PHOTOGRAPH INDEXING AND RETRIEVAL USING STAR-GRAPHS

Jean Martinet, Yves Chiaramella, Philippe Mulhem

MRIM - CLIPS-IMAG - UJF
Grenoble, FRANCE
First.Last@imag.fr

Iadh Ounis

IR Group - University of Glasgow
United Kingdom
ounis@dcs.gla.ac.uk

ABSTRACT

We present in this paper a relational approach for indexing and retrieving photographs from a collection. Instead of using simple keywords as an indexing language, we propose to use *star-graphs* as document descriptors. A *star-graph* is a conceptual graph that contains a single relation, with some concepts linked to it. They are elementary pieces of information describing combinations of concepts. We use star-graphs as descriptors - or index terms - for image content representation. This allows for relational indexing and expression of complex user needs, in comparison to classical text retrieval, where simple keywords are generally used as document descriptors. We present a document representation model, a weighting scheme for star-graphs inspired by the *tf.idf* used in text retrieval. We have applied our model to image retrieval, and show the system evaluation results.

1. INTRODUCTION

The classical models of Information Retrieval (IR) consider that a document is described by a set of representative index terms. In text retrieval for instance, index terms are keywords extracted from the collection. Because all index terms in a document do not describe it equally, they are assigned numerical weights. The purpose of a weighting scheme is to give emphasis to important terms, quantifying how well they semantically describe and discriminate documents.

When dealing with image documents, we can use keywords to describe main elements appearing on an image, e.g. "MAN" or "SKY". However some information contained in the image cannot be expressed or modeled by keywords themselves [10], such as spacial relations between objects, or object attributes. Expression of the spacial relations between objects have been considered in research by Bertino and Catania [2] and Di Sciascio and colleagues [15], where the authors define languages to represent shapes and their position on the image. This allows for relational image indexing and querying. In order to encapsulate the complex knowledge related to either an image description or a user need, a formalism that supports relations is required. In particular, the knowledge representation formalism of *conceptual graphs*, introduced by Sowa [16], has been used for image representation by Mechkour [7] and image retrieval by Ounis and Pasca [10]. We use conceptual graphs to index the images in the collection, assigning some numerical values to elementary sub-graphs in order to represent the importance of a sub-graph in the index.

The conceptual graphs formalism, as many knowledge representation formalisms in artificial intelligence, allows only boolean truth values. Numerous research has been have been carried out to extend the conceptual graphs formalism. Morton [8] extended Sowa's conceptual graph theory to fuzzy concepts, fuzzy referents, and fuzzy operations. Wuwongse and colleagues [18] have extended Morton's fuzzy conceptual graphs to take into account fuzzy conceptual relations. Maher [5] has proposed a similarity measurement for matching simple fuzzy conceptual graphs, based on fuzzy similarity measurement for concepts. Those works aim at assigning a non-boolean truth value to a conceptual graph, in order to take uncertainty into account when facing imprecise knowledge. The purpose of our work is to define and weight simple pieces of conceptual graphs and to match them for IR purpose. The weighting scheme for our index terms is inspired by the *term frequency* and *inverse document frequency* (*tf.idf*) weighting scheme used in text retrieval[14, 13]. Among the extensions to *tf.idf* in image retrieval, Wang and Du [17] introduced the *region frequency* and *inverse picture frequency* (*rf.ipf*), a region-based measure for image retrieval purposes. The images in the collection are segmented into regions, employing several features (e.g. colour components). The *rf* measures how frequently a region feature occurs in a picture, and the *ipf* attaches a higher value to region features that occur in few images, therefore considered good discriminators. All regions are assigned an *rf.ipf* weight, that is stored for the image matching process. Research by Wang is based on signal analysis; it does not imply symbolic meaning of the segmented regions, and no formalism is applied. In our work, we combine signal and symbolic approaches, using a measure of image region importance which is based on visual parame-

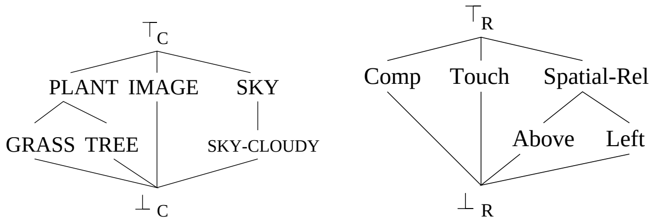

Fig. 1. Concept type and relation type lattices.

Fig. 2. A conceptual graph indexing an image.

ters determined by the user's perception [9], and an extension of the *idf* using a new definition of a *term occurrence* in a document. The classical *idf* includes all documents indexed by a given term. The basic idea behind our extension is that a document containing a term t', where t' is a specific of t, is also *about* t, and thereby the document should be taken into account in the *idf* of t. The generic/specific relations of the concepts are defined by a lattice. Plachouras and Ounis [11] have defined a probability measure for a concept in a collection of documents based on the same idea, using a hierarchical structure of concepts. They adopted this probability measure to estimate a *query scope* in the context of the Web, which indicates how specific or generic a query is, given the hierarchy of concepts.

This paper describes an image representation model based on star-graphs. The remaining sections are organized as follows: in Section 2 we briefly describe how documents are represented, in Section 3 we examine the elements involved for weighting a star-graph. Current experiments are presented and discussed in Section 4.

2. IMAGE REPRESENTATION

Our image representation model shows the images that were originally represented by conceptual graphs. The conceptual graphs are split into elementary pieces of information, that are considered to be the document descriptors.

2.1. Conceptual Graphs

The formalism of conceptual graph provides an expressive framework that supports relational image content indexing. A conceptual graph (CG) is a bipartite oriented graph, composed with concepts and relations [16]. Concept and relation are organized in lattices, which indicate generic/specific relationships. Figure 1 shows an example of lattices. We use conceptual graphs to represent images content. Each segmented region of the image is associated with a concept, where the conceptual relations describe relationships between these regions. The conceptual graph, Figure 2, is the index of the picture on the right. This corresponds to the symbolic, structural, and spatial facets of $EMIR^2$, Mechkour's image representation model [7].

The graphs are built from a set of elementary graphs: the canonical basis. To create a CG, a restricted graph from the canonical basis is joined together with another, using the *join* operation. The *join* operation is defined as follows [16]: if two concept nodes have the same label with the same marker, we can join them and obtain a single concept node. We break-up the conceptual graphs into elementary pieces of information. These pieces of information will constitute our indexing terms. For that, we use the *split* operation.

The *split* operation on CGs is defined as follow [1]: let $[c]$ be a concept node of a conceptual graph.

1. Introduce a new node $[x]$.
2. Replace $[c] \rightarrow (r) \rightarrow [c_1]$ (or $[c_1] \rightarrow (r) \rightarrow [c]$) with $[x] \rightarrow (r) \rightarrow [c_1]$ (or $[c_1] \rightarrow (r) \rightarrow [x]$), where $[c_1]$ is an arbitrary concept in the graph.
3. Replace $[x]$ with $[c]$.

An example of the application of *split* on a graph is given, Figure 3. This operation introduces a node in the graph, while the number of edges remains constant. It results in two connected, well-formed subgraphs. Let's now consider a graph on which *split* is iterated until each node has exactly one adjacent edge. At each step, the total number of edges remains constant and the number of adjacent edges for one of the concepts decreases. Hence this procedure terminates. A set of well-formed sub-graphs containing only one relation is obtained at the end, which is called a *table of graphs* [1]. This set contains restricted elements from the canonical basis. A table of graphs can be considered a set of star-graphs. A *star-graph* is a conceptual graph that contains a single relation. It consists of a single conceptual relation r, every arc that belongs to r, and every concept c that is linked by some arc (r, c) that belongs to r. We call them star-graph because they are star-shaped conceptual graphs: the relation is the center of the star, and the arcs linking the concepts to the relation form the branches of the star.

When possible existential quantifiers are eliminated from the graphs by the mean of introducing new unique constants (or *witnesses*) [10], *split* and *join* are reverse operations. Hence, it is possible to build the original graph back by iterating the *join* operation on the split concepts of the resulting star-graphs. A conceptual graph with k relations can

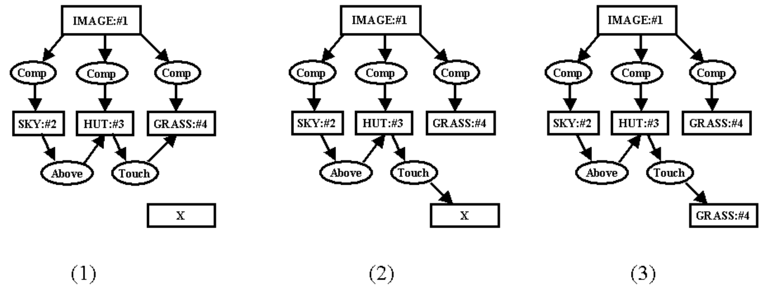

Fig. 3. Application of the Split operation on a conceptual graph.

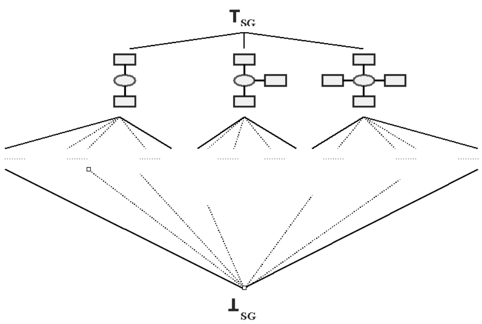

Fig. 4. General outline of the star-graph lattice.

be constructed from k star-graphs (building blocks), one for each conceptual relation.

2.2. Star-graphs as index terms

The set of star-graphs (SGs) extracted from the collection of conceptual graphs has a lattice structure. Its partial order relation $\leq$, is the graph projection [16]. This lattice is limited at the top by $\top_{SG}$ and at the bottom by $\perp_{SG}$. It has the general outline shown in Figure 4. In this figure, there are three sub-lattices, containing star-graphs respectively with two, three, and four concepts.

The star-graphs obtained at the end of the *split* are considered document descriptors. From now on, star-graphs will be referred to as *index terms*. As for text documents, the weight of a descriptor should represent how important this descriptor is as a part of the document's index and how rare it is within the whole collection. For instance, we want to know how well the term $[SKY] \rightarrow (Above) \rightarrow [HUT]$ describes the document shown in Figure 1 and to what extent this term allows for the discrimination of the documents across the collection. The following section describes a weighting scheme for an SG, based on a combination of a local analysis of elements in a document (a *document-related value*) and a global analysis of the indexed collection (a *collection-related value*).

3. TERM WEIGHTING

In this section, we address the extension of the classical *tf.idf* for star-graphs, in the context of image retrieval. We define:

- a local analysis for image index terms, corresponding to the *tf* for text. It consists of representing the importance of a term in a document,
- a global analysis corresponding to the *idf*. Terms are assigned a value according to their distribution in the collection, using some information from the lattice. According to the definition of a lattice, the more specific a term is, the more information it contains, hence the greater it's important to retrieve relevant documents. This global analysis is aimed at emphasizing the impact of more specific terms of the lattice, while moderating the effect of more generic terms.

3.1. Local analysis

We define a *Local* value for the index terms. A star-graph consists of a set of concepts with a relation. In order to assign a local value to a star-graph, we need its concepts to be assigned local values. These values are related to a user's perception that depends on parameters such as the relative surface of the segmented region, its contrast and position on the image [9]. We assume that every concept c of a document d is assigned a value $Local_{c,d}$. The *Local* value of an index term t is related by a function f to the importance of its composing concepts:

$$Local_{t,d} = f(Local_{c_1,d}, Local_{c_2,d}, \cdots, Local_{c_n,d})$$

where n is the number of concepts in the star-graph t (i.e the relation of the star-graph is n-adic). For instance, if we assume that the importance of a term is the one of the most important concepts composing it, we can set f to be max:

$$Local_{t,d} = \max_{c_j \in t} Local_{c_j,d} \quad (1)$$

The *Local* value for the term $[SKY] \rightarrow (Above) \rightarrow [HUT]$ would be the maximum of the *Local* values of $[SKY]$ and $[HUT]$.

3.2. Global analysis

The objective is to assign a collection-dependent value to a term, depending on its distribution in the collection. The idea is to consider not only the documents in which the term appears, but also the documents containing some terms specific to it. Indeed, the documents that are *about* $[SKY] \rightarrow (Above) \rightarrow [GRASS]$, are also *about* $[SKY] \rightarrow (Above) \rightarrow [PLANT]$. Consequently, if a query contains $[SKY] \rightarrow (Above) \rightarrow [PLANT]$, then the documents containing the term $[SKY] \rightarrow (Above) \rightarrow [GRASS]$ are relevant to it.

The occurrence of a term t' in a document includes the *implicit* occurrence of the terms $\{t|t' \leq t\}$ in this document. If $[SKY] \rightarrow (Above) \rightarrow [GRASS]$ appears in a document, then $[SKY] \rightarrow (Above) \rightarrow [PLANT]$ appears implicitly. We define the occurrence ratio $p(t)$ of a term t as the number of documents in which it appears implicitly, divided by the the number of documents in which $\top_{SG}$ appears implicitly:

$$p(t) = \frac{card\{d \in D|t' \leq t, t' \in d\}}{card\{d \in D|t' \leq \top_{SG}, t' \in d\}}$$

where d is a document of the collection D. If we call n_T the number of documents in which t implicitly appears and N the total number of documents in the collection, then we may define:

$$p(t) = \frac{n_t}{N}$$

We illustrate this idea with an example, in which terms are just concepts, for the sake of simplicity. Consider the collection containing 6 documents described in Table 1. To calculate $p([PLANT])$, all the documents containing $[PLANT]$ or any concept specific to $[PLANT]$ (see lattice in Figure 1) are counted: $p([PLANT]) = 6/6 = 1$. The occurrence ratio of $[TREE]$, which is a term specific to $[PLANT]$, is $1/6$. The occurrence ratio of $[PLANT]$ is higher than the one of $[TREE]$, although they both appear in the same number of documents.

Let's interpret $\neg t$ as the union of all the terms *not* below t in the star-graph lattice:

$$\neg t = \cup_{\neg(t \leq t)} t'$$

With this interpretation, $p(t)$ verifies all classical Kolmogorov properties of a probabilistic distribution and therefore is isomorphic to a probability. The *Global* value of a term t, in a collection, is defined as the amount of information held by the term:

$$Global_t = -log(p(t)) \tag{2}$$

We can compare this value to the classical *idf*: it gives higher values to terms holding more information, while giving lower values to terms holding more information.

We have defined a local and a global analysis for the star-graphs. Based on the above equations, we can now define the weight of a star-graph.

3.3. Weighting a star-graph

The weight W of a term should depend on its importance in the graph and on its distribution in the collection. Based on Equations (1) and (2), a term can be weighted as a combination (e.g. multiplication) of both local and global analysis:

$$W_{t,d} = Local_{t,d} \times Global_t \tag{3}$$

This weight is the result of a local and global analysis, inspired by the use of the *tf.idf*. In the section, we have defined a weight for a star-graph (Equation (3)). In the following section, we describe an implementation and an evaluation for this weighting scheme, on a collection of images.

4. EVALUATION

In this section, we present the results of the evaluation of our approach for image retrieval. Instead of using the classical operation to match conceptual graphs (projection operator [16]), we use star-graphs as index terms for the vector space model of IR [12].

4.1. Image collection

We used a image test collection [4] providing 498 personal photographs and their indexes according to Mechkour's image representation model [7], as well as the concept and relation lattices. The concept lattice is composed with 104 concepts, such as $[SKY]$, $[BUILDING]$, or $[WATER]$, and has a maximal depth of 6. In this collection, the concepts are weighted only according to their relative surface. For instance, if a building occupies 30% of an image, it will be weighted 0.3. The relation lattice contains 46 relations, such as $(Above)$, or (On_The_Left), and has a maximal depth of 3. A sample of the photographs is given in Figure 5. CGs are composed with an average of 29 SGs. In total, 3854 terms are extracted from the index. This corresponds to the number of dimensions of the vector space model. The test collection provides a set of 38 queries with relevant documents association. Figure 6shows one of the queries, describing "a person between two buildings". Queries are *propagated* in order to allow a meaningful matching [6]. That is to say, terms specific to the original query terms are added. Indeed, a query containing a term t should match with a document containing a term t', $t' \leq t$, The complexity of this operation depends on the size of the lattices: the

Document 1	Document 2	Document 3	Document 4	Document 5	Document 6
[PLANT]	[SKY] [GRASS]	[BUILDING] [GRASS]	[SKY] [GRASS]	[SKY] [TREE]	[GRASS]

Tab. 1. Collection of 6 documents.

Fig. 5. A sample of images from the collection.

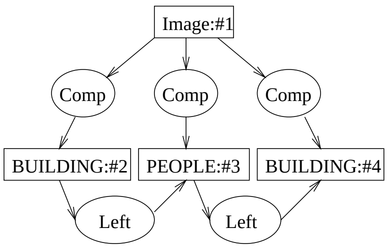

Fig. 6. A query from the test collection.

wider and the deeper they are, the more terms are likely to be added.

The system was built on top of SMART [12], that we used as an evaluation kernel. We are currently evaluating different aspects of our work, and present some results for the following evaluations:

1. Evaluation of the impact of the weighting scheme defined in the equation (3), in comparison to a boolean weighting scheme.

2. Evaluation of the impact of taking into account the relations between the concepts for an IR task, compared to using concepts (or keywords) only.

3. Comparison of the vector space matching with the CG matching.

4.2. Impact of the weighting scheme

This experiment is aimed at evaluating the performance of our weighting scheme, that is integrated to the system. We ran four experiments. Our weighting scheme is decomposed, in order to evaluate the effects of individual elements.

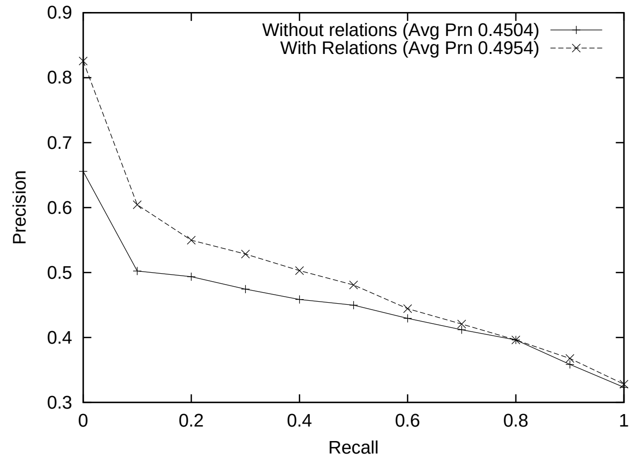

Fig. 7. Recall-Precision curves for the system with and without considering the relations between terms.

We also compared the results with the classical *idf*, calculated over the terms. For each run, we used different weighting schemes. The results are presented Table 2.

We can see that integrating the *Local* value in the document index yields a slight improvement of the average precision across this collection. The use of the *Global* value did improve the average precision as well. In the indexes as in the queries, almost only the most specific concepts (i.e. concepts having no specific in the lattice except $\perp_C$) where used. As the aim of the *Global* value is to moderate the impact of generic terms, while emphasizing more specific ones, the effect of the *Global* value is not as important as we expected, on this test collection. However, we stress that it would be efficient on collections where more generic terms are used in the index as well as in the queries. Further experiments are still being carried out to determine this impact more precisely, especially on the FERMI collection [3].

4.3. Integrating the relations

This simple experiment compares the results of the system with results that we would obtain using *only* concepts, without using the relations between them. The results are shown in Figure 7. The curves have similar precision value for large recall values. This is due to the fact that in case of relational indexing, the term $[Image] \rightarrow (Comp) \rightarrow [OBJECT]$ appears in every document containing the con-

Document weights -	Query weights	Average precision	Relative change in %-age
Boolean	Boolean	0.4850	0.0
$Local$	Boolean	0.4893	+ 0.89
$Local \times idf$	idf	0.4924	+ 1.53
$Local \times Global$	$Global$	0.4954	+ 2.14

Tab. 2. Average precision for 4 different weighing schemes.

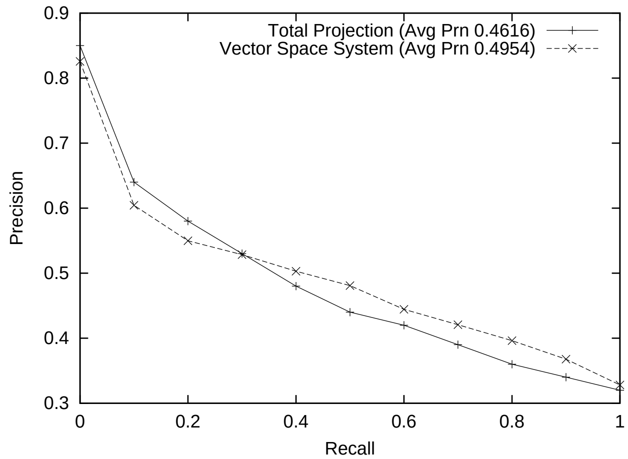

Fig. 8. Recall-Precision curves for the graph and vector matching systems.

cept $[OBJECT]$ and all the documents in which $[OBJECT]$ appears are retrieved. Hence, for high recall, the same relevant documents are retrieved by both systems. The relative improvement on average precision is 9.99%, which is important. Taking into account that relations helps the user to retrieve more relevant documents, for this collection, yields a real improvement compared to considering an image as a set of concepts only. This confirm previous results shown in [10].

4.4. Vector matching versus CG matching

The aim of this experiment is to highlight that the conceptual graph matching can be simulated with a vector space matching. To illustrate, we have compared the result of our system with a system implanting the total projection operator for conceptual graphs. This system uses the relative surface of objects only for concept weighting where the relevance value is based on the sum of the weights of the SGs on which the query is projected, inspired from [10]. The results are show in Figure 8. We can see from the curves that for low recall values, the vector space matching system is less precise than the graph matching system. There may be a model-dependent reason and a collection-dependent reason for that.

First, the use of SGs as documents descriptors in the context of a vector space model does not make it possible to consider join information between SGs. Hence the system retrieves less relevant documents, while the system implementing the total projection, which is precision-oriented, retrieves more relevant documents. Second, the image representation lacks semantic. That is to say, for instance, it does not represent the relation $(Between)$ for three concepts, but two relations (On_The_Left). Hence the vector space system would retrieve some documents containing only one relation (On_The_Left) (which satisfies a partial projection), that are not relevant to the user. For high recall values, the vector matching system achieves better results. Indeed, as the vector matching system implements a partial matching, it is more flexible than the graph matching system and consequently retrieves *partially* relevant documents, that are not retrieved by the other system.

Processing a query requires an average of 6 s. of CPU-time for the graph matching system, and less than 1 s. for the vector space system, with a Pentium III 733MHz processor. The relative gain of average precision is of 7.32%, moreover the time required to process a query is divided by 6. The average precision is higher for our system, however there is a loss of precision for low recall values. Further research is required to determine which factors make a document relevant to the user, and make a document relevant to a query in image retrieval.

5. CONCLUSION

In this paper, we defined an image representation model that allows relational indexing of image content. This model is based on star-graphs. We showed how the weight a star-graph for the purposes of IR, employing a *Local* and a *Global* analysis for a star-graph. Our approach introduces a novel framework to combine an expressive knowledge representation formalism with a classical IR model.

We applied our research to image retrieval, in the context of a vector space model. Experiments on an image test collection have shown that integrating our weighting scheme yields improvements in average precision. Using star-graphs as index terms allows for relational indexing, and expression of complex user needs. As a consequence, it helps users to retrieve more relevant documents compared to using only keywords for image indexing and query representation. The results of a vector space system using star-graphs as index terms are comparable to those of a system implementing the total projection for conceptual graphs.

Moreover, the vector space system is faster. Further experiments are still being carried out to, especially on the larger FERMI image test collection [3].

Although we have applied the use of SGs as index terms to image retrieval in the context of a vector space model, we think that it could be used in any other media. Moreover, the approach could extend the classical textual IR systems, providing a more powerful indexing language.

6. REFERENCES

[1] G. Amati and I. Ounis. Conceptual graphs and first order logic. *The Computer Journal*, 43(1):1–12, 2000.

[2] E. Bertino and B. Catania. A constraint-based approach to shape management in multimedia databases. *ACM Multimedia Journal*, 6(1):2–16, 1998.

[3] Y. Chiaramella and M. Mechkour. *Indexing an image test collection.* Technical Report, FERMI BRA 8134, 1997.

[4] J.-H. Lim. Building visual vocabulary for image indexation and query formulation. *Pattern Analysis and Applications (Special Issue on Image Indexation)*, 4(2/3):125–139, 2001.

[5] P. E. Maher. Conceptual graphs - a framework for uncertainty management. In *NAFIPS'91*, pages 106–110, 1991.

[6] J. Martinet, Y. Chiaramella, and P. Mulhem. Un modèle vectoriel étendu de recherche d'information adapté aux images. In *INFORSID'02*, pages 337–348, 2002.

[7] M. Mechkour. *Un modèle étendu de représentation et de correspondance d'images pour la recherche d'informations*. Ph.D. Thesis, Joseph Fourier University, Grenoble, 1995.

[8] S. Morton. *Conceptual Graphs and Fuzziness in Artificial Intelligence.* PhD thesis, University of Bristol, 1987, 1987.

[9] I. Ounis. A flexible weighting scheme for multimedia documents. In *Database and Expert Systems Applications*, pages 392–405, 1999.

[10] I. Ounis and M. Pasca. Relief: Combing expressiveness and rapidity into a single system. In *SIGIR'98*, pages 266–274, 1998.

[11] V. Plachouras and I. Ounis. Query-biased combination of evidence on the web. In *Mathematical/Formal Methods in Information Retrieval Workshop, ACM SIGIR*, 2002.

[12] G. Salton. *The SMART Retrieval System*. Prentice Hall, 1971.

[13] G. Salton and C. Buckley. Term-weighting approaches in automatic text retrieval. In *Information Processing and Management*, pages 513–523, 1988.

[14] G. Salton and M. McGill. *Introduction to Modern Information Retrieval*. McGraw-Hill, 1983.

[15] E. Di Sciascio, F. M.Donini, and M. Mongiello. Structured knowledge representation for image retrieval. *Journal of Artificial Intelligence Research*, 16:209–257, 2002.

[16] J. F. Sowa. *Conceptual Structures*. Addison-Wesley, Reading, MA, 1984.

[17] J. Z. Wang and Y. Du. Rf*ipf: A weighting scheme for multimedia information retrieval. In *ICIAP*, pages 380–385, 2001.

[18] V. Wuwongse and M. Manzano. Fuzzy conceptual graphs. In *ICCS'93*, pages 430–449, 1993.