motivation in conducting the research presented here is that *a reconciliation between these new developments and traditional IR is needed*, in order to foster cross-fertilization and promote the development of a more mature technology, able to enhance the respective approaches via their integration.

As a first step towards this reconciliation, we now propose a general definition of IR that preserves the meaning underlying the tradition, while being consistent with new developments in multimedia. We will regard information retrieval as *the task of identifying documents in a collection on the basis of properties ascribed to the documents by the user requesting the retrieval.* That is, a document d is to be retrieved in response to a request r issued by a certain user if (and only if) that user would recognize d as having the property expressed by r. The many different types of retrieval that can be envisaged on a multimedia document can then be seen as special cases of the operation just defined, via an appropriate categorization of the properties that may be ascribed to a document. This categorization is outlined in the next section.

2.2 Dimensions of multimedia documents

We will call a document *simple* if it cannot be further decomposed into other documents. In the present context, images and pieces of text are (the only) simple documents. A simple document is an arrangement of symbols that carry information via meaning, thus concurring in forming what is called the *content* of the document. In the case of text, the symbols are words (or their semantically significant fractions, such as stems, prefixes or suffixes), whereas for images the symbols are colors and textures.

We will thus characterize simple documents as having two parallel dimensions, that of *form* (or *syntax*, or *symbol*) and that of *content* (or *semantics*, or *meaning*). As we have just remarked, the form of a simple document is dependent on the medium that carries the document. On the contrary, we take the content dimension to be medium-independent, as we assume an objective view of meaning: the meaning of a simple document is the set of states of affairs (or "worlds") in which it is true. For instance, the meaning of a piece of text is the set of (spatio-temporally determined) states of affairs in which the assertions made are true, and the meaning of an image is the set of such states of affairs in which the scene portrayed in the image indeed occurs.

Complex documents (or simply documents) are structured sets of simple documents. This leads to the identification of *structure* as the third dimension of our characterization of documents. For the sake of simplicity, we assume the structure of documents to be hierarchical, but this is no serious limitation, as extensions to other structures are well-known and can be included in this framework without any major conceptual shift. Notice that this notion of structure only applies to complex documents and should not be confused with the notion of structure that is embodied in the syntax of simple documents, such as the structure of an image as the particular arrangement of color regions that the image exhibits.

Finally, documents, whether simple or complex, exist as independent entities characterized by (meta-)attributes (often called *metadata* in the recent literature on digital libraries), which describe the relevant properties of such entities. The set of such attributes is usually called the *profile* of a document, and constitutes the fourth (and last) document dimension that our model considers.

Corresponding to the four dimensions of documents just introduced, there can be four categories of retrieval, each one being a projection of the general notion of retrieval defined in Section 2.1 onto a specific dimension. The usefulness of retrieval based on document structure or profile mostly lies in the possibility of using these categories of retrieval *in conjunction* with the other categories, which are discussed in the following section.

2.3 Form- and content-based multimedia information retrieval

The retrieval of information based on form addresses the syntactic properties of documents. In particular, form-based retrieval methods automatically create the document representations to be used in retrieval by extracting low-level features from documents, such as the number of occurrences of a certain word in a text, or the energy level in a certain region of an image. The resulting representations are abstractions which retain that part of the information originally present in the document that is considered sufficient to characterize the document for retrieval purposes. User queries to form-based retrieval engines may be documents themselves (this is especially true in the non-textual case, as this allows to overcome the medium mismatch problem), from which the system builds abstractions analogous to those of documents. Document and query abstractions are then compared by an appropriate function, aiming at assessing their degree of relatedness. A document ranking results from these comparisons, in which the documents with the highest scores occur first.

In the case of text, form-based retrieval includes most of the traditional IR methods, ranging from simple string matching (as used in popular Web search engines) to the classical *tf-idf* term weighting method, to the most sophisticated algorithms for similarity measurement. Some of these methods make use of information structures, such as thesauri, for increasing retrieval effectiveness [78]; however, what makes them form-based retrieval methods is their relying on a fully automatic indexing. In the case of images, form-based retrieval includes all similarity-based image retrieval methods, such as those employed by system like QBIC [31] or Virage [8].

On the contrary, semantic-based retrieval methods rely on symbolic representations of the meaning of documents, that is descriptions formulated in some suitable knowledge representation language, spelling out the truth conditions of the involved document. Various languages have been employed to this end, ranging from net-based to logical. Retrieval by reasoning on the spatial relationships between image objects (see, *e.g.* [40]) falls under this category. For reasons of space this kind of retrieval is not dealt with in this paper. However, integration of spatial retrieval in the present framework can be accomplished as illustrated in [2]. Typically, meaning representations are constructed manually, perhaps with the assistance of some automatic tool; as a consequence, their usage on text in not viable because of the remarkable size (up to millions of documents) that collections of textual documents may reach.

While semantic-based methods explicitly apply when a connection in meaning between documents and queries is sought, the status of form-based methods is, in this sense, ambiguous. On one hand, these methods may be viewed as pattern recognition tools that assist an information seeker by providing associative access to a collection of signals. On the other hand, form-based methods may be viewed as an alternative way to approach the same problem addressed by semantic-based methods, that is deciding relevance, in the sense of connection in meaning, between documents and queries. This latter, much more ambitious view, can be justified only by relying on the assumption that there be a systematic correlation between "sameness" in low-level signal features and "sameness" in meaning. Establishing the systematic correlation between the expressions of a language and their meaning is precisely the goal of a *theory of meaning* (see, *e.g.* [23]), a subject of the philosophy of language that is still controversial, at least as far as the meaning of natural languages is concerned. So, pushed to its extreme consequences, the ambitious view of form-based retrieval leads to viewing a MIR system *as an algorithmic simulation of a theory of meaning*, in force of the fact that the sameness assumption is relied upon in every circumstance, not just in the few, happy cases in which everybody's intuition would bet on its truth. At present, this assumption seems more warranted in the case of text than in the case of non-textual media, as the representations employed by form-based textual retrieval methods (*i.e.* vectors of weighted words) come much closer to a semantic representation than the feature vectors employed by similarity-based image retrieval methods. Anyway, irrespectively of the tenability of the sameness assumption, the identification of the alleged syntactic-semantic correlation is at the moment a remote possibility, so we subscribe to the weaker view of form-based retrieval and present a model where this is just one of the possible ways to access multimedia information.

3 Relation to previous work

A model of structured multimedia documents finalized to retrieval is proposed in [19]. This model has been further developed [67] and subsequently used for building the PRIME information retrieval system [9]. Although subscribing to the logical view of information retrieval, this model is indeed expressed in terms of Sowa's conceptual graphs [87]. This fact makes the comparison with our model hard, due to the more algebraic/operational, as opposed to logical/declarative, nature of conceptual graphs. From a pragmatic point of view, the two models can be considered to share some basic intuitions, notably a categorization of documents into orthogonal dimensions. However, the fully formal status of our model enables us to address fundamental questions such as the computational complexity of the retrieval problem and the relation between form- and content-based retrieval.

Logical models (in a more orthodox sense) of information retrieval have been actively investigated in the last ten years ([22] is a recent source-book on this). Within this research trend, the work closest in spirit to this is the work by Fuhr and colleagues on Probabilistic Datalog (see *e.g.* [34]), an extension of Stratified Datalog by probabilistic features. It is similar in that a logic with a model-theoretic semantics is used as a deductive tool in which to encode independently defined models of IR, but it is different in most other respects. For instance, Fuhr and colleagues do not define independently motivated "mereologies" for the various types of media (see Sections 5.3 and 5.1), and do not cater for the integration, within their formalism, of approaches to MIR originating from different fields (such as those based on digital signal processing), as we instead do.

In the area of image processing, a considerable effort has been devoted to the investigation of effective methods for form-based image retrieval[1]. This effort has led to the development of a number of systems (see *e.g.* [37, 66, 85]) which, in some cases, have also been turned into commercial products [8, 31]. These systems, and in general the methods found in the literature, differ in the aspect of image form that is considered, in the features that are used to capture each aspect, in the algorithms employed to compute features, in the function adopted to match features for assessing similarity. While an exhaustive review is outside the scope of this paper, a general account of form-based image retrieval is given later, in Section 5.2, when we will consider the representation and retrieval of image forms. A survey of current trends can be found in [42]. Given the context in which [42] originated, it is not surprising that its unifying trait is the lack of a proper representation and use of (what *we* call) the content of images.

Indexing methods based on statistical [75] or probabilistic [33] methods may be viewed as attempts to capture (perhaps shallow) semantic properties of the document, as they abstract from supposedly useless properties of a word (*e.g.* its contexts of occurrence in the document) to concentrate on properties of it that are deemed more significant from a semantic point of view (*e.g.* the number of times it occurs in a document).

More daring approaches to capturing more document semantics are those based on natural language processing (NLP – see *e.g.* [82] for a discussion of these approaches). So far, no real attempt has been made to capture document semantics through NLP-based *understanding* of the document. This is unsurprising, given the substantial distance that still separates NLP from achieving real text "understanding". More surprisingly, also less ambitious NLP-based efforts to capture document semantics have failed to yield improved retrieval effectiveness. The paradigmatic case is the attempt to index documents by noun phrases rather than by individual words; although intuitively capturing "more semantics" than the latter, the former have been experimentally shown to be less effective indexing units, perhaps because of their different statistical properties (see *e.g.* [48]).

Methods based on either statistical or probabilistic indexing have been applied to the retrieval of images via textual annotations (see *e.g.* [50, 90, 83]), in some cases supported by the use of thesauri to semantically connect the terms occurring in the text [3, 41, 69]. The resulting methods have proved effective only in very narrow contexts, and do not fully exploit the capabilities of human memory and the potentiality of the visual language in supporting such capabilities.

Image retrieval methods based on both textual annotation and visual similarity have also been investigated as a way of enhancing retrieval performance and system usability [86]. While very naive in the representation of image semantics, the resulting systems "sell" form-based text retrieval for retrieval based on image semantics. As a consequence, they face the problem of how to combine the results of two sources of imprecision each addressing the same aspect, *i.e.* document form, in a different way.

[1] The kind of image retrieval that we call "form-based" to contrast it with the retrieval based on the semantics of images, has been called "content-based retrieval" by the image processing community, to contrast it with the retrieval based on externally attributed properties of images ("metadata"). This terminological mismatch is a typical inconvenient of the integration between different worlds. We have decided to live with it in order to preserve the connotation underlying our reference model, as discussed in Section 2.2.

Models and methods for MIR developed in the database community tend to focus exclusively on the symbolic representation and usage of the content of *e.g.* images, regarding form-based retrieval just as a challenging application area for fast access methods to secondary storage [30]. In the classification of multimedia systems outlined in [39], this would fall under the category of retrieval termed as "based on logical data units", where a logical data unit is a multimedia data structure determined a priori. A paradigmatic case is the model presented in [52], where visual aspects of images are treated at the symbolic level as semantic properties, and visual similarity is not provided by the model's query language. Incidentally, this view of MIR has been pursued also by relying on a description logic as modelling tool [35].

As already argued, our view of MIR is that the requirements of applications do not mirror the partition of the field that has been induced in practice by the different backgrounds of researchers. The application areas that have been mentioned at the beginning of this paper do require an information retrieval functionality able to address all the document dimensions and, most importantly, able to address each dimension *in its own modality*. In order to fulfill this goal, integration is the key concept, and, indeed, the basis of our approach. In this sense, our model can be seen as a generalization of current MIR methods. This does not mean that every functionality found in any text or image retrieval model/system is also provided by the model being presented. Rather, it means that our model provides the basis for integrating retrieval methods pertaining to different media and document dimensions. In so doing, it relies on a standard set of functionalities for the various kinds of retrieval considered. It will be shown in due course that these functionalities can be made significantly more sophisticated without altering the architecture of the model.

This model is the result of a research effort spanning almost a decade. The requirements of a suitable MIR model were outlined in [55], also based on the insights gained through the MULTOS Project [97]. A first formulation of this model based on a description logic was given in [59]. Starting from that formulation, two parallel streams of development have been undertaken. On the one hand, we have worked on the tool, aiming at the definition of a description logic more tightly coupled with the task of information retrieval [18, 58, 60, 80, 91, 92, 93, 94]. On the other hand, we have worked on the application of this tool to image retrieval [54, 57], and successively generalized the model to MIR [56]. In order to simplify the presentation, the present paper does not include the full-blown logical tool resulting from the former stream of research, but rather focuses on the results of the latter stream. A preliminary version of this model can be found in [56].

4 A fuzzy description logic

Description Logics (DLs, for short – see *e.g.* [12, 27]) are contractions of the predicate calculus that descend from the formalization of early semantic network- or frame-based knowledge representation languages. DLs have an "object-oriented" character that makes them especially suitable for reasoning about hierarchies of structured objects.

DL systems have been used for building a variety of applications including systems supporting configuration management [53], software management [26], browsing and querying of networked information sources [29], data archaeology [16], plan recognition [100], natural language understanding [11] and multimedia data description and classification [35]. The grandfather of DL systems was KL-ONE [15]. Nowadays, a whole family of DL-based knowledge representation systems has been built, like BACK [68], CLASSIC [14], KRIS [7], and LOOM [51]. For most of these systems, the computational complexity of the underlying reasoning problems is known. The systems mostly differ for the expressiveness of the language and the completeness of the inferential algorithms.

The specific DL that we use to express our model is $\mathcal{ALC}$ [79]. $\mathcal{ALC}$ is universally considered a significant representative of a family of expressive DLs and is therefore regarded as a convenient

$$\begin{array}{lll}\langle concept\rangle ::= & \top \mid & \texttt{\% top concept} \\ & \bot \mid & \texttt{\% bottom concept} \\ & \langle primitive\text{-}concept\rangle \mid & \texttt{\% primitive concept} \\ & \langle concept\rangle \sqcap \langle concept\rangle \mid & \texttt{\% concept conjunction} \\ & \langle concept\rangle \sqcup \langle concept\rangle \mid & \texttt{\% concept disjunction} \\ & \neg\langle concept\rangle \mid & \texttt{\% concept negation} \\ & \forall\langle role\rangle.\langle concept\rangle \mid & \texttt{\% universal quantification} \\ & \exists\langle role\rangle.\langle concept\rangle & \texttt{\% existential quantification}\end{array}$$

Figure 1: Grammar rules for $\mathcal{ALC}$ concepts

workbench for carrying out any kind of logical work of an experimental nature. However, we stress that our model is not tied in any way to this particular choice. Indeed, most implemeted systems rely on DLs that are less expressive than $\mathcal{ALC}$. On the other hand, IR models based on more expressive DLs have been also studied [59].

4.1 A quick look at $\mathcal{ALC}$

Concepts, roles and *individual constants* are the basic building blocks of $\mathcal{ALC}$. Concepts describe sets of objects, such as "Italian musician", or, in $\mathcal{ALC}$ notation, $\mathsf{Italian}\sqcap\mathsf{Musician}$. Roles give binary properties, such as Friend. Individual constants are simple names for individuals, such as tim. From a data modelling point of view, concepts correspond to classes, roles to attributes and individual constants to basic objects. From a logical point of view, concepts can be seen as (possibly complex) unary predicate symbols obtained by lambda-abstraction, roles as binary predicate symbols and individual constants as constant symbols.

Formally, we assume three alphabets of symbols, called *primitive concepts, primitive roles* and *individual constants.* The *concepts* of the language $\mathcal{ALC}$ are formed out of primitive concepts according to the syntax rules given in Figure 1. In $\mathcal{ALC}$ roles are always primitive. Other DLs have instead also role-forming operators.

For example, the complex concept $\mathsf{Musician}\sqcap\forall\mathsf{Plays}.\neg\mathsf{ElectricInstrument}$ is obtained by combining the primitive concepts $\mathsf{Musician}$ and $\mathsf{ElectricInstrument}$ and the primitive role Plays by the conjunction ($\sqcap$), universal quantification ($\forall$) and negation ($\neg$) constructors. Under the intended interpretation and in a way that will be formalized soon, such concept denotes the musicians who do not play any electric instrument.

It is immediate to verify that $\mathcal{ALC}$ is a notational variant of a (conservative) contraction of predicate calculus, determined by the very limited usage of quantifiers and variables, the latter always implicit[2]. Accordingly, the semantics of DLs is the restriction of the Tarskian semantics for the predicate calculus corresponding to this syntactical contraction. An *interpretation* $\mathcal{I}$ is a pair $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ consisting of a non-empty set $\Delta^{\mathcal{I}}$ (called the *domain*) and of an *interpretation function* $\cdot^{\mathcal{I}}$. Following the intuitive meaning of constants, roles and concepts, given at the beginning of this Section, $\cdot^{\mathcal{I}}$ maps different individual constants into different elements of $\Delta^{\mathcal{I}}$, primitive concepts into subsets of $\Delta^{\mathcal{I}}$ and primitive roles into subsets of $\Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$. The interpretation of complex concepts

[2] A calculus C is a *contraction* of a calculus C' when the language L and the set of valid formulas V of the former are, respectively, subsets of the language L' and of the set of valid formulas V' of the latter. The contraction is *conservative* when $(V' - V)$ does not contain any formula which is expressible solely by means of L.

is defined by structural induction via the following rules, where C, C_1, C_2 stand for concepts and R for roles:

$$
\begin{aligned}
\top^{\mathcal{I}} &= \Delta^{\mathcal{I}} \\
\bot^{\mathcal{I}} &= \emptyset \\
(C_1 \sqcap C_2)^{\mathcal{I}} &= C_1^{\mathcal{I}} \cap C_2^{\mathcal{I}} \\
(C_1 \sqcup C_2)^{\mathcal{I}} &= C_1^{\mathcal{I}} \cup C_2^{\mathcal{I}} \\
(\neg C)^{\mathcal{I}} &= \Delta^{\mathcal{I}} \setminus C^{\mathcal{I}} \\
(\forall R.C)^{\mathcal{I}} &= \{d \in \Delta^{\mathcal{I}} \mid \forall d' \in \Delta^{\mathcal{I}}.(d,d') \in R^{\mathcal{I}} \text{ implies } d' \in C^{\mathcal{I}}\} \\
(\exists R.C)^{\mathcal{I}} &= \{d \in \Delta^{\mathcal{I}} \mid \exists d' \in \Delta^{\mathcal{I}}.(d,d') \in R^{\mathcal{I}} \text{ and } d' \in C^{\mathcal{I}}\}.
\end{aligned}
$$

For instance, $(\forall R.C)^{\mathcal{I}}$ is the result of viewing $\forall R.C$ as the first order formula $\forall y.R(x,y) \to C(y)$. Similarly, $(\exists R.C)^{\mathcal{I}}(d)$ is the result of viewing $\exists R.C$ as $\exists y.R(x,y) \wedge C(y)$. As a consequence, it can be verified that for all interpretations $\mathcal{I}$

$$
\begin{gathered}
(\neg(C_1 \sqcap C_2))^{\mathcal{I}} = (\neg C_1 \sqcup \neg C_2)^{\mathcal{I}} \\
(\exists R.C)^{\mathcal{I}} = (\neg(\forall R.\neg C))^{\mathcal{I}}.
\end{gathered}
$$

$\mathcal{ALC}$ concepts and roles can be used for making *crisp assertions* about individual constants (metavariables a, a_1, a_2), *i.e.* expressions belonging to one of the following categories:

1. $C(a)$, asserting that a is an instance of C; for example, $(\mathsf{Musician} \sqcap \mathsf{Teacher})(\mathsf{tim})$ makes the individual constant tim a Musician and a Teacher;

2. $R(a_1, a_2)$, asserting that a_1 is related to a_2 by means of R (*e.g.* $\mathsf{Friend}(\mathsf{tim},\mathsf{tom})$);

3. $C_1 \sqsubseteq C_2$, asserting that C_1 is more specific than C_2 (*e.g.* $\mathsf{Pianist} \sqsubseteq (\mathsf{Artist} \sqcap \exists\mathsf{Plays}.\mathsf{Piano})$).

Assertions of type 1 and 2 are called *simple assertions*, and have identical analogues in the predicate calculus. An assertion of type 3 is called an *axiom*, and its predicate calculus analogue is the sentence $\forall x.C_1(x) \to C_2(x)$. By stating both $C_1 \sqsubseteq C_2$ and $C_2 \sqsubseteq C_1$, the primitive concept C_1 is defined to be equivalent to C_2.

Semantically, the assertion $C(a)$ (resp. $R(a,b)$ and $C_1 \sqsubseteq C_2$) is *satisfied* by $\mathcal{I}$ iff $a^{\mathcal{I}} \in C^{\mathcal{I}}$ (resp. $(a^{\mathcal{I}}, b^{\mathcal{I}}) \in R^{\mathcal{I}}$ and ${C_1}^{\mathcal{I}} \subseteq {C_2}^{\mathcal{I}}$). A set Σ of assertions will be called a *knowledge base* (KB). An interpretation $\mathcal{I}$ *satisfies* (*is a model of*) a KB Σ iff $\mathcal{I}$ satisfies each element in Σ. A KB Σ *entails* an assertion α (written $\Sigma \models \alpha$) iff every model of Σ also satisfies α. For instance, if Σ is:

$$
\Sigma \quad = \quad \{\mathsf{Italian} \sqsubseteq \mathsf{European}, (\mathsf{Italian} \sqcap \mathsf{Pianist})(\mathsf{tom}), \mathsf{Friend}(\mathsf{tim}, \mathsf{tom})\}
$$

then

$$
\Sigma \models (\exists\mathsf{Friend}.(\mathsf{Pianist} \sqcap \mathsf{European}))(\mathsf{tim}),
$$

i.e. tim has a friend who is a European pianist.

4.2 Fuzzy $\mathcal{ALC}$

In order to deal with the imprecision inherent in information retrieval, we extend $\mathcal{ALC}$ with fuzzy capabilities [94]. The extension of DLs to this end is not new. Yen [102] was the first to introduce imprecision into a simple DL; the resulting language has interesting features: it allows the definition of imprecise concepts by means of explicit membership functions over a domain, and it introduces *concept modifiers*, like Very or $\mathsf{Slightly}$, by means of which concepts like "very low pressure" can be defined. This last idea has been generalized to $\mathcal{ALC}$ in [98], where a certain type of concept modifiers are allowed. The result is a more expressive language than just fuzzy $\mathcal{ALC}$, with radically different computational properties, though.

From a syntactical point of view, *fuzzy* $\mathcal{ALC}$ provides *fuzzy assertions*, *i.e.* expressions of type $\langle \alpha, n\rangle$, where α is a crisp assertion and $n \in [0,1]$. We will use the terms *fuzzy simple assertion*, *fuzzy axiom* and *fuzzy KB* with the obvious meaning. Then, $\langle \exists\mathsf{About.Piano}(\mathsf{i}), .7\rangle$ is a fuzzy simple assertion with intended meaning (see below) "the membership degree of individual constant i to concept $\exists\mathsf{About.Piano}$ is at least .7", or, in less technical terms, "i is likely to be about a piano."

According to Zadeh [103], a *fuzzy set* X with respect to a set S is characterized by a membership function $\mu_X : S \to [0,1]$, assigning a X-membership degree, $\mu_X(s)$, to each element s in S. This membership degree gives us an estimation of how much s belongs to X. Typically, if $\mu_X(s) = 1$ then s definitely belongs to X, while $\mu_X(x) = .7$ means that s is "likely" (with degree of likelihood .7) to be an element of X. Membership functions have to satisfy the following three restrictions (for all $s \in S$ and for all fuzzy sets X, Y with respect to S):

$$\begin{aligned}\mu_{X\cap Y}(s) &= \min\{\mu_X(s), \mu_Y(s)\}\\ \mu_{X\cup Y}(s) &= \max\{\mu_X(s), \mu_Y(s)\}\\ \mu_{\overline{X}}(s) &= 1 - \mu_X(s)\end{aligned}$$

where $\overline{X}$ is the complement of X in S, *i.e.* $S \setminus X$. Other membership functions have been proposed in the literature (the interested reader can consult *e.g.* [28, 46]).

In fuzzy $\mathcal{ALC}$, concepts and roles are interpreted as fuzzy sets, thus becoming *imprecise*. Formally, a *fuzzy interpretation* is a pair $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$, where $\Delta^{\mathcal{I}}$ is, as for the crisp case, the *domain*, whereas $\cdot^{\mathcal{I}}$ is an *interpretation function* mapping:

1. different individual constants to different elements of $\Delta^{\mathcal{I}}$, as for the crisp case:
2. $\mathcal{ALC}$ concepts into a membership degree function $\Delta^{\mathcal{I}} \to [0,1]$, and
3. $\mathcal{ALC}$ roles into a membership degree function $\Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}} \to [0,1]$.

Therefore, if C is a concept then $C^{\mathcal{I}}$ will be interpreted as the *membership degree function* of the fuzzy set which is denoted by C w.r.t. $\mathcal{I}$, *i.e.* if d is an object of the domain $\Delta^{\mathcal{I}}$ then $C^{\mathcal{I}}(d)$ gives us the degree of membership of d in the denotation of C under the interpretation $\mathcal{I}$. Similarly for roles.

In order to reflect intuition, $\cdot^{\mathcal{I}}$ has to satisfy the following equations (for all $d \in \Delta^{\mathcal{I}}$):

$$\begin{array}{lcl}
\top^{\mathcal{I}}(d) & = & 1 \\
\bot^{\mathcal{I}}(d) & = & 0 \\
(C_1 \sqcap C_2)^{\mathcal{I}}(d) & = & \min\{{C_1}^{\mathcal{I}}(d), {C_2}^{\mathcal{I}}(d)\} \\
(C_1 \sqcup C_2)^{\mathcal{I}}(d) & = & \max\{{C_1}^{\mathcal{I}}(d), {C_2}^{\mathcal{I}}(d)\} \\
(\neg C)^{\mathcal{I}}(d) & = & 1 - C^{\mathcal{I}}(d) \\
(\forall R.C)^{\mathcal{I}}(d) & = & \min_{d' \in \Delta^{\mathcal{I}}}\{\max\{1 - R^{\mathcal{I}}(d,d'), C^{\mathcal{I}}(d')\}\} \\
(\exists R.C)^{\mathcal{I}}(d) & = & \max_{d' \in \Delta^{\mathcal{I}}}\{\min\{R^{\mathcal{I}}(d,d'), C^{\mathcal{I}}(d')\}\}.
\end{array}$$

Again, $(\forall R.C)^{\mathcal{I}}(d)$ is the result of viewing $\forall R.C$ as the first order formula $\forall y.R(x,y) \rightarrow C(y)$, where $F \rightarrow G$ is $\neg F \vee G$ and the universal quantifier $\forall$ is viewed as a possibly infinite conjunction over the elements of the domain (in the literature, several different definitions of the fuzzy implication connective $\rightarrow$ have been proposed–see *e.g.* [46] for a discussion). The overall effect is, informally:

$$(\forall R.C)(d) \equiv \bigwedge_{d' \in \Delta^{\mathcal{I}}} \neg R(d,d') \vee C(d') \equiv \bigwedge_{d' \in \Delta^{\mathcal{I}}} \bigvee\{\neg R(d,d'), C(d')\}.$$

By applying (in this order) the rules for conjunction, disjunction and negation to the last formula, the min-max expression above yields. Similarly, $(\exists R.C)^{\mathcal{I}}(d)$ is the result of viewing $\exists R.C$ as $\exists y.R(x,y) \wedge C(y)$, where the existential quantifier $\exists$ is considered as a possibly infinite disjunction over the elements of the domain (see also [47]).

Also in this case, it is verified that for all interpretations $\mathcal{I}$ and individual constants $d \in \Delta^{\mathcal{I}}$, $(\neg(C_1 \sqcap C_2))^{\mathcal{I}}(d) = (\neg C_1 \sqcup \neg C_2)^{\mathcal{I}}(d)$ and $(\neg(\forall R.C))^{\mathcal{I}}(d) = (\exists R.\neg C)^{\mathcal{I}}(d)$. Moreover, every "crisp" interpretation $\mathcal{I}$ can be seen as a fuzzy interpretation with membership degree in $\{0,1\}$ rather than in $[0,1]$.

An important operator of DLs concerns number restrictions on role fillers [17]. The interested reader can find the fuzzy semantic rules for number restriction operators in [102].

An interpretation $\mathcal{I}$ *satisfies* (*is a model of*) a fuzzy assertion $\langle C(a), n\rangle$ (resp. $\langle R(a_1,a_2), n\rangle$ and $\langle C_1 \sqsubseteq C_2, n\rangle$) iff $C^{\mathcal{I}}({a_1}^{\mathcal{I}}) \geq n$ (resp. $R^{\mathcal{I}}({a_1}^{\mathcal{I}}, {a_2}^{\mathcal{I}}) \geq n$ and $\min_{d \in \Delta^{\mathcal{I}}}\{\max\{1 - {C_1}^{\mathcal{I}}(d), {C_2}^{\mathcal{I}}(d)\}\} \geq n$). The satisfiability condition $\min_{d \in \Delta^{\mathcal{I}}}\{\max\{1 - {C_1}^{\mathcal{I}}(d), {C_2}^{\mathcal{I}}(d)\}\} \geq n$ for $\langle C_1 \sqsubseteq C_2, n\rangle$ is a consequence of viewing $C_1 \sqsubseteq C_2$ as the formula $\forall x.C_1(x) \rightarrow C_2(x)$, or $\forall x.\neg C_1(x) \vee C_2(x)$.

An interpretation $\mathcal{I}$ *satisfies* (is *a model of*) a fuzzy KB Σ iff $\mathcal{I}$ satisfies each element of Σ. A fuzzy KB Σ *entails* a fuzzy assertion γ (written $\Sigma \models \gamma$) iff every model of Σ also satisfies γ. Given a fuzzy KB Σ and a fuzzy assertion α, we define the *maximal degree of truth* of α with respect to Σ (written $Maxdeg(\Sigma, \alpha)$) to be $\max\{n > 0 \mid \Sigma \models \langle \alpha, n\rangle\}$ ($\max \emptyset = 0$). Notice that $\Sigma \models \langle \alpha, n\rangle$ iff $Maxdeg(\Sigma, \alpha) \geq n$.

For example, suppose we have two images i1 and i2 indexed as follows:

$$\begin{array}{lcl}
\Sigma_{\mathsf{i1}} & = & \{\langle \mathsf{About(i1,tim)}, .9\rangle, \langle \mathsf{Tall(tim)}, .8\rangle, \langle \mathsf{About(i1,tom)}, .6\rangle, \langle \mathsf{Tall(tom)}, .7\rangle\} \\
\Sigma_{\mathsf{i2}} & = & \{\langle \mathsf{About(i2,joe)}, .6\rangle, \langle \mathsf{Tall(joe)}, .9\rangle\}.
\end{array}$$

Moreover, let the background KB be

$$\begin{array}{lcl}
\Sigma_{\mathsf{B}} & = & \{\langle \mathsf{Image(i1)}, 1\rangle, \langle \mathsf{Image(i2)}, 1\rangle, \\
 & & \langle \mathsf{Musician(tim)}, 1\rangle, \langle \mathsf{Musician(tom)}, 1\rangle, \langle \mathsf{Musician(joe)}, 1\rangle, \\
 & & \langle \mathsf{Tall} \sqsubseteq \mathsf{Adult}, .9\rangle\},
\end{array}$$

and let

$$\begin{aligned}\Sigma_1 &= \Sigma_{\mathsf{i1}} \cup \Sigma_{\mathsf{B}},\\ \Sigma_2 &= \Sigma_{\mathsf{i2}} \cup \Sigma_{\mathsf{B}}\end{aligned}$$

Our intention is to retrieve all images in which there is an adult musician. This can be formalized by means of the query concept

$$C = \mathsf{Image} \sqcap \exists \mathsf{About}.(\mathsf{Adult} \sqcap \mathsf{Musician}).$$

It can be verified that $Maxdeg(\Sigma_1, C(\mathsf{i1})) = .9$, whereas $Maxdeg(\Sigma_1, C(\mathsf{i2})) = .6$. Therefore, in retrieving both images we will rank i1 before i2.

The pivotal role that fuzzy $\mathcal{ALC}$ has in the context of our model will become clear in the next sections. The connection between logical reasoning in fuzzy $\mathcal{ALC}$ and non-logical computation through medium-specific document processing techniques will be realized by identifying a number of *special* $\mathcal{ALC}$ *individual constants and predicate symbols* and imposing that their semantics be not a *generic* subset of $\Delta^{\mathcal{I}}$ (or $\Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$) but one that complies with the results of the document processing analysis.

5 Form

We now proceed to discussing the "form" dimension of simple documents. We present models for *image layouts* and *text layouts*, which consist of the symbolic representations of the form-related aspects of an image or text, respectively. Each notion is endowed with a *mereology*, *i.e.* a theory of parts, based on notions such as *atomic region*, *region* and *grounded region.* Each of these three notions will be defined *twice*, once for images and once for text. The context will tell which notion is meant from time to time. Note also that the term "layout" is used elsewhere in the literature in a different sense, namely to denote the rendering of a document on a display device. In order to query image and text models, we introduce special predicate symbols, which will be used in the unified query language discussed in Section 8. The reader will note the evident parallelism, even down to many details, in our treatment of image form and text form; given that form is the only medium-specific aspect of documents, this shows the potential of this model for extension to other media.

5.1 Modelling image layouts

In order to make the paper self-contained, some elementary notions from digital geometry are briefly recalled below (see also *e.g.* [72, Chapter 11]).

Let $\mathbb{N}$ be the set of natural numbers. A *zone* is any subset of $\mathbb{N}^2$, *i.e.* a set of *points.* A zone S is *aligned* if

$$min\{x \mid \langle x, y\rangle \in S\} = 0 \text{ and } min\{y \mid \langle x, y\rangle \in S\} = 0.$$

The *neighbors* of a point $P = \langle x, y\rangle$, when both x and y are non-zero, are the points $\langle x-1, y\rangle$, $\langle x, y-1\rangle$, $\langle x, y+1\rangle$, and $\langle x+1, y\rangle$. If only one of P's coordinates is 0, then P has only 3 neighbors; $\langle 0, 0\rangle$ has only two neighbors. Two zones are said to be *neighbor* to each other if they are disjoint and a point in one of them is a neighbor of a point in the other one. A *path* of length n from point P to point P' is a sequence of points $P = P_0, P_1, \ldots, P_n = P'$ such that $P_0 = P$, $P_n = P'$ and P_i is a neighbor of P_{i-1}, $1 \leq i \leq n$. Let S be a zone and P and P' points of S: P is *connected* to P' in S if there is a path from P to P' consisting entirely of points of S. For any P in S, the set of points

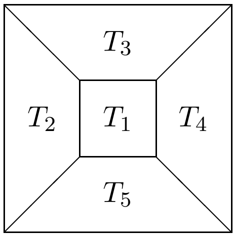


Figure 2: The atomic regions of a simple image

that are connected to P in S is called a *connected component* of S. If S has only one connected component, it is called a *connected* zone.

We now formalize the notion of an image layout. Given a set of *colors* $\mathcal{C}$, an *image layout* is a triple $i = \langle A^i, \pi^i, f^i \rangle$, where:

- A^i, the *domain*, is a finite, aligned, rectangular zone;
- π^i is a partition of A^i into non-empty connected zones $\{T_1, ..., T_n\}$, called *atomic regions*;
- f^i is a total function (the *color function*) from π^i to $\mathcal{C}$ assigning a color to each atomic region in such a way that no two neighbor atomic regions have the same color.

Figure 2 shows the partition π underlying a simple image layout. From an image processing point of view, an image layout is a segmentation, based on a color uniformity criterion. As a consequence, in our model "atomic region" is synonymous to "color region". The reason behind this choice is that the mereology induced by this criterion permits us:

- to model a widely used class of queries on image form, namely those addressing single color regions as well as shapes;
- to model regions corresponding to objects within images, as these regions can be seen as aggregations of color regions.

This choice, on the other hand, has no impact on global image similarity, such as color- or texture-based similarity, because these kinds of similarity are not expressed nor computed in terms of single image regions. Also, it is important to notice that other image segmentation criteria, and the consequent mereologies, can be accommodated into the model by generalizing the definition of image layout in the following way: $i = \langle A^i, \langle \pi^i_1, f^i_1 \rangle, \langle \pi^i_2, f^i_2 \rangle, \ldots, \langle \pi^i_n, f^i_n \rangle \rangle$, where each pair $\langle \pi^i_j, f^i_j \rangle$ captures a different segmentation criterion. To keep the model simple, and the paper readable, we consider only one segmentation criterion.

The calculation of the partition π^i from a pixel representation is notoriously difficult and, in fact, still an open problem. For this reason, we next provide a more general notion of image region, which is defined on top of the notion of atomic region, but which could as well be assumed as primitive, thus reconciling the model with the current status of image processing.

The *regions* of an image layout $i = \langle A^i, \pi^i, f^i \rangle$ are defined to be the set:

$$\pi^i_e = \{S \mid \exists\, T_1, ..., T_k \in \pi^i, k \geq\ 1,\ S = \cup^k_{j=1} T_j, S \text{ connected}\}.$$

A region is a connected zone obtained by the union of one or more atomic regions. The fact that we allow S to have holes enables the model to deal with partial occlusion (*e.g.* the area of an image showing a goal-keeper partly covered by an approaching ball counts as a region). Accordingly, the *extended color function* of an image layout $i = \langle A^i, \pi^i, f^i \rangle$ is defined as the function f_e^i that assigns to each region S the *color distribution* induced by the atomic regions that make up S. Technically, if $S = \cup_{i=1}^{k} T_i$, $f_e^i(S)$ is a mapping from $\mathcal{C}$ to [0,1] defined as follows (for all $c \in \mathcal{C}$ and letting $|A|$ be the cardinality of A):

$$f_e^i(S)(c) = \frac{\sum_{T \in Z_c} |T|}{|S|}, \text{ where } Z_c = \{T \in \{T_1, \ldots, T_k\} \mid f^i(T) = c\}.$$

Since Z_c is the subset of the atomic regions making up S that have color c, $f_e^i(S)(c)$ gives the percentage of S that has color c. In fact, it is immediately verified that:

$$\sum_{c \in \mathcal{C}} f_e^i(S)(c) = 1, \text{ for all regions } S.$$

For any given region S, let $\phi(S)$ stand for the *shape* of the region, that is the closed curve that delimits S. We will let $\mathcal{S}$ be the set of closed curves in $\mathbb{N}^2$.

By definition, a region may occur in more than one image, since it is defined in a purely extensional way. For instance, whenever an object shows up in two images in exactly the same way, those two images will share at least one region, namely the portion of the image domain containing the object. In general, the same region will occur in all images having at least one atomic region in common. In order to evaluate image queries, though, we need a more selective notion of region, bound to the specific image where a region belong. To this end, we introduce the notion of *grounded image region*, which we define as a pair $\langle i, S \rangle$ where $i = \langle A^i, \pi^i, f^i \rangle$ is an image layout and $S \in \pi_e^i$. For simplicity of notation, in what follows we will directly refer to the shape of a grounded image region $\langle i, S \rangle$, actually meaning the shape of its component region S. Formally, we extend the function ϕ by stipulating that:

$$\phi(\langle i, S \rangle) = \phi(S).$$

Analogously, we define the function f_e on grounded image regions $\langle i, S \rangle$ as follows:

$$f_e(\langle i, S \rangle)(c) = f_e^i(S)(c).$$

Finally, we define *the image universe* $\mathcal{IU}$ as the set of all possible image layouts of any domain. The set of all grounded image regions, denoted as $\mathcal{R}$, is defined on top of the image universe as:

$$\mathcal{R} = \{\langle i, S \rangle \in (\mathcal{IU} \times 2^{\mathbb{N}^2}) \mid S \in \pi_e^i\}.$$

5.2 Querying image layouts

Queries referring to the form dimension of images are called *visual* queries, and can be partitioned as follows (for a taxonomy of approaches to image retrieval, see *e.g.* [38]):

1. *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; the addressed aspect of similarity may concern color [8, 31, 95, 96], texture [49, 69, 84], appearance [71]: combinations of these are gaining ground [73, 20];

2. *abstract visual queries:* these are artificially constructed image elements (hence, "abstractions" of image layouts) that address specific aspects of image similarity; they can be further categorized into:

(a) *color queries:* specifications of color patches, used to indicate a request to retrieve those images in which a similar color patch occurs [24, 31];

(b) *shape queries:* specifications of one or more shapes (closed simple curves in the 2D space), used to indicate a request to retrieve those images in which the specified shapes occur as contours of significant objects [25, 43, 70];

(c) combinations of the above [44].

As mentioned in Section 2.3, visual queries are processed by matching a vector of features extracted from the query image, with each of the homologous vectors extracted from the images candidate for retrieval. For concrete visual queries, the features are computed on the whole image, while for abstract visual queries only the features indicated in the query (such as shape or color) are represented in the vectors involved. For each of the above categories of visual queries, a number of different techniques have been proposed for performing image matching, depending on the features used to capture the aspect addressed by the category, or the method used to compute such features, or the function used to assess similarity. For instance, a color similarity criterion for concrete visual query is captured by the following function [95]:

$$s(i,i') = \sum_{j \in HSB} w_{1j}|m^i_{1j} - m^{i'}_{1j}| + w_{2j}|m^i_{2j} - m^{i'}_{2j}| + w_{3j}|m^i_{3j} - m^{i'}_{3j}| \tag{1}$$

where:

- HSB is the set of color channels, $HSB = \{H, S, B\}$;
- m^g_{kj} is the k-th moment of the true color histogram of image layout g, for the channel j of the HBS color space, that is:

  $$m^g_{1j} = \frac{1}{N}\sum_{l=1}^{N} p^g_{j,l} \qquad m^g_{2j} = \sqrt{\frac{1}{N}\sum_{l=1}^{N}(p^g_{j,l} - m^g_{1j})^2} \qquad m^g_{3j} = \sqrt[3]{\frac{1}{N}\sum_{l=1}^{N}(p^g_{j,l} - m^g_{1j})^3}$$

  where $p^g_{j,l}$ is the value of the channel j in the point l of the image layout g;
- $w_{ij} \geq 0$, $(1 \leq j, i \leq 3)$ are user specified weights.

As the last example shows, the inference carried out by a similarity retrieval engine is heavily based on numerical techniques, hence the least apt to be captured by logic. For this reason, the model does not provide the machinery for defining similarity functions, and indeed assumes that similarity reasoning will *not* be performed by means of logic.

However, against current practice, *we argue that visual queries are expressions of a formal language*, and that logic is a most suitable tool for the specification of such a language. Of course, the primitive elements of the language for querying image forms we are about to introduce, will be specifiable in a visual way, being visual in nature. So, at the surface level, any system based on this model can be made to look identical to the similarity retrieval engines in current practice nowadays. But the existence of the query language marks an important difference, since it permits, among other things, the integration of form-based image retrieval with other kinds of retrieval, on images as well as on documents pertaining to any other medium.

The categorization of visual queries pointed out above, will be used as the guideline for the definition of the query language on image layouts. The building blocks of the query language are, as anticipated at the end of Section 4.2, *special $\mathcal{ALC}$ individual constants and predicate symbols* (abbreviated in SICs and SPSs, respectively). In particular, in order to express visual queries, two kinds of symbols are called for:

1. symbols for denoting image layouts, their component regions, the properties of regions (such as colors and shapes), and the appropriate relationships among these entities; we will call these symbols *mereological* SICs and SPSs;
2. symbols for denoting similarity between whole images (for concrete visual queries) and image components (for abstract visual queries); we will call these symbols *similarity* SPSs.

As far as mereological SICs are concerned, the following disjoint countable alphabets are introduced, each consisting of description logic individual constants:

- Ω_I, the names of image layouts (metavariable i, optionally followed by a natural number);
- Ω_R, the names of grounded image regions (r);
- Ω_C, the names of colors (c);
- Ω_S, the names of shapes (s).

Even though at a first sight the above alphabets may look as an unnecessary complication imposed by the formalism in which this model is stated, it is worth to observe that they are indeed of everyday use. For instance, one could think of Ω_I as the set of image URLs. Ω_C can be thought of as naming the elements of one of the many color spaces proposed in the literature; for instance, in the RGB space each such name is a triple giving the level of energy of a pixel on the corresponding channel. Analogously, Ω_S may be understood as any suitable notation for representing contours, such as, for instance, the 8-contour notation, given by elements of the set $\{0, 1, \ldots, 7\}^+$. Finally, each element in Ω_R could consist of the composition of an image name, a region shape and a point for locating the shape within the image range, thereby uniquely identifying a region of the image. Formally, the intended semantics of these SICs is given by conditions which constraint the generic fuzzy interpretation $\mathcal{I}$ to use a specific function to interpret each of them. In particular, we will treat each of the above introduced individual constants as a *rigid designator* (*i.e.* interpretation-independent name, compliant with the given intuitive account of the alphabets) for a corresponding semantic entity. These conditions require $\cdot^{\mathcal{I}}$ to be a total bijective mapping from:

$$\Omega_I \text{ to } \mathcal{IU} \tag{2}$$

$$\Omega_R \text{ to } \mathcal{R} \tag{3}$$

$$\Omega_C \text{ to } \mathcal{C} \tag{4}$$

$$\Omega_S \text{ to } \mathcal{S}. \tag{5}$$

In this way, for instance, each image layout has a name in the language, and each name in the language names a different layout.

Furthermore, the following mereological SPSs are assumed, each having the syntactical status of a description logic role:

- HAIR(i,r) (standing for <u>H</u>as <u>A</u>tomic <u>I</u>mage <u>R</u>egion): relates the image layout i to one of its grounded atomic regions r;
- HIR(i,r) (<u>H</u>as <u>I</u>mage <u>R</u>egion): relates the image layout i to one of its grounded regions r;
- HS(r,s) (<u>H</u>as <u>S</u>hape): relates a grounded image region r to its shape s;
- HC(r,c) (<u>H</u>as <u>C</u>olor): relates a grounded image region r to its color c.

The semantic conditions on mereological SPSs are as follows (for all image layouts i, i', regions S, shapes s and colors c):

$$\mathsf{HAIR}^{\mathcal{I}}: \quad \mathcal{IU} \times (\mathcal{IU} \times 2^{\mathbb{N}^2}) \to \{0,1\}, \text{ s.t. } \mathsf{HAIR}^{\mathcal{I}}(i, \langle i', S\rangle) = \begin{cases} 1 & \text{if } i = i' \text{ and } S \in \pi^i \\ 0 & \text{otherwise} \end{cases} \tag{6}$$

$$\mathsf{HIR}^{\mathcal{I}}: \quad \mathcal{IU} \times (\mathcal{IU} \times 2^{\mathbb{N}^2}) \to \{0,1\}, \text{ s.t. } \mathsf{HIR}^{\mathcal{I}}(i, \langle i', S\rangle) = \begin{cases} 1 & \text{if } i = i' \text{ and } S \in \pi^i_e \\ 0 & \text{otherwise} \end{cases} \tag{7}$$

$$\mathsf{HS}^{\mathcal{I}}: \quad (\mathcal{IU} \times 2^{\mathbb{N}^2}) \times \mathcal{S} \to \{0,1\}, \text{ s.t. } \mathsf{HS}^{\mathcal{I}}(\langle i, S\rangle, s) = \begin{cases} 1 & \text{if } S \in \pi^i_e \text{ and } s = \phi(S) \\ 0 & \text{otherwise} \end{cases} \tag{8}$$

$$\mathsf{HC}^{\mathcal{I}}: \quad (\mathcal{IU} \times 2^{\mathbb{N}^2}) \times \mathcal{C} \to [0,1], \text{ s.t. } \mathsf{HC}^{\mathcal{I}}(\langle i, S\rangle, c) = f^i_e(S)(c) \tag{9}$$

HAIR and HIR give raise, as expected, only to crisp assertions, that are valuated as true (that is, 1) just in case the grounded image region given as second argument belongs to the image layout given as first argument. The reason why we have included both these SPSs in the language, despite the obvious subsumption relation that links them (*i.e.* $\mathsf{HAIR} \sqsubseteq \mathsf{HIR}$), is that color and shape queries, while always allowed on atomic image regions, will be allowed on extended regions only in restricted cases. The rationale for this choice is of a computational nature and will be discussed in detail later, in Section 8.2. Also HS is a crisp role, true only if the given closed curve is the contour of the given grounded image region. Finally, HC is a truly fuzzy role, assigning to each pair (grounded image region, color) the percentage of the latter that occurs in the former. Notice that, in order to compute that percentage, the color function must be known, hence it is mandatory to have grounded image regions, in this case. Clearly, HC behaves as a crisp role on atomic regions, on which it is true just in case the given color is indeed the color of the region, otherwise HC is false, that is 0.

As far as similarity SPSs, these are categorized into two groups, mirroring the categorization of visual queries:

- *global* similarity SPSs; in general, there will be a family of such SPSs, each capturing a specific similarity criterion. Since from the conceptual viewpoint these SPSs form a uniform class, this model provides just one of them, to be understood as a representative of the whole class. Any other symbol of the same sort can be added without altering the structure and philosophy of the language. So, for global similarity matching we use the role:
  - $\mathsf{SI}(i, i')$ (Similar Image): denotes the similarity between the given image layouts;
- *local* similarity SPSs, assessing the similarity between individual features of images. Similarly for what we have done for global similarity, we include in the language one SPS for each type of abstract visual query. So we have:
  - $\mathsf{SC}(c, c')$ (Similar Color): denotes the similarity between the given colors;
  - $\mathsf{SS}(s, s')$ (Similar Shape): denotes the similarity between the given shapes.

The semantic clauses for both global and local similarity SPSs are defined on the basis of corresponding functions σ_i, σ_c and σ_s, that measure in the [0,1] interval the degree of similarity of two image layouts, colors and shapes, respectively:

$$\mathsf{SI}^{\mathcal{I}}: \quad \mathcal{IU} \times \mathcal{IU} \rightarrow [0,1], \text{ s.t. } \mathsf{SI}^{\mathcal{I}}(i,i') = \sigma_i(i,i') \tag{10}$$

$$\mathsf{SC}^{\mathcal{I}}: \quad \mathcal{C} \times \mathcal{C} \rightarrow [0,1], \text{ s.t. } \mathsf{SC}^{\mathcal{I}}(c,c') = \sigma_c(c,c') \tag{11}$$

$$\mathsf{SS}^{\mathcal{I}}: \quad \mathcal{S} \times \mathcal{S} \rightarrow [0,1], \text{ s.t. } \mathsf{SS}^{\mathcal{I}}(s,s') = \sigma_s(s,s') \tag{12}$$

The prototype of the model introduced in Section 12 uses, as image similarity function σ_i, a close relative of the function (1). This function can be calculated from an image layout, once a channel structure on the color set $\mathcal{C}$ is imposed. We remark that this particular choice for σ_i is not fundamental, as the present work is not a study in similarity image retrieval. In selecting σ_i we have just picked a function that is considered as a reasonable measure for image similarity by the digital image processing experts. The weights $w_{1H}, \ldots, w_{3B}$ will play an important role in relevance feedback, hence we postpone the complete definition of σ_i until Section 11. The functions σ_c and σ_s used for the prototype are given in Section 12.

The semantic clauses introduced in this section capture the desired behavior of the special symbols that have been defined to query image layouts. In order to turn the desired behavior into the actual behavior, we restrict the semantic universe of our logic to only those interpretations that satisfy these conditions. A fuzzy interpretation $\mathcal{I}$ will thus be called an *image interpretation* if it satisfies conditions (2) to (12).

From the semantic point of view this modeling style is an application of the so-called *method of concrete domains* [6], by which certain symbols are treated as having a fixed, interpretation-independent meaning. From the practical point of view, it results in interpreting every occurrence of the SPSs in question not as the occurrence of an uninterpreted role, but as a call to a routine that implements the corresponding image processing technique. In knowledge representation, this would be called a *procedural attachment* [63]. Section 10 will show how procedural attachment works in our model.

5.3 Modelling text layouts

In order to strictly parallel conceptual uniformity with modelling uniformity, we now define the notion of text layout as a semantic entity that conforms, as much as possible, to the notion of image layout. To this end, we will allow ourselves a moderate amount of overloading, and use some of the names introduced for modelling image form also for the corresponding notions pertaining to text form. Ambiguity will always be resolved by context.

In the single-dimensional space where text layouts belong, the notion of connected zone corresponds to the notion of interval. We define an interval $S \subset \mathbb{N}$ to be *aligned* iff

$$min\{x \mid x \in S\} = 0.$$

Given the set of *words* on an alphabet Λ, we define a *text layout* to be a triple $t = \langle A^t, \pi^t, f^t \rangle$ where:

- A^t (the *domain*) is a finite aligned interval,
- π^t is a partition of A^t into non-empty intervals $\{T_1, ..., T_n\}$ called *atomic regions*, and
- f^t is a total function (the *word function*) assigning a word to each atomic region.

In paralleling mono-chromatic image regions with the words of text, we do not intend to suggest any cognitive correspondence between these concepts. Rather, we consider as convenient to assume them as elementary notions of the corresponding layout model.

The *regions* of a text layout $t = \langle A^t, \pi^t, f^t \rangle$ are defined as the set

$$\pi_e^t = \{S \mid \exists T_1, ..., T_k \in \pi^t, k \geq 1,\ S = \cup_{i=1}^k T_i,\ S \text{ is an interval}\}$$

i.e. a region is the interval obtained by the union of one or more pairwise-adjacent atomic regions. Similarly to the case of images, a region S is not bound to a particular text layout, but is just a "window" that can be opened on many of them. This binding is realized in the notion of *grounded text region*, which we define to be a pair $\langle t, S \rangle$, where $t = \langle A^t, \pi^t, f^t \rangle$ is a text layout and $S \in \pi_e^t$.

Finally, we define *the text universe* $\mathcal{TU}$ as the set of all possible text layouts of any domain, and will use the symbol $\mathcal{E}$ to denote the set of all grounded text regions.

5.4 Querying text layouts

Similarly to what has been done for images, we will introduce the query language on the form of text by giving an elementary text mereology. This consists of two more sets of SICs (one for text layouts and the other for grounded text regions) and one SPS whose function is to allow queries (see below) to be addressed to a portion of a text layout, rather than to the text layout as a whole. These symbols are as follows:

- Ω_T, the names of text layouts (metasymbol t, optionally followed by a natural number);
- Ω_E, the names of grounded text regions (r); and
- HTR(t,r) (Has Text Region): a role relating the text layout t to one of its grounded regions r.

The semantic conditions for these symbols parallel those for their image analogs and will not be spelled out for brevity. Notice that no SPS is provided to address atomic text regions, as the textual analogue to queries on color patches or shapes (*i.e.*, full-text queries, see below) will be handled in a different way.

We distinguish between two categories of queries addressing text layouts:

1. *full-text* queries, each consisting of a *text pattern*, which denotes, in a deterministic way, a set of texts; when used as a query, the text pattern is supposed to retrieve any text layout belonging to its denotation;
2. *similarity* queries, each consisting of a text, and aimed at retrieving those text layouts which are similar to the given text.

In a full-text query, the text pattern can be specified in many different ways, *e.g.* by enumeration, via a regular expression, or via *ad hoc* operators specific to text structure such as proximity, positional and inclusion operators (for instance, in the style of the model for text structure presented in [64]). As in the case of images (Section 5.2), the choice as to what sub-language $\mathcal{L}_p$ for text patterns to adopt is not crucial for the rest of the model, so we will leave this piece of the query language unspecified and limit ourselves to specifying how to link it to the main body of the language.

The basic idea is to consider each well-formed expression $\varepsilon \in \mathcal{L}_p$, as representative of a language in its own right. This language, which will be denoted χ_ε, consists of those text layouts that "match" ε. For instance, if we adopted the language of regular expressions as $\mathcal{L}_p$, then the expression *␣info* would be the name of a language consisting of the text layouts in which at least one word with "info" as a prefix occurs. Furthermore, in accordance with its syntactical status, each $\varepsilon \in \mathcal{L}_p$ is assumed to be an SPS, that is an $\mathcal{ALC}$ concept having as instances the individual constants naming the text layouts in χ_ε. In order to query text layouts, therefore, countably many SPSs are introduced, as follows.

- ε is the generic member of the language for text pattern $\mathcal{L}_p$

The semantics of these symbols is the following:

$$\varepsilon^{\mathcal{I}}: \ \ \mathcal{TU} \rightarrow \{0,1\}, \text{ s.t. } \varepsilon^{\mathcal{I}}(t) = \begin{cases} 1 & \text{if } t \in \chi_\varepsilon \\ 0 & \text{otherwise;} \end{cases} \tag{13}$$

Similarity queries involve a similarity matching between text layouts that parallels image similarity matching. These queries are processed on the basis of automatically constructed abstractions of documents and queries, typically sets of weighted terms occurring in the text which, based on statistical properties of language and of the particular collections in which documents belong, are deemed significant for assessing similarity. To this end, these abstractions are compared by appropriate similarity assessing functions, leading to a document ranking on the basis of a best match criterion. For instance, a by now standard text similarity function is the cosine function [74], given by:

$$cos(\vec{t},\vec{t'}) = \frac{\sum_{k=1}^{m} v_{tk} \cdot v_{t'k}}{\sqrt{\sum_{k=1}^{m} v_{tk}^2} \cdot \sqrt{\sum_{k=1}^{m} v_{t'k}^2}}, \tag{14}$$

where $\vec{e}$ is the *index* of text layout e, and is given by a vector of weights $(v_{e1},\ldots,v_{em})$. v_{ei} is the weight of the i-th term. The index $\vec{e}$ can be directly computed from the layout $\langle A^e, \pi^e, f^e \rangle$.

As for images, the model supports similarity of text layouts by endowing the query language with a specific class of SPSs, each modelling textual similarity. The generic representative of this class is the $\mathcal{ALC}$ role ST, realizing the text similarity function σ_t. Syntax and semantics of ST are given below:

- $\mathsf{ST}(t,t')$ (standing for Similar Text): denotes the degree of similarity between the given text layouts.

Formally,

$$\mathsf{ST}^{\mathcal{I}}: \ \ \mathcal{TU} \times \mathcal{TU} \rightarrow [0,1], \text{ s.t. } \mathsf{ST}(t,t') = \sigma_t(t,t') \tag{15}$$

A suitable choice could be: $\sigma_t(t,t') = cos(\vec{t},\vec{t'})$. An image interpretation $\mathcal{I}$ will be called a *basic document interpretation* if it satisfies the semantic conditions for the SPSs introduced in this section.

6 Content

We take the content of a simple document (be it a text or an image) to be a *situation*, *i.e.* the set of all worlds or states of affairs that are compatible with the information carried by the document. For instance, the content of an image will be the set of worlds where the facts depicted in the image hold, irrespective of every other fact that is not visually evident, such as what the people represented in the image are thinking of, or what is taking place outside the setting of the image. Analogously, the content of a text will consist of all worlds in which the sentences making up the text simultaneously hold[3]. This view is inspired to a simple and by now classical notion of semantics, taking an "objective" stance on the meaning of natural language. Other views are possible, of

[3] Indeed, the semantics of a text may be a fairly articulated world structure, if, for instance, different time instants are referenced in the text. We believe that these structures need not be considered for retrieval purposes, due to the fact that the content representations involved in a retrieval system can realistically be expected to be gross abstractions of the "full" text semantics. We have elaborated on this theme more deeply elsewhere [58].

course. For instance, one could more subjectively understand the content of an image in terms of the impressions caused on the image viewer. For obvious reasons of generality, we choose the objective view.

The objectivity of the view, of course, does not imply the objectivity of the descriptions used to represent document semantics. The reason is that access to semantics is always through *interpretation*, a subjective endeavor by definition. In addition, the identification of what counts as the content of a document, although necessary, is not sufficient by itself to determine a model of document content. The reason is that the amount of facts that ought to be represented to exhaustively describe, *e.g.* an image, is typically too large to be considered as a step of an information retrieval methodology. This latter fact is even more evident from the famous saying "A picture is worth a thousand words". An inevitable prerequisite is the selection of a suitable abstraction of the documents at hand, to be described via the elements of a corresponding ontology. Such ontology would be a small subset of the user's "real" ontology, but central to the purpose of retrieval.

The selection of an appropriate ontology is neutral to that of the tool for representing document contents, as far as the latter is powerful enough to allow the description of any document candidate to retrieval. For the reasons pointed out in Sections 1 and 4, we have adopted a fuzzy description logic as a content representation language. The rest of this section is devoted to illustrate how fuzzy $\mathcal{ALC}$ is to be used to represent, and to retrieve by content, multimedia information.

6.1 Modelling content

Let l be a layout (either text or image) uniquely identified by the individual constant l, *i.e.* $\mathsf{l}^{\mathcal{I}} = l$ for any basic document interpretation $\mathcal{I}$. In this model, l may have an arbitrary number of associated *content descriptions*. Each such content description δ is a set of fuzzy assertions, given by the union of four component subsets:

1. the *layout identification*, a singleton with a fuzzy assertion of the form

$$\langle \mathsf{Self}(\mathsf{l}), 1 \rangle$$

whose role is to associate a content description with the layout it refers to. The layout identification is the same for all content descriptions of the same layout. In what follows, we will let $\sigma(\mathsf{l})$ denote the set of the content descriptions associated to the layout l, *i.e.*

$$\delta \in \sigma(\mathsf{l}) \text{ iff } \langle \mathsf{Self}(\mathsf{l}), 1 \rangle \in \delta;$$

2. the *object anchoring*, a set of fuzzy assertions of the form

$$\langle \mathsf{Represents}(\mathsf{r}, \mathsf{o}), n \rangle$$

where r is an individual constant that uniquely identifies a grounded region r of l and o is an individual constant that uniquely identifies the object represented in the region r. The combined effect of components 1 and 2 could have been achieved by eliminating the Self concept and making $\mathsf{Represents}$ a ternary predicate symbol, *i.e.* $\mathsf{Represents(l,r,o)}$. While extended DLs capable of dealing with predicate symbols of arity greater than 2 do exist, we prefer to use Ockham's razor and stick to the simpler, orthodox DLs of which $\mathcal{ALC}$ is the standard representative;

3. the *situation anchoring*, a set of fuzzy assertions of the form

$$\langle \mathsf{About}(\mathsf{l}, \mathsf{o}), n \rangle$$

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 (where none of the symbols Self, $\mathsf{Represents}$ or About occur), describing important facts stated in the layout about the individual constants identified by assertions of the previous two kinds.

The task of components 1 to 3 is actually that of binding the form and content dimensions of the same layout, thus allowing form- and content-based retrieval to be simultaneously performed on the same text or image.

As an example, let us consider a photograph showing a singer, Kiri, performing as Zerlina in Mozart's "Don Giovanni". Part of a plausible content description for this image, named i, could be (of course, the truth-values of the assertions reflect the best of the image indexer's knowledge):

$$\begin{aligned}\{\ & \langle \mathsf{Self}(\mathsf{i}), 1\rangle, \\ & \langle \mathsf{Represents}(\mathsf{r}, \mathsf{Kiri}), .7\rangle, \\ & \langle \mathsf{About}(\mathsf{i}, \mathsf{o}), .8\rangle, \\ & \langle \mathsf{DonGiovanni}(\mathsf{o}), 1\rangle, \langle \mathsf{Plays}(\mathsf{Kiri}, \mathsf{Zerlina}), .6\rangle\}\end{aligned} \tag{16}$$

Since there may be more than one content description for the same layout l, our model permits to consider a simple document under multiple viewpoints. In Section 9 we will see that, as a result of this, the "retrieval status values" of a layout to a query resulting from different content descriptions do not add up. Any of components 2 to 4 can be missing in a content description.

6.2 Querying content

Queries pertaining to content are called *content-based* queries, and involve conditions on the semantics of a text or image. Since content description is, as remarked at the beginning of this section, ontology-neutral, there are no SPSs specific to content-based queries.

Reasoning about content is performed directly (*i.e.* without procedural attachments) by fuzzy $\mathcal{ALC}$ on the content descriptions illustrated in the previous Section. The results of this logical reasoning activity will, if needed, be transparently merged to the results of non-logical computations (obtained through the procedural attachments to the various SPSs) by fuzzy $\mathcal{ALC}$ using the other components of content descriptions.

7 Documents

As mentioned in the introduction, we take multimedia documents to be complex objects consisting, in general, of a hierarchically structured set of simple documents, which may in turn be either chunks of text or images. It is just natural, then, to allow this model to deal not only with the features of simple documents, but also with the way these are structured into a complex document. We hence define the notions of *document* and *document structure*, along with a set of SPSs for addressing both within queries.

In order to reflect the "objective", domain- and application-independent nature of the notion of document, the latter is defined in the semantic universe of our model, as it is the form dimension of documents. More precisely, the model considers documents to be just structurally organized layouts, and for this reasons the two notions (document and structure) are defined together. In this way, the content dimension is left out of the document definition, as something to be considered only in the context of specific, hence subjective, domain- and application-dependent, document bases. Indeed, the link between documents and their content representations will be established precisely upon defining document bases.

7.1 Modelling structured documents

The kind of structure that is offered by the model is intentionally simple. It is designed having in mind documents such as newspaper articles or books, including images, possibly captioned. The operators for navigating these documents directly reflect their hierarchical structure, and so are the expected ones, with the additional expressive power permitted by the logical structure of the $\mathcal{ALC}$ language. More complex structures or more complex navigation operators can be considered, of course (a review of research work on richer document structures and associated query languages can be found in [1]). However, in order to keep the complexity of this model at a reasonable level, we have preferred to limit ourselves to hierarchical structures, as a realistic case which gives us the opportunity of introducing a modelling style that can be applied to more sophisticated structures.

The model views a document as a sequence of simple documents. A structure is imposed on this sequence by grouping contiguous simple documents into aggregates that are not allowed to partially overlap. Each aggregate determines a structural element of d.

Formally, a *document* is a 4-tuple $d = \langle n, B, w, R \rangle$, where:

1. $n \in \mathbb{N}_+$ is the *order* of the document, that is the number of basic components in d;
2. $B \subset (\mathcal{IU} \cup \mathcal{TU})$ is any finite set of (image or text) layouts, constituting the basic components of the document;
3. $w : [1, n] \rightarrow B$ is the total and surjective function that gives, for each $i \in [1, n]$, the i-th basic component of d;
4. $R = \{\rho_1, \ldots, \rho_m\}$ is a set of intervals, in fact sub-intervals of $[1, n]$, each of which is used to define, in a way to be seen soon, a structural element of d; to this end, R must satisfy the following conditions:
   (a) for all $1 \leq i \leq m$, $\rho_i \subseteq [1, n]$;
   (b) $[1, n] \in R$, that is, the "whole document" is a distinguished member of the document structure;
   (c) for all $\rho_i, \rho_j \in R$, either $\rho_i \subseteq \rho_j$ or $\rho_j \subseteq \rho_i$ or $\rho_i \cap \rho_j = \emptyset$.

This definition permits to model a simple document as a document. In particular, any image or text layout l is to be viewed as the document $\langle 1, \{l\}, w_1, R_1 \rangle$ where:

- $w_1(1) = l$ and
- $R_1 = \{[1, 1]\}$.

The *structure* of a document $d = \langle n, B, w, R \rangle$ is defined by the pair $S_d = \langle R, E \rangle$, where E is given by:

$$E = \{(\rho_1, \rho_2) \in R^2 \mid \rho_2 \subset \rho_1 \text{ and there is no } \rho_3 \in R \text{ such that } \rho_2 \subset \rho_3 \subset \rho_1\}.$$

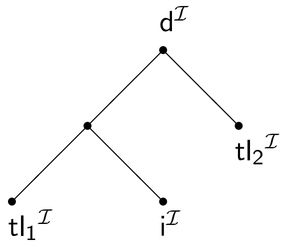

$\mathsf{d}^{\mathcal{I}}$
$\mathsf{tl_2}^{\mathcal{I}}$
$\mathsf{tl_1}^{\mathcal{I}}$
$\mathsf{i}^{\mathcal{I}}$

Figure 3: A document.

$\mathsf{d}^{\mathcal{I}} = \langle 3, \{\mathsf{tl_1}^{\mathcal{I}}, \mathsf{tl_2}^{\mathcal{I}}, \mathsf{i}^{\mathcal{I}}\}, w, R\rangle$
$w(1) = \mathsf{tl_1}^{\mathcal{I}}$, $w(2) = \mathsf{i}^{\mathcal{I}}$, $w(3) = \mathsf{tl_2}^{\mathcal{I}}$
$R = \{[1,3],[1,2],[3,3],[1,1],[2,2]\}$
$S_{\mathsf{d}^{\mathcal{I}}} = \langle R, E\rangle$
$E = \{([1,3],[1,2]),([1,3],[3,3]),([1,2],[1,1]),([1,2],[2,2])\}$

Figure 4: The representation of document d.

As anticipated, S_d is completely determined by d. It can be verified that S_d is a tree having R as nodes and E as edges. Needless to say, the root of S_d is $[1, n]$. Note that in the case of a document consisting of one simple document, E is empty.

As for document form, also for document structure the notion of grounded region is introduced, to enable the reference to structural elements within queries. A *grounded region* of a document $d = \langle n, B, w, R\rangle$ is defined as a pair $\langle d, \rho\rangle$ such that $\rho \in R$[4]. The *extent* of a grounded region $\langle d, \rho\rangle$ is defined as the set of image or text layouts to which elements in ρ are mapped by w, that is

$$\{w(i) \mid i \in \rho\}$$

Finally, we let $\mathcal{D}$ be the set of all documents and $\mathcal{G}$ that of all grounded regions.

Let us consider a document about the opera Don Giovanni, consisting of two parts: one part shows, as an image of the opera, the image layout i introduced at the end of Section 6.1, with an accompanying caption $\mathsf{tl_1}$; the other part is just a text layout reporting a (positive) critical review of the opera. The document is schematically represented in Figure 3, whereas its symbolic representation according to the just introduced model is given in tabular form in Figure 4.

The document has 5 grounded regions $\langle \mathsf{d}^{\mathcal{I}}, \rho\rangle$, for each $\rho \in R$, corresponding to the nodes of the tree shown in Figure 3. The extent of the grounded region $\langle \mathsf{d}^{\mathcal{I}}, [1,2]\rangle$ is the set of layouts $\mathsf{tl_1}^{\mathcal{I}}, \mathsf{i}^{\mathcal{I}}\}$. The E component of $\mathsf{d}^{\mathcal{I}}$'s structure has four elements, corresponding to the edges of the tree in Figure 3.

7.2 Querying structured documents

In querying structured documents, the following kinds of operations are typically performed:

[4] So, a grounded region is *not* either a grounded image region or a grounded text region, as the standard usage of natural language would maybe suggest. The reason for this somehow counterintuitive choice is to keep the model's lexicon as small as possible.

- navigation along the structure of documents; SPSs for expressing this kind of operation will be called *structural* SPSs;
- access to the basic constituents of a grounded region, *i.e.* the image and text layouts that are in the extent of that region; SPSs for expressing these accesses will be termed *extensional* SPSs;
- query the image and text layouts. These queries (called *ground queries*) are to be expressed by means of the SPSs introduced in Sections 5 and 6.

Structural symbols, in turn, can be categorized as *generic* and *positional* symbols. The former allow one to denote documents, their grounded regions and the relationships between these. We have two sets of generic SICs:

- Ω_D naming documents, and
- Ω_G naming grounded regions.

As customary, the intended meaning of these alphabets is formally captured by having $\cdot^{\mathcal{I}}$ as a total bijective mapping from Ω_D to $\mathcal{D}$ and from Ω_G to $\mathcal{G}$. Furthermore, we introduce one generic SPS, that is:

- HN (standing for <u>H</u>as <u>N</u>ode), relating a document to one of its grounded regions.

The semantics of HN is given by (for all documents d and sets of natural numbers ρ):

$$\mathsf{HN}^{\mathcal{I}}: \quad \mathcal{D} \times (\mathcal{D} \times 2^{\mathbb{N}}) \to \{0,1\}, \text{ such that } \mathsf{HN}^{\mathcal{I}}(d, \langle d', \rho\rangle) = \begin{cases} 1 & \text{if } d = d' \text{ and } \rho \in R \\ 0 & \text{otherwise.} \end{cases} \tag{17}$$

In conformance with the semantics of the structural symbols defined on layouts, that of HN assigns 1 only to the pairs ⟨document, grounded region⟩ such that the latter is a grounded region of the former.

Positional SPSs, on the other hand, allow one to navigate in the structure of the document. Among the many primitives that might be envisaged in order to model tree navigation, we propose the following SPSs:

- Root, the concept denoting roots of document structures;
- Leaf, the concept denoting leaf nodes of document structures;
- HCh (standing for <u>H</u>as <u>Ch</u>ild), a role denoting the link between nodes and their children nodes;
- HP (<u>H</u>as <u>P</u>arent), a role denoting the link between nodes and their parent node;
- HD (<u>H</u>as <u>D</u>escendant), the transitive closure of HCh;
- HA (<u>H</u>as <u>A</u>ncestor), the transitive closure of HP.

We just show the semantics of two positional symbols, leaving that of the others to the reader. In what follows, d, d' denote documents, ρ, ρ' denote sets of natural numbers, the structure of d is $S_d = \langle R, E\rangle$ and, as customary, E^+ indicates the transitive closure of E.

$$\begin{aligned}
\mathsf{Leaf}^{\mathcal{I}}: \quad & \mathcal{D} \times 2^{\mathbb{N}} \to \{0,1\}, \text{ such that} \\
& \mathsf{Leaf}^{\mathcal{I}}(\langle d, \rho \rangle) = \begin{cases} 1 & \text{if for no } \rho',\ \langle \rho, \rho' \rangle \in E \\ 0 & \text{otherwise} \end{cases}
\end{aligned} \tag{18}$$

$$\begin{aligned}
\mathsf{HD}^{\mathcal{I}}: \quad & (\mathcal{D} \times 2^{\mathbb{N}}) \times (\mathcal{D} \times 2^{\mathbb{N}}) \to \{0,1\}, \text{ such that} \\
& \mathsf{HD}^{\mathcal{I}}(\langle d, \rho \rangle, \langle d', \rho' \rangle) = \begin{cases} 1 & \text{if } d = d' \text{ and } \langle \rho, \rho' \rangle \in E^{+} \\ 0 & \text{otherwise} \end{cases}
\end{aligned} \tag{19}$$

As for extensional SPSs, we include two of them in the language, relating a grounded region of a document to the image and text layouts it contains. These SPSs are: HasImage and HasText. The semantics of the former is:

$$\begin{aligned}
\mathsf{HasImage}^{\mathcal{I}}: \quad & (\mathcal{D} \times 2^{\mathbb{N}}) \times \mathcal{IU} \to \{0,1\}, \text{ such that} \\
& \mathsf{HasImage}^{\mathcal{I}}(\langle d, \rho \rangle, l) = \begin{cases} 1 & \text{if } \rho \in R \text{ and } w(k) = l \text{ for some } k \in \rho \\ 0 & \text{otherwise} \end{cases}
\end{aligned} \tag{20}$$

while that of the latter is perfectly analogous. A basic document interpretation $\mathcal{I}$ will be called a *document interpretation* if it satisfies the semantic conditions for the SPSs (17) to (20) (as well as the conditions not explicitly stated for brevity).

8 A unified query language

We are now in the position to define the *query language* of the model. This language satisfies the two basic requirements necessary for complying with the philosophy of this model, namely:

1. It is the language of a description logic, so that matching queries against document bases can be done in the logical framework defined in Section 4.

2. Its syntax is a restriction of the general DL syntax that reflects the intended meaning of the SPSs for addressing form, content and structure previously introduced. More specifically, the syntax rules to be introduced shortly, rule out those concepts that would be meaningless according to the conditions on interpretations (2) to (20). As an example, let us consider the concept

   $$\mathsf{Root} \sqcap \exists \mathsf{HAIR}.\mathsf{Leaf}$$

   denoting the basic objects that are, at the same time, root nodes of document structures and images, the latter having an atomic region that is a leaf node. No such basic object may exist, a fact that is captured by the semantics of the involved SPSs. Even though, from a strictly logical viewpoint, concepts like the above would cause no problems, when used as queries (since they are satisfied by no interpretation, no documents would be returned by the system in response to them), licensing them would go against intuition and ultimately be misleading.

$$\begin{aligned}
\langle \textit{document-query} \rangle &::= \exists \mathsf{HN}.\langle \textit{node-concept} \rangle \mid \\
&\quad\ \langle \textit{document-query} \rangle \sqcap \langle \textit{document-query} \rangle \mid \\
&\quad\ \langle \textit{document-query} \rangle \sqcup \langle \textit{document-query} \rangle \\
\langle \textit{node-concept} \rangle &::= \langle \textit{extent-concept} \rangle \mid \\
&\quad\ \langle \textit{positional-concept} \rangle \mid \\
&\quad\ \exists \langle \textit{positional-role} \rangle.\langle \textit{node-concept} \rangle \mid \\
&\quad\ \langle \textit{node-concept} \rangle \sqcap \langle \textit{node-concept} \rangle \mid \\
&\quad\ \langle \textit{node-concept} \rangle \sqcup \langle \textit{node-concept} \rangle \\
\langle \textit{extent-concept} \rangle &::= \exists \mathsf{HasImage}.\underline{\langle \textit{image-query} \rangle} \mid \exists \mathsf{HasText}.\underline{\langle \textit{text-query} \rangle} \\
\langle \textit{positional-concept} \rangle &::= \mathsf{Root} \mid \mathsf{Leaf} \\
\langle \textit{positional-role} \rangle &::= \mathsf{HCh} \mid \mathsf{HP} \mid \mathsf{HD} \mid \mathsf{HA}
\end{aligned}$$

Figure 5: Grammar rules for document queries

Before dwelling into the technical details, we would like to emphasize that the language that is about to be introduced is a rigorous notation for expressing information needs as queries. How queries will be specified by users is an entirely different matter, which will be addressed in Section 12 and should not be confused with the question of what queries are.

The language will be presented in the remainder of this section, following a top-down style, starting from concepts addressing documents and their structure, and proceeding down to the queries addressing the basic components of documents, *i.e.* text and image layouts.

8.1 Document queries

The grammar given in Figure 5 defines the *document query language.* The categories capturing text and image queries will be defined later and are underlined for the reader's convenience.

A document query is a combination, via the conjunction and disjunction constructors, of concepts of the form $\exists \mathsf{HN}.C$ where C is a *node-concept.* Technically, the above concept reads as "the individual constants related through the role HN to an individual constant that is a C". Given the semantics of HN, this becomes "the documents having a node that is a C". From a pragmatic viewpoint, the prefix $\exists \mathsf{HN}$ introduces a concept that specifies the characteristics to be satisfied by a structural component (*i.e.*, a node) of the sought documents. By combining the above concept, conditions on several, possibly different, nodes of the same document may be stated. For instance,

$$(\exists \mathsf{HN}.C_1) \sqcup (\exists \mathsf{HN}.C_2)$$

expresses two conditions (C_1 and C_2) on two nodes of a document which may stand in any structural relationship, or even be the same node. On the contrary, if the nodes to be addressed are known to be structurally related in a specific way, then the whole query is to be stated as a *node-concept*, as it will be illustrated below (see concept (21)).

The reason why the negation constructor is not allowed here, as well as in other parts of the query language, is twofold. From one hand, this operator would make query evaluation an expensive operation. For instance, the simple query $\neg\exists \mathsf{HN}.C$ when evaluated for a document d, would imply to check, for each structural component n of d, that n is a $\neg C$. This is a consequence of the fact that, according to the semantics of fuzzy $\mathcal{ALC}$, the above query is equivalent to $\forall \mathsf{HN}.\neg C$ Ultimately, all components of all documents would have to be considered in answering the query. From the other hand, it is difficult to grasp the intuitive meaning of the negation applied to a similarity symbol. In other words, if the formal semantics of fuzzy $\mathcal{ALC}$ entitles one say that the maximal degree of truth of $\neg C(a)$ is .3 when that of $C(a)$ is .7, intuition may find meaningless this attribution when applied to a complex notion such as similarity. For these reasons, negation is only allowed in those parts of queries that are concerned with content. In such cases, its intuitive interpretation poses no problem, at least to those who accept the philosophy of fuzzy logic, as we do.

A *node-concept* may be comprised of several conditions, combined via the $\sqcap$ or $\sqcup$ operators. In its basic form, the syntax of a *node-concept* reflects two possibilities: whether or not the condition has a structural clause.

If no structural clause is present, a *node-concept* takes the form of an *extent-concept*, which addresses directly the basic constituents of the document without mentioning any condition on the document structure. An *extent-concept*, in turn, may take one of two forms, depending on the kind of basic component that it addresses. If the addressed basic component is a text layout, then the HasText role is used. For instance, the concept:

$$\exists \mathsf{HN}.\exists \mathsf{HasText}.C_t$$

denotes the documents having a node with a textual basic component that is a C_t. If the basic component is an image layout, the HasImage roles is used in a perfectly analogous way. Otherwise, a *node-concept* states structural conditions, which are couched in terms of the structural symbols introduced in Section 7.2. The simplest structural conditions are *positional-concepts*, which regard whether a node is a Root or a Leaf. More complex conditions involve *positional-roles* and recursively introduce other *node-concepts*. For instance, the concept:

$$\exists \mathsf{HN}.(\mathsf{Root} \sqcap \exists \mathsf{HCh}.(\exists \mathsf{HasImage}.C_i))$$

denotes the documents whose root has a child with an image that is a C_i, whereas:

$$\exists \mathsf{HN}.((\mathsf{Leaf} \sqcap \exists \mathsf{HasText}.C_t) \sqcap (\exists \mathsf{HA}.\exists \mathsf{HasImage}.C_i)) \tag{21}$$

denotes the documents having a node satisfying two conditions: (1) it is a leaf containing a text that is a C_t, (2) it has an ancestor with an image that is a C_i.

Before closing the structure topic, we would like to stress again that this model does not purport to be an advanced one in its treatment of structure. On the contrary, it merely aims at showing what is the role of structure in the general framework of document modeling, and how structure can be included in the query language. Having done this, the way is open to the consideration of more sophisticated structural models, a good example of which is presented in [65].

8.2 Image queries

The syntax of image queries is given in Figure 6. The first thing to observe is the presence, in the clauses defining *image-, color-* and *shape-concepts,* of a new DL concept constructor of the form $\{a\}$ where a is an individual constant, which may be a *layout-*, a *color-* or a *shape-name,* respectively. In the DL terminology, this constructor is called *singleton*, and represents, as expected, a concept having only the individual constant a as instance. From a semantics point of view, an interpretation has to satisfy the following condition: for all $d \in \Delta^{\mathcal{I}}$

$$
\begin{array}{rcl}
\langle \textit{image-query} \rangle & ::= & \langle \textit{image-concept} \rangle \mid \\
 & & \langle \textit{image-query} \rangle \sqcap \langle \textit{image-query} \rangle \mid \\
 & & \langle \textit{image-query} \rangle \sqcup \langle \textit{image-query} \rangle \\
\langle \textit{image-concept} \rangle & ::= & \exists \mathsf{About}.\langle \textit{content-concept} \rangle \mid \\
 & & \exists \mathsf{SI}.\{\langle \textit{layout-name} \rangle\} \mid \\
 & & \exists \mathsf{HAIR}.(\exists \mathsf{HC}.\langle \textit{color-concept} \rangle) \mid \\
 & & \exists \mathsf{HIR}.(\exists \mathsf{Represents}.\langle \textit{content-concept} \rangle) \mid \\
 & & \exists \mathsf{HIR}.((\exists \mathsf{Represents}.\langle \textit{content-concept} \rangle) \sqcap \langle \textit{region-concept} \rangle) \\
\langle \textit{region-concept} \rangle & ::= & \exists \mathsf{HC}.\langle \textit{color-concept} \rangle \mid \\
 & & \exists \mathsf{HS}.\langle \textit{shape-concept} \rangle \mid \\
 & & ((\exists \mathsf{HC}.\langle \textit{color-concept} \rangle) \sqcap (\exists \mathsf{HS}.\langle \textit{shape-concept} \rangle)) \\
\langle \textit{color-concept} \rangle & ::= & \{\langle \textit{color-name} \rangle\} \mid \exists \mathsf{SC}.\{\langle \textit{color-name} \rangle\} \\
\langle \textit{shape-concept} \rangle & ::= & \{\langle \textit{shape-name} \rangle\} \mid \exists \mathsf{SS}.\{\langle \textit{shape-name} \rangle\} \\
\langle \textit{layout-name} \rangle & ::= & \langle \textit{individual-constant} \rangle \\
\langle \textit{color-name} \rangle & ::= & \langle \textit{individual-constant} \rangle \\
\langle \textit{shape-name} \rangle & ::= & \langle \textit{individual-constant} \rangle
\end{array}
$$

Figure 6: Grammar rules for image queries

$$
\{a\}^{\mathcal{I}}(d) = \begin{cases} 1 & \text{if } d = a^{\mathcal{I}} \\ 0 & \text{otherwise.} \end{cases}
$$

Image queries are thus concepts of the DL $\mathcal{ALCO}$, which extends $\mathcal{ALC}$ with the singleton constructor. The additional expressive power of $\mathcal{ALCO}$ over $\mathcal{ALC}$ has no impact on the complexity of the image retrieval problem, as it will be argued later. All names in the query language are defined as individual constants, whose syntax is left unspecified as unnecessary.

An image query is a combination (through $\sqcap$ and $\sqcup$) of *image-concepts*, each of which may have one of four forms, illustrated in the following in the same order as they appear in Figure 6.

First, an *image-concept* may be a query on some content object, explicitly asserted to be related to the sought images through an About role assertion (termed "situation anchoring" in Section 6.1). In the query, the object in question is required to be an instance of *content-concept*, that is an $\mathcal{ALCO}$ concept built with the symbols used for situation descriptions. The grammar rule for *content-concept* is thus that for generic $\mathcal{ALC}$ concepts (given in Figure 1, Section 4), with the addition of the rule for the singleton operator presented above. For instance, under the obvious lexicon, the images about an Italian musician are retrieved via the query

$$\exists \mathsf{About}.(\mathsf{Musician} \sqcap \exists \mathsf{Born}.\mathsf{Italy}).$$

Second, an *image-concept* may be a concrete visual query, according to the terminology defined in Section 5.2. In this case, a prototype image layout l is to be provided with the query; this is done by specifying the singleton with the layout name l in the scope of the existential quantification on the SPS SI. By so doing, the similarity with l is captured in the query.

Third, an *image-concept* may be a query on the color of an atomic region, that is a color abstract visual query, expressed via an existential quantification on the HC SPS, followed by a *color-concept*; the latter is a singleton with the name of the color, optionally preceded by a color similarity predicate.

Finally, an *image-concept* may be a query on an image region. This kind of queries come in two forms:

1. The first form is meant to address the content dimension, and just consists of a $\mathsf{Represents}$ clause. In order to qualify for this kind of queries, an image must have an associated content description containing a $\mathsf{Represents}$ role assertion (object anchoring), relating a region of the image to an individual constant that is an instance of the *content-concept* that follows.

2. The second form extends the first with an additional condition (*region-concept*) on the color or the shape (or both) of the involved region. A shape condition is expressed via a *shape-concept*, which is strictly analogous to *color-concept*. The reason why such conditions are allowed only in conjunction with a $\mathsf{Represents}$ clause is that, in this way, their evaluation only involves the regions that have been the subject of an object anchoring. If this restriction is removed, then the evaluation of this type of queries would require, in the worst case, as many checks as the possible regions of an image, a number that is exponential in the number of atomic regions. As a verification of this latter fact, consider again Figure 2, showing the atomic regions of a simple image. This image has a region connected with region T_1 for each subset of the set $\{T_2, T_3, T_4, T_5\}$.

As an instance of an image query, let us consider the query asking for the images showing a cylindric reddish hat. This query can be expressed by the following image concept:

$$\exists \mathsf{HIR}.((\exists \mathsf{Represents}.\mathsf{Hat}) \sqcap (\exists \mathsf{HC}.\exists \mathsf{SC}.\{\mathsf{red}\}) \sqcap (\exists \mathsf{HS}.\{\mathsf{cylinder}\}))$$

The above query presents an interesting case of mixed form- and content-based image retrieval. In particular, the $\mathsf{Represents}$ 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. It is important to realize that herered is just a name (possibly given by the system) to a visual entity, namely a color, specified by the user via a convenient facility, such as the selection from a color palette. Analogously, $\mathsf{cylinder}$ is a system name for a shape that the user has perhaps drawn on the screen or selected from a palette of common shapes.

Let us reconsider the example introduced in Section 6. The images about the opera Don Giovanni are retrieved by the query $\exists \mathsf{About}.\{\mathsf{DonGiovanni}\}$. Those showing the singer Kiri are described by $\exists \mathsf{HIR}.\exists \mathsf{Represents}.\{\mathsf{Kiri}\}$. Turning to visual queries, the request to retrieve the images similar to a given one, named this, is expressed by $\exists \mathsf{SI}.\{\mathsf{this}\}$, and can be combined with any conceptual query, *e.g.* yielding $\exists \mathsf{SI}.\{\mathsf{this}\} \sqcup \exists \mathsf{About}.\{\mathsf{DonGiovanni}\}$, which would retrieve the images that are either similar to the given one or are about Don Giovanni. As for abstract visual queries, the images in which there is a blue region whose contour has a shape similar to a given curve s are retrieved by $\exists \mathsf{HAIR}.(\exists \mathsf{HC}.\{\mathsf{blue}\} \sqcap \exists \mathsf{HS}.\exists \mathsf{SS}.\{\mathsf{s}\})$. Finally, the user interested in retrieving the

$$\begin{array}{rcl}
\langle \textit{text-query} \rangle & ::= & \exists\mathsf{About}.\langle \textit{content-concept} \rangle \mid \\
 & & \exists\mathsf{HTR}.\exists\mathsf{Represents}.\langle \textit{content-concept} \rangle \mid \\
 & & \langle \textit{text-pattern} \rangle \mid \\
 & & \exists\mathsf{ST}.\{\langle \textit{text-layout-name} \rangle\} \mid \\
 & & \langle \textit{text-query} \rangle \sqcap \langle \textit{text-query} \rangle \mid \\
 & & \langle \textit{text-query} \rangle \sqcup \langle \textit{text-query} \rangle \\
\langle \textit{text-layout-name} \rangle & ::= & \langle \textit{individual-constant} \rangle
\end{array}$$

Figure 7: Grammar rules for text queries

images in which Kiri plays Zerlina and wears a blue-ish dress, can use the query

$$\exists\mathsf{HIR}.\exists\mathsf{Represents}.(\{\mathsf{Kiri}\} \sqcap \exists\mathsf{Plays}.\{\mathsf{Zerlina}\}) \sqcap (\exists\mathsf{HC}.\exists\mathsf{SC}.\{\mathsf{blue}\}). \tag{22}$$

8.3 Text queries

The syntax of text queries is given in Figure 7. A text query is a combination, via the usual $\sqcap$ and $\sqcup$ constructors, of concepts, each of which may have one of the following forms (following the order of presentation in Figure 7):

1. It may be a semantic query on the whole text layout, which is required to be About the *content-concept* that follows, or on a portion of it, which is introduced by the quantification on the mereological SPS HTR and which $\mathsf{Represents}$ an instance of the *content-concept* that follows.

2. It may be an exact match query driven by *text-pattern*.

3. It may be a similarity match request, syntactically similar to the analogous image query category.

As an example of the last category, the layouts that are about (in a traditional text retrieval sense) "successful representations of Mozart's operas", are retrieved by the query:

$$\exists\mathsf{ST}.\{\mathsf{tl}\} \tag{23}$$

where tl is a text layout consisting of exactly the above quoted words.

9 Document bases and document retrieval

The behavior of our query language is specified by formally defining the notion of a document base and of document retrieval. We model a document base as having three main components:

1. a collection of documents, *i.e.* structured aggregates of layouts, which collectively form the "objective" level of the document base;

2. a collection of content descriptions associated to the layouts of the structured documents; these descriptions collectively form the "subjective" level of the document base;

3. a knowledge base providing definitions of the concepts (in the form of fuzzy DL axioms – see Section 4) employed in content representations, as well as general knowledge (in the form of fuzzy DL assertions or axioms – see Section 4) on the domain of discourse that applies to the whole document base; this latter component may be thought of as representing the "conceptual context" in which the document base lives.

More formally, a *document base* is a triple $DB = \langle D, \Sigma_C, \Sigma_D \rangle$ where

- D is a set of documents (as defined in Section 7), *i.e.*:

  $$D = \{\langle n_i, B_i, w_i, R_i \rangle \mid i = 1, \ldots, N\}$$

  We will let B_{DB} stand for the set of layouts of the documents in DB, *i.e.*:

  $$B_{DB} = \cup_{i=1}^{N} B_i$$

- Σ_C is the set of content descriptions of the layouts in the document base, that is:

  $$\Sigma_C = \cup_{\mathsf{l}^{\mathcal{I}} \in B_{DB}} \sigma(\mathsf{l})$$

  where $\sigma(\mathsf{l})$ is defined (in Section 6.1) as the set of content descriptions associated to the layout l.

- Σ_D is a set of fuzzy assertions and axioms giving the document base context.

In response to a query Q addressed to a document base $DB = \langle D, \Sigma_C, \Sigma_D \rangle$, each document is attributed a *retrieval status value* (RSV), which is the numerical value according to which the document is ranked against all others. The RSV m of a given document $\mathsf{d}^{\mathcal{I}} = \langle n, \{\mathsf{l_1}^{\mathcal{I}}, \ldots, \mathsf{l}_n{}^{\mathcal{I}}\}, w, R \rangle$ is determined in the following way. Let $\Gamma(\mathsf{d})$ be the Cartesian product $\sigma(\mathsf{l}_1) \times \ldots \times \sigma(\mathsf{l}_n)$. Each tuple $\tau = \langle \delta_1, \ldots, \delta_n \rangle \in \Gamma(\mathsf{d})$ represents a choice of a content description for each layout in d. Let n_τ be the value:

$$n_\tau = \overline{Maxdeg}(\Sigma_D \cup \bigcup_{1 \le j \le n} \delta_j, Q(\mathsf{d}))$$

where $\overline{Maxdeg}$ is the same as the $Maxdeg$ function discussed in Section 4 except for the fact that it is calculated with respect to document interpretations only. n_τ can be interpreted as the RSV of d to Q calculated on the specific choice of content descriptions represented by τ. The RSV of d is then simply obtained by taking the maximum over all such choices, *i.e.*:

$$m = max_{\tau \in \Gamma(\mathsf{d})} \{n_\tau\}. \tag{24}$$

As an example, let us consider the document base $DB = \langle D, \Sigma_C, \Sigma_D \rangle$, where: (i) D contains the document d of Figure 3, *i.e.* $\mathsf{d} \in D$; (ii) Σ_C includes the image i content description (16); and (iii) Σ_D includes the following axioms

$$\langle \mathsf{DonGiovanni} \sqsubseteq \mathsf{EuropeanOpera}, 1 \rangle$$

$$\langle \mathsf{WestSideStory} \sqsubseteq \mathsf{AmericanOpera}, 1 \rangle$$

$$\langle \mathsf{EuropeanOpera} \sqsubseteq \mathsf{Opera} \sqcap (\exists \mathsf{ConductedBy.European}), .9 \rangle$$

$$\langle \mathsf{AmericanOpera} \sqsubseteq \mathsf{Opera} \sqcap (\exists \mathsf{ConductedBy.European}), .8 \rangle.$$

Suppose we are interested in documents containing images about operas conducted by a European director. To this end, we can use the query:

$$\exists \mathsf{HN}.\exists \mathsf{HasImage}.\exists \mathsf{About}.(\mathsf{Opera} \sqcap \exists \mathsf{ConductedBy}.\mathsf{European}) \tag{25}$$

The RSV attributed to d in response to this query, is .8, because: (i) d, with truth-value 1, has the node $[1,3]$; (ii) this node, with truth-value 1, has the image i; (iii) image i, with truth-value .8, is about o; (iv) o is an instance of the concept $\mathsf{Opera} \sqcap \exists \mathsf{ConductedBy}.\mathsf{European}$, with truth-value .9. This latter fact is a consequence of the axioms $\langle \mathsf{DonGiovanni} \sqsubseteq \mathsf{EuropeanOpera}, 1\rangle$, $\langle \mathsf{EuropeanOpera} \sqsubseteq \mathsf{Opera} \sqcap (\exists \mathsf{ConductedBy}.\mathsf{European}), .9\rangle$ and of the assertion $\langle \mathsf{DonGiovanni}(\mathsf{o}), 1\rangle$. Combining the evidence (i) – (iv) according to the semantic rule for conjunction, we obtain $.8 = \min\{1, 1, .8, .9\}$.

10 Implementing the model

As pointed out in the introduction, the model presented so far has a twofold role: on one hand, it aims at presenting multimedia information retrieval as a unique discipline, endowed with its own goals and techniques. On the other hand, the model aims at guiding the design of systems supporting a wider class of multimedia information retrieval applications than those supported by current models. The latter role, though, can be legitimately advocated only if the model proves implementable with state-of-the-art technology.

The rest of the paper elaborates on this aspect. First, in the present Section, the implementation of the retrieval capability of the model is addressed, to the end of proving that such implementation can indeed be achieved with off-the-shelf techniques borrowed from text and image retrieval, and knowledge representation. The proof hinges on the query evaluation procedure, which is first informally presented in Section 10.1, then fully specified in Section 10.2. Finally, in keeping with the formal style adopted all along the paper, the soundness and completeness of the procedure are stated in Section 10.3 and proved in the Appendix. It is important to notice that the query evaluation procedure is not a mere formal device, but an effective technique that could (in fact, should) be adopted in concrete implementations of the model, as we have done in our prototype (see below).

Once an implementation strategy is laid down, we will be able to present, in Section 11, a technique for performing relevance feedback, a crucial aspect of form-based retrieval.

Finally, Section 12 presents the prototypical implementation of a substantial part of the model, namely that dealing with form- and content-based image retrieval. The aim of this implementation is not to demonstrate the feasibility of the model: for this purpose, the query evaluation procedure is enough. The aim of this implementation is, first, to show an effective realization of the query evaluation procedure, and, second, to make concrete the benefits of our work by presenting a retrieval engine endowed with a semantic-content based capability that largely surpasses the functionalities of analogous similar systems. For this latter goal, we have chosen the *medium* most difficult to handle (*i.e.* images), leaving aside text and structure that are easier because of their more consolidated status. Our prototypical system has also a practical value, as it can be used for the rapid prototyping of specifications built according to the model.

10.1 The query evaluation procedure: an example

The query evaluation procedure is schematically given in Figure 8 (boxes standing for data, ovals for functions). For greater clarity, this procedure will be illustrated through an example. The query employed to this end is "documents with a critical review on a successful representation of a Mozart's opera with an Italian conductor, and with a picture showing Kiri in a blue-ish dress, playing Zerlina". This query is in fact the composition of queries that have been introduced in previous Sections, namely queries 23, 22 and 25, which are here recollected:

$$\begin{array}{ll}(A) & \exists \mathsf{ST}.\{\mathsf{tl}\} \\ (B) & \exists \mathsf{HIR}.\exists \mathsf{Represents}.(\{\mathsf{Kiri}\} \sqcap \exists \mathsf{Plays}.\{\mathsf{Zerlina}\}) \sqcap (\exists \mathsf{HC}.\exists \mathsf{SC}.\{\mathsf{blue}\}) \\ (C) & \exists \mathsf{About}.(\mathsf{Opera} \sqcap \exists \mathsf{ConductedBy}.\mathsf{European})\end{array}$$

Using the abbreviations above, the sample query is expressed as:

$$\exists \mathsf{HN}.(\exists \mathsf{HasText}.A \sqcap \exists \mathsf{HasImage}.(B \sqcap C)) \tag{26}$$

Following Figure 8, this query is input, along with each document to be evaluated doc, to the "Query Assertion Builder", which produces the query assertion $Q(\mathsf{doc})$. For our example, we will consider, as doc, the document d introduced in Figures 3 and 4, so that the query assertion turns out to be:

$$\exists \mathsf{HN}.(\exists \mathsf{HasText}.A \sqcap \exists \mathsf{HasImage}.(B \sqcap C))(\mathsf{d}). \tag{27}$$

This assertion is input to "Query Decomposition & Evaluation", which is realized by the function Φ. This is a central step of the procedure, which we now introduce in an informal way, leaving the precise definition of Φ, as well as the proof of its soundness and completeness, for the forthcoming Sections. In essence, Φ "scans" the query assertion with the aim of generating knowledge, in the form of fuzzy assertions, to be passed to the DL theorem prover (TP) for evaluating the query assertion. The assertions generated by Φ concern the SPSs of the model and can therefore be considered as "domain knowledge", where the domain in question is MIR and a specific document base. Once these assertions are provided to the DL TP, the latter can proceed to calculate the RSV of each considered document *just performing standard fuzzy logical reasoning.*

At each step Φ is given an assertion: at step 1, that is the query assertion; at step n, $n \geq 2$, that is one of the assertions generated in the previous step. Φ analyzes the out-most part of the given assertion and performs two tasks:

1. the *generative* task, in which it generates the fuzzy assertions that are necessary to the DL TP to reason about the analyzed part, and,
2. the *recursive* task, in which it provides for the continuation of the procedure by recursively applying itself to the remaining part of the assertion.

Let us now see this practically on the assertion 27. From a decomposition point of view, this assertion can be analyzed as $\exists \mathsf{HN}.\alpha(\mathsf{d})$, saying that d has a node that is an instance of concept α. Consequently, in its generative task, Φ produces all the assertions that will inform the DL TP about what are d's nodes, namely:

$$\begin{array}{c}\langle \mathsf{HN}(\mathsf{d}, \langle \mathsf{d}, [1,3] \rangle), 1 \rangle \\ \langle \mathsf{HN}(\mathsf{d}, \langle \mathsf{d}, [1,2] \rangle), 1 \rangle \\ \langle \mathsf{HN}(\mathsf{d}, \langle \mathsf{d}, [3,3] \rangle), 1 \rangle \\ \langle \mathsf{HN}(\mathsf{d}, \langle \mathsf{d}, [1,1] \rangle), 1 \rangle \\ \langle \mathsf{HN}(\mathsf{d}, \langle \mathsf{d}, [2,2] \rangle), 1 \rangle\end{array} \tag{28}$$

Needless to say, in order to calculate the above assertions Φ must have access to a database storing document structure; this aspect concerns *how* Φ works, and will be discussed in Section 12; for

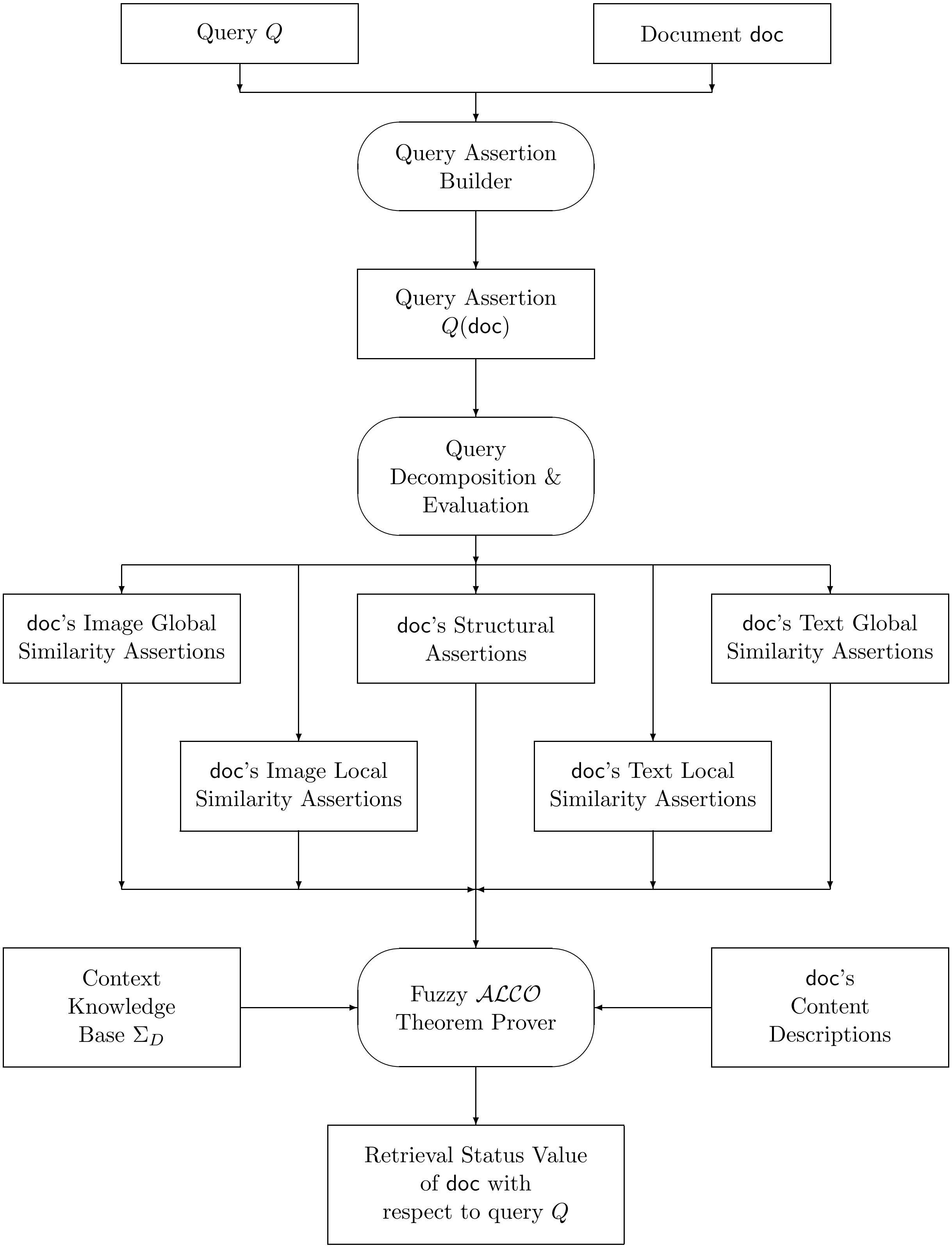

Query Q
Document doc
Query Assertion Builder
Query Assertion $Q(\mathsf{doc})$
Query Decomposition & Evaluation
doc's Image Global Similarity Assertions
doc's Structural Assertions
doc's Text Global Similarity Assertions
doc's Image Local Similarity Assertions
doc's Text Local Similarity Assertions
Context Knowledge Base Σ_D
Fuzzy $\mathcal{ALCO}$ Theorem Prover
doc's Content Descriptions
Retrieval Status Value of doc with respect to query Q

Figure 8: The query evaluation procedure

this Section, we will confine ourselves to *what* Φ does. In its recursive task, Φ applies itself to the following assertions:

$$\begin{array}{l}
\exists\mathsf{HasText}.A \sqcap \exists\mathsf{HasImage}.(B \sqcap C)(\langle \mathsf{d}, [1,3]\rangle) \\
\exists\mathsf{HasText}.A \sqcap \exists\mathsf{HasImage}.(B \sqcap C)(\langle \mathsf{d}, [1,2]\rangle) \\
\exists\mathsf{HasText}.A \sqcap \exists\mathsf{HasImage}.(B \sqcap C)(\langle \mathsf{d}, [3,3]\rangle) \\
\exists\mathsf{HasText}.A \sqcap \exists\mathsf{HasImage}.(B \sqcap C)(\langle \mathsf{d}, [1,1]\rangle) \\
\exists\mathsf{HasText}.A \sqcap \exists\mathsf{HasImage}.(B \sqcap C)(\langle \mathsf{d}, [2,2]\rangle)
\end{array}$$

The combined affect of the generative and recursive tasks seen so far, can be compactly described by letting $\Phi(\exists\mathsf{HN}.(\exists\mathsf{HasText}.A \sqcap \exists\mathsf{HasImage}.(B \sqcap C))(\mathsf{d}))$ be defined as follows:

$$\begin{array}{l}
\{\langle \mathsf{HN}(\mathsf{d}, \langle \mathsf{d}, [1,3]\rangle), 1\rangle, \ldots, \langle \mathsf{HN}(\mathsf{d}, \langle \mathsf{d}, [2,2]\rangle), 1\rangle\} \cup \\
\Phi(\exists\mathsf{HasText}.A \sqcap \exists\mathsf{HasImage}.(B \sqcap C)(\langle \mathsf{d}, [1,3]\rangle)) \cup \\
\ldots \cup \\
\Phi(\exists\mathsf{HasText}.A \sqcap \exists\mathsf{HasImage}.(B \sqcap C)(\langle \mathsf{d}, [2,2]\rangle)).
\end{array}$$

Intuitively, only the applications involving intervals [1,3] and [1,2] will generate non-zero fuzzy assertions, as they address the only nodes of d having both a text and an image, *i.e.* the root node and its left descendant. So, in the following we will consider only the decomposition of these two assertions. Since each of them is a conjunction, not surprisingly, Φ will handle it by generating no assertions (as no special knowledge is required by the TP to handle conjunction), while recursively applying itself to the single conjuncts:

$$\begin{array}{rcl}
\Phi(\exists\mathsf{HasText}.A \sqcap \exists\mathsf{HasImage}.(B \sqcap C))(\langle \mathsf{d}, [1,3]\rangle) & = & \Phi(\exists\mathsf{HasText}.A(\langle \mathsf{d}, [1,3]\rangle)) \cup \qquad (29) \\
 & & \Phi(\exists\mathsf{HasImage}.(B \sqcap C))(\langle \mathsf{d}, [1,3]\rangle))) \quad (30)
\end{array}$$

and the same for interval [1,2]. Let us consider the first application 29. Fully stated, it is as follows:

$$\Phi(\exists\mathsf{HasText}.\exists\mathsf{ST}.\{\mathsf{tl}\}(\langle \mathsf{d}, [1,3]\rangle)) \qquad (31)$$

Following the same approach as above, Φ will let the DL TP know what are the text layouts in the grounded region $\langle \mathsf{d}, [1,3]\rangle$ by generating the assertions:

$$\langle \mathsf{HasText}(\langle \mathsf{d}, [1,3]\rangle, \mathsf{tl}_1), 1\rangle \qquad \langle \mathsf{HasText}(\langle \mathsf{d}, [1,3]\rangle, \mathsf{tl}_2), 1\rangle$$

while proceeding on to compute $\Phi(\exists\mathsf{ST}.\{\mathsf{tl}\}(\mathsf{tl}_1))$ and $\Phi(\exists\mathsf{ST}.\{\mathsf{tl}\}(\mathsf{tl}_2))$. In order to handle the text similarity addressed by these two last assertions, Φ "calls" each time the function σ_t, in order to obtain the degree of similarity between the query layout tl and each layout tl_i. The result of this call, denoted as usual $\sigma_t(\mathsf{tl}, \mathsf{tl}_i)$, is then embodied into an apposite assertion for the DL TP. Therefore, the results of the last two applications of Φ are:

$$\begin{array}{rcl}
\Phi(\exists\mathsf{ST}.\{\mathsf{tl}\}(\mathsf{tl}_1)) & = & \{\langle \mathsf{ST}(\mathsf{tl}, \mathsf{tl}_1), \sigma_t(\mathsf{tl}, \mathsf{tl}_1)\rangle\} \\
\Phi(\exists\mathsf{ST}.\{\mathsf{tl}\}(\mathsf{tl}_2)) & = & \{\langle \mathsf{ST}(\mathsf{tl}, \mathsf{tl}_2), \sigma_t(\mathsf{tl}, \mathsf{tl}_2)\rangle\}.
\end{array}$$

This concludes application 31. In working out the analogous assertion on interval [1,2], *i.e.* $\exists\mathsf{HasText}.\exists\mathsf{ST}.\{\mathsf{tl}\}(\langle \mathsf{d}, [1,2]\rangle)$), Φ re-generates the assertions concerning the text layout tl_1, which is the only text layout contained in the region $\langle \mathsf{d}, [1,2]\rangle$, giving no contribution to the overall process. Let us now consider application 30. Following the same behaviour seen so far on structure:

$$\Phi(\exists\mathsf{HasImage}.(B \sqcap C))(\langle \mathsf{d}, [1,3]\rangle))) = \{\langle \mathsf{HasImage}(\langle \mathsf{d}, [1,3]\rangle, \mathsf{i}), 1\rangle\} \cup \Phi((B \sqcap C)(\mathsf{i})). \qquad (32)$$

The application of Φ to the analogous assertion on region $\langle \mathsf{d}, [1,2]\rangle$, is not going to produce anything new, since both these regions have only image i in their extent. So, we will not consider such application. Turning back to 32:

$$\begin{array}{rcl}\Phi((B \sqcap C)(\mathsf{i})) & = & \Phi(B(\mathsf{i})) \cup \Phi(C(\mathsf{i})) \\ & = & \Phi(\exists \mathsf{HIR}.\exists \mathsf{Represents}.(\{\mathsf{Kiri}\} \sqcap \exists \mathsf{Plays}.\{\mathsf{Zerlina}\}) \sqcap (\exists \mathsf{HC}.\exists \mathsf{SC}.\{\mathsf{blue}\})(\mathsf{i})) \quad (33) \\ & \cup & \Phi(\exists \mathsf{About}.(\mathsf{Opera} \sqcap \exists \mathsf{ConductedBy}.\mathsf{European})(\mathsf{i})).\end{array}$$

The last application is easily handled: since it is a semantic assertion, Φ has no job to do on it: the DL TP, using the axioms in Σ_D as well as the content descriptions associated to i, is perfectly able to evaluate it. Therefore:

$$\Phi(\exists \mathsf{About}.(\mathsf{Opera} \sqcap \exists \mathsf{ConductedBy}.\mathsf{European})(\mathsf{i})) \quad = \quad \emptyset.$$

On the contrary, the processing of assertion 33 is much more elaborated. In handling the HIR role, Φ must identify the grounded image regions of i which have been annotated with a $\mathsf{Represents}$ assertion. Looking at the content description of i (16 in Section 6.1), only region $\langle \mathsf{i}, \mathsf{r}\rangle$ is the case, thus, the assertion:

$$\langle \mathsf{HIR}(\mathsf{i}, \langle \mathsf{i}, \mathsf{r}\rangle), 1\rangle \quad (34)$$

is generated, along with application:

$$\Phi(\exists \mathsf{Represents}.(\{\mathsf{Kiri}\} \sqcap \exists \mathsf{Plays}.\{\mathsf{Zerlina}\}) \sqcap (\exists \mathsf{HC}.\exists \mathsf{SC}.\{\mathsf{blue}\})(\langle \mathsf{i}, \mathsf{r}\rangle))$$

which turns out to be:

$$\Phi(\exists \mathsf{Represents}.(\{\mathsf{Kiri}\} \sqcap \exists \mathsf{Plays}.\{\mathsf{Zerlina}\})) \cup \Phi(\exists \mathsf{HC}.\exists \mathsf{SC}.\{\mathsf{blue}\})(\langle \mathsf{i}, \mathsf{r}\rangle)).$$

The first term of this last union is a semantic application, and thus will produce the empty set. What is left is then the second term, whose argument $\exists \mathsf{HC}.\exists \mathsf{SC}.\{\mathsf{blue}\}(\langle \mathsf{i}, \mathsf{r}\rangle)$ is logically equivalent to a disjunction, ranging on all colors l, of the assertion:

$$\mathsf{HC}(\langle \mathsf{i}, \mathsf{r}\rangle, l) \sqcap \mathsf{SC}(l, \mathsf{blue}). \quad (35)$$

According to the rules of fuzzy semantics, the truth-value of the last assertion is given by the minimum of its conjuncts' truth-values, which are in turn established by the semantic clauses for the SPSs HC and SC. Considering that each such assertion is embedded in a global disjunction, we have the following:

$$\Phi(\exists \mathsf{HC}.\exists \mathsf{SC}.\{\mathsf{blue}\})(\langle \mathsf{i}, \mathsf{r}\rangle)) = \{\langle \exists \mathsf{HC}.\exists \mathsf{SC}.\{\mathsf{blue}\})(\langle \mathsf{i}, \mathsf{r}\rangle), n\rangle\} \quad (36)$$

where

$$n = max_{l \in \mathcal{C}}\{min\{f_e(\mathsf{r}^{\mathcal{I}})(l), \sigma_c(l, \mathsf{blue}^{\mathcal{I}})\}\}. \quad (37)$$

And this concludes the decomposition and evaluation of the query assertion. For convenience, all the assertions that are generated in this stage are classified (as indicated in Figure 8) into the following categories:

- *structural assertions*, *i.e.* the assertions involving structural SPSs;

- *image global similarity assertions*, *i.e.* SI assertions;
- *image local similarity assertions*, *i.e.* image structural HAIR, HIR assertions, and HC, HS assertions;
- *text global similarity assertions*, *i.e.* ST assertions; and
- *text local similarity assertions*, *i.e.* text structural HTR assertions, and ε assertions.

As shown in Figure 8, all these assertions are input to the fuzzy $\mathcal{ALCO}$ theorem prover TP for the computation of m, along with the context knowledge base Σ_C and all combinations of d's content descriptions. The graphical illustration of this latter input in Figure 8 has been simplified by omitting the feeding loop. Clearly, if at most one content description is provided for each layout in d, then $\Gamma(\mathsf{d})$ is a singleton and its consideration adds no complexity to the query evaluation process. Otherwise, the size of $\Gamma(\mathsf{d})$ is exponential in the number of d's layouts and its consideration heavily impacts on the complexity of the query evaluation procedure. This is the price to be paid for considering multiple content descriptions.

For space reasons, we cannot discuss the TP here. The interested reader is referred to the papers quoted in Section 3 for detailed descriptions of the various aspects involved in the design and evaluation of the TP. From a computational complexity viewpoint, the implication problem for fuzzy $\mathcal{ALCO}$ is proved to be PSPACE-complete. This result, albeit negative, is expected to have no significant impact on the tool being described for two basic reasons. First, it is a worst case result; things do not necessarily go as bad in practical cases. Second, the size of the prototypical systems for which our tool is designed is expected to be limited. Experimental results with an implementation of the TP show that the system is reactive with KBs of the order of a few thousands assertions.

10.2 Query decomposition and evaluation

We now proceed to the formal specification of the function Φ. For greater clarity, being a function defined on the query language, Φ will be introduced by following the query language syntactical structure.

10.2.1 Decomposition and evaluation of document queries

In the most general case, a query begins by addressing document structure, so the definition of Φ begins, in Figure 9, from document queries, whose syntax is given in Figure 5. In illustrating Φ, we will follow the order established by Figure 9 and the forthcoming Figures that make up Φ's definition.

Composite queries, consisting of conjunctions and disjunctions of document queries, are separately treated by Φ, each in fact being a query of its own. This way of handling conjunctions and disjunctions will be applied whenever correctness is preserved.

Upon operating on the simplest document queries, *i.e.* assertions of the form $(\exists\mathsf{HN}.C)(\mathsf{d})$, Φ generates an assertion $\langle\mathsf{HN}(\mathsf{d},\mathsf{n}),1\rangle$, for each grounded region n of d, and recursively applies itself to the assertion $C(\mathsf{n})$. This is one of the few basic principles that inspire the definition of Φ, and we believe that the informal explanation given in the previous Section is sufficient to make it clear. The treatment of assertions of the form $(\exists\mathsf{HasImage}.C)(\mathsf{n})$ is analogous. For each image layout i_j of n, the assertion $\langle\mathsf{HasImage}(\mathsf{n},\mathsf{i}_j),1\rangle$ is generated while continuing the evaluation on $C(\mathsf{i}_j)$. The same, *mutatis mutandis*, is done for assertions involving the other positional roles or concepts.

x	$\Phi(x)$
$((\exists\mathsf{HN}.C)\sqcap(\exists\mathsf{HN}.D))(\mathsf{d})$ $((\exists\mathsf{HN}.C)\sqcup(\exists\mathsf{HN}.D))(\mathsf{d})$	$\Phi((\exists\mathsf{HN}.C)(\mathsf{d}))\cup\Phi((\exists\mathsf{HN}.D)(\mathsf{d}))$
$(\exists\mathsf{HN}.C)(\mathsf{d})$	$\{\langle\mathsf{HN}(\mathsf{d},\mathsf{n}_1),1\rangle,\ldots,\langle\mathsf{HN}(\mathsf{d},\mathsf{n}_k),1\rangle\}\cup\Phi(C(\mathsf{n}_1))\cup\ldots\cup\Phi(C(\mathsf{n}_k))$, $(k\geq 1)$, for all grounded regions n_i $(1\leq i\leq k)$ such that $\mathsf{HN}^{\mathcal{I}}(\mathsf{d}^{\mathcal{I}},\mathsf{n}_i{}^{\mathcal{I}})=1$
$(C\sqcap D)(\mathsf{n})\quad(C\sqcup D)(\mathsf{n})$	$\Phi(C(\mathsf{d}))\cup\Phi(D(\mathsf{d}))$
$(\exists\mathsf{HasImage}.C)(\mathsf{n})$	$\{\langle\mathsf{HasImage}(\mathsf{n},\mathsf{i}_1),1\rangle,\ldots,\langle\mathsf{HasImage}(\mathsf{n},\mathsf{i}_k),1\rangle\}\cup\Phi(C(\mathsf{i}_1))\cup\ldots\cup\Phi(C(\mathsf{i}_k))$, $(k\geq 0)$, for all image layouts i_j $(1\leq j\leq k)$ such that $\mathsf{HasImage}^{\mathcal{I}}(\mathsf{n}^{\mathcal{I}},\mathsf{i}_j{}^{\mathcal{I}})=1$ (analogously for the role $\mathsf{HasText}$)
$\mathsf{Root}(\mathsf{n})$	$\begin{cases}\{\langle\mathsf{Root}(\mathsf{n}),1\rangle\} & \text{if } \mathsf{Root}^{\mathcal{I}}(\mathsf{n}^{\mathcal{I}})=1\\ \emptyset & \text{otherwise.}\end{cases}$ (analogously for the concept Leaf)
$(\exists\mathsf{HCh}.C)(\mathsf{n})$	$\{\langle\mathsf{HCh}(\mathsf{n},\mathsf{n}_1),1\rangle,\ldots,\langle\mathsf{HCh}(\mathsf{n},\mathsf{n}_k),1\rangle\}\cup\Phi(C(\mathsf{n}_1))\cup\ldots\cup\Phi(C(\mathsf{n}_k))$, $(k\geq 0)$, for all grounded regions n_j $(1\leq j\leq k)$ such that $\mathsf{HCh}^{\mathcal{I}}(\mathsf{n}^{\mathcal{I}},\mathsf{n}_j{}^{\mathcal{I}})=1$ (analogously for the roles HP, HD and HA)

Figure 9: The decomposition function for structural queries

10.2.2 Decomposition and evaluation of image sub-queries

The decomposition and evaluation of image queries is presented in Figure 10. 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, and is guided by the fulfillment of the query decomposition principle.

Queries on situation anchoring, formulated in terms of the About SPS, are, as already remarked in the previous Section, just ignored by Φ, as the knowledge for their evaluation is already part of the document base, namely it is contained in the content descriptions collected in Σ_C and in the context knowledge base Σ_D. The same applies to queries on object anchoring, formulated in terms of the Represents SPS. These queries may stand alone (*i.e.* be of the form $(\exists\mathsf{Represents}.C)(\mathsf{r})$) or be conjoined to a *region concept* (*i.e.* $((\exists\mathsf{Represents}.C)\sqcap D)(\mathsf{r})$); in both cases, the content sub-query gives no contribution to $\Phi(C(\mathsf{d}))$.

Abstract visual queries come in two sorts, depending on the kind of image region addressed. The first and simplest sort address exclusively atomic regions. It consists of color queries and aim at retrieving images having a patch of a specified color. These queries 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.

x	$\Phi(x)$
$(C \sqcap D)(\mathsf{i}) \quad (C \sqcup D)(\mathsf{i})$	$\Phi(C(\mathsf{i})) \cup \Phi(D(\mathsf{i}))$
$(\exists \mathsf{SI}.\{\mathsf{qi}\})(\mathsf{i})$	$\{\langle \mathsf{SI}(\mathsf{i}, \mathsf{qi}), \sigma_i(\mathsf{i}^{\mathcal{I}}, \mathsf{qi}^{\mathcal{I}})\rangle\}$ if $\sigma_i(\mathsf{i}^{\mathcal{I}}, \mathsf{qi}^{\mathcal{I}}) > 0$, $\emptyset$ otherwise
$(\exists \mathsf{About}.C)(\mathsf{i})$	$\emptyset$
$(\exists \mathsf{HAIR}.\exists \mathsf{HC}.\{\mathsf{c}\})(\mathsf{i})$	$\{\langle x, 1\rangle\}$ if for some atomic region T of $\mathsf{i}^{\mathcal{I}}$, $f(T) = \mathsf{c}^{\mathcal{I}}$, $\emptyset$ otherwise
$(\exists \mathsf{HAIR}.\exists \mathsf{HC}.\exists \mathsf{SC}.\{\mathsf{c}\})(\mathsf{i})$	$\{\langle x, n\rangle\}$ if $n = max_{T \in \pi}\{\sigma_c(f(T), \mathsf{c}^{\mathcal{I}})\} > 0$, $\emptyset$ 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 grounded image regions $\mathsf{r}_j{}^{\mathcal{I}}$ $(1 \leq j \leq k)$ such that $\mathsf{HIR}^{\mathcal{I}}(\mathsf{i}^{\mathcal{I}}, \mathsf{r}_j{}^{\mathcal{I}}) = 1$ and $\mathsf{Rep}(\mathsf{r}_j, \mathsf{o}) \in \delta$ for some $\delta \in \sigma(\mathsf{i})$ and individual constant o
$(\exists \mathsf{Rep}.C)(\mathsf{r})$	$\emptyset$
$((\exists \mathsf{Rep}.C) \sqcap D)(\mathsf{r})$	$\Phi(D(\mathsf{r}))$
$(\exists \mathsf{HC}.\{\mathsf{c}\})(\mathsf{r})$	$\{\langle \mathsf{HC}(\mathsf{r}, \mathsf{c}), n\rangle\}$ if $n = f_e(\mathsf{r}^{\mathcal{I}}, \mathsf{c}^{\mathcal{I}}) > 0$, $\emptyset$ otherwise
$(\exists \mathsf{HC}.\exists \mathsf{SC}.\{\mathsf{c}\})(\mathsf{r})$	$\{\langle x, n\rangle\}$ if $n = max_{l \in \mathcal{C}}\{min\{f_e(\mathsf{r}^{\mathcal{I}})(l), \sigma_c(l, \mathsf{c}^{\mathcal{I}})\}\} > 0$, $\emptyset$ otherwise
$(\exists \mathsf{HS}.\{\mathsf{s}\})(\mathsf{r})$	$\{\langle \mathsf{HS}(\mathsf{r}, \mathsf{s}), 1\rangle\}$ if $\mathsf{s}^{\mathcal{I}} = \phi(\mathsf{r}^{\mathcal{I}})$, $\emptyset$ otherwise
$(\exists \mathsf{HS}.\exists \mathsf{SS}.\{\mathsf{s}\})(\mathsf{r})$	$\{\langle x, n\rangle\}$ if $n = \sigma_s(\phi(\mathsf{r}^{\mathcal{I}}), \mathsf{s}^{\mathcal{I}}) > 0$, $\emptyset$ otherwise

Figure 10: The decomposition function for image queries

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, yielding queries of the form $(\exists \mathsf{HAIR}.\exists \mathsf{HC}.\exists \mathsf{SC}.\{\mathsf{c}\})(\mathsf{i})$. 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, 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; 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.

The second sort of abstract visual queries address both atomic and non-atomic regions and takes the general 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, for the computational reasons that have been illustrated in Section 8.2, C is bound to include a Represents clause, which, of course, restricts the candidate regions to all and only those referenced by object anchoring. 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 10 for brevity. Let us quickly review the first two cases:

x	$\Phi(x)$
$(C \sqcap D)(\mathsf{t})$ $(C \sqcup D)(\mathsf{t})$	$\Phi(C(\mathsf{t})) \cup \Phi(D(\mathsf{t}))$
$(\exists \mathsf{ST}.\{\mathsf{qt}\})(\mathsf{t})$	$\{\langle \mathsf{ST}(\mathsf{t},\mathsf{qt}), \sigma_t(\mathsf{t}^{\mathcal{I}}, \mathsf{qt}^{\mathcal{I}})\rangle\}$
$(\exists \mathsf{About}.C)(\mathsf{t})$	$\emptyset$
$(\exists \mathsf{HTR}.(\exists \mathsf{Rep}.C))(\mathsf{t})$	$\{\langle \mathsf{HTR}(\mathsf{t},\mathsf{r}_1), 1\rangle, \ldots, \langle \mathsf{HTR}(\mathsf{t},\mathsf{r}_k), 1\rangle\}$, $(k \geq 0)$, for all grounded text regions r_j $(1 \leq j \leq k)$ such that $\mathsf{HTR}^{\mathcal{I}}(\mathsf{t}^{\mathcal{I}}, \mathsf{r}_j{}^{\mathcal{I}}) = 1$ and $\mathsf{Rep}(\mathsf{r}_j, \mathsf{o}) \in \delta$ for some $\delta \in \sigma(\mathsf{t})$ and individual constant o
$\varepsilon(\mathsf{t})$	$\begin{cases} \{\langle \varepsilon(\mathsf{t}), 1\rangle\} & \text{if } \mathsf{t}^{\mathcal{I}} \in \chi_\varepsilon \\ \emptyset & \text{otherwise.} \end{cases}$

Figure 11: The decomposition function for text queries

- $(\exists \mathsf{HC}.\{\mathsf{c}\})(\mathsf{r})$: is evaluated by generating the corresponding fuzzy assertion, having as degree of truth the percentage of color c in the region r.
- $(\exists \mathsf{HC}.\exists \mathsf{SC}.\{\mathsf{c}\})(\mathsf{r})$: this case has been already discussed in the previous Section; it is easy to verify that this is a generalization over the previous case.
- $(\exists \mathsf{HS}.\{\mathsf{s}\})(\mathsf{r})$: if the shape of r equals s, the evaluation of this query yields the corresponding assertion with degree 1; otherwise, no assertion is generated.
- $(\exists \mathsf{HS}.\exists \mathsf{SS}.\{\mathsf{s}\})(\mathsf{r})$: 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.

The decomposition and evaluation of text queries (described in Figure 11) is performed in an analogous way, and is not further discussed.

10.3 Foundations

Last but not least, we provide formal foundation to what has been so far justified on a purely intuitive basis. The intuitive justification for Φ given in Section 10.1, is that Φ generates all the knowledge needed to reduce document retrieval to standard fuzzy logical reasoning. Here, the word "standard" means "treating the SPSs (resp. SICs) as standard DL roles (individuals)", or, put in another way "ignoring the special semantics of the special symbols of the query language". In technical terms, this amounts to say that the assertions generated by Φ permit to perform the computation of the Retrieval Status Value on standard interpretations rather than on document interpretations. Formally, this is stated as follows:

Proposition *For all document bases, the RSV of the document* d *to the query* Q *is given by the maximal degree of truth of* $Q(\mathsf{d})$ *with respect to the knowledge base consisting of: the decomposition of query* Q*,* $\Phi(Q(\mathsf{d}))$*, the context knowledge base* Σ_D *and the set of* d*'s content descriptions. That is:*

$$\overline{Maxdeg}(\Sigma_D \cup \bigcup_{1 \leq j \leq n} \delta_j, Q(\mathsf{d})) = Maxdeg(\Sigma_D \cup \Phi(Q(\mathsf{d})) \cup \bigcup_{1 \leq j \leq n} \delta_j, Q(\mathsf{d})).$$

This proposition, whose proof is given in the Appendix, captures both soundness and completeness of the query evaluation procedure (QEP). From the soundness point of view, it says that the RSV computed by the QEP is indeed correct. From the completeness point of view, it says that, if m is the RSV of a document d with respect to query Q, then the QEP computes precisely it.

11 Relevance feedback

Traditional (*i.e.* text) Information Retrieval systems are interactive. A most interesting aspect of this interactivity concerns the possibility for the user to give the system indications about the relevance (or irrelevance) of certain documents, and have the system take into account these indications to improve retrieval performance. This is the basis of a mechanism called "Relevance Feedback" (hereafter simply RF for brevity).

Techniques for performing RF in text retrieval, date back to the SMART system [74], but this is still an active research field, as witnessed by more recent work [4, 10, 32, 36, 45, 81, 99]. RF has attracted also the interest of researchers in image retrieval. However, this interest dates to the very recent past, hence the results obtained so far are less numerous and consolidated [13, 76, 101, 73, 20].

In the following, we will enrich the model presented so far with a RF capability that applies to global similarity retrieval, either on text or image layouts. We first present, in section 11.1, our approach to the problem in general terms, discussing how a feedback mechanism can be embedded in the model. Then, we move on to illustrate, in Sections 11.2 and 11.3, specific RF techniques for each of the two considered *media.* These techniques, analogously to all the other specific text or image techniques used by the model, are not new; rather, they are borrowed from the corresponding field and imported into the model in order to concretely show how integration amongst the various fields involved in MIR can be achieved. In this sense, the main contribution of this Section is Section 11.1, which presents how RF works in our model independently of any *medium*-dependent technique.

11.1 The approach

RF typically pertains to the form dimension of MIR, as it addresses the imprecision inherent in the usage of a layout (whether text or image) as a query. The application of RF to semantic-based retrieval is not appropriate, since this kind of retrieval hinges on logical reasoning, which is quite opposite to the "best match" inference on which form-based retrieval relies. This does not mean that imprecision is not present in semantic retrieval, but only that it is to be handled at the logical level, by appropriately defining the logical implication relation of the model. We have carried out work in this sense, and refer the interested reader to [58]. For the same reason, we do not consider RF on the structure dimension of retrieval, which is in fact a kind of exact retrieval, hence outside the scope of techniques such as RF.

Whether text or images are considered, RF consists of a few, basic steps, which are here outlined:

1. the user submits a query to the system;
2. the system returns the top k documents, $D_1, \ldots, D_k$, ordered according their RSVs. If the user is satisfied, then the retrieval session is over. Otherwise
3. on the top k documents, the user performs a relevance assessment, by indicating whether each document is relevant or irrelevant, and possibly to what extent. The user may also express no judgment on a document;
4. the system takes into account the user judgments by changing its internal status;

5. Step 2 is repeated in order to determine the top k documents according to the new system status.

The difference between text and images emerges in the way the system status is modified, but for the moment we can leave this aside, to the end of focusing on a general RF procedure for our model. Global, form-based retrieval, is captured in the model's query language via concrete visual queries or text similarity queries, respectively given by:

$$\exists \mathsf{SI}.\{\mathsf{qi}\} \quad \text{and} \quad \exists \mathsf{ST}.\{\mathsf{qt}\}.$$

In the query evaluation procedure, these queries generate, through the function Φ detailed in the last Section, global similarity assertions, *i.e.*assertions of the form:

- $\langle \mathsf{SI}(\mathsf{i},\mathsf{qi}), \sigma_i(\mathsf{i}^{\mathcal{I}},\mathsf{qi}^{\mathcal{I}})\rangle$, where qi is the query image, and i is an image of the document being evaluated that the decomposition of the query assertion has identified as a candidate image; or
- $\langle \mathsf{ST}(\mathsf{t},\mathsf{qt}), \sigma_t(\mathsf{t}^{\mathcal{I}},\mathsf{qt}^{\mathcal{I}})\rangle$, where qt and t are analogous to qi and i.

It follows then that, in the context of our model, relevance judgments impact on global similarity assertions, which, as argued, are the only assertions that reflect the kinds of queries suitable to RF.

Based on this consideration, Figure 12 shows the basic working of a relevance feedback mechanism for the model presented in the previous parts of this paper, relatively to a document doc. This Figure outlines the n-th RF iteration, and follows the same conventions as Figure 8, to which it directly relates.

RF begins with the user performing an assessment of relevance on a ranked list of documents (of course, this assessment is performed only once for each iteration, and not repeated for each document doc). In the first RF iteration, *i.e.* $n = 1$, the ranked list considered by the user is the one produced by the retrieval stage; for $n > 1$, the ranked list on which relevance is assessed is the result of the previous RF iteration.

As Figure 12 shows, the assessment of relevance produces two different judgement sets, those on text and those on images. In case the retrieved documents are simple, *i.e.* texts or images, then, clearly, one of these two sets is empty. But our model permits the retrieval of structured documents embodying texts and images, thus, in the general case, the user is able to express judgement on both images and texts. The case may arise in which the query contains more than one global similarity condition. There is no reason to rule out RF in this case: it suffices to assume that each similarity condition in the query is associated with the relevance judgements that pertain to it.

In illustrating the rest of the RF iteration following Figure 12, we will only refer to one *medium,* say images, leaving as understood that what we say applies as well to text. The central task of an RF iteration is carried out by the Image RF Module. This Module takes as input: (a) the relevance judgements on images, and (b) the current, *i.e.* n-th, doc's image global similarity assertions $\langle \mathsf{SI}(\mathsf{i},\mathsf{qi}), \sigma_i^{(n)}(\mathsf{i}^{\mathcal{I}},\mathsf{qi}^{\mathcal{I}})\rangle$ (for $n = 1$, these latter are the image global similarity assertions deriving from $\Phi(Q(\mathsf{doc})$, otherwise they result from the previous RF iteration). As output, the Image RF Module produces the $(n+1)$-th doc's image global similarity assertions $\langle \mathsf{SI}(\mathsf{i},\mathsf{qi}), \sigma_i^{(n+1)}(\mathsf{i}^{\mathcal{I}},\mathsf{qi}^{\mathcal{I}})\rangle$. Each RF iteration leaves therefore unchanged the logical part of each similarity assertion, *i.e.* $\mathsf{SI}(\mathsf{i},\mathsf{qi})$, which guides the DL TP to resolve the query assertion, while the truth-value, *i.e.* $\sigma_i^{(n)}(\mathsf{i}^{\mathcal{I}},\mathsf{qi}^{\mathcal{I}})$, is updated in order to reflect the user's relevance judgements. Sections 11.2 and 11.3 will present methods to compute the new truth-values for text and images, respectively.

After computing the new global similarity assertions, the procedure is in same conditions as the query evaluation procedure after the query decomposition and evaluation stage. Hence, the newly

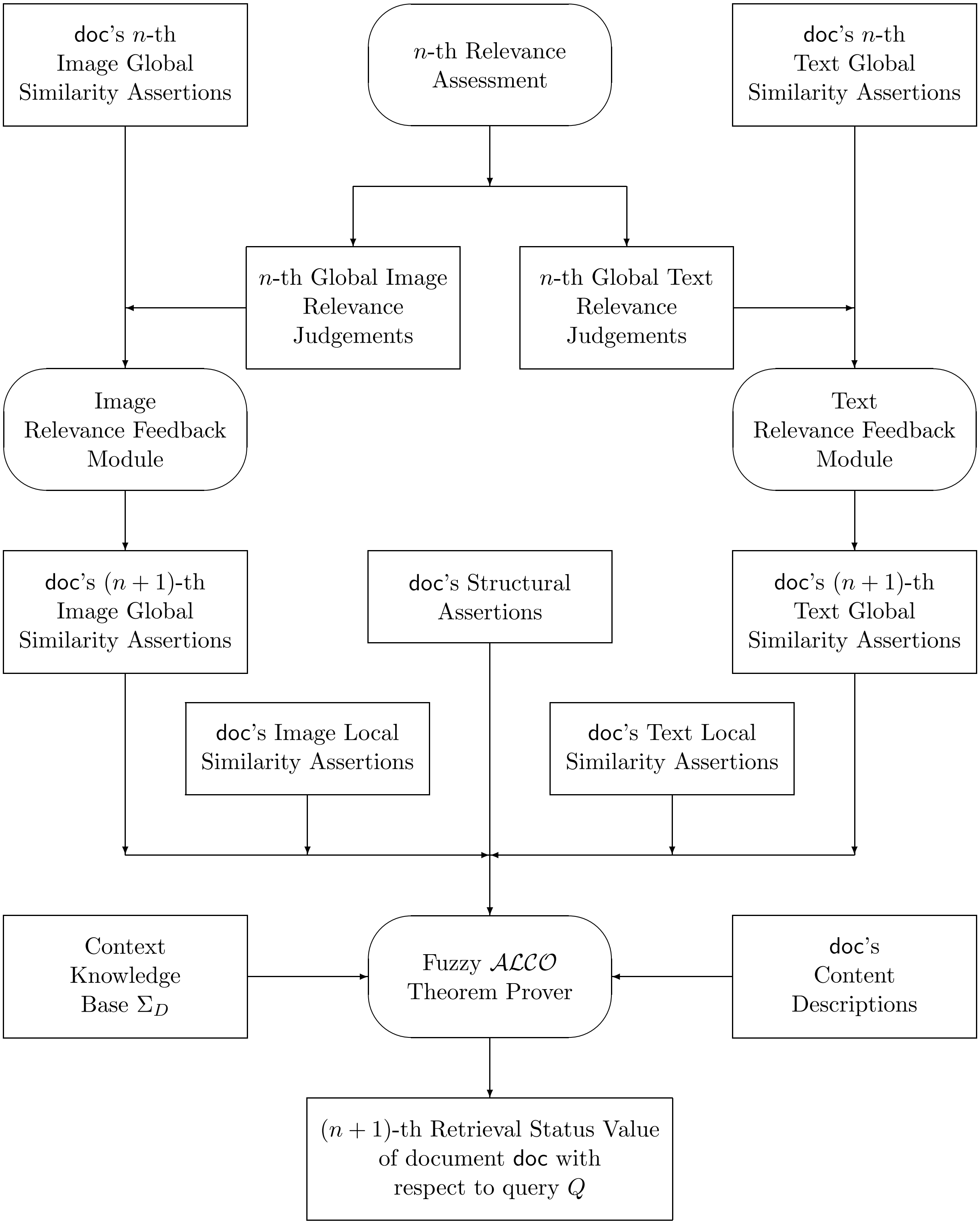

doc's n-th Image Global Similarity Assertions
n-th Relevance Assessment
doc's n-th Text Global Similarity Assertions
n-th Global Image Relevance Judgements
n-th Global Text Relevance Judgements
Image Relevance Feedback Module
Text Relevance Feedback Module
doc's $(n+1)$-th Image Global Similarity Assertions
doc's Structural Assertions
doc's $(n+1)$-th Text Global Similarity Assertions
doc's Image Local Similarity Assertions
doc's Text Local Similarity Assertions
Context Knowledge Base Σ_D
Fuzzy $\mathcal{ALCO}$ Theorem Prover
doc's Content Descriptions
$(n+1)$-th Retrieval Status Value of document doc with respect to query Q

Figure 12: The n-th relevance feedback iteration ($n \geq 1$).

calculated assertions are input to the DL TP, along with structural and local similarity assertions, which are untouched by RF. Clearly, it is possible that RF be centered around some local image similairty criterion, such as shape similarity; in this case, the RF iteration will produce, through an appropriate module, new local (shape) similarity assertions.

As the last step, the DL TP re-evaluates the query to produce the $(n+1)$-th document ranking.

11.2 Text relevance feedback

The aim of this Section is to come up with a suitable value for $\sigma_t^{(n+1)}(\mathsf{t}^{\mathcal{I}}, \mathsf{qt}^{\mathcal{I}})$, to be computed by the Text RF Module during the n-th RF iteration.

Typically, in text RF, relevance judgments are ternary, *i.e.* for each one of the top k documents, a user may:

1. indicate relevance; in this case the document ends up into a set RT_n of documents judged relevant at the n-th RF iteration;

2. indicate not relevance; in this case the document ends up into a set NT_n of documents judged not relevant at the n-th RF iteration;

3. give no indication; in this case the document is ignored by RF.

These judgments can be taken into account by the system in one of two ways: (1) by expanding the query with the addition of terms from the relevant document; or (2) by re-weighting the query terms based on the user relevance judgment. We opt for the latter choice, and so must specify how the weights of the query layout index are re-computed on the basis of the sets RT_n and NT_n. The 3 best options that exist on this aspect have been shown to offer similar performance improvements: among them we pick the Rocchio's formula.

Consequently, the function σ_t that computes the degree of similarity between two text layouts as a result of the n-th RF iteration, where the 0-th RF iteration is considered to be the retrieval stage, can be stated as follows:

$$\sigma_t^{(n+1)}(\mathsf{t}^{\mathcal{I}}, \mathsf{qt}^{\mathcal{I}}) = cos(\vec{\mathsf{t}}^{\mathcal{I}}, \vec{\mathsf{qt}}^{\mathcal{I}}_{(n+1)}).$$

The dynamic component of σ_t is the query layout, whose index, as said above, is re-computed at each RF iteration on the basis of the user judgments. Initially, for the retrieval stage, the query layout index is determined via one of the many indexing methods for text, such as the $tf-idf$ method mentioned earlier:

$$\vec{\mathsf{qt}}^{\mathcal{I}}_{(1)} = \vec{\mathsf{qt}}^{\mathcal{I}}.$$

The query layout index computed during the $n+1$-th RF iteration ($n \geq 1$), according to the Rocchio formula, is given by:

$$\vec{\mathsf{qt}}^{\mathcal{I}}_{(n+1)} = \alpha \cdot \vec{\mathsf{qt}}^{\mathcal{I}}_{(n)} + \beta \cdot \frac{1}{|RT_n|} \cdot \sum_{\mathsf{tl} \in RT_n} \vec{\mathsf{tl}}^{\mathcal{I}} - \gamma \cdot \frac{1}{|NT_n|} \cdot \sum_{\mathsf{tl} \in NT_n} \vec{\mathsf{tl}}^{\mathcal{I}}. \quad (38)$$

for all terms i, $1 \leq i \leq m$, where α, β and γ are suitable constants such that $\alpha + \beta - \gamma \leq 1$. In words, 38 says that the i-th term weight of the $(n+1)$-th query index is a linear combination of: the i-th term weight of the previous query index, the average i-th term weight of the layouts judged relevant, the average i-th weight of the layouts judged irrelevant. This very intuitive re-weighting scheme directly stems from the best query vector that can be used for retrieval in the ideal situation in which all relevant documents are known. Typically, α is set to 1, while β is chosen greater than γ as the information contained in judged relevant documents is usually more important than that given by documents judged not relevant.

Indeed, it has been experimentally shown that the relevance feedback process can account for improvements in retrieval effectiveness of up to 50 percent in precision for high-recall searches, and of approximately 20 percent in precision for low-recall searches.

11.3 Image relevance feedback

RF techniques for image retrieval differ substantially from those for text, due to the fact that both image indexes and image similarity functions are much more elaborated than the text ones. While decades of study and experimentation have established term weights-based representations and the cosine function as very reasonable, if not optimal, choices for text similarity, analogously solid choices for image similarity have still to be found. In fact, both the intuitive fact that images have a much richer perceptual content than text, as well as the experimental evidence gathered so far, seem to indicate that a representation of images that is, at the same time, as simple and as effective for retrieval as that of text, is unlikely to exist.

For these reasons, image indexes, in the most general case, are collections of features (*e.g.* color, texture, shape), each feature possibly being itself multiply represented (*e.g.* histogram and moments for color), each representation being a multidimensional object, such as a vector, of its own. As a consequence, image similarity functions are typically linear combinations of distance measures relative to the single features. Under these circumstances, RF can be used to calculate what emphasis should be given to each feature, to each representation of the same feature, and to each component with each feature representation [73]. Other usages of the user judgments are possible, such as for altering the query index [20]; however, given the composite nature of such indexes, emphasizing and de-emphasizing their components according to the user preferences would seem more appropriate than numerically manipulating them.

In order to concretely illustrate RF on images, in the rest of this Section we will introduce the image similarity function used by the prototype Arianna (fully described in Section 12), and show how the RF technique presented in [73] can be applied to this function. Although the function only considers one feature (color), singularly represented via moments, what follows will suffice to demonstrate how an image RF technique can be integrated in our model.

As for text, the function σ_i for computing global image similarity is the same, whether the calculation is performed in the retrieval stage or in the course of a RF iteration, the former being considered, as usual, the 0-th RF iteration. Then, the similarity between two image layouts i and i' resulting from the n-th RF iteration, is given by:

$$\sigma_i(i,i')^{(n+1)} = \sum_{j \in HSB} w_{1j}^{(n+1)} \cdot |\mu_{1j}^{i} - \mu_{1j}^{i'}| + w_{3j}^{(n+1)} \cdot |\mu_{2j}^{i} - \mu_{2j}^{i'}| + w_{3j}^{(n+1)} \cdot |\mu_{3j}^{i} - \mu_{3j}^{i'}|, \tag{39}$$

where μ_{kj} is the normalization of m_{kj}, performed in order to assign equal emphasis on each color moment, and $w_{ij}^{(n+1)}$ are the $(n+1)$-th weights. The dynamic components of $\sigma_i(i,i')^{(n+1)}$ are, as announced, its weights $w_{ij}^{(n+1)}$, which are affected by RF. Initially, for the retrieval stage, the weights are fixed according to an "objective" criterion, assigning equal emphasis to each color moment. Hence:

$$w_{ij}^{(1)} = \tfrac{1}{9} \qquad \text{for } 1 \leq i,j \leq 3.$$

The resulting $\sigma_i^{(1)}$ is the truth-value associated with each image global similarity assertion generated during query evaluation. Let us now see how $w_{ij}^{(n+1)}$ is computed, for $n \geq 1$.

In the n-th relevance assessment, the user is presented the top k images, on which relevance judgment is expressed, resulting in the set RI_n, analogous to the set RT_n obtained for text[5]. The idea is to consider the vector $V_{ij}^{(n)}$ containing the values of the color moment μ_{ij} for all the images in RI_n. If all such images have similar values for μ_{ij}, then μ_{ij} can be considered as a good indicator of the user's information need; instead, if the $V_{ij}^{(n)}$ values differ significantly, μ_{ij} does not look like as a good indicator. Letting $\tau_{ij}^{(n)}$ stand for the standard deviation of $V_{ij}^{(n)}$, we then set:

$$w_{ij}^{(n+1)} = \frac{1}{\tau_{ij}^{(n)} \cdot W}$$

where W is a normalization factor given by the sum of all weights $w_{ij}^{(n+1)}$.

Experimental results reported in [73] show that this RF technique offers a significant improvement in retrieval performance. These improvements mostly emerge after one RF iteration, while successive iterations only produce marginal benefits.

12 A prototypical implementation of the model's image retrieval capability

The prototype system that we have developed, named Arianna, implements the form- and content-based retrieval of images, thus addressing one of the fundamental traits of the model, namely the integration of several kinds of image retrieval into a unique framework. Arianna consists of two main modules: the *indexing* module (hereafter IM for short) supporting the acquisition of images and the creation of the various representations needed for performing retrieval, and the *query* module (QM), performing query evaluation.

12.1 The indexing module

Figure 13 illustrates the various operations comprising image acquisition in what may be considered as the typical sequence (in this figure, rectangular boxes represent data, while ovals represent modules).

Filtering and Size Reduction Acquisition begins from an *Input Image*, which in our case may be any image in GIF or JPEG format. As a first step, the input image is reduced, if necessary, to the size handled by the system, which is a parameter currently fixed to 128×128 pixels. The reduction is performed by means of a re-sampling technique. After size reduction, the RGB color space is abandoned in favor of the HSB space, and noise reduction is performed on the image, by applying a color median filter. As a result, the *Basic Image* is produced. Figure 14 presents a sample input image (left) and the corresponding basic image.

Segmentation The task of the Segmentation Module is to derive the Image Layout from the Basic Image. The Image Layout is used solely to support the user in specifying the image regions that are to be annotated via Represents assertions. This may be surprising to the uninitiated to form-based image retrieval: given that the model supports retrieval by color patches, it is natural to expect that this kind of retrieval be implemented on top of the Image Layout. Unfortunately,

[5] In [73], a richer model is presented, able to cope with multi-featured and multi-representations similarity functions, and allowing 5-ary relevance assessment. Since our similarity model is much simpler, the extra judgment levels would not be used, hence we have not considered them.

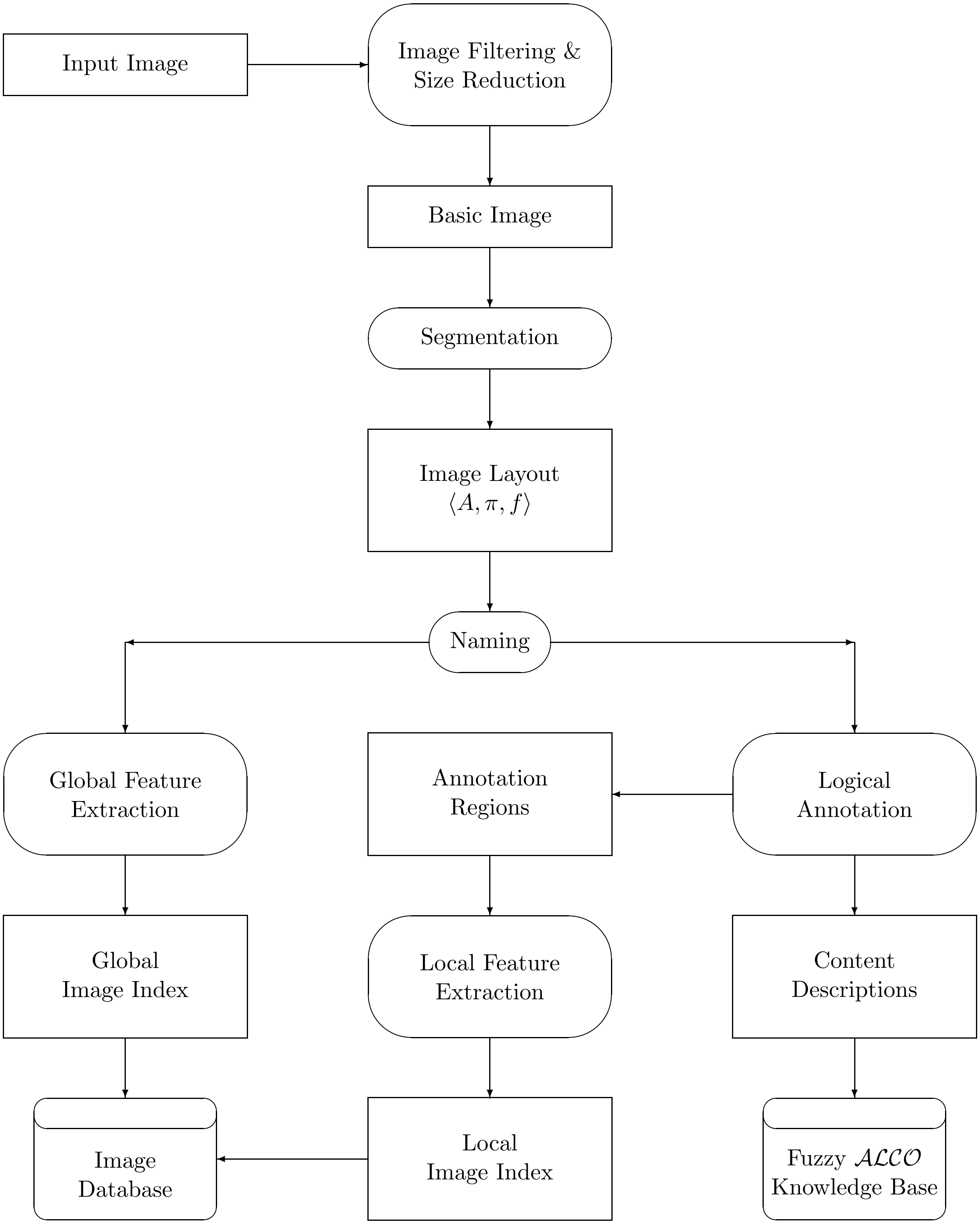


Figure 13: Image acquisition in Arianna

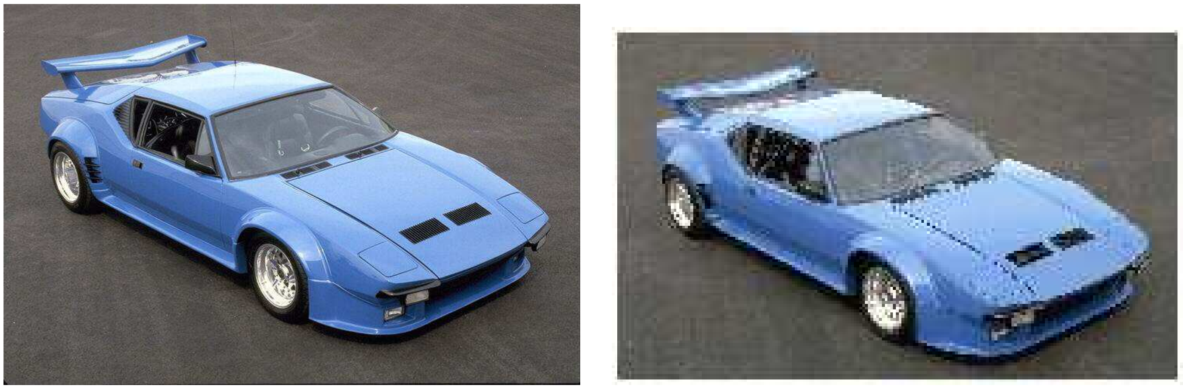


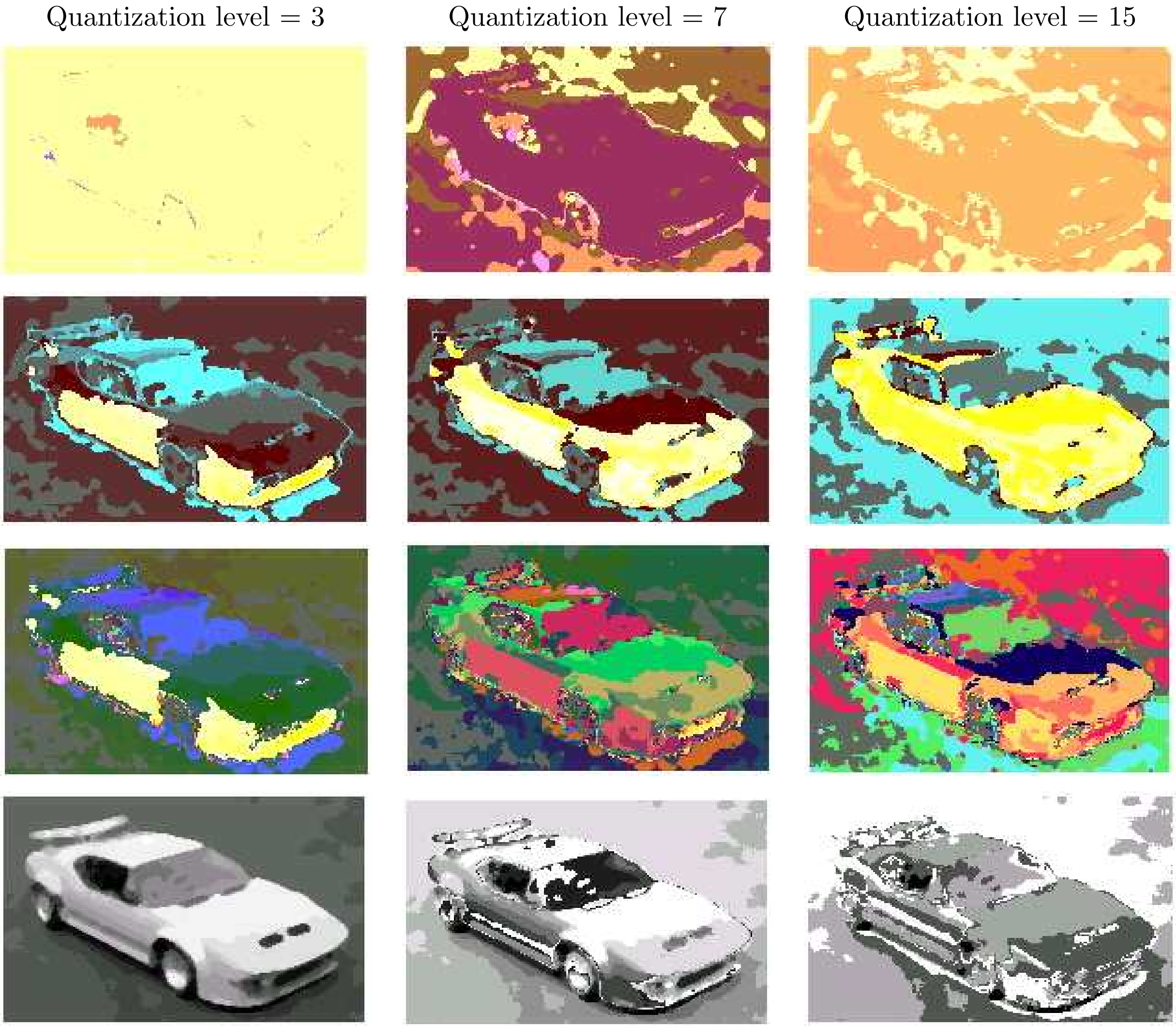
Figure 14: A Sample Input Image and the corresponding Basic Image

the number of atomic regions in an image tends to be very large, often it is of the same order of magnitude as the image size, as it is very well known in the image retrieval context. In fact, image retrieval systems implement retrieval by color patches by relying on various approximations, aiming at cutting down the size of the computational space. Arianna is no exception.

The derivation of the Image Layout implies two operations: segmentation and color quantization. These operations are strictly related, and in fact they are both performed by the Segmentation Module. As already pointed out in Section 5.1, the segmentation of color images is still an open problem, for which no universally valid algorithm is currently known. Successful techniques have been developed for specific image types. However, 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 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 14 are showed in Figure 15, 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 three levels of quantization: 2, 7 and 15 colors. Edge detection techniques have been employed to obtain these segmentations (see Figure 16). Finally, a segmentation on texture is derived, at two levels of quantization (see Figure 17). 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 18 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; in particular, the central image is the input image, and is surrounded by the 7 different segmentations introduced above.

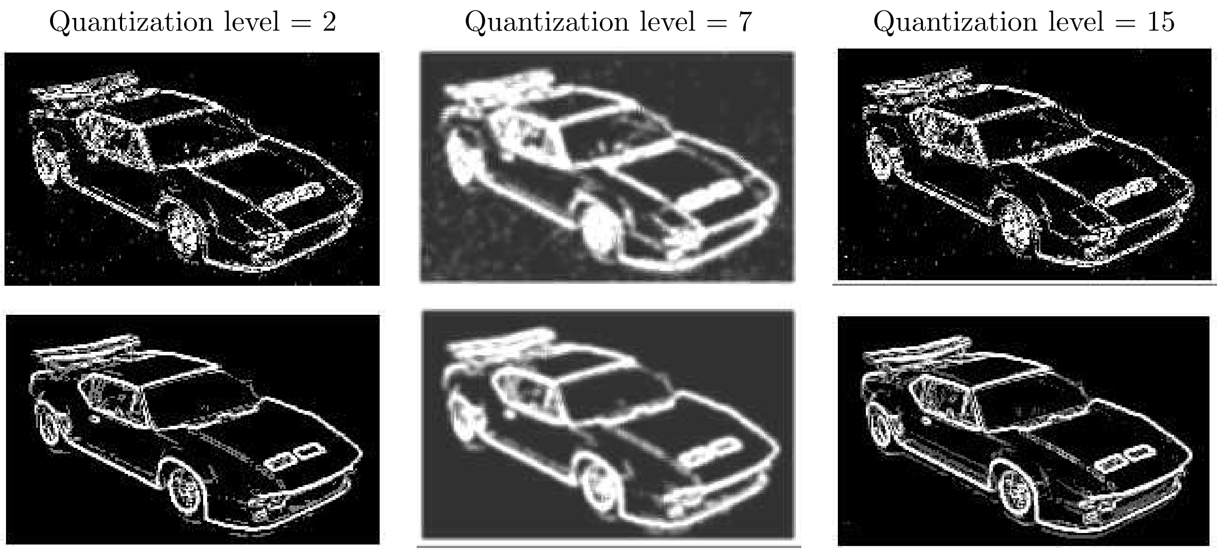


Quantization level = 3

Quantization level = 7

Quantization level = 15

Figure 15: Segmentation by color, saturation, color and saturation, and brightness at 3 levels of quantization.



Quantization level = 2

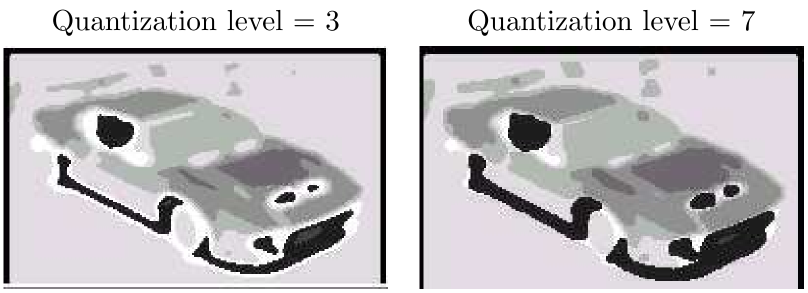
Quantization level = 7

Quantization level = 15

Figure 16: Segmentations by edge at 3 levels of quantization.

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.

Naming Prior to any indexing operation, the derived Image Layout must be identified as an individual of fuzzy $\mathcal{ALCO}$, and this is the objective of the Naming operation. When this operation is requested, the user is asked to give a name for the image being acquired; the system validates the proposed name by checking that it is not used as the name of another image. From that point on, the name becomes the unique image identifier and two operations are possible: *Global Feature*



Quantization level = 3

Quantization level = 7

Figure 17: Segmentations by texture at 2 levels of quantization.

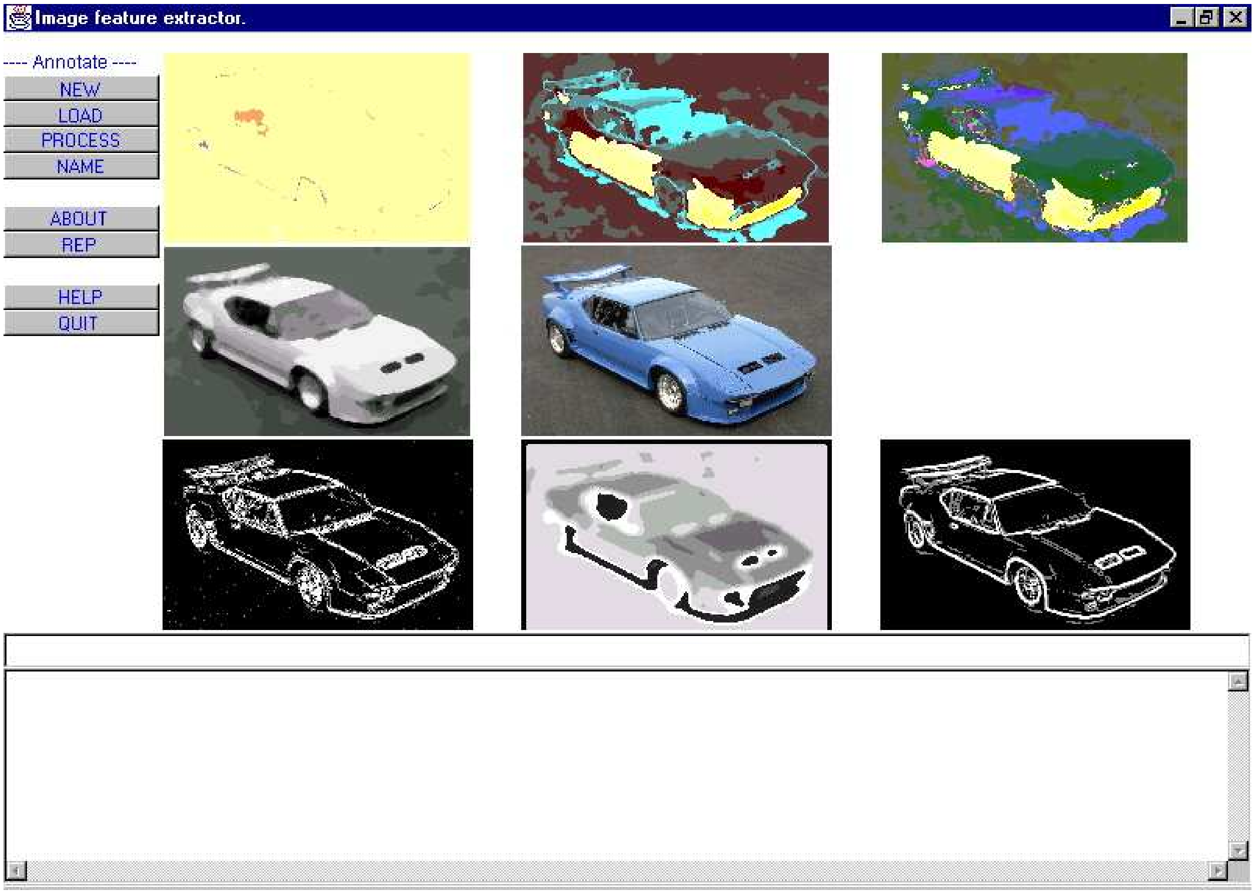


Figure 18: The screen shown by IM after segmenting the sample image.

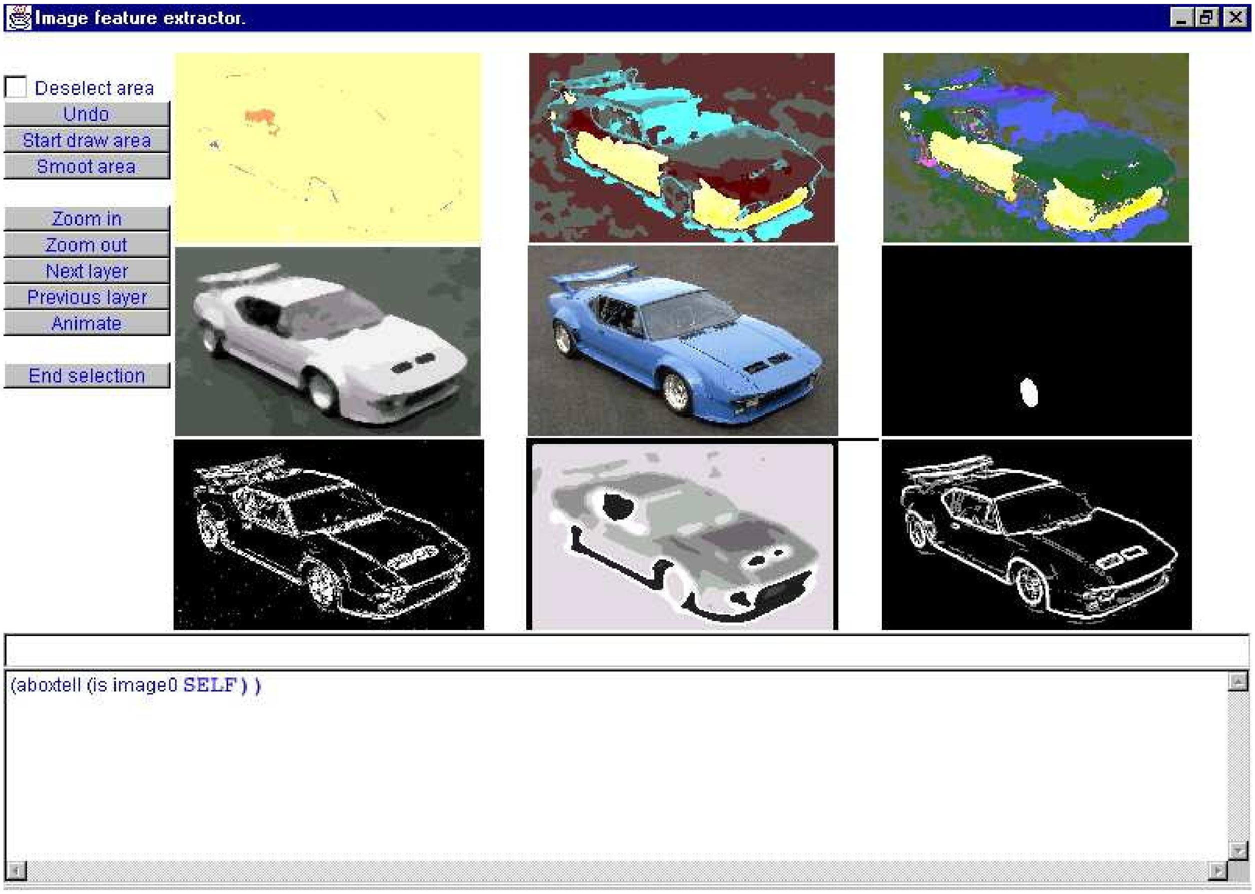
Extraction and *Logical Annotation*.

Global Feature Extraction This operation 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*. According to the definition of the Φ function given in Figure 10, the following features are extracted from the Basic Image named i:

- The first three moments of the image (true) color histogram for each channel of the HBS color space. These features are extracted in order to compute the image similarity function σ_i, given in Section 5.2.

- The list of colors occurring in the image; this is used in order to process color queries on atomic regions, *i.e.* $(\exists \mathsf{HAIR}.\exists \mathsf{HC}.\{\mathsf{c}\})(\mathsf{i})$.

- In order to evaluate queries on color similarity (*i.e.* $(\exists \mathsf{HAIR}.\exists \mathsf{HC}.\exists \mathsf{SC}.\{\mathsf{c}\})(\mathsf{i})$), the vector V is extracted, defined as follows. 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, *i.e.* $max_{T \in \pi}\{\sigma_c(f(T), \mathsf{c}^{\mathcal{I}})\}$, as required



(aboxtell (is image0 SELF))

Figure 19: The screen shown by the IM during region selection.

by Φ. The distance measure used as color similarity function σ_c is the normalization in the [0,1] interval of:

$$\sum_{c} \sum_{k=1}^{3} |m_{kc}^{\mathsf{i}^{\mathcal{I}}} - m_{kc}^{\mathsf{j}^{\mathcal{I}}}|$$

where c ranges on the 3 channels of the color space.

Logical Annotation This operation permits the specification of one or more *Content Descriptions* for named images. Upon requesting it, IM automatically creates the layout identification assertion (*i.e.* $\langle \mathsf{Self}(\mathsf{i}), 1\rangle$), and supports the creation of the other kinds of assertions. In particular:

- 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.

- 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 Represents 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 19 shows the IM screen during the selection of a region to be annotated via a Represents 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*, which is filed in an apposite archive of the *Image Database.* The structure of the local image index, relatively to the annotation region r, is as follows (see Figure 10):

- The name of the region r.
- The region color histogram, used to process color queries on the region (*i.e.* $(\exists\mathsf{HC}.\{\mathsf{c}\})(\mathsf{r})$). 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.
- 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 $\mathsf{c}^{\mathcal{I}}$, $T_{\mathsf{c}^{\mathcal{I}}}$, gives the maximum, for all colors l, of the values $v(\mathsf{c}^{\mathcal{I}}, l)$, each of which is the minimum between the percentage of l in the region r and the similarity between l and $\mathsf{c}^{\mathcal{I}}$, *i.e.* $T_{\mathsf{c}^{\mathcal{I}}} = max_{l \in \mathcal{C}}\{min\{f_e(\mathsf{r}^{\mathcal{I}})(l), \sigma_c(l, \mathsf{c}^{\mathcal{I}})\}\}$, as given in Figure 10.
- the shape of the region represented by the 8-contour and by 7 invariant moments [61]. 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.* $(\exists\mathsf{HS}.\exists\mathsf{SS}.\{\mathsf{s}\})(\mathsf{r})$). In this latter case, the similarity function σ_s is the Euclidean distance between the 7 moments, normalized in the [0,1] interval.

12.2 The query module

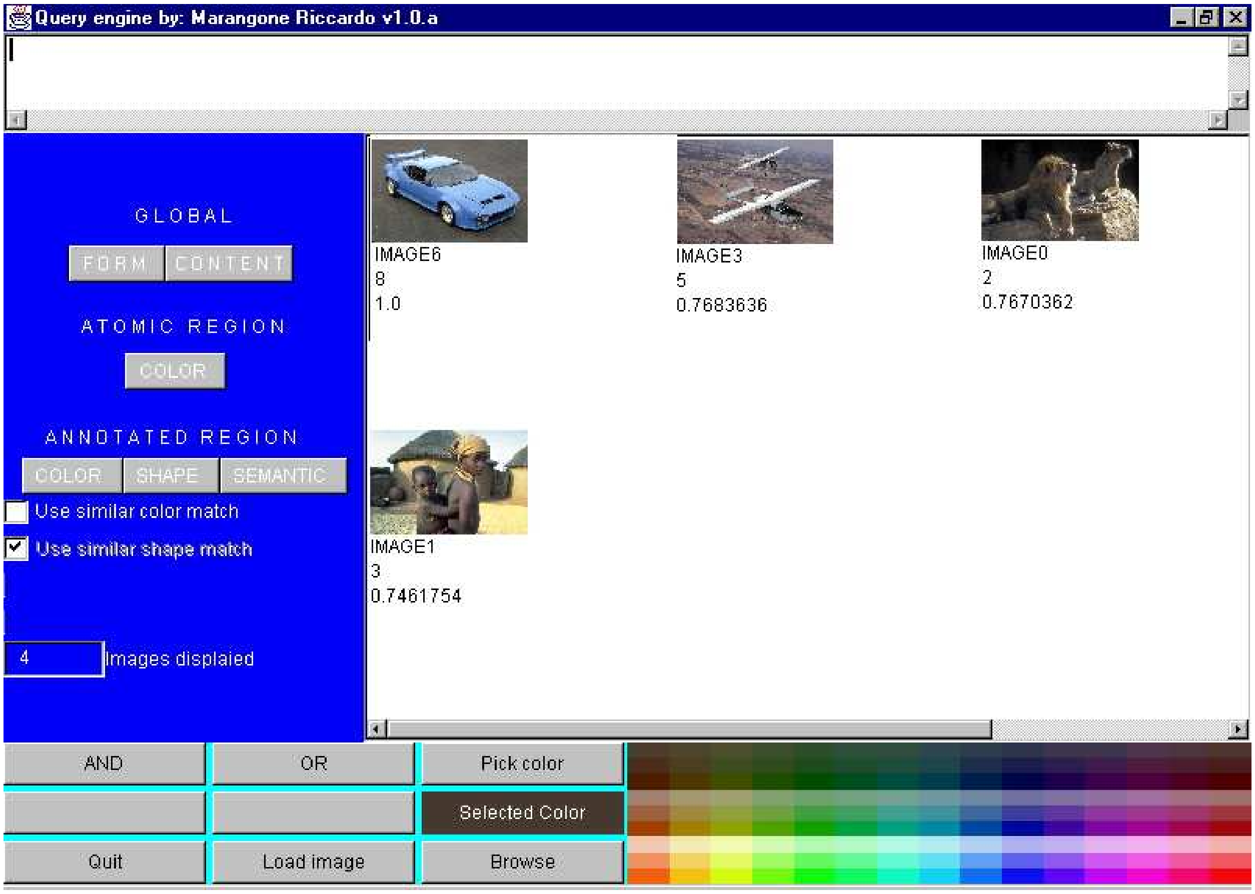
The query module QM provides two basic services: query specification and query evaluation.

12.2.1 Query specification

Query specification supports the construction of image queries, according to the syntax given in Figure 6. The specification is performed by the user via the interface shown in Figure 20. During query specification, the system keeps track of the indications given by the user, so that, at the end of the specification, it can automatically construct the query.

Following the syntax of the image query language (Figure 6, from the top down), there are two buttons (labeled "AND" and "OR") in the bottom left part of the query panel, which the user must first select in case the query involves conjunctions or disjunctions image concepts, which are the elementary queries. Image concepts are specified in the middle left area of the panel, which is divided in three main parts that strictly reflect the language syntax:

- a part labeled "GLOBAL", which is to be used for specifying concepts involving a whole image: these, in turn, can be of two types, corresponding to the two buttons in this area of the panel:



☐ Use similar color match

☑ Use similar shape match

4 Images displaied

IMAGE6
8
1.0

IMAGE3
5
0.7683636

IMAGE0
2
0.7670362

IMAGE1
3
0.7461754

Figure 20: The QM displaying the result of a query.

- "Content", *i.e.* $\exists \mathsf{About}.C$, where C is a content concept, *i.e.* a standard $\mathcal{ALCO}$ concept on image contents. The specification of such concept is given in the top part of the query panel by inputing the concept in textual form.
- "Form", *i.e.* $\exists \mathsf{SI}.\{\mathsf{qi}\}$. The query image qi is indicated by selection from the display area (the widest area in the panel), where it can be loaded via the "Load Image" button (bottom left corner).

- a part labelled "ATOMIC REGION", which is to be used for specifying a query of the form $\exists \mathsf{HAIR}.(\exists \mathsf{HC}.D)$, where D is a color concept. In fact, there is only one button in this area, tagged "COLOR". If this button is used with the "Use similar color match" box not checked, then it is understood that D is just a color specification, *i.e.* $\{\mathsf{c}\}$. If, on the other hand, the "COLOR" button is clicked with the "Use similar color match" box checked, then it is understood that D is of the form $\exists \mathsf{SC}.\{\mathsf{c}\}$. In either case, c is specified by clicking on the "Pick color" button and then selecting a color in the color space depicted next to this button.

- a part labeled "ANNOTATED REGION", which is to be used for specifying queries on image regions that have been previously annotated with a Represents assertions. The three buttons in this area, have the following usage:
  - clicking on the "SEMANTIC" button, implies that an image concept of the form

$\exists\mathsf{HIR}.(\exists\mathsf{Represents}.C)$ is being specified, where C is a content concept that the user is asked to textually input in the topmost part of the query panel;

- clicking on the "COLOR" button, implies that an image concept of the form $\exists\mathsf{HIR}.(\exists\mathsf{Represents}.(C \sqcup \exists\mathsf{HC}.D)$ is being specified, where C is a content concept and D is a color concept (the specification of these two kinds of concepts has been illustrated above);
- clicking on the "SHAPE" button, implies that an image concept of the form $\exists\mathsf{HIR}.(\exists\mathsf{Represents}.(C \sqcup \exists\mathsf{HS}.D)$ is being specified, where C is a content concept and D is a shape concept. Concepts of the latter kind are input to the system in a perfectly analogous way to color concepts, except that the user is asked to draw the shape.

12.2.2 Query evaluation

Query evaluation is performed by the Image Query Evaluation Procedure, which is in fact a restriction of the Query Evaluation Procedure previously illustrated (see Figure 8) to image queries. More specifically, given an image query Q, the Image Query Assertion Builder iteratively produces a query assertion $Q(\mathsf{i})$ for each acquired image i. The image assertion 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 10.

Essentially, in deriving $\Phi(Q(\mathsf{i}))$, which in the present case only includes Image Assertions, the IQDE accesses the Image Database in order to fetch the Global Image Index of i and the Local Image Index of each of i's annotation regions, both built during i's acquisition. 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 $T_{\mathsf{c}\mathcal{I}}$ vector position, part of r's local index.

The last step of the evaluation procedure is the invocation of the TP, to which the query assertion $Q(\mathsf{i})$ is sent with the purpose of computing its m value against: (a) the context knowledge base Σ_D, (b) i's content descriptions (both these are part of the Fuzzy $\mathcal{ALCO}$ Knowledge Base maintained by the TP); and (c) the just computed $\Phi(Q(\mathsf{i}))$.

13 Conclusions

We have presented a view of multimedia information retrieval that reconciles in a unique, well-founded framework the many functionalities that are found under the MIR label. The view has been introduced both at the informal and formal level, the latter taking the shape of a logical model endowed with a syntax and a semantics. Implementation of the model has been discussed, and a simple prototype of multi-modal image retrieval has been presented.

At the technical level, the model makes two important contributions. First, at single-medium level, it makes full and proper use of semantics and knowledge in dealing with the retrieval of images and text, 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. More importantly, all these forms of retrieval coexist in a well-founded framework, which combines in a neat way the different techniques, notably digital signal processing and semantic information processing, required to deal with the various aspects of the model. Secondly, at the multimedia level, the model addresses the retrieval of structural aggregates of images and texts, casting the single medium models in a framework informed by the same, few principles. 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.* [40]) 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 [21]. Analogously, the model can be enhanced with the treatment of texture-based similarity image retrieval.

We believe that the presented model can open the way to a novel approach to the modelling of multimedia information, leading to the development of retrieval systems able to cope in a formally neat and practically adequate way with documents including text and images. More research is needed to attack delay-sensitive media, such as audio and video, but we think that the present model offers a suitable framework for the development of conceptual models for these media.

14 Acknowledgments

Partial support form the European Union was kindly provided under the MIRO Working Group (n. 6175) and the FERMI Action (n. 8134), both part of the ESPRIT Basic Research Programme. We thank our teammates in these two actions for the many stimulating discussions. Finally, our thanks to Riccardo Marangone, who developed ARIANNA, and to Antonio Lopreiato, who developed the fuzzy $\mathcal{ALCO}$ theorem prover.

References

[1] S. Abiteboul, S. Cluet, V. Christophides, T. Milo, G. Moerkotte, and J. Simeon. Querying documents in object databases. *International Journal on Digital Libraries*, 1(1):5–19, 1997.

[2] M. Aiello, C. Areces, and M. Rijke. Spatial reasoning for image retrieval. In *Proceedings of the International Workshop on Description Logics*, pages 23–27, Linkoeping, Sweden, 1999.

[3] Y. Alp Aslandogan, C. Thier, C. Yu, J. Zou, and N. Rishe. Using semantic contents and WordNet in image retrieval. In *Proceedings of SIGIR-97, 20th ACM Conference on Research and Development in Information Retrieval*, pages 286–295, Philadelphia, US, 1997.

[4] G. Amati, F. Crestani, and F. Ubaldini. A learning system for selective dissemination of information. In *Proceedings of IJCAI-97, 15th International Joint Conference on Artificial Intelligence*, pages 764–769, Nagoya, Japan, 1997.

[5] A. Anderson and N. Belnap. *Entailment - the logic of relevance and necessity.* Princeton University Press, Princeton, US, 1975.

[6] F. Baader and P. Hanschke. A schema for integrating concrete domains into concept languages. In *Proceedings of IJCAI-91, International Joint Conference on Artificial Intelligence*, pages 452–457, Sydney, AU, 1991.

[7] F. Baader and B. Hollunder. KRIS: Knowledge representation and inference system - system description. *ACM SIGART Bulletin*, 2(3):8–14, 1991.

[8] J. R. Bach, C. Fuller, A. Gupta, A. Hampapur, B. Horowitz, R. Humphrey, R. Jain, and C.-F. Shu. The Virage image search engine: An open framework for image management. In *Proceedings of SPIE-96, 4th SPIE Conference on Storage and Retrieval for Still Images and Video Databases*, pages 76–87, San Jose, US, 1996.

[9] C. Berrut, F. Fourel, M. Mechkour, P. Mulhem, and Y. Chiaramella. *Indexing, navigation and retrieval of multimedia structured documents : the PRIME information retrieval system.* ESPRIT III Basic Research Action n. 8134 (*FERMI: Formalization and Experimentation in the Retrieval of Multimedia Information*), Deliverable D11, 1997.

[10] K. Bharat, T. Kamba, and M. Albers. Personalized, interactive news on the web. *Multimedia Systems*, (6):349–358, 1998.

[11] T. Bollinger and U. Pletat. The LILOG knowledge representation system. *ACM SIGART Bulletin*, 2(3):22–27, 1991.

[12] A. Borgida. Description logics in data management. *IEEE Transactions on Knowledge and Data Engineering*, 7:671–682, 1995.

[13] M. Bouet and C. Djeraba. Visual content based retrieval in image databases with relevance feedbacks. In *Proceedings of the International Workshop on Multi-Media Database Management Systems*, pages 98–105, Dayton, USA, August 1998.

[14] R. J. Brachman, D. L. McGuinness, P. F. Patel-Schneider, L. Alperin Resnick, and A. Borgida. Living with CLASSIC: when and how to use a KL-ONE-like language. In J. F. Sowa, editor, *Principles of Semantic Networks*, pages 401–456. Morgan Kaufmann, Los Altos, US, 1991.

[15] R. J. Brachman and J. G. Schmolze. An overview of the KL-ONE knowledge representation system. *Cognitive Science*, 9(2):171–216, 1985.

[16] R. J. Brachman, P. G. Selfridge, L. G. Terveen, B. Altman, A. Borgida, F. Halpern, T. Kirk, A. Lazar, D. L. McGuinness, and L. A. Resnick. Knowledge representation support for data archaeology. In Y. Yesha, editor, *Proceedings of CIKM-92, International Conference on Information and Knowledge Management*, pages 457–464, 1992.

[17] M. Buchheit, F. M. Donini, and A. Schaerf. Decidable reasoning in terminological knowledge representation systems. pages 704–709, Chambery, France, 1993. Morgan Kaufmann, Los Altos, US.

[18] P. Buongarzoni, C. Meghini, R. Salis, F. Sebastiani, and U. Straccia. Logical and computational properties of the description logic MIRTL. In A. Borgida, M. Lenzerini, D. Nardi, and B. Nebel, editors, *Proceedings of DL-95, 4th International Workshop on Description Logics*, pages 80–84, Roma, IT, 1995.

[19] Y. Chiaramella, P. Muhlem, and F. Fourel. A model for multimedia information retrieval. Technical report, ESPRIT III Basic Research Action n. 8134 (*FERMI: Formalization and Experimentation in the Retrieval of Multimedia Information*) Technical Report, 1996.

[20] R. Ciocca and R. Schettini. Using a relevance feedback mechanism to improve content-based image retrieval. In *Proceedings of Vsual99, International Conference on Visual Information Systems*, number 1614 in Lecture Notes in Computer Science, pages 107–114. Springer Verlag, 1999.

[21] A. G. Cohn. Calculi for qualitative spatial reasoning. In *Proceedings of AISMC-93, the 3rd International Conference on Artificial Intelligence and Symbolic Mathematical Computation*, Lecture Notes in Computer Science, Steyr, AT, 1996. Springer Verlag, Heidelberg, DE.

[22] F. Crestani, M. Lalmas, and C. J. van Rijsbergen, editors. *Logic and uncertainty in information retrieval: Advanced models for the representation and retrieval of information.* Kluwer Academic Publishing, Dordrecht, NL, 1998. Forthcoming.

[23] D. Davidson. Truth and meaning. collected essays by donald davidson. In *Inquiries into truth and interpretation*, pages 17–36. Clarendon Press, Oxford, UK, 1991.

[24] A. Del Bimbo, M. Mugnaini, P. Pala, F. Turco, and L. Verzucoli. Image retrieval by color regions. In A. Del Bimbo, editor, *Proceedings of ICIAP'97, 9th International Conference on Image Analysis and Processing*, number 1311 in Lecture notes in computer science, pages 180–187, Firenze, IT, 1997. Springer Verlag, Heidelberg, DE.

[25] A. Del Bimbo and P. Pala. Visual image retrieval by elastic matching of user sketches. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(2):121–132, 1997.

[26] P. Devanbu, R. J. Brachman, P. J. Selfridge, and B. W. Ballard. LASSIE: A knowledge-based software information system. *Communications of the ACM*, 34(5):36–49, 1991.

[27] DL. Description Logic Web Home Page: `http://dl.kr.org/dl`.

[28] D. Dubois and H. Prade. *Fuzzy Sets and Systems.* Academic Press, New York, US, 1980.

[29] M. O. Duschka and Y. A. Levy. Recursive plans for information gathering. In *Proceedings of IJCAI-97, 15th International Joint Conference on Artificial Intelligence*, pages 778–784, Nagoya, JP, 1997.

[30] C. Faloutsos. *Searching Multimedia Databases by Content.* Kluwer Academic Publishers, Dordrecht, NL, 1996.

[31] C. Faloutsos, R. Barber, M. Flickner, J. Hafner, and W. Niblack. Efficient and effective querying by image content. *Journal of Intelligent Information Systems*, 3:231–262, 1994.

[32] L. Fitzpatrick and M. Dent. Automatic feedback using past queries: Social searching? In *Proceedings of SIGIR-97, 20th ACM Conference on Research and Development in Information Retrieval*, pages 306–313, Philadelphia,PA, July 1997.

[33] N. Fuhr and C. Buckley. A probabilistic learning approach for document indexing. *ACM Transactions on Information Systems*, 9:223–248, 1991.

[34] N. Fuhr and T. Rölleke. Information retrieval with Probabilistic Datalog. In F. Crestani, M. Lalmas, and C. J. van Rijsbergen, editors, *Logic and Uncertainty in Information Retrieval: Advanced models for the representation and retrieval of information.* Kluwer Academic Publishing, Dordrecht, NL, 1998. Forthcoming.

[35] C. A. Goble, C. Haul, and S. Bechhofer. Describing and classifying multimedia using the description logic GRAIL. In *Proceedings of SPIE-96, 4th SPIE Conference on Storage and Retrieval for Still Images and Video Databases*, pages 132–143, San Jose, US, 1996.

[36] A. Goker and T. McCluskey. Towards an adaptive information retrieval system. In Z. W. Ras and M. Zemenkova, editors, *Proc. of the 6th Int. Sym. on Methodologies for Intelligent Systems (ISMIS-91)*, number 542 in Lecture Notes in Artificial Intelligence, pages 349–357. Springer Verlag, 1991.

[37] V. Gudivada and V. Raghavan. Special issue on content-based image retrieval. *IEEE Computer*, 28(9), 1995.

[38] V. Gudivada and V. Raghavan. Modeling and retrieving images by content. *Information Processing and Management*, 33(4):427–452, 1997.

[39] V. N. Gudivada. Multimedia systems – an interdisciplinary perspective. *ACM Computing Surveys*, 27(4):545–548, 1995.

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

[41] E. J. Guglielmo and N. C. Rowe. Natural-language retrieval of images based on descriptive captions. *ACM Transactions on Information Systems*, 14(3):237–267, 1996.

[42] A. Gupta, S. Santini, and R. Jain. In search of information in visual media. *Communications of the ACM*, 40(12):34–42, 1997.

[43] K. Hirata and T. Kato. Query by visual example. In *Proceedings of EDBT-92, 3rd International Conference on Extending Database Technology*, pages 56–71, Wien, AT, 1992.

[44] A. Jain and A. Vailaya. Image retrieval using color and shape. *Pattern Recognition*, 29(8):1233–1244, 1996.

[45] T. Kindo, H. Yoshida, T. Morimoto, and T. Watanabe. Adaptive personal information filtering system that organizes personal profiles automatically. In *Proceedings of IJCAI-97, 15th International Joint Conference on Artificial Intelligence*, pages 716–721, Nagoya, Japan, 1997.

[46] S. Kundu and J. Chen. Fuzzy logic or Łukasiewicz logic: A clarification. In Z. W. Ras and M. Zemenkova, editors, *Proceedings of ISMIS-94, 8th International Symposium on Methodologies for Intelligent Systems*, number 869 in Lecture Notes in Artificial Intelligence, pages 56–64. Springer Verlag, 1994.

[47] R. C. T. Lee. Fuzzy logic and the resolution principle. *Journal of the ACM*, 19(1):109–119, 1972.

[48] D. D. Lewis. An evaluation of phrasal and clustered representations on a text categorization task. In *Proceedings of SIGIR-92, 15th ACM International Conference on Research and Development in Information Retrieval*, pages 37–50, Kobenhavn, DK, 1992.

[49] F. Liu and R. Picard. Periodicity, directionality, and randomness: Wold features for image modelling and retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(7):722–733, 1996.

[50] Z. Liu and J. Sun. Structured image retrieval. *Journal of Visual Languages and Computing*, 8:333–357, 1997.

[51] R. MacGregor. Inside the LOOM description classifier. *SIGART Bulletin*, 2(3):88–92, 1991.

[52] S. Marcus and V. Subrahmanian. Foundations of multimedia information systems. *Journal of the ACM*, 43(3):474–523, 1996.

[53] D. L. McGuinness and J. Wright. An industrial strength description logic-based configurator platform. *IEEE Intelligent Systems*, 13(4):69–77, July/August 1998.

[54] C. Meghini. An image retrieval model based on classical logic. In *Proceedings of SIGIR-95, 18th ACM Conference on Research and Development in Information Retrieval*, pages 300–308, Seattle, US, 1995.

[55] C. Meghini, F. Rabitti, and C. Thanos. Conceptual modelling of multimedia documents. *IEEE Computer. Special Issue on Multimedia Systems*, 24(10):23–30, 1991.

[56] C. Meghini, F. Sebastiani, and U. Straccia. Modelling the retrieval of structured documents containing texts and images. In C. Peters and C. Thanos, editors, *Proceedings of ECDL-97, the First European Conference on Research and Advanced Technology for Digital Libraries*, number 1324 in Lecture notes in computer science, Pisa, IT, 1997. Springer Verlag, Heidelberg, DE.

[57] C. Meghini, F. Sebastiani, and U. Straccia. The terminological image retrieval model. In A. Del Bimbo, editor, *Proceedings of ICIAP'97, 9th International Conference On Image Analysis And Processing*, number 1311 in Lecture notes in computer science, pages 156–163, Firenze, IT, 1997. Springer Verlag, Heidelberg, DE.

[58] C. Meghini, F. Sebastiani, and U. Straccia. Mirlog: a logic for multimedia information retrieval. In F. Crestani, M. Lalmas, and C. J. van Rijsbergen, editors, *Logic and Uncertainty in Information Retrieval: Advanced models for the representation and retrieval of information.* Kluwer Academic Publishing, Dordrecht, NL, 1998. Forthcoming.

[59] C. Meghini, F. Sebastiani, U. Straccia, and C. Thanos. A model of information retrieval based on a terminological logic. In *Proceedings of SIGIR-93, 16th ACM Conference on Research and Development in Information Retrieval*, pages 298–307, Pittsburgh, US, 1993.

[60] C. Meghini and U. Straccia. A relevance terminological logic for information retrieval. In *Proceedings of SIGIR-96, 19th ACM Conference on Research and Development in Information Retrieval*, pages 197–205, Zürich, CH, 1996.

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

[62] S. Mizzaro. Relevance: The whole history. *Journal of the American Society for Information Science*, 48:810–832, 1997.

[63] K. L. Myers. Hybrid reasoning using universal attachment. *Artificial Intelligence*, 67:329–375, 1994.

[64] G. Navarro and R. Baeza-Yates. A language for queries on structure and contents of textual databases. In *Proceedings of SIGIR-95, 18th ACM Conference on Research and Development in Information Retrieval*, pages 93–101, Seattle, US, 1995.

[65] G. Navarro and R. Baeza-Yates. Proximal nodes: A model to query document databases by content and structure. *ACM Transactions on Information Systems*, 15(4):400–435, 1997.

[66] S. Orphanoudakis, C. Chronaki, and S. Kostomanolakis. I^2C: A system for the indexing, storage, and retrieval of medical images by content. *Medical Informatics*, 19(2):109–122, 1994.

[67] I. Ounis and J.-P. Chevallet. Using conceptual graphs in a multifaceted logical model for information retrieval. In *Proceedings of DEXA'96, 7th Database and Expert system Applications Conference*, volume 1134 of *Lecture Notes in Computer Science*, pages 812–823, Zürich, CH, 1996. Springer Verlag.

[68] C. Peltason. The BACK system – an overview. *SIGART Bulletin*, 2(3):114–119, 1991.

[69] A. Pentland, R. Picard, and S. Sclaroff. Photobook: Content-based manipulation of image databases. In *Proceedings of SPIE-94, 2nd SPIE Conference on Storage and Retrieval for Still Images and Video Databases*, San Jose, US, 1994.

[70] E. G. Petrakis and C. Faloutsos. Similarity searching in medical image databases. *IEEE Transactions on Data and Knowledge Engineering*, 9(3):435–447, 1997.

[71] S. Ravela and R. Manmatha. Image retrieval by appearance. In *Proceedings of SIGIR-97, 20th ACM Conference on Research and Development in Information Retrieval*, pages 278–285, Philadelphia, US, 1997.

[72] A. Rosenfeld and A. C. Kak. *Digital picture processing.* Academic Press, New York, UK, 2nd edition, 1982.

[73] Y. Rui, T. Huang, M. Ortega, and S. Mehrotra. Relevance feedback: a power tool for interactive content-based image retrieval. *IEEE Trans. on Circuits and Systems for Video Technology*, 8(5):644–655, September 1998.

[74] G. Salton. *Automatic Text Processing: The Tranformation, Analysis, and Retrieval of Information by Computer.* McGraw-Hill, New York, 1983.

[75] G. Salton and C. Buckley. Term-weighting approaches in automatic text retrieval. *Information processing and management*, 24:513–523, 1988. Also reprinted in [88], pp. 323–328.

[76] S. Santini, A. Gupta, and R. Jain. User interfaces for emergent semantics in image databases. In *Proceedings of the 8th IFIP Working Conference on Database Semantics (DS-8)*, Rotorua, New Zealand, 1999.

[77] T. Saracevic. Relevance: a review of and a framework for the thinking on the notion in information science. *Journal of the American Society for Information Science*, 26:321–343, 1975. Also reprinted in [88], pp. 143–165.

[78] P. Schäuble and D. Knaus. The various roles of information structures. In O. Opitz, B. Lausen, and R. Klar, editors, *Proceedings of the 16th Annual Conference of the Gesellschaft für Klassifikation*, pages 282–290. Dortmund, DE, 1992. Published by Springer Verlag, Heidelberg, DE, 1993.

[79] M. Schmidt-Schauß and G. Smolka. Attributive concept descriptions with complements. *Artificial Intelligence*, 48:1–26, 1991.

[80] F. Sebastiani. A probabilistic terminological logic for modelling information retrieval. In W. B. Croft and C. J. van Rijsbergen, editors, *Proceedings of SIGIR-94, 17th ACM International Conference on Research and Development in Information Retrieval*, pages 122–130, Dublin, IE, 1994. Published by Springer Verlag, Heidelberg, DE.

[81] A. Singhal and M. M. Buckley. Learning routing queries in a query zone. In *Proceedings of SIGIR-97, 20th ACM Conference on Research and Development in Information Retrieval*, pages 25–32, Philadelphia,PA, July 1997.

[82] A. F. Smeaton. Using NLP or NLP resources for information retrieval tasks. In T. Strzalkowski, editor, *Natural language information retrieval*. Kluwer Academic Publishers, Dordrecht, NL, 1997.

[83] A. F. Smeaton and I. Quigley. Experiments on using semantic distances between words in image caption retrieval. In *Proceedings of SIGIR-96, 19th International Conference on Research and Development in Information Retrieval*, pages 174–180, Zürich, CH, 1996.

[84] J. R. Smith and S.-F. Chang. Transform features for texture classification and discrimination in large image databases. In *Proceedings of the 1st IEEE International Conference on Image Processing*, pages 407–411, Austin, US, 1994.

[85] J. R. Smith and S.-F. Chang. Visualseek: A fully automatic content-based image query system. In *Proceedings of the ACM International Conference on Multimedia*, pages 87–93, New York, US, 1996. ACM Press.

[86] J. R. Smith and S.-F. Chang. Visually searching the Web for content. *IEEE Multimedia*, pages 12–20, July-September 1997.

[87] J. F. Sowa. *Conceptual structures: information processing in mind and machine.* Addison Wesley, Reading, US, 1984.

[88] K. Sparck Jones and P. Willett, editors. *Readings in information retrieval.* Morgan Kaufmann, San Mateo, US, 1997.

[89] D. Sperber and D. Wilson. *Relevance. Communication and cognition.* Basil Blackwell, Oxford, UK, 1986.

[90] R. Srihari. Automatic indexing and content-based retrieval of captioned images. *IEEE Computer*, 28(9):49–56, 1995.

[91] U. Straccia. Document retrieval by relevance terminological logics. In I. Ruthven, editor, *Proceedings of MIRO-95, Workshop on Multimedia Information Retrieval*, Glasgow, UK, 1996. Springer Verlag, Heidelberg, DE.

[92] U. Straccia. A four-valued fuzzy propositional logic. In *Proceedings of IJCAI-97, 15th International Joint Conference on Artificial Intelligence*, pages 128–133, Nagoya, JP, 1997.

[93] U. Straccia. A sequent calculus for reasoning in four-valued description logics. In *Proceedings of TABLEAUX-97, International Conference on Analytic Tableaux and Related Methods*, number 1227 in Lecture Notes in Artificial Intelligence, pages 343–357, Pont-à-Mousson, FR, 1997.

[94] U. Straccia. A fuzzy description logic. In *Proceedings of AAAI-98, 15th Conference of the American Association for Artificial Intelligence*, pages 594–599, Madison, US, 1998.

[95] M. Stricker and M. Orengo. Similarity of color images. In *Proceedings of SPIE-95, 3rd SPIE Conference on Storage and Retrieval for Still Images and Video Databases*, pages 381–392, 1995.

[96] M. Swain and D. Ballard. Color indexing. *International Journal of Computer Vision*, 7(1):11–32, 1991.

[97] C. Thanos, editor. *Multimedia Office Filing. The MULTOS Approach.* North Holland, Amsterdam, NL, 1990.

[98] C. Tresp and R. Molitor. A description logic for vague knowledge. In *Proc. of the 13th European Conf. on Artificial Intelligence (ECAI-98)*, Brighton (England), August 1998.

[99] B. Vélez, R. Weiss, M. Sheldon, and D. K. Gifford. Fast and effective query refinement. In *Proceedings of SIGIR-97, 20th ACM Conference on Research and Development in Information Retrieval*, pages 6–15, Philadelphia,PA, July 1997.

[100] R. Weida and D. Litman. Terminological reasoning with constraint networks and an application to plan recognition. In *Proceedings of KR-92, 3rd International Conference on Principles of Knowledge Representation and Reasoning*, pages 282–293. Morgan Kaufmann, Los Altos, US, 1992.

[101] M. Wood, N. Campbell, and B. Thomas. Iterative refinement by relevance feedback in content-based digital image retrieval. In *Proceedings of Multimedia-98, the 6th ACM International Conference on Multimedia*, pages 13–20, N.Y., September 1998. ACM Press.

[102] J. Yen. Generalizing term subsumption languages to fuzzy logic. In *Proc. of the 12th Int. Joint Conf. on Artificial Intelligence (IJCAI-91)*, pages 472–477, Sydney, AU, 1991.

[103] L. A. Zadeh. Fuzzy sets. *Information and Control*, 8(3):338–353, 1965.

A Proof

Proposition *For all document bases:*

$$\overline{Maxdeg}(\Sigma_D \cup \bigcup_{1 \leq j \leq n} \delta_j, Q(\mathsf{d})) = Maxdeg(\Sigma_D \cup \Phi(C(\mathsf{d})) \cup \bigcup_{1 \leq j \leq n} \delta_j, Q(\mathsf{d})). \tag{40}$$

Proof: Let α be a fuzzy assertion. With $\Sigma \models_{\mathsf{di}} \alpha$ we denote the case where α is satisfied by all document interpretations $\mathcal{I}$ satisfying Σ. We will show that for all $n \in [0, 1]$,

$$\Sigma_D \cup \bigcup_{1 \leq j \leq n} \delta_j \models_{\mathsf{di}} \langle Q(\mathsf{d}), n \rangle \text{ iff } \Sigma_D \cup \Phi(C(\mathsf{d})) \cup \bigcup_{1 \leq j \leq n} \delta_j \models \langle Q(\mathsf{d}), n \rangle. \tag{41}$$

From (41), (40) quickly follows (just take n being the maximal degree). Hence, let us prove (41).

First of all, we show that for all $\alpha \in \Phi(C(\mathsf{d}))$,

$$\Sigma_D \cup \bigcup_{1 \leq j \leq n} \delta_j \models_{\mathsf{di}} \alpha \tag{42}$$

follows. The proof consists in a case analysis through the tables in Figure 9, Figure 10 and Figure 11. Due to space limitations, we will consider some of these cases only.

Consider Figure 10 and consider a document interpretation $\mathcal{I}$ satisfying $\Sigma_D \cup \bigcup_{1 \leq j \leq n} \delta_j$.

- Let x be $(\exists \mathsf{SI}.\{\mathsf{qi}\})(\mathsf{i})$ and let $\alpha \in \Phi(x)$ be $\langle \mathsf{SI}(\mathsf{i}, \mathsf{qi}), \sigma_i(\mathsf{i}^{\mathcal{I}}, \mathsf{qi}^{\mathcal{I}})\rangle$. By definition of document interpretations, $\mathsf{SI}^{\mathcal{I}}(\mathsf{i}^{\mathcal{I}}, \mathsf{qi}^{\mathcal{I}}) = \sigma_i(\mathsf{i}^{\mathcal{I}}, \mathsf{qi}^{\mathcal{I}})$. Therefore, $\mathcal{I}$ satisfies $\langle \mathsf{SI}(\mathsf{i}, \mathsf{qi}), \sigma_i(\mathsf{i}^{\mathcal{I}}, \mathsf{qi}^{\mathcal{I}})\rangle$.
- Let x be $(\exists \mathsf{HIR}.C)(\mathsf{i})$ and let $\alpha \in \Phi(x)$ be $\langle \mathsf{HIR}(\mathsf{i}, \mathsf{r}_j), 1\rangle$. By definition of $\Phi(x)$, $\mathcal{I}$ satisfies α.
- Let x be $(\exists \mathsf{HC}.\exists \mathsf{SC}.\{\mathsf{c}\})(\mathsf{r})$ and let $\alpha \in \Phi(x)$ be $\langle x, n\rangle$, where

$$n = max_{c \in \mathcal{C}}\{min\{f_e(\mathsf{r}^{\mathcal{I}})(c), \sigma_c(c, \mathsf{c}^{\mathcal{I}})\}\}$$

By definition of document interpretations, for all $c \in \mathcal{C}$,

$$\begin{aligned} &\mathsf{HC}^{\mathcal{I}}(\mathsf{r}^{\mathcal{I}}, c) = f_e(\mathsf{r}^{\mathcal{I}})(c), \text{ and} \\ &\mathsf{SC}^{\mathcal{I}}(c, \mathsf{c}^{\mathcal{I}}) = \sigma_c(c, \mathsf{c}^{\mathcal{I}}). \end{aligned}$$

Therefore,

$$\begin{array}{ll} (\exists \mathsf{HC}.\exists \mathsf{SC}.\{\mathsf{c}\})^{\mathcal{I}}(\mathsf{r}^{\mathcal{I}}) & = \\ max_{c \in \mathcal{C}}\{\min\{\mathsf{HC}^{\mathcal{I}}(\mathsf{r}^{\mathcal{I}}, c), (\exists \mathsf{SC}.\{\mathsf{c}\})^{\mathcal{I}}(c)\}\} & = \\ max_{c \in \mathcal{C}}\{\min\{f_e(\mathsf{r}^{\mathcal{I}})(c), max_{c' \in \mathcal{C}}\{\min\{\mathsf{SC}^{\mathcal{I}}(c, c'), \{\mathsf{c}\}^{\mathcal{I}}(c')\}\}\}\} & = \\ max_{c \in \mathcal{C}}\{\min\{f_e(\mathsf{r}^{\mathcal{I}})(c), \mathsf{SC}^{\mathcal{I}}(c, \mathsf{c}^{\mathcal{I}})\}\} & = \\ max_{c \in \mathcal{C}}\{\min\{f_e(\mathsf{r}^{\mathcal{I}})(c), \sigma_c(c, \mathsf{c}^{\mathcal{I}})\}\}. & \end{array}$$

As a consequence, $\mathcal{I}$ satisfies α.

All the other cases can be proved by similar arguments, thus obtaining the proof of (42).

An immediate consequence of (42) is that all document interpretations satisfying $\Sigma_D \cup \bigcup_{1 \leq j \leq n} \delta_j$ also satisfy $\Sigma_D \cup \Phi(C(\mathsf{d})) \cup \bigcup_{1 \leq j \leq n} \delta_j$, and vice-versa. Therefore, for all $n \in [0, 1]$

$$\Sigma_D \cup \bigcup_{1 \leq j \leq n} \delta_j \models_{\mathsf{di}} \langle Q(\mathsf{d}), n\rangle \text{ iff } \Sigma_D \cup \Phi(C(\mathsf{d})) \cup \bigcup_{1 \leq j \leq n} \delta_j \models_{\mathsf{di}} \langle Q(\mathsf{d}), n\rangle. \tag{43}$$

Finally, we have to show that for all $n \in [0, 1]$

$$\Sigma_D \cup \Phi(C(\mathsf{d})) \cup \bigcup_{1 \leq j \leq n} \delta_j \models_{\mathsf{di}} \langle Q(\mathsf{d}), n\rangle \text{ iff } \Sigma_D \cup \Phi(C(\mathsf{d})) \cup \bigcup_{1 \leq j \leq n} \delta_j \models \langle Q(\mathsf{d}), n\rangle, \tag{44}$$

which combined with (43) proves (41).

(only if) Consider $n \in [0, 1]$ and assume that $\Sigma_D \cup \Phi(C(\mathsf{d})) \cup \bigcup_{1 \leq j \leq n} \delta_j \models \langle Q(\mathsf{d}), n\rangle$. Let $\mathcal{I}$ be a document interpretations satisfying $\Sigma_D \cup \Phi(C(\mathsf{d})) \cup \bigcup_{1 \leq j \leq n} \delta_j$. Since $\mathcal{I}$ is an interpretation, by hypothesis it follows that $\mathcal{I}$ satisfies $\langle Q(\mathsf{d}), n\rangle$.

(if) Consider $n \in [0,1]$ and assume that $\Sigma_D \cup \Phi(C(\mathsf{d})) \cup \bigcup_{1 \leq j \leq n} \delta_j \models_{\mathsf{di}} \langle Q(\mathsf{d}), n\rangle$. Let $\mathcal{I}$ be an interpretation, not necessarily being a document interpretation, satisfying $\Sigma_D \cup \Phi(C(\mathsf{d})) \cup \bigcup_{1 \leq j \leq n} \delta_j$. We show that $\mathcal{I}$ satisfies $\langle Q(\mathsf{d}), n\rangle$. Consider the restriction of $\mathcal{I}$ to all the symbols appearing in Σ_D, $\Phi(C(\mathsf{d}))$, $\bigcup_{1 \leq j \leq n} \delta_j$ and $Q(\mathsf{d})$. Since (i) Q can not query any negative information about SPSs, *i.e.* the negation connective $\neg$ may involve *content concepts only* and there is no universal quantification on SPSs; and (ii) $\mathcal{I}$ satisfies $\Phi(Q(\mathsf{d}))$, it follows that the interpretation $\mathcal{I}$, w.r.t. the restricted symbols, is a document interpretation. From hypothesis, $\mathcal{I}$ satisfies $\langle Q(\mathsf{d}), n\rangle$ follows, completing the proof.

Q.E.D.