Representing and Reasoning on Conceptual Queries Over Image Databases

Mohand-Saïd Hacid[1]* and Christophe Rigotti[2]

[1] LuFg Theoretical Computer Science
RWTH Aachen, Ahornstraße 55, 52074 Aachen, Germany
hacid@@cantor.informatik.rwth-aachen.de
[2] LISI-INSA Lyon, Bâtiment 501,
F-69621 Villeurbanne Cedex
crig@@lisi.insa-lyon.fr

Abstract. The problem of content management of multimedia data types (e.g., image, video, graphics) is becoming increasingly important with the development of advanced multimedia applications. In this paper we develop a knowledge-based framework for modeling and retrieving image data. To represent the various aspects of an image object's characteristics, we propose a model which consists of three layers: **(1)** Feature and Content Layer, intended to contain image visual features such as contours, shapes, etc.; **(2)** Object Layer, which provides the (conceptual) content dimension of images; and **(3)** Schema Layer, which contains the structured abstractions of images. We propose two abstract languages on the basis of description logics: one for describing knowledge of the object and schema layers, and the other, more expressive, for making queries. Queries can refer to the form dimension (i.e., information of the Feature and Content Layer) or to the content dimension (i.e., information of the Object Layer). As the amount of information contained in the previous layers may be huge and operations performed at the Feature and Content Layer are time-consuming, resorting to the use of materialized views to process and optimize queries may be extremely useful. For that, we propose a formal framework for testing containment of a query in a view expressed in our query language.

1 Introduction

With recent progress in compression technology, it is possible for a computer to store huge amount of images, audio and even video. If such media are widely used in today's communication (e.g. in the form of home movies, education and training, scholarly research, and corporate enterprise solutions), efficient computer exploitation is still lacking. Many databases should be created to face the increasing development of advanced applications, such as digital libraries, archival and processing of images captured by remote-sensing satellites and air

* Work supported in part by the Deutscher Akademischer Austausch Dienst (DAAD) grant A/98/05793.

photos, training and education, entertainment, medical databases, virtual reality, Internet video, interactive TV, group-ware applications, etc. Though only a partial list, these advanced applications indicate that the problem of efficiently and user-friendly accessing to image/video data is widely encountered in real-life applications and solutions to it is significant. An important feature to be considered is content-based retrieval of the multimedia data types. For example, there are two essential questions associated with content-based query systems for imaging data [8]: (1) How to specify queries, and (2) How to access the intended data efficiently for given queries. These queries may be formulated in terms of a number of different image features, and can be grossly classified into three categories [14]: (1) *form queries*, addressing images on the basis of color, texture, sketch, or shape specifications; (2) *content queries*, focusing on domain concepts, spatial constraints, or various types of attributes; (3) *mixed queries*, which combine the two previous categories.

Despite some proposals on finding appropriate representations of image data and systems architectures able to support such representations, there is little research work on finding semantic foundations for query optimization in image databases. This paper is a contribution in this direction. We take a new look at the problem of modeling and querying image data and find that knowledge representation and reasoning techniques for concept languages developed in Artificial Intelligence provide an interesting angle to attack such problems. We exploit the possibility of using two languages: one for defining the schema (i.e. the structure) of an image database and populating it, and the other, more expressive, for querying the database through the schema.

We build on work by Chu et al. [7] to propose three layers for representing image content:

(1) *Feature and Content Layer:* It contains image features such as contours, spatial relationships, etc. This layer is characterized by a set of techniques allowing to retrieve images based on the similarity of physical features such as region, color, and shape.

(2) *Object Layer:* This layer contains objects of interest, their descriptions, and relationships among objects based on the extracted features[1]. This layer constitutes what we call the extensional part of an image database. Objects in an image are represented in the object layer as visual entities. Instances of visual objects consist of conventional attributes (e.g. name, patientID, date, etc.) as well as visual attributes (e.g. shape, size, etc.) of objects contained in the feature and content layer.
The interface between the *Feature and Content Layer* and the *Object Layer* is determined by the *Feature and Content Layer* itself and a set of predicates (e.g., similar-to predicates) over this Feature and Content Layer. Objects of the *Object Layer* are related to objects of the Feature and Content Layer via attributes.

(3) *Schema Layer:* This layer is intended to capture the structured abstractions and knowledge that are needed for image retrieval. It contains a general

[1] Features can be extracted *manually*, *semi-automatically* or *automatically.*

schema about the classes of objects stored in the object layer, their general properties and mutual relationships. In this layer, visual entities can be classified into a hierarchical structure known as a concept hierarchy on the basis of both conventional and visual attributes. This layer is well suited for integrating domain knowledge.

The part of a query that pertains to the Feature and Content Layer is processed by specialized signal processing procedures, and hence are time consuming. In addition, the amount of information contained in the object layer is huge. To enable quick response to the queries, the strategy based on the use of materialized[2] views to compute answers to queries can turn out to be useful. Supporting materialized views to process and optimize queries is the topic of much recent work on data-intensive applications (e.g., see [13]). Reasoning on queries in the case of image databases is not only relevant to determine views that can be used for answering queries, but it can be applied to organize large sets of queries into taxonomies which can be important to support navigation. For that, we develop an algorithm for checking the *containment*[3] between a query and a view (which is seen as a query as well) expressed in our query language.

Although, in the basic form that we give here, the languages do not account for all aspects of image data, they constitute kernels to be extended. Showing how we can model and reason about the structure of image databases and queries is useful and significant.

Paper outline: In Section 2, we develop our languages, give their Tarski-style extensional semantics, and a calculus for query containment. Section 3 discusses related work. We conclude in Section 4 by anticipating on the necessary extensions.

2 The Languages

Before we give the syntax and semantics of our *abstract languages*, we define *concrete domains*, which are used to incorporate application-specific domains (i.e., strings, reals, integers, etc.) into the abstract domain of individuals.

Definition 1. (Concrete Domains) A concrete domain $\mathcal{D} = (\mathsf{dom}(\mathcal{D}), \mathsf{pred}(\mathcal{D}))$ consists of:

- the domain $\mathsf{dom}(\mathcal{D})$,
- a set of predicate symbols $\mathsf{pred}(\mathcal{D})$, where each predicate symbol $P \in \mathsf{pred}(\mathcal{D})$ is associated with an arity n and an n-ary relation $P^{\mathcal{D}} \subseteq \mathsf{dom}(\mathcal{D})^n$,

In many applications (in particular when querying databases), one would like to be able to refer to concrete domains and predicates on these domains when defining queries. An example of a concrete domain could be the set of (nonnegative) integers with comparisons $(=, <, \leq, \geq, >)$.

[2] A materialized view is a query whose a physical copy of each instance, answer to the query, is stored and maintained.

[3] Containment of queries is the problem of checking whether the result of one query is contained in what another query produces [1].

2.1 Schema Language ($\mathcal{SL}$)

We now introduce a simple description logic that will be used for describing the structure of an image data. Starting from atomic concepts and roles, complex concepts are built by using the universal quantification ($\forall$) and predicate restrictions. The syntax and the semantics of this description logic are given below.

Definition 2. (Syntax) Let N_C, N_R, N_f be three pairwise disjoint sets of concept names, role names, and feature (i.e., functional role) names respectively, $\mathcal{D}_1, \ldots, \mathcal{D}_k$ be concrete domains. Let P be a role name, $f, f_1, \ldots, f_n$ be feature names, A be a concept name, A' be a concept name or a concrete domain name, and P_r be an n-ary predicate name. Concept terms C, D are defined by the following rules:

$$\begin{array}{lll} C, D \longrightarrow & A \mid & \text{(primitive concept)} \\ & \forall P.A \mid & \text{(typing of role)} \\ & \forall f.A' \mid & \text{(typing of feature)} \\ & P_r(f_1, \ldots, f_n) & \text{(predicate restriction)} \end{array}$$

Let A, A_1, A_2 be concept names, A_3 be a concept name or a concrete domain name, D be a concept term, P be a role name, and f be a feature name. Then $A \dot{\preceq} D$ (we say A is a subconcept of D), $P \dot{\preceq} A_1 \times A_2$, $f \dot{\preceq} A_1 \times A_3$ are called *axioms*. A $\mathcal{SL}$ schema $\mathcal{S}$ consists of a finite set of axioms. In the following, we consider only *acyclic* schemas. A schema $\mathcal{S}$ is acyclic if no concept name occurs–neither directly nor indirectly–within its own specification.

Definition 3. (Semantics) The semantics is given by an interpretation $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$, which consists of an (abstract) interpretation domain $\Delta^{\mathcal{I}}$, and an interpretation function $\cdot^{\mathcal{I}}$. The abstract domain has to be disjoint from any given concrete domain, i.e., $\Delta^{\mathcal{I}} \cap \mathsf{dom}(\mathcal{D}_i) = \emptyset$ for all concrete domain $\mathcal{D}_i$ ($i \in [1, k]$), the concrete domains are pairwise disjoint, and $\mathsf{pred}(\mathcal{D}_i) \cap \mathsf{pred}(\mathcal{D}_j) = \emptyset$ for $i \neq j$. The interpretation function $\cdot^{\mathcal{I}}$ associates each concept C with a subset $C^{\mathcal{I}}$ of $\Delta^{\mathcal{I}}$, each role P with a binary relation $P^{\mathcal{I}}$ on $\Delta^{\mathcal{I}}$, and each feature name f with a partial function $f^{\mathcal{I}} : \Delta^{\mathcal{I}} \to (\Delta^{\mathcal{I}} \cup (\bigcup_{i=1}^{k} \mathsf{dom}(\mathcal{D}_i)))$. Additionally, $\mathcal{I}$ has to satisfy the following equations:

$$\begin{aligned} (\forall P.A)^{\mathcal{I}} &= \{d \in \Delta^{\mathcal{I}} \mid \forall d' \in \Delta^{\mathcal{I}} : \\ &\qquad (d^{\mathcal{I}}, d'^{\mathcal{I}}) \in P^{\mathcal{I}} \to d'^{\mathcal{I}} \in A^{\mathcal{I}}\} \\ (\forall f.A')^{\mathcal{I}} &= \{d \in \Delta^{\mathcal{I}} \mid \text{ if } f^{\mathcal{I}}(d^{\mathcal{I}}) \text{ is defined then} \\ &\qquad f^{\mathcal{I}}(d^{\mathcal{I}}) \in A'^{\mathcal{I}}\} \\ (P_r(f_1, \ldots, f_n))^{\mathcal{I}} &= \{d \in \Delta^{\mathcal{I}} \mid (f_1^{\mathcal{I}}(d^{\mathcal{I}}), \ldots, f_n^{\mathcal{I}}(d^{\mathcal{I}})) \in P_r^{\mathcal{D}}\} \end{aligned}$$

An interpretation $\mathcal{I}$ satisfies the axiom $A \dot{\preceq} D$ iff $A^{\mathcal{I}} \subseteq D^{\mathcal{I}}$, the axiom $P \dot{\preceq} A_1 \times A_2$ iff $P^{\mathcal{I}} \subseteq A_1^{\mathcal{I}} \times A_2^{\mathcal{I}}$, and the axiom $f \dot{\preceq} A_1 \times A_3$ iff $f^{\mathcal{I}} \subseteq A_1^{\mathcal{I}} \times A_3^{\mathcal{I}}$. If A_3 is a concrete domain name then $A_3^{\mathcal{I}}$ stands for the domain of A_3 (i.e., $\mathsf{dom}(A_3)$) for all $\mathcal{I}$. An interpretation $\mathcal{I} = (\Delta^{\mathcal{I}}, .^{\mathcal{I}})$ is a model, also called a valid interpretation, of a schema $\mathcal{S}$ iff it satisfies every axiom in $\mathcal{S}$.

An interpretation $\mathcal{I}$ that satisfies all axioms in $\mathcal{S}$ is called an $\mathcal{S}$-interpretation.

The language introduced previously allows to describe knowledge about classes of individuals and relationships between these classes. We can now turn our attention to the extensional level, which we call the *ABox*. The *ABox* essentially allows one to specify instance-of relations between individuals and classes (concepts), and between pairs of individuals and roles or features.

Definition 4. Let N_I and N_D be two disjoint alphabets of symbols, called abstract individual names and concrete individual names respectively. Instance-of relationships are expressed in terms of *membership assertions* of the form: $a : C$, $(a, b) : P$, $(a, b) : f$, $(a, z) : f$, and $(z_1, \ldots, z_n) : P_r$, where a and b are abstract individual names, $z, z_1, \ldots, z_n$ are concrete individual names, C is a concept name or an arbitrary concept, P is a role name, and P_r is an n-ary predicate name of a concrete domain. Intuitively, the first form states that a is an instance of C, and the second form states that a is related to b by means of the role P (we also say b is a P-*successor* of a).

An interpretation $\mathcal{I}$ is easily extended to individuals and membership assertions.

An *ABox* $\mathcal{A}$ is a finite set of membership assertions. An interpretation $\mathcal{I}$ is a model for an *ABox* $\mathcal{A}$ iff $\mathcal{I}$ satisfies all the assertions in $\mathcal{A}$.

2.2 Query Language ($\mathcal{QL}$)

Querying a database means retrieving stored objects that satisfy certain *conditions* or *qualifications* and hence are interesting for a user. In the case of relational databases, queries are constructed by means of algebra expressions defined on relations from the database. As a property, answers are also relations (i.e., sets of tuples). This correspondence between database entities and answer formats presents advantages that lead to the design and development of query optimization techniques. In object-oriented databases, classes are used to represent sets of objects. By analogy with the relational approach, classes can be used for *describing query results*. If such a possibility exists, then we can consider some kind of *reasoning* on the structure[4] of classes that will lead to reveal, for example, *subsumption* relationships between queries.

In this paper, we follow this approach. Queries are represented as concepts in our abstract language.

Definition 5. (Syntax) Let A be a concept name, P be an atomic role, d be an abstract individual name, $f_1, f_2, \ldots, g_1, g_2, \ldots$ be feature names, $P_r \in \mathsf{pred}(\mathcal{D}_i)$ for some $i \in [1, k]$ be an n-ary predicate name, and $P_{r_j} \in \mathsf{pred}(\mathcal{D}_i)$ for some $i \in [1, k]$ ($j \in [1, m]$) be a binary predicate name. Concepts C, D and roles R, R' can be formed by means of the following syntax:

$$\begin{aligned} C, D &\longrightarrow \top \mid A \mid C \sqcap D \mid \{d\} \mid \exists R.C \mid P_r(f_1, \ldots, f_n) \mid \\ &\qquad \Theta(C, D, \{\langle f_1, P_{r_1}, g_1\rangle, \ldots, \langle f_m, P_{r_m}, g_m\rangle\}) \\ R, R' &\longrightarrow P \mid P^- \mid R \circ R' \end{aligned}$$

[4] And hence the semantics of class hierarchies.

Definition 6. (Semantics) $\mathcal{I}$ is defined as in definition 3. Additionally, $\mathcal{I}$ has to satisfy the following equations:

$$\begin{aligned}
\top^{\mathcal{I}} &= \Delta^{\mathcal{I}} \\
(\exists R.C)^{\mathcal{I}} &= \{d \in \Delta^{\mathcal{I}} \mid \exists d' : (d^{\mathcal{I}}, d'^{\mathcal{I}}) \in R^{\mathcal{I}} \wedge d'^{\mathcal{I}} \in C^{\mathcal{I}}\} \\
P_r(f_1, \ldots, f_n)^{\mathcal{I}} &= \{d \in \Delta^{\mathcal{I}} \mid (f_1^{\mathcal{I}}(d^{\mathcal{I}}), \ldots, f_n^{\mathcal{I}}(d^{\mathcal{I}})) \in P^{\mathcal{D}}\} \\
\{d\}^{\mathcal{I}} &= \{d^{\mathcal{I}}\} \\
(R \circ R')^{\mathcal{I}} &= \{(d, d') \in \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}} \mid \exists c \in \Delta^{\mathcal{I}} \text{ such that} \\
&\qquad (d^{\mathcal{I}}, c^{\mathcal{I}}) \in R^{\mathcal{I}} \wedge (c^{\mathcal{I}}, d'^{\mathcal{I}}) \in R'^{\mathcal{I}}\}
\end{aligned}$$

$$\begin{aligned}
&(\Theta(C, D, \{\langle f_1, P_{r_1}, g_1\rangle, \ldots, \langle f_m, P_{r_m}, g_m\rangle\}))^{\mathcal{I}} = \\
&\quad \{d \in \Delta^{\mathcal{I}} \mid d \in C^{\mathcal{I}} \text{ and } \exists d'\ d' \in D^{\mathcal{I}} \text{ such that} \\
&\quad (f_1^{\mathcal{I}}(d^{\mathcal{I}}), g_1^{\mathcal{I}}(d'^{\mathcal{I}})) \in P_{r_1}^{\mathcal{D}_{P_{r_1}}} \wedge \ldots \wedge (f_m^{\mathcal{I}}(d^{\mathcal{I}}), g_m^{\mathcal{I}}(d'^{\mathcal{I}})) \in P_{r_m}^{\mathcal{D}_{P_{r_m}}}\}
\end{aligned}$$

2.3 Query Containment

Definition 7. (Containment) Given a $\mathcal{SL}$ schema $\mathcal{S}$, a query Q and a view V in $\mathcal{QL}$ language, are the answers to Q also answers to V for any database state obeying the schema $\mathcal{S}$.

A query Q is $\mathcal{S}$-satisfiable if there is an $\mathcal{S}$-interpretation $\mathcal{I}$ such that $Q^{\mathcal{I}} \neq \emptyset$. We say that Q is $\mathcal{S}$-*contained* in V (written $Q \dot{\preceq}_{\mathcal{S}} V$) if $Q^{\mathcal{I}} \subseteq V^{\mathcal{I}}$ for every $\mathcal{S}$-interpretation $\mathcal{I}$.

We have devised an algorithm (see [10]) for deciding containment of a query in a view. The basic idea for deciding the *containment* of a query Q in a view V is drawn from [4]. We take an object o and transform Q into a prototypical database state where o is an answer to Q. We do so by generating individuals, entering them into concepts in the schema, and relating them through roles and features. If o belongs to the answer of V, then Q is contained in V. If not, we have a state where an individual is in the answer to Q but not in the answer to V and therefore V does not contain Q. The details of the algorithm and the proof of the following theorem can be found in [10].

Theorem 8. *Containment of a query in a view in our query language can be decided in time polynomial to the size of Q, V and $\mathcal{S}$.*

3 Related Work

Our work relates to several fields of research in databases and Artificial Intelligence. We shortly discuss the relationship to modeling and retrieving image data by content, multimedia databases and query optimization.

Modeling and Retrieving Image Data by Content. Modeling and retrieving image data by content has been considered from both database and artificial intelligence points of views. Meghini *et al.* [15] have investigated the use of a description logic as a conceptual tool for modeling and querying image data. The problem with this language is that subsumption between concepts is $PSPACE$-complete. In addition, They do not consider predicate restrictions over concrete domains. Hsu *et al.* [11] proposed a knowledge-based approach for retrieving images by content. The knowledge-based query processing is based on a query relaxation technique which exploits a Type Abstraction Hierarchy of image features. The query language is an extension of OQL[5] to include specific predicates (e.g., *similar-to* predicates).

Multimedia. Goble *et al.* [9] proposed a description logic, called, GRAIL, for describing the image and video semantic content. A set of dedicated constructors are used to capture the structural part of these media objects. The aim is to support the coherent and incremental development of a coarse index on the semantic annotations of media documents. Lambrix and Padgham [12] described an extended description logic for representing and retrieving documents. The description logic includes part-of relations and allows for ordering information between the parts.

In these two proposals, the underlying query languages support only queries based on the structure of the documents (i.e., conceptual queries). None of them supports visual queries. Together with [15] they do not take into account predicate restrictions over concrete domains, which are extremely useful when querying multimedia repositories. In addition, they did not address the questions of decidability and complexity of reasoning services in their languages.

Query Optimization in Multimedia databases. The problem of optimizing queries over multimedia repositories has been addressed in recent works (see, among others, [6]). In summary, these works consider the indexes used to search the repository and user-defined filter conditions to define an execution space that is search-minimal. Semantic query optimization considers semantic knowledge for constructing query evaluation plans, and the framework for testing query containment presented in this paper is relevant due to the incorporation of schema knowledge in our algorithm. In multimedia applications where metadata play an important role [17], these two kinds of query optimization have to cohabit.

4 Conclusion

There is now intense interest in multimedia systems. These interests span across vast areas in computer science, such as computer networks, databases, distributed computing, data compression, document processing, user interfaces, artificial intelligence, etc. In the long run, we expect that intelligent-solving systems will access information stored in a variety of formats, on a wide variety of media.

Multimedia information is inherently complex. Traditional database techniques do not apply since, for example, they do not deal with content-based retrieval. We believe that the combination of database techniques and intelligent information retrieval will contribute to the realization of intelligent multimedia systems. Artificial intelligence, and more specifically knowledge representation, will play an important role in this task.

Our work focuses on a fundamental problem, namely, a content-based retrieval of image data. We have merely laid a formal and flexible framework which is appropriate for modeling and reasoning about meta-data and queries in image databases. Expressiveness and services of the meta data schema are crucial for image database quality[5]. The framework is general in that little needs to be changed when making extensions or taking other constructors for the abstract languages. In addition, this framework is appropriate for supporting semantic indexing [16], conceptual queries [18] and intensional queries [3]. Indeed, as the information structure that is contained in image databases is usually complicated and amount of information is huge, users may prefer to express queries with more general and abstract information instead of primitive terms directly based on the data stored in a database.

There are many interesting directions to pursue. **(1)** An important direction of active research is to significantly extend this framework to support part-whole relations. The result reported in [2] constitutes a nice basis; **(2)** Due to the visual nature of the data, a user may be interested in results that are similar to the query, thus, the query system should be able to perform exact as well as partial or fuzzy matching.

We are investigating these important research directions.

References

1. Serge Abiteboul, Richard Hull, and Victor Vianu. *Foundations of Databases.* Addison-Wesley, 1995.
2. Alessandro Artale, Enrico Franconi, Nicola Guarino, and Luca Pazzi. Part-Whole Relations in Object-Centered Systems: An Overview. *Data & Knowledge Engineering*, 20(3):347–383, December 1996.
3. Sonia Bergamaschi, Claudio Sartori, and Maurizio Vincini. Description Logic Techniques for Intensional Query Answering in OODBs. In Franz Baader, Martin Buchheit, Manfred Jeusfeld, and Werner Nutt, editors, *Proceedings of the 2nd Workshop on Knowledge Representation meets DataBases (KRDB'95), Bielefeld, Germany*, September 1995.
4. Martin Buchheit, Manfred A. Jeusfeld, Werner Nutt, and Martin Staudt. Subsumption Between Queries to Object-Oriented Databases. In *Proceedings of the 4th International Conference on Extending Database Technology (EDBT'94), Cambridge, UK*, March 1994. (Also in Information Systems 19(1), pp. 33-54, 1994).
5. Rick Cattel. *The Object Database Standard: ODMG-93.* Morgan Kaufmann, San Mateo, CA, 1994.
6. Surajit Chaudhuri and Luis Gravano. Optimizing Queries over Multimedia Repositories. In H.V. Jagadish and Inderpal Singh Mumick, editors, *Proceedings of the 1996 ACM SIGMOD International Conference on Management of Data (SIGMOD'96), Montréal, Québec, Canada,*, pages 91–102, June 1996.

[5] Quality implies accessibility, optimization, validation, etc.

7. Wesley W. Chu, Alfonso F. Cárdinas, and Ricky K. Taira. Knowledge-Based Image Retrieval with Spatial and Temporal Constructs. In Zbigniew W. Raś and Andrzej Skowron, editors, *Proceedings of the 10th International Symposium on Methodologies for Intelligent Systems (ISMIS'97), Charlotte, North Carolina, USA*, LNAI - 1325, pages 17–34. Springer, October 1997.
8. Tzi cker Chiueh. Content-Based Image Indexing. In Jorge Bocca, Matthias Jarke, and Carlo Zaniolo, editors, *Proceedings of the 20th International Conference on Very Large Data Bases (VLDB'94), Santiago, Chile*, pages 582–593, September 1994.
9. C. A. Goble, C. Haul, and S. Bechhofer. Describing and Classifying Multimedia Using the Description Logic GRAIL. In Ishwar K. Sethi and Ramesh C. Jain, editors, *Storage and retrieval for image and video database IV (SPIE'96), San Jose, California*, pages 132–143, February 1996.
10. Mohand-Said Hacid and Christophe Rigotti. Representing and Reasoning on Conceptual Queries Over Image Databases. Research Report TCS-99-02, LuFG Theoretical Computer Science, RWTH-Aachen, Germany, 1999.
11. Chih-Cheng Hsu, Wesley W. Chu, and Ricky K. Taira. A Knowledge-Based Approach for Retrieving Images by Content. *IEEE Transactions on Knowledge and Data Engineering*, August 1996.
12. Patrick Lambrix and Lin Padgham. A Description Logic Model for Querying Knowledge Bases for Structured Documents. In Zbigniew W. Raś and Andrzej Skowron, editors, *Proceedings of the 10th International Symposium on methodologies for Intelligent Systems (ISMIS'97), Charlotte, North Carolina, USA*, LNAI 1325, pages 72–83. Springer, October 1997.
13. Alon Y. Levy, Alberto O. Mendelzon, Yehoshua Sagiv, and Divesh Srivastava. Answering Queries Using Views. In *Proceedings of the 1995 Symposium on Principles of Database Systems (PODS'95), San Jose, CA, USA*, pages 95–104, May 1995.
14. Carlo Meghini. Towards a Logical Reconstruction of Image Retrieval. In Ishwar K. Sethi and Ramesh C. Jain, editors, *Storage and retrieval for image and video database IV (SPIE'96), San Jose, California*, pages 108–119, February 1996.
15. Carlo Meghini, Fabrizio Sebastiani, and Umberto Straccia. The Terminological Image Retrieval Model. In Alberto Del Bimbo, editor, *Proceedings of ICIAP'97, 9th International Conference On Image Analysis And Processing*, volume II, pages 156–163, Florence, I, September 1997.
16. Albrecht Schmiedel. Semantic Indexing Based on Description Logics. In Franz Baader, Martin Buchheit, Manfred Jeusfeld, and Werner Nutt, editors, *Proceedings of the 1st Worshop on Knowledge Representation meets DataBases (KRDB'94), Saarbrücken, Germany*, September 1994.
17. Amit Sheth and Wolfgang Klas. *Multimedia Data Management: Using Metadata to Integrate and Apply Digital Media.* Mc Graw Hill, 1998.
18. S.C. Yoon. Towards Conceptual Query Answering. In Zbigniew W. Raś and Andrzej Skowron, editors, *Proceedings of the 10th International Symposium on methodologies for Intelligent Systems (ISMIS'97), Charlotte, North Carolina, USA*, LNAI 1325, pages 187–196. Springer, October 1997.