A framework for Mixed Symbolic-based and Feature-based Query by Example Image Retrieval

Philippe Mulhem[1], Emmanuel Debanne[2]

[1] CLIPS-IMAG
BP 53
38041 Grenoble Cedex 9
Philippe.Mulhem@imag.fr

[2] I^{2}R
21 Heng Mui Keng Terrace
Singapore 119613

Abstract

This paper defines and studies the use of query by example (QBE) in the context of photograph retrieval. The novelty of our approach lies in considering an automatic indexing process of photographs as well as a representation of the features of image regions in a single knowledge representation formalism. Both symbolic and feature based representation are used during the query by example process. More precisely, the QBE process is able to take into account the symbolic descriptors of the images but also the extracted features from image under the form of histograms. The aim of this process is to detect the relative importance of the symbolic elements and of the feature elements according to the user's query by example. We experiment the query by example process on two collections of a total of 1100 photographs. The precision measures we have obtained are as good as a baseline defined as explicit textual queries processing.

Keyword: Symbolic Image Indexing, Image Retrieval, Conceptual Graphs.

I. Introduction

Retrieval of still photographs, as described in [27], is a difficult task because computer systems do not really know yet how to accurately link visual features to symbols. That is why many image retrieval systems are based solely on signal features and not on symbols, even though some attempts are considering this difficult task. The problem we address in this paper is to describe and to experiment an integrated model which is able to manage several modes of image retrieval based on symbols and signal based features: queries typed as simple texts and queries by examples where the user selects examples close to the desired photographs in the corpus. One interesting characteristic of the work described here is to take into account the fact that the symbols describing the image may be extracted automatically or manually, thereby enabling the management of large quantities of data.

An information retrieval system requires simple yet accurate interaction. Relevance Feedback (RF) techniques (see pages 140-145 of [26]) are known to be simple for a user, because she/he does not need to know the vocabulary that describes document content to obtain satisfactory results. RF allows a user to select relevant and/or non-relevant documents from the results of an initial query, and then generates a new query according to the content of the selected documents and the original query. Query By Example (QBE) is a specific case of RF, assuming an initial query that retrieves all the documents in the corpus. RF or QBE processing in existing image retrieval systems is mostly based on signal features and not symbols, leading to a mismatch between user's considerations and computer manipulated data. In the work described here, the users as well as the system make explicit use of symbols: the images are indexed using symbols, and when a user looks at images she/he thinks with symbols, so the gap between the two actors of the retrieval is narrowed. Our opinion is that the system is then able to communicate more effectively with the user.

The two following examples explain in more details our goals and motivations. These examples assume both errorless symbolic description of objects in the photographs and color descriptions of these objects.

- Example 1: Consider that a user selects the two photographs of Fig. 1 as relevant to her/his information need. On the left image of Fig. 1 we see a red orchid and on the right-hand sided photograph we see a red circus capital. Because the objects are of different kinds, the system in this case should consider the description of the photographs (a red orchid and a red circus capital) to generate a query that represents the fact that the user is in fact looking for red objects, whatever the object are. In this case, the symbolic description of the images should have less importance than the signal-based color description.

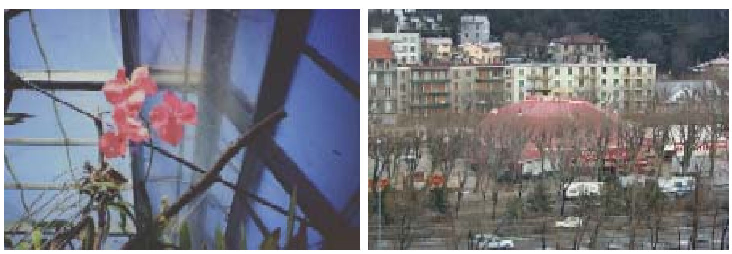


Figure 1. Two photographs selected, color-based QBE retrieval.

- Example 2: In this case, a user selects the two photographs presented Fig. 2 as a Query By Example. These two photographs contain flowers, but their colors are very different: on the photograph on the left-hand side the flower is yellow and in the photograph on the right-hand side the flower is red. In this case, the system should consider that the information the user is interested in is primarily flowers (i.e. we consider more the symbolic representation of the image), and then consider the color as a less important factor.

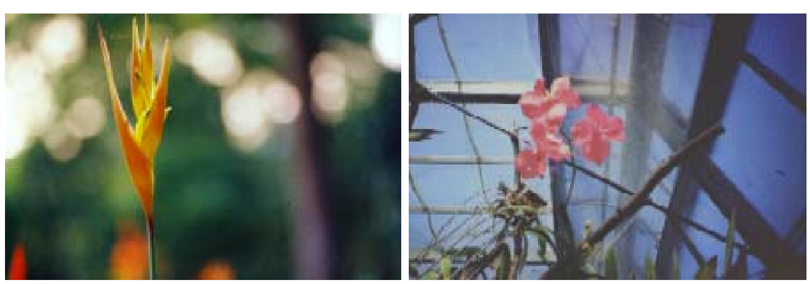


Figure 2. Two photographs, symbolic-based QBE retrieval.

According to the examples above, we define models and processes that allow a user to express her/his needs by examples and that ensure effective balancing between the symbols and/or color features of images. Besides, uncertain automatic generation of symbolic image descriptions is taken care in our proposal, as well as the weights that reflect, as in textual information retrieval, the importance of the components in an image content representation.

We describe in section II the works related to our concern, namely image retrieval systems and relevance feedback techniques. In section III we focus on the model of images that supports the retrieval. Section IV presents an overview of our proposal, while section V focuses on providing the fundamentals on the relevance status value computation during the graph matching process. The section VI is dedicated to the query by example processing. Experimental results on two corpuses are presented in section VII, and we conclude in section VIII.

II. Related Works

Existing content-based image retrieval systems differ on their definition of what is an "image content":

- The first category of approaches, namely signal-based, considers the raw digital information (i.e. the matrix of pixels) as image content. For signal-based indexing, query by example (image or sketch) are extensively used, as the symbolic description of images is not addressed. QBIC [7], VisualSeek [28, 29] and Blobwords [2] are example of such systems. They usually incorporate relevance feedback techniques. Other works [19], specifically dedicated to relevance feedback on signal-based descriptions of images, perform well but do not intend to fill the gap between image features and symbols. With such system, the execution gulf during the retrieval task [20] between the user's need and the expression of the query is huge, because the user has to translate his information need into a signal-based description, hoping that the system uses adequate feature matching. Compared to such approaches, our concern here is to allow query by example based both on semantic and feature based description of images in a way to ease the interaction between the user and the system.
- The second category of approaches considers the explicit semantic interpretation of images. Among other works, the content description of MPEG-7 [17] and the Dublin Core Metadata Initiative [34] fit into this category. Such approaches take into account the fact that when people describe images [9, 11, 12] they use symbols and not signal features. Symbolic-based descriptions are able to manage complex representations [8, 18], but as pointed out in [24], symbolic descriptions lack of scalability, are tedious to use and subject to inconsistencies due to human intervention in the indexing process. In this case, the execution gulf is smaller, and the system has to fill accurately the gap between signal and symbols. Approaches has been achieved in learning symbols from image feature regions using a priori samples [14, 15, 33] or relevance feedback [35], but in our work we consider that simple lists of labels do not represent adequately image content, hence the use of graphs. Other results [6] propose ways to extract and label salient objects, as well as to label whole images, with still a limitation on the vocabulary size (less than 40 keywords). Regarding approaches that intend to automatically link images and words, like [10, 23, 32] for instance, the lack of explicit relationships between regions and the keywords forbid the use of real integration between both low-level features and

symbolic descriptions of the images; on the other side, the vocabulary that describes images is larger than with the region labeling approaches above.

When we consider image retrieval systems, we need to focus on the possible retrieval modes. Relevance feedback approaches are well known in the information retrieval community since the 70s with the work of Rocchio [25]. When using the classification of relevance feedback modes studied in [13], the signal-based systems are mainly able to manage *opaque* RF (i.e. the user selects relevant and/or non-relevant documents and then see the revised ranking without any other action possible). In their experiments, Koenemann and Belkin found out that other relevance feedback interactions, namely *transparent* (the system displays the new query generated from the documents selected by the user, allowing a check before actually running the RF query) and *penetrable* (the system allows modification of the generated query before query processing) slightly increase the quality of the results. We believe that transparent and penetrable interactions are only manageable when using symbolic data; this is why we make use of symbolic descriptions of images in this work.

III. The Image Model

The goal of the image model presented here is to describe the relevant content of the images based on objects, on relations between them, and also on histograms related to image regions associated to the visible objects. Such histograms may represent colors because colors are very important in describing image content.
The conceptual graph formalism [30, 31] is used to support our image model. This formalism has already been applied on photograph content representation [15, 18], and has also been shown to be compatible with an inverted file implementation [21] and with a vector space implementation [16].
Conceptual graphs are oriented bipartite finite graphs composed of concept nodes and of relation nodes. Concepts node are composed of a concept type and a referent (generic or individual). A generic referent denotes the existence of a referent, while an individual refers to one instance of the concept type. In our case, the concept types represent the objects of the real world visible in the photographs, the image or the histograms; they are organized in a lattice that reflects generalization/specialization relationships. Fig. 6 presents a concept lattice where for instance the concept type *palm* is a sub-type of type *tree*, noted *palm≤tree*. In the image model, absolute and relative spatial relationships are defined. Absolute spatial relationships link the image and the object concepts (coming from an existing labeling process, outside the scope of the paper) and indicate the position of the center of gravity of a region corresponding to one object by a couple of integers between 0 and 4. For instance, the relation Center22 between an image and a visible object concept denotes that the region in which the object is visible has its center of mass at the center of the image. Relations are also organized into a lattice, indicating for instance that the relationship "touch" is a specific of the relation "close_to".

A conformance relation, noted "::", links concept types and valid referents for each type. For instance "palm::#p1" denotes the fact that the referent #p1 is a valid referent for the type palm. If we consider the concept type hierarchy of Fig. 6 then "tree::#p1" holds also, because palm is a sub-type of tree. In the following, we use the minimal conformance relation, noted "$::_m$", defined as: "$x::_m y$" exists if "x::y" and no concept type x' so that "x'≤x and x'::y" exists.

Syntactically correct graphs are built upon a canonical base of graph. The canonical base is a set of general graphs that express the possible relationships and the possible concepts that may occur in syntactically correct graphs. The syntactically correct graphs are built upon the canonical base using the building operators [30] copy, restriction, joint and simplification.

In this work, we use a weighting scheme that supports the computation of the relevance value of images according to queries. The weighting scheme we propose is inspired from [22]. For textual documents, the well known *tf.idf* (term frequency multiplied by inverse document frequency) values [26] model the importance of a term in a document with respect to a document collection. We limit ourselves to weights that compute visual term frequencies, because the advantage of *idf* in the case of image retrieval has never been proved so far. We however consider the certainty of the recognition of the concepts, because automatic recognition processes are not errorless. So, we associate each concept that corresponds to a visible object in the image with two values:

- A weight *w* of a concept represents the importance of a visible object in the photograph. Many parameters may influence the weight of the objects. We compute the weight of an object as the probability that one pixel of the photograph may be in its region: *w = surface(object_region) / surface(image)*.
- A certainty of recognition *c* of a concept coming from a labeling process. In case of manual indexing, this certainty is equal to 1.

A concept corresponding to an indexed visible object in an image is then represented as: [type: referent | *w* | *c*]. Fig. 3 shows a concept [Sky: #sky1 | 0.32 | 1.0] that indicates that the occurrence "#sky1" of the concept "Sky" has a weight of 0.32 and a certainty of recognition of 1.0. In the following, we call symbolic concepts the concepts that do not represent features extracted from the images. In Fig. 3, such concepts are [Sky: #sky1 | 0.32 | 1.0] and [Water: #water1 | 0.27 | 0.5].
The image model also considers extracted features from an image. The concept type Histo in Fig. 3 is not symbolic but feature-based: the referent of such concept type is an instance of a histogram. For such concepts, the weight *w* is the same than the visible object related to the region on which the histogram is computed, and the certainty of recognition equals 1.0 . For the sake of simplicity, we only consider one concept type Histo, color histograms, in the remaining of this paper, but the extension to several histogram concept types (for instance one for colors and one for textures) is almost straightforward. Concepts occur in arches: an arch *a* is a triplet (c^{in}, r, c^{out}) where c^{in} and c^{out} are concepts, and r is a relation linking c^{in} to c^{out}.

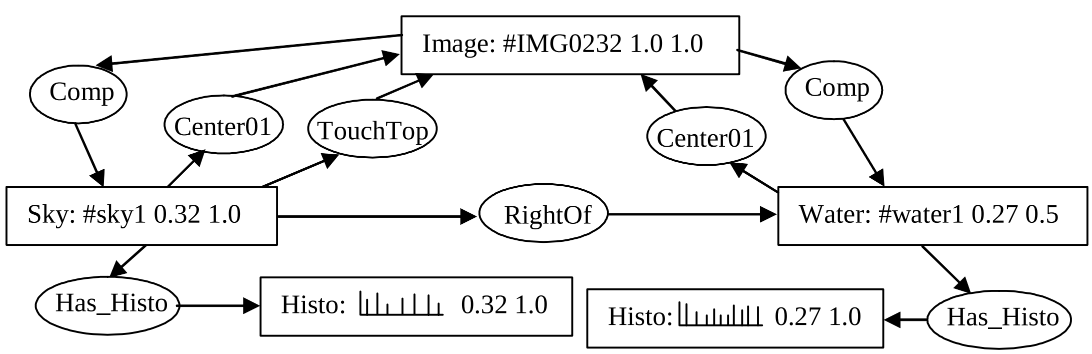


Figure 3. The Conceptual Graph representing an image.

We differentiate in part 6 the arches that contain symbolic concept only (called symbolic arches) from the arches that contain one histogram concept and one symbolic concept (called non-symbolic arches). The relations considered in this paper are binary, without loss of generality. The *Image* concept has a weight of 1.0 and a certainty of recognition of 1.0.

In a way to retrieve relevant images according to a query, we make use of the projection operator defined by [30]. The projection of a graph v into a graph u, noted $\pi_v(u)$, is the set of sub-graphs of u so that each concept c of v is associated to a specific (or equal) concept noted πc in u, and each relation r between concepts c^{in} and c^{out} of v is associated to a specific (or equal) relation noted π_r in u that relates πc^{in} and πc^{out}. The projection may not be unique, that is why the projection gives a set of graphs as result. In our case, the weights and certainty of recognitions will be integrated to achieve ranked results during the QBE process.

IV. Overview of the proposal

This section summarizes our proposal and presents one example, using the image model described in section III. We briefly show how we manage both symbolic and non-symbolic features (namely color histograms) of image elements during Query by Example processing.

Consider the photographs presented in Fig. 1. The one on the left contains a flower and the one on the right a capital of a circus. Consider the (simplified) graphs that correspond respectively to each of these photographs in Fig. 4. In these graphs, we present only three-dimensional histograms in RGB space, one bin for red, one bin for green and one bin for blue. For instance, the histogram of the left graph in Fig. 4 is composed of 70% of red, 10% of green and 20% of blue. The concept types "flower" and "capital" are part of the concept type hierarchy of Fig. 6.

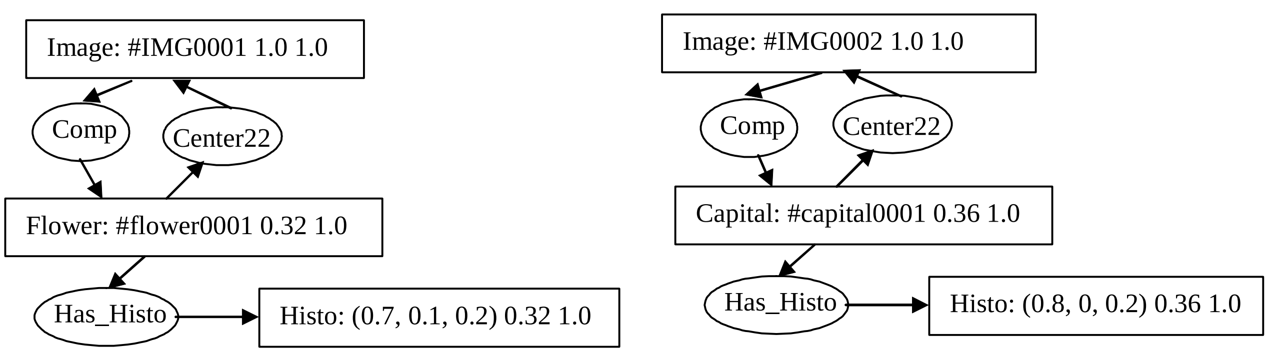


Figure 4. Simplified graphs of respective photographs (left and right) of Fig. 1.

If the two images of Fig. 1 are selected as a Query By Example, and if the other images seen by the user are dissimilar in colors, our proposal assumes that the symbolic concepts flower and capital (as well as their generics in the concept type hierarchy: vegetation and construction) are around half as important to express the query than the color part of the generated query (see part 6). This comes from the fact that the colors of selected images are similar, whereas the symbolic concepts of the image descriptions are very dissimilar according to the concept type hierarchy. So, the system will retrieve in first places other images that have similar color elements to both of the histograms of the two images selected (if such images exist) when processing the generated query.

Consider now the case of the images in Fig. 2. According to similar steps than for Fig. 4, we give in Fig. 5 a graph representation of the content of such images.

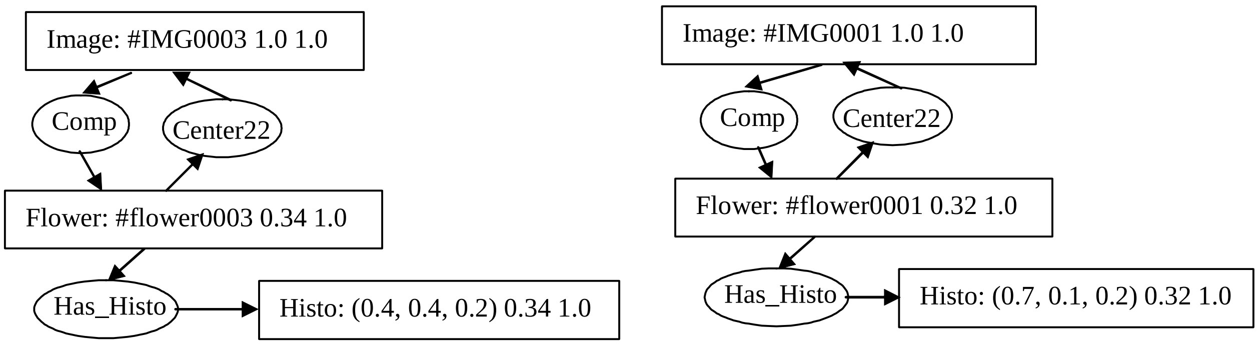


Figure 5. Simplified graphs of respective photographs (left and right) of Fig. 2.

If the two images of Fig. 5 are selected with no other images with flowers seen, our approach will consider that the flower concept type is very important in the generated query. The two color histograms will bee kept in the generated query, and because they are dissimilar (Euclidian distance of 0.42) no other image will have one histogram close to both of these two histograms. This leads to the fact that the symbolic aspect of the two selected images (i.e. flower) will be considered more important during the Query By Example matching process.

V. Basics on image graph matching

We introduce now the elements that are used during the retrieval of images described by graphs, using queries by example. The retrieval process considers the concepts (symbolic and non-symbolic), but also the relations, that occur in an image *d* and in a generated query *q*. Our approach is somewhat similar to the work of Iadh Ounis in [22]: because of the strong separation between symbols and feature-based characteristics, the work of Ounis is unable to fit in the different cases related to query by example as described in the introduction. In [1], the authors focus on efficiency of matching on graphs models of image descriptions; the Attributed Relational Graphs do not reflect the features of relations and concepts lattices, and we believe that these extensions are a must for effective retrieval of images and visual media data.
To process the matching between a query and an image, we firstly apply a filter using the projection operator on the image graphs (see end of section III). A second step is related to the computation of the retrieval status value (RSV) of the image graphs according to the generated query. To achieve this second step we use the following elements:

- The importance of the image concepts and arches
- The importance (called *wish* in the following) of the concepts and arches in the generated query.
- The similarity between the concepts and arches of images and queries.

V.1. Matching of concepts

The matching of concepts depends on the similarity between a concept of a document and a concept of a query. We describe here how we compute such a value, using the function $match_c$. The importance $impc_d$ of any image concept (i.e. either symbolic or histogram) is defined as the product of its weight and its certainty of recognition:

$$impc_d(c) = w_c \cdot c_c \qquad (1)$$

Formula (1) has one main characteristic: when the weight w_C (resp. c_C) is near to zero, then the importance of the concept *c* is also near zero.
The matching of a query concept c_q and an image concept c_d is defined as:

$$match_c(c_d, c_q) = impc_d(c_d) \cdot simc_q(c_d, c_q) \quad (2)$$

In formula (2), the similarity $simc_q$ denotes the matching of the complex[1] concept c_q of the generated query and the image concept c_d. This function is described in section VI for symbolic and non-symbolic concepts.

V.2. Matching of arches

Consider an arch $a_d(c^{in},r,c^{out})$ from an image description graph. We assume that the importance of such an arch is related to the importance of its concepts. This is why we use the standard definition of fuzzy logic conjunction to express the importance of an arch:

$$impa_d(a_d(c^{in}, r, c^{out})) = \min(impc_d(c^{in}), impc_d(c^{out})) \quad (3)$$

According the formula (3), the importance of an arch cannot be greater (resp. smaller) than the higher (resp. lower) importance value of its related concepts.
The matching value between an images arch a_d and a query arch a_q is defined as:

$$match_a(a_d, a_q) = impa_d(a_d) \cdot sima_q(a_d, a_q) \quad (4)$$

As for the concepts, the similarity between the image and query arches $sima_q$ (defined in part 6) also manages the complex arches of a query.

V.3. Matching of graphs

The matching of an image graph g_d and a query graph g_q is intended to compute the Relevance Status Value (RSV) of an image according to the generated query from the query by example interaction. This RSV is based on concepts and arches matching values, as in [1]:

[1] See section VI for details.

$$RSV(g_d, g_q) = \max_{p_q \in \pi_{g_q}(g_d)} \Big(\sum_{c_d \text{ concept of } g_d} match_c(c_d, \pi c_d) + \sum_{\substack{a_d \text{ arch of } g_d \text{ that} \\ \text{is corresponding to} \\ a_q \text{ of } p_q}} match_a(a_d, a_q) \Big) \quad (5)$$

The formula (5) may be compared to a dot product where the actual importance of arches and concepts of the image graph is one vector and the matching of the image parts and query parts is a second vector. We do not differentiate in this formula the symbolic parts and the histogram parts of the graphs, hence making clear that these parts are considered in a similar way.

VI. The Query by Example Process

A QBE retrieval interaction process presents a part of the corpus, and the user is required to select images that are representative of his need. The system is then expected to generate an accurate representation of the users' need from the content of the selected images. The problem is somewhat related to the learning by induction process from a learning sample composed by image representations. An additional element that we need to extract from the content of the selected images is the expected *wish* of the user for a concept or an arch to appear in the generated query. To do so, we consider that this wish is in fact composite: it is related to the concepts that are present in the graphs representing the selected images, as well as to their relationships (supported by the arches). More specifically, the query generation is achieved by building a *Compound Least Common Generalizing* (CLCG) graph, i.e., a synthetic representation of the concepts and arches that occur in the graphs representing the selected images.

We describe in the following sub-sections the QBE process in three steps:

1. We focus first on the query generation on a collection where each image is represented by a single concept. This allows us to define the notion of Compound Least Common Generalization (CLCG) of concept for symbolic objects, and CLCG of concepts for feature-based components of the graphs through the example of the color histograms.
2. We explain then the building of CLCG of whole graphs and the additional required steps in order to deal with QBE on images represented by complex graphs (a complex graph is connected graph containing more than one concept). In this case, we integrate the use of the relationships occurring in arches.
3. We define the matching between a generated CLCG and an image graph description.

VI.1. Query generation on Simple Graphs

VI.1.1. Query generation on simple Symbolic Concepts

VI.1.1.1. Compound Least Common Generalization of Symbolic Concepts

A CLCG on symbolic concepts aims at expressing the relative importance of the elements present in the selected images by the user.

As described in Section III, concept types and relationships are organized into lattices. For the ease of the explanation, we only consider tree-like hierarchies instead of lattices for concept type and relation in the following (the absurd concept ⊥ being introduced only to fit the conceptual graph model). In this part only concepts are considered, we use the concept types hierarchy presented in Fig. 6, where the abbreviation of each concept type is between brackets.

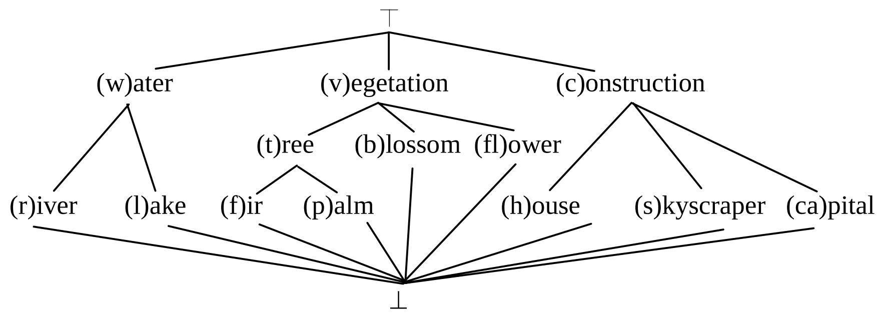


Figure 6. A concept type hierarchy.

Consider a corpus of 15 documents (numbered 1 to 15) indexed by only one concept from the hierarchy of Fig. 6 as described in the last two lines of Table 1 (without considering weights or certainty of recognition). We assume that the referent are related to the "$::_m$" conformance relation to the concept type of the corresponding document (for instance "fir$::_m$#f1" holds). If a user is looking for photographs containing *trees*, we assume in the following that she/he selects the relevant images to her/his need as shown in the second line of Table 1.

Doc Id.	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Selected	√	√	√	√	√										
Concept type	f	f	t	t	v	f	t	b	b	c	b	c	w	w	w
Concept referent	#f1	#f2	#t1	#t1	#v1	#f3	#t2	#b1	#b2	#c1	#b3	#c2	#w1	#w2	#w3

Table 1. Relevance judgments.

According to Table 1, the system should find out that the user focuses on *vegetation*, more specifically on *trees*. We summarize the relevance judgments information by computing the CLCG of the relevant graphs.

In a way to define the CLCG of single concepts, we use as a basis the "concept mixed referent/type hierarchy". Such hierarchy is a set of sub-hierarchies composed of couples [concept type: concept referent] and of a relationship $R_{\leq}$. The concept types of the couples present in the mixed hierarchy are the concept occurring in the canonical base and their specific types, except the absurd type ⊥. The referent that occurs in the couples composed of a concept type x are: i) the generic referent "*", or ii) an individual referents y so that "x$::_m$y". A relation $R_{\leq}$ exists between couples [x:y] and [x':y'] if:

$$[x{:}y\}\ R_{\leq}\,[x'{:}y'] \begin{cases} x \leq x' \text{ and } y = y' = * \\ x = x' \text{ and } y \neq * \text{ and } y' = * \end{cases} \qquad (6)$$

Such hierarchy is composed of independent sub-hierarchies, because it does not contain the generic type T. According to the parameters described earlier, the mixed hierarchy using the concept type hierarchy of Fig. 6 and the referents from table 1 is presented in Fig. 7.

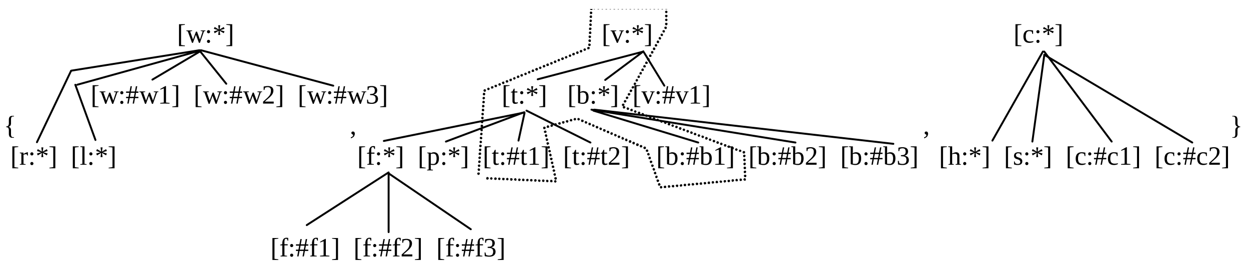


Figure 7: A mixed referent/type hierarchy.

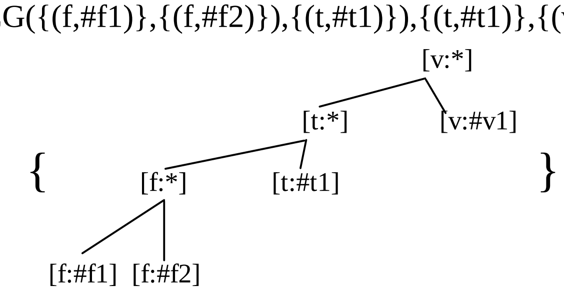
The CLCG of two concepts is built upon the Least Common Generalization (LCG) of two concepts. A LCG of two concepts c_1 and c_2, according to a mixed referent/type hierarchy M, is the set of lowest common ancestors of c_1 and c_2 in M. In the case of tree hierarchies, this set is a singleton or an empty set. Using one of the examples of Fig. 7, we obtain:

LCG([tree:#t1],[blossom:#b1]) = {[vegetation:*]}

A CLCG is the smallest (in term of node cardinality) sub-hierarchy of M that contains c1, c2, and the LCG(c1, c2). For instance, the CLCG([tree:#t1],[blossom:#b1]) is surrounded by a dotted line in Fig. 7.
More generally, the CLCG of two sets of concept hierarchies T_1 and T_2, (sets of sub-hierarchies of the mixed hierarchy), according to a mixed referent/type hierarchy M, is defined as the set of sub-hierarchies of the mixed hierarchy containing T_1, T_2 and the LCG of the roots of each of their components when they exist. The top of a CLCG T (noted top(T)) is the set of the roots (i.e. the more generic concepts) of T.

When considering our previous example with simple graphs, we compute iteratively the CLCG for each of the selected documents. The iterative CLCG of concept types the five marked relevant documents considered in their increasing id number is:

CLCG(CLCG(CLCG(CLCG({(f,#f1)},{(f,#f2)}),{(t,#t1)}),{(t,#t1)},{(v.#v1)})=



The definition of a CLCG is qualitative: it does not put any weight on the concepts but only the concepts that may be interesting to the user.

VI.1.1.2. Valuation

After selecting in the previous part the concepts that are supposed to be useful for the query, the goal of the valuation is to assign a *wish* value of each of the concepts of the CLCG. This value quantifies the interest that the user has in the concept considered.
For each of the concept t of the final CLCG, we weight its relevance based on two values:

$$\alpha_{tq} = \frac{\left|Sel \cap Corpus(t_q)\right|}{\left|Sel\right|} \qquad \beta_{tq} = \frac{\left|(Viewed \setminus Sel) \cap Corpus(t_q)\right|}{\left|Viewed \setminus Sel\right|} \qquad (7)$$

- α_{tq} is the trend for selected image documents (*Sel*) to be indexed by the concept t_q or a concept specific of t_q, where *Corpus(t_q)* is the set of documents indexed by t_q, and *Sel* is the set of documents marked relevant.
- β_{tq} is the trend for **non** selected graphs to contain t_q or a specific of t_q, where *Viewed* is the set of visualized documents during the selection processed. Thus β_{tq} / α_{tq} is as a measure of the significance of the selection of the concept t_q.

Finally the estimated *wish* of a user to retrieve a concept t_q is expressed as:

$$wish(t_q) = u(\alpha_{tq}) \,.\, v\left(\frac{\beta_{tq}}{\alpha_{tq}}\right) \qquad (8)$$

Where *u* and *v* are real functions constrained by the following properties:

- A concept t_q whose presence in the selected graphs is weak ($\alpha_{tq} \leq 0.5$) is considered not relevant.
- If a concept is more forgotten than selected ($\alpha_{tq} \leq \beta_{tq}$), it is considered as non relevant. Otherwise t_q is almost certainly relevant.

Sigmoid-based functions may be used to generate *u* and *v* as presented in Fig. 8.

Considering our previous example, we obtain the results in Table 2. It appears that [vegetation:*], with a wish value of 0.83, and above all [tree:*] with a wish value of 0.93, are evaluated relevant by the system.

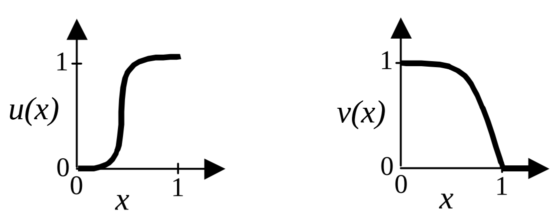


Figure 8: Typical shapes of *u* and *v* functions.

t_q	α_{tq}	β_{tq}	β_{tq} / α_{tq}	wish(t_q)
[vegetation:*]	5/5	5/10	1/2	≈1·0.83≈0.83
[vegetation:#v1]	1/5	0/10	0	≈0.05·1≈0.05
[tree:*]	4/5	2/10	1/4	≈0.97·0.98≈0.95
[tree:#t1]	2/5	0/10	0	≈0.1·1≈0.1
[fir:*]	2/5	1/10	1/4	≈0.1·0.98≈0.1
[fir:#f1]	1/5	0/10	0	≈0.05·1≈0.05
[fir:#f2]	1/5	0/10	0	≈0.05·1≈0.05

Table 2. Valuation of a CLCG.

VI.1.1.3. Similarity Matching

The similarity matching process intends to evaluate the "closeness" between a query CLCG and a concept from a document. This similarity is used in the formula (2) of section V.1. The system is then able to rank the results according to the obtained value.

The matching between a query generated as a CLCG T_q and documents that contain a document concept t_d with a weight *w* and a certainty *c* (cf. section III) is defined as:

$$simc_q(T_q, t_d) = \max_{t_q \in T_q \text{ and } t_q \text{ ancestor of } t_d} wish(t_q) \qquad (9)$$

The similarity of formula (9) is only computed when one of the concepts of T_q is a generic of the concept t_d, otherwise its value is zero.
This formula leads to the following similarities according to our example of table 2:
$simc_q$(T,[vegetation:*|1|1])=0.83, and $simc_q$(T,[tree:*|1|1])=$simc_q$(T,[fir:*|1|1])=0.95.
Images indexed by the concept [tree:*|1|1] are then, as we were expecting, considered more relevant than those indexed by [vegetation:*|1|1].

VI.1.2. Query generation on Simple Histogram Concepts

VI.1.2.1. Compound Least Common Generalization of Histogram Concepts

The creation of CLCGs for non-symbolic concepts is simpler than the one for symbolic concepts, because we do not need to handle several concepts types: we only use the Histo concept type, which is like having a specific hierarchy composed only of the Histo concept type and the bottom ⊥ and top T of the hierarchy. For the CLCG of histogram concepts, the concept [Histo:*] has no interest because we consider that each non-symbolic concept is of this type. We keep however the referents that correspond to the content of the histogram in a mixed concept/referent hierarchy similar to the one described in part 6.1.1.1., leading to consider the CLCG of non-symbolic concepts as a union of the concepts under consideration.

We explain the process on image contents represented as histograms with 4 bins: we then represent an histogram that has x for the first bin, y for the second bin, z for the third bin and t for the fourth bin, a weight of p and a certainty of c with [Histo:[x,y,z,t]|p|c], or compactly [Histo:H|p|c] with H=[x,y,z,t]. Consider that

- 15 images D_i, with $1 \le i \le 15$, indexed each by one histogram concept [Histo:H_i|1|1], have been seen by the user,
- The first five images D_j, with $1 \le j \le 5$, are marked relevant by the user.

The CLCG computed on the five relevant images is then:

CLCG(CLCG(CLCG(CLCG({[Histo:H_1|1|1]},{[Histo:H_2|1|1)}),
{[Histo:H3|1|1]}),{[Histo:H4|1|1]}),{[Histo:H5|1|1]})=
{[Histo:H_1], [Histo:H_2], [Histo:H_3], [Histo:H_4], [Histo:H_5]}

So the CLCG of several histograms keeps the different histograms, and represents then all the histograms that correspond to the relevant images. The valuation determines the importance of the different histograms of the CLCG.

VI.1.2.2. Valuation

During the valuation of the CLCG of histogram concepts, we do not make use of referent equality, but we employ a similarity function between histograms, namely F_h. F_h equals 1 for equal histograms.

Like the α_{tq} and β_{tq} of formula (7), the α_{hq} and β_{hq} values provide a basis to compute the importance of a query histogram concept hq:

$$\alpha_{hq} = \frac{\sum_{h \in Sel/\{hq\}} F_h(hq,h)}{|Sel|} \qquad \beta_{hq} = \frac{\sum_{h \in Viewed \setminus Sel} F_h(hq,h)}{|Viewed \setminus Sel|} \qquad (10)$$

- The α_{hq} value expresses the trend for the selected documents (i.e., Sel) to have a content that is similar in colors.
- The β_{hq} expresses the trend that the non-selected documents are similar to h_q.

Having defined the α_{hq} and β_{hq} values, we estimate the wish for a user to retrieve a given histogram concept h_q by the formula (11):

$$wish(h_q) = u(\alpha_{hq}) \cdot v\left(\frac{\beta_{hq}}{\alpha_{hq}}\right) \qquad (11)$$

This function is similar to the one defined of formula (8) (see part VI.1.2.).

VI.1.2.3. Similarity Matching

The evaluation of the matching between one histogram of an image of the database and the CLCG of histogram concepts of the query has to give a low value if the CLCG contains very different colors, indicating that the color aspect is not important in itself. To enforce this property, we propose the expression of formula (12) for histogram concepts, H_q being the set of histogram concepts from the selected images, and h_d being the concept histogram considered:

$$simc_q(H_q, h_d) = \frac{\sum_{h_q \in H_q} F_h(h_d, h_q).wish(h_q)}{|H_q|} \quad (12)$$

The similarity is the average of the product of the query histogram wishes and the similarity between the query and image histogram concepts. If a histogram of an image is very similar to colors that have large wish values, then the similarity is large. If a histogram of an image is very similar to colors that have small wish values, then the similarity is small.

VI.2. QBE on Complex Graphs

The explanation of sub-section VI.1 is limited to simple graphs. We explain now its generalization on complex graphs.

VI.2.1. CLCG of Graphs

Complex graphs are composed of concepts and relations. We explained above how to handle single concepts. However, when dealing with complex graphs the question of which concepts are to be used for the definition of the CLCGs has to be solved. Our concern in this work is to avoid NP-complete sub-graph computations, knowing that the QBE process has to give fast answers to the image retrieval system users. That is why we select one-to-one concept maxima as a basis for the CLCG of complex graphs.

The following algorithm defines the CLCG G of two graphs *A* and *B*. We call C_X (resp. R_X) the CLCG concepts (resp. relations) of the graph *X*, with $X \in \{A, B\}$. c_X and r_X are the sets of roots of the CLCGs C_X and R_X. The CLCG of a relation is computed in the same way than as CLCG of symbolic concepts.

1. To each concept C_A, we associate all the concepts C_B which minimize $d(C_A, C_B)$: for a symbolic concept d is the number of nodes between C_A and $LCG(C_A, C_B)$, and for an histogram concept, d is equal to $1\text{-}F_h$. We emphasize on concepts because they are the most important components of conceptual graphs.
2. To each R_A belonging to all the sub-graphs of pattern $[C_A^{in}] \rightarrow (R_A) \rightarrow [C_A^{out}]$, we associate all the relations R_B from the sub-graphs of pattern $[C_B^{in}] \rightarrow (R_B) \rightarrow [C_B^{out}]$ which minimize $d(R_A, R_B)$, with C_A^{in} and C_B^{in} associated as well as C_A^{out} and C_B^{out}. The result is the association of all the relationships that are somewhat similar.
3. We build the best substitution between concepts of *A* and *B* based on
    - The higher number of association of concepts. We consider this condition first because the concepts are the most important elements in conceptual graphs.

- If equal, the higher number of association of relations with their concepts *in* and *out*[2] belonging to the substitution. We consider the substitution that leads to the maximum of associated relationships.
- If equal, the maximal following value is used:

$$\sum_{\substack{c \text{ in concepts of } B \\ \text{in the substitution}}} impc_d(c) \quad + \quad \sum_{\substack{c \text{ in arches of } B \\ \text{in the substitution}}} impa_d(c)$$

This substitution allows building the CLCG graph *G* composed of the CLCG concepts CLCG(C_A, C_B) and of the CLCG relations CLCG(R_A, R_B).

4. Concepts and relations from *A* and *B* which do not belong to this substitution are added to *G* in order to keep the whole information of *A* and *B.*

As we seen in step 4 above, a CLCG of two graphs may generate a set of graphs. So, an iteratively built CLCG that uses a set of graph as *A* and an image graph as *B* apply the above steps for each graphs, in the set *A*.

We give an example of the CLCG of ten complex graphs limited to symbolic concepts, noted G_i for $1 \leq i \leq 10$. In a way to ease the explanation, we show only non-connected parts of real graphs, represented as set of arches, as well as we do not present the certainty and weights, because they are not used during the CLCG generation:

G_1 = { tree:#t1 → close to → construction:#c1 }
G_2 = { fir:#f2 → touch → house:#h2, palm:#p2 → close to → lake:#l2 }
G_3 = { fir:#f3 → touch → house:#h3, river:#r3 }
G_4 = { fir:#f4 → intersect → mill:#m4 , river:#r4 }
G_5 = { blossom:#b5 → touch → skyscraper:#s5 }
G_6 = { blossom:#b6 → touch → house:#h6, palm:#p6 → touch → fir:#f6 }
G_7 = { lake:#l7 → close to → skyscraper:#s7 }
G_8 = { lake:#l8 → intersect → fir:#f8 }
G_9 = { palm:#p9 → close to → lake:#l9 }
G_{10} = { lake:#l10 → close to → blossom:#b10 }

Consider that a user selects for her/his query by example the images described by the graphs G_1, G_2, G_3 and G_4. According to the description above, the computed CLCG is presented in Fig. 9.

VI.2.2. Valuation

The valuation of CLCGs of single concepts is similar to what was described above in part 6.1., except that we use all the concepts of each graph of the selected and viewed images.
We focus on arches valuation in the following.

[2] Where *in* and *out* correspond to the concepts C_B^{in} and C_B^{out} of the second step.

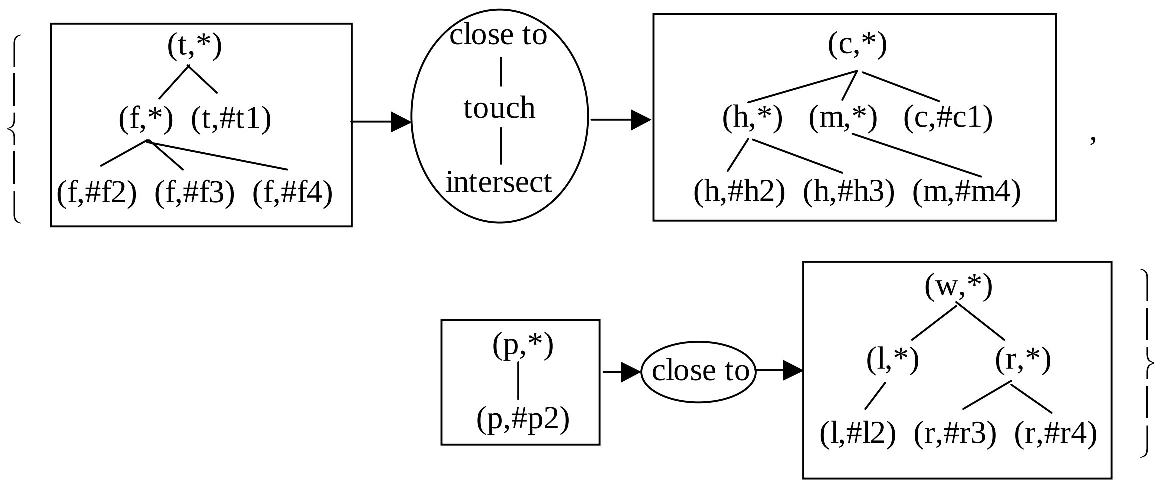


Figure 9: A CLCG for complex graphs.

For an arch $a_q(C^{in},r,C^{out})$ (with C^{in} and C^{out} CLCG concepts) we define α_{aq} and β_{aq} in formula (13):

$$\alpha_{aq} = \frac{|Sel \cap Corpus(a_q)|}{|Sel|} \qquad \beta_{aq} = \frac{|(Viewed \setminus Sel) \cap Corpus(a_q)|}{|Viewed \setminus Sel|} \qquad (13)$$

Where :

- α_{aq} is the trend of the selected images to contain the considered arch. Corpus(a_q) is the set of images that are indexed by a specific of the arches ($C_1 \in top(C^{in}),r,C_2 \in top(C^{out})$) for symbolic arches (arches composed of two symbolic concepts), and the set of images that are indexed by a specific of the symbolic concept of the arch and by one of the histograms of the a_q arch according to a threshold given on the F_h value for the non-symbolic arches.
- β_{asq} is the trend for the graphs of the non selected (but viewed) images to contain the considered arch.

Then, the wish of an arch is defined as in formula (14):

$$wish(a_{sq}) = u(\alpha_{asq}) \,.\, v\left(\frac{\beta_{asq}}{\alpha_{asq}}\right) \qquad (14)$$

In Fig. 10, we show the α_{tq}, β_{tq} and *wish* values for each of the concepts and the α_{aq}, β_{aq} and *wish* values for the arches of the CLCG above. The values that correspond to the arches are displayed along with the relations:

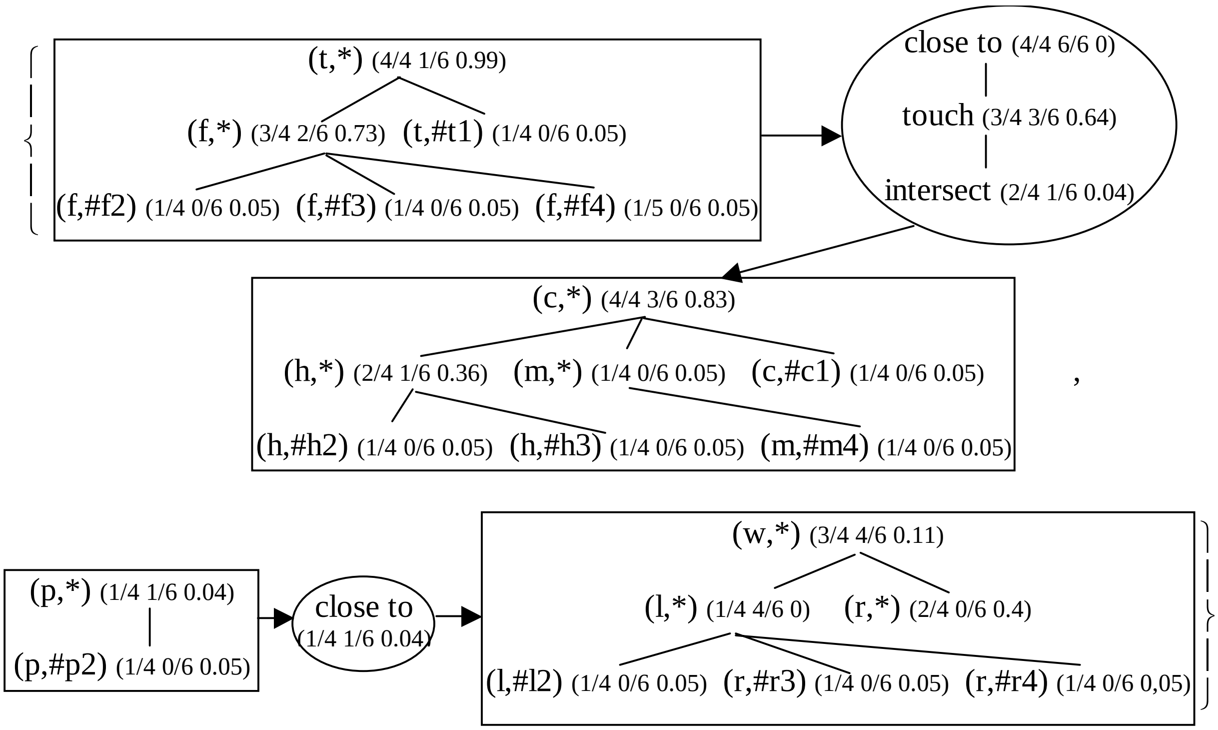


Figure 10: A decorated CLCG of complex graphs.

VI.2.3. Matching of arches

The goal of the matching of symbolic arches is to give the best wish of the query arch that matches an image arch. The matching of an arch composed (see formula (15)) of two symbolic concepts is similar to the matching function for symbolic concepts (formula (9)):

$$sima_q(A_q, a_d) = \max_{a_q \in A_q \text{ and } a_q \text{ ancestor of } a_d} wish(a_q) \qquad (15)$$

In the above formula, a query arch $x(c_x^{in}, r_x, c_x^{out})$ of a CLCG is called a ancestor of an image arch $y(c_y^{in}, r_y, c_y^{out})$ if the CLCG concept c_x^{in} (resp. c_x^{out}) of x contains at least one generic of the concept c_y^{in} (resp. c_y^{out}) of y, and if the relationship r_x of x is a generic of the relationship r_y of y.
We use a very similar approach for histogram arches, except that for the histogram concepts we only consider the existence of a histogram concept in the query arch and we do not use the top of the histogram CLCG.

VII. Experiments

The experiments that evaluate the accuracy of our approach were conducted on two collections representing a total of more than 1100 images. The interface of the DIESKAU (Digital Image rEtrieval based on Symbolic Knowledge and histogrAm featUres) system developed is displayed in

Fig. 11. In this figure, we see in the upper-left part the text-based query region, where a history shows the previous queries. Under this region we have the part dedicated to the QBE, where the user can see the query generated after selection of images, and where the generated query can be executed. The parameters part (bottom left) is dedicated to tune the matching process. The right part of the interface displays images and allows the selection of relevant images for the QBE process.

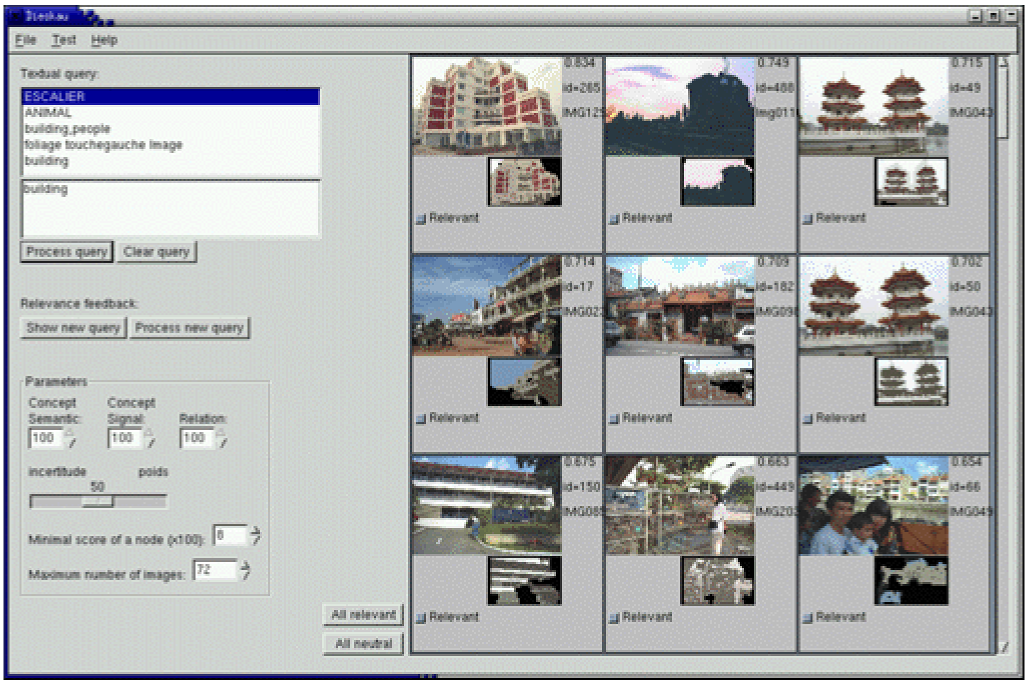


Figure 11. DIESKAU query interface.

VII.1. Histograms generation and matching

The histograms used in the experiments are based on the work of Comaniciu and Meer [3] on color image segmentation to define using [4, 5] low dimensions colors descriptions for the segmentation of regions. This approach allows the association of descriptors H (a histogram) to image regions:

$$H = \{(c_i, p_i);\ i=1,\ldots,N\}$$

Where N is the number of colors that describe the region, c_i a color and p_i is the importance of the color in the description ($\Sigma p_i=1$).
The distance D^2 between two color histograms H_1 and H_2 is defined as:

$$D^2(H_1,H_2) = \sum_{i=1}^{N_1} p_{1i}^2 + \sum_{j=1}^{N_2} p_{2j}^2 - \sum_{i=1}^{N_1}\sum_{j=1}^{N_2} 2\, a_{1i,2j}\, p_{1i}\, p_{2j}$$

where the color similarity coefficients $a_{k,l}$ between the colors c_k and c_l are defined as:

$$a_{k.l} = \begin{cases} 1 - d_{k,l} / d_{\max} & \text{if } d_{k,l} \le T_d \\ 0 & \text{if } d_{k,l} > T_d \end{cases}$$

Where $d_{k,l}$ is the Euclidian distance between two colors, and T_d the threshold to consider two colors as similar, accordingly to [5].

The similarity between an image histogram h_d concept and a query histogram concept (that contains several histograms) h_q is defined as:

$$F_h(h_d, h_q) = \sum_{i=1}^{n} (1 - \frac{D(h_d, h_i)}{\sqrt{2}})$$

As explained before, the certainty of each histogram concept is fixed to 1 and the importance value of the histogram concept is equal to the relative size of the region in the image.

VII.2. Matching Tradeoffs

To achieve fast computation of the generated queries, we apply a pruning process that keeps the concepts and relations that are important during the CLCGs building process. The matching process uses simplification of the projection operators, according to the fact that all image graphs follow certain patterns that do not have to be tested if we assume that the graphs are syntactically correct. Moreover the CLCG process may be optimized by taking into account only the concepts having an importance larger than a given threshold. These optimizations allow retrieving images in our tests in less than 5 seconds in average on a Pentium III processor 700 MHz with 128Mb of RAM, and the query generation takes between 1 and 13 seconds, depending on the pruning used. This test was achieved on purpose on a low-end configuration, knowing that on a high-end computer (like Pentium IV processor 3.4 MHz with 1 GB of RAM) the response time is divided by more than 3.

VII.3. Test Collections

The first collection, Col_1, is composed of 498 home photographs automatically indexed according to the work of [14]. The number of concept types is 105 and the number of relations is 47. On the collection Col_1, we defined 38 queries involving the labels as well as spatial relationships of labels. Because the labeling is automatically generated, the descriptions are not very accurate. An assessment made by 3 persons defined the relevant documents for each query. Queries include objects ("building"), relative positions (like "at the left of") and absolute relations (like "touch top"). This test collection allows us to evaluate the QBE using symbolic and non-symbolic concepts, in the case of simple symbolic descriptions.

The second collection, Col_2, is the one used in [21] and is composed of 653 grayscale photographs. The number of concepts types is 5945, and the number of relations is 78. The photographs were manually indexed. As we guess according to the complexity of the concept type hierarchy, the descriptions of the photographs are very precise. For the collection Col_2 we do not use any histogram since all the photographs are black and white. The evaluation made for Col_2 is based on

30 queries and on assessments made by 4 people. This test collection allows us to validate our proposal in the context of complex symbolic descriptions.
For each of the tests, each query by example was built by selecting randomly relevant images, and 10 runs were made after removing the selected images from the set of relevant images. The results are then averaged.

VII.4. Results

Precision-recall curves in Fig. 12 shows the results obtained for a query built with 1 (qbe1), 3 (qbe3) and 5 (qbe5) example images on the collection Col_1. The results with a textual query (Qtext) are also presented so as to compare the QBE process. The left part of Fig. 12 shows the results obtained using descriptions of images that include histograms. The right part of Fig. 12 compares the results obtained without and with histograms for the same query by example using 5 images.

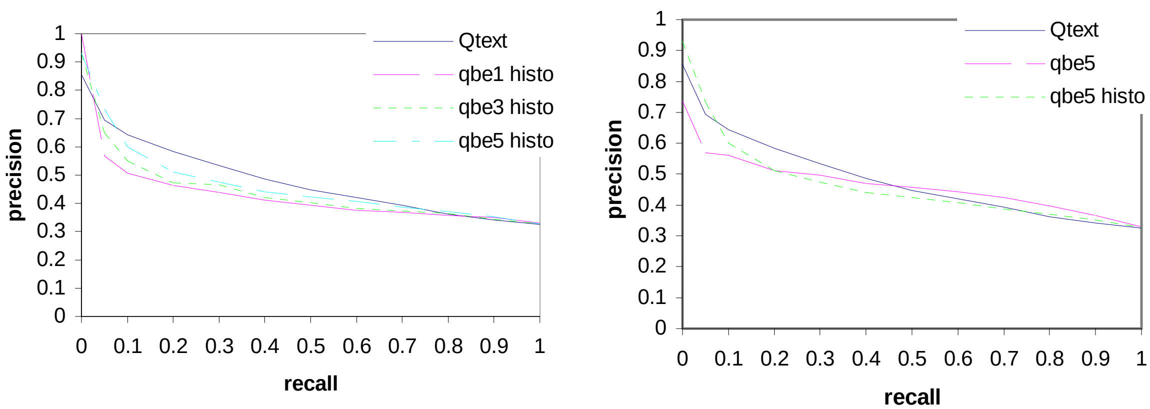


Figure 12: Results for the collections Col_1 (left), with/without histograms (right)

	Qtext	Qbe1 histo	Qbe2 histo	Qbe5 histo	Qbe5
Prec. At 5 docs	0.6211	0.5632	0.6842	0.7158	0.4947
Prec. At 10 docs	0.6026	0.4921	0.5737	0.6079	0.4526
Avg. precision	0.4616	0.4094	0.4274	0.4501	0.4423

Table 3. Precision at 5 and 10 documents, average precision on Col_1.

We see on the left part of Fig. 12 that the relevance of the generated query increases with the number of images selected. This validates the shape of *u* and *v* functions.
Considering the results obtained in table 3 for the collection Col_1, the QBE process (average precision of 0.4501 for 5 image queries) is almost as effective as the textual query process (average precision of 0.4616). Compared to the results obtained without histograms (right part of Fig. 12), the results with histograms is more accurate, this aspect is also reflected in Table 3 where the precision at 5 and 10 documents is much higher with than without histograms. This is due to the fact that the labeling errors spoil both the built query and the matching process, but the histogram integration use helps to recover the errors. The Fig 12 shows that the QBE process recall/precision curves are lower than the textual query curve between 0.1 and 0.5 recall values. However when

considering medium to low values of recall, the QBE processes with 3 and 5 input images have slightly higher precision values.

For the collection Col_2 (see results in Fig. 13 and Table 4), the QBE process significantly outperforms (at alpha=0.05) the textual query performance by a gain of average precision of 3% (paired t-test with null hypothesis H_0 that the two result sequences have the same mean values, P-value equals 0.033). For this collection, we notice the obvious interest in taking 5 images for the QBE, as the precision values for recall between 0 and 0.3 are much higher.
To summarize, the QBE based on 5 documents provides results that are better than, or very close to, the results obtained using textual queries. As one would have expected, the accuracy of indexes impacts the quality of the overall QBE results: when the indexes of images are already very reliable, the query by example process proposed is able to find out the relevant information that the user is looking for; when the indexes of images are uncertain, the quality of the results obtained is as accurate as with textual query, even if the textual input is less subject to errors.

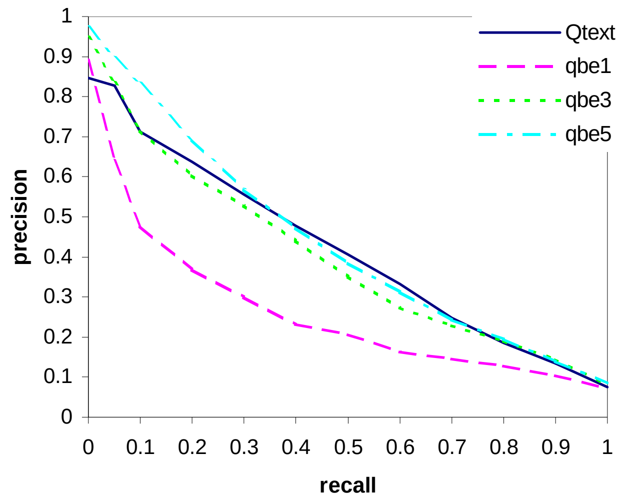


Figure 13. Recall/precision curve for collection Col_2.

	Qtext	Qbe1	Qbe3	Qbe5
Prec. At 5 docs.	0.6467	0.46	0.68	0.8133
Prec. At 10 docs	0.6	0.3633	0.57	0.65
Avg. precision	0.3961	0.2393	0.3806	0.425

Table 4. Precision at 5 and 10 documents, average precision on Col_2.

VIII. Conclusion

We have presented in this paper a way to define Query By Example processes on images described by conceptual graphs. The process is able to take into account symbolic concepts that come from a labeling process as well as histograms that come from the feature analysis of images. The query generation is based on the indexes of the images selected by a user, and considers the importance of each element (concept as well as relations) in the query, according to their frequency in the selected images index. The interest of managing both features and symbols is that we can integrate easily labeling results, metadata information as well as feature extracted data. As a consequence, our approach supports query by example as well as text-based query in an integrated way, which is from our point of view a great advantage.

The results obtained on two collections provide encouraging results. On the automatically indexed collection the QBE process performs as well as the textual queries but the gap between these results is not very large. For the manually indexed collection, the QBE process outperforms the textual input query results. These results mean that a user can choose the mode of query input that she/he wants (between QBE and text-based) without impacting a lot the system performances. From our point of view, being able to provide different interaction modes to image retrieval with the same quality of retrieval is a very important result obtained.

The QBE process proposed here does not consider the certainty of recognition of labels when building the query. It is obvious that if we put in the generated query elements that were uncertain in the selected images, then erroneous elements may be included in the query. This point is also interesting in a way to increase the speed of the query generation process. Although the approach proposed gives good results, we have to focus on speed to be able to manage thousands of images in less than 10 seconds in a way to be usable. As we said before, a pruning step increases the speed but neglects elements of the image representation, so the question on how to balance between both aspects has to be studied. For now, our system does not provide penetrable but transparent query by example interface: a penetrable relevance feedback process [13] presents the query generated and the user can modify the query before running it. Because our approach is based on symbols and on simple representations of extracted feature (histograms), such penetrable process is possible and will be developed in the future.

References

[1] S. Berretti, A. Del Bimbo and E. Vicario. Efficient Matching and Indexing of Graph Models in Content-Based Retrieval. IEEE Transactions on Pattern Analysis Machine Intelligence, 23(10), pp. 1089-1105.

[2] C. Carson, S. Belongie, H. Greenspan and J. Malik, Blobworld: Color- and Texture-Based Image Segmentation Using EM and Its Application to Image Querying and Classification, Trans. on Pattern Analysis and Machine Intelligence, Vol. 24, No. 8, August 2002, pp. 1026-1038.

[3] D. Comaniciu and P. Meer. Robust analysis of feature spaces: Color image segmentation. In Proceeedings of the IEEE Conference on Computer Vision and Pattern Recognition, Puerto Rico, pages 750-755, 1997.

[4] Y. Deng, C. Kenney, M. S. Moore and B. S. Manjunath. Peer group filtering and perceptual color image quantization. In Proc. of IEEE Intl. Symposium on Circuits and Systems (ISCAS), vol. 4, p. 21-24, 1999.

[5] Y. Deng, B. S. Manjunath, C. Kenney, M. S. Moore and H. Shin. An Efficient Color Representation for Image Retrieval. In IEEE Transaction on Image Processing, 10(1), pp. 140-147.

[6] J. Fan, Y. Gao, H. Luo, G. Xu, Automatic image annotation by using concept-sensitive salient objects for image content representation. ACM SIGIR 2004, Sheffield, UK, 2004, pp. 361-368.

[7] M. Flickner, H. Sawhney, W. Niblack, J Ashley, Q. Huang, B. Dom. M. Gorkani, J. Hafner, D. Petrovic, D. Steel amd P. Yanker. Query by image and video content: The QBIC System. IEEE Computer, 28(9): 23-32, September 1995.

[8] A. Gupta, T. E. Weymouth and R. Jain. Semantic Queries with Pictures: the VIMSYS model. In Proceedings of the Very Larage Database Systems Conference, pages 69-79, 1991.

[9] L. Hollink, A. Th. Screiber. B. J. Wielinga, and M. Worring, Classification of user image descriptions, Intl. Journal of Human-Computer Studies, 61, 2004, pp. 601-626.

[10] J. Jeon, V. Lavrenko, and R. Manmatha, Automatic image annotation and retrieval using cross-media relevance models, ACM SIGIR 2003, Toronto, Canada, pp. 119-126.

[11] C. Jörgensen, Indexing Images: Testing an Image Desription Template. In Proceedings of the 58th ASIS Annual Conference, USA, pages 209-213, 1996.

[12] C. Jörgensen. Attributes of Images in Describing Tasks. Information Processing and Management, 34(2/3): 161-174, 1998.

[13] J. Koenemann and N. J. Belkin. A case for interaction: A study of interactive information retrieval behaviour and effectiveness. In Proceeding of Computer Human Interaction Conference, pages 205-212, 1996.

[14] J.-H. Lim. Photograph retrieval and classification by visual keywords and thesaurus. In New Generation Computing, 18:147-156, 2000.

[15] J. H. Lim, Q. Tian and P. Mulhem , Home Photo Content Modeling for Personalized Event-Based Retrieval, IEEE Multimedia, Special Issue on Multimedia Content Modeling and Personalization, Vol. 10, No .4, October-December, 2003, pp. 28-37.

[16] J. Martinet, Y. Chiaramella. P. Mulhem and I. Ounis, *Photograph indexing and retrieval using star-graphs*, in Third International Workshop on Content-Based Multimedia Indexing (CBMI'03), Rennes, France, pp. 335-341, 22-24 September, 2003.

[17] J. M. Martínez. Overview of the MPEG-7 Standard (version 5.0). ISO/IEC JTC1/SC29/WG11 N4031, March 2001.

[18] M. Mechkour. EMIR2: an extended model for image representation and retrieval. In Proceeding of Database and Expert Systems Applications Conference, London, pages 395-404, 1995.

[19] C. Meilhac and C. Nastar. Relevance feedback and category search in image databases. In IEEE International Conference on Multimedia Computing and Systems, volume 1, pages 512-517, 1999.

[20] P. Mulhem and L. Nigay. Interactive Information Retrieval Systems: From User Centred Interface Design Design to Software Design. In ACM Special Interest Group in Information Retrieval Conference, pages 326-334, 1996.

[21] I. Ounis and M. Pasça. RELIEF: Combining expressiveness and rapidity into a single system. In Proceedings of the ACM Special Interest Group in Information Retrieval Conference, pages 266-274, 1998.

[22] I. Ounis. A flexible weighting scheme for multimedia documents. In Proceedings of the Database and Expert Systems Applications Conference, pages 392-405, 1999.

[23] S. Paek, C. L. Sable, V. Hatzivassiloglou, A. Jaimes, B. H. Schiffman, S.-F. Chang and K. R. McKeown, Integration of Visual and Text based Approaches for the Content Labeling

and Classification of Photographs, ACM SIGIR'99, Workshop on Multimedia Indexing and Retrieval. Berkeley, CA. August, 1999.

[24] E. Rasmussen. Indexing Images. ARIST, 32:169-196, 1997.

[25] J. J. Rocchio. Relevance Feedback in Information Retrieval. In G. Salton, editor, The SMART Retrieval System: Experiments in Automatic Document Processing, pp. 313-323. Prentice-Hall, Englewood Cliffs, NJ, 1971.

[26] G. Salton and M. J. McGill. Introduction to Modern Information Retrieval. McGraw-Hill, New York (NY), USA, 1983.

[27] A. Smeulders, M. Worring, S. Santini, A. Gupta and R. Jain. Content-Based Retrieval Image Retrieval at the end of the Early Years. IEEE Transactions on Pattern Analysis and Machine Intelligence, 22(12), pp:1349-1380.

[28] J. Smith and S.-F. Chang. Tools and techniques for color image retrieval. In IST&SPIE Proceedings, Storage and retrieval for Image and Video Databases IV, USA, volume 2670, pages 87-98, 1996.

[29] J. Smith and S.-F. Chang, Content-based image query system. In ACM Multimedia Conference, pages 2670, 1996.

[30] J. F. Sowa. Conceptual Structures: Information Processing in Mind and Machines. Addison-Wesley, Reading (MA), USA, 1984.

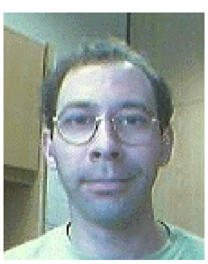
[31] J. F. Sowa. Knowledge Representation: Logical, Philosophical, and Computational Foundations. Brookes/Cole, Pacific Grove (CA), USA, 2000.

[32] D. L. Swets, Y. Pathak, and J. J. Weng, A system for combining traditional alphanumeric queries with content-based queries by example in image databases, Tech. Rep. CPS-96-03, Michigan State University, January 1996.

[33] C. P. Town and D. Sinclair. Content Based Image Retrieval using Semantic Visual Categories. AT&T Laboratories Cambridge Report 2000.14, 2000.

[34] S. Weibel, J. Kunze, C. Lagoze, M. Wolf. Dublin Core Metadata for Resource Discovery. IETF #2413. The Internet Society, September 1998.

[35] M. E. J. Wood, N. W. Campbell, and B. T. Thomas. Iterative Refinement by Relevance Feedback in Content-Based Digital Image Retrieval. In ACM Multimedia 1998, pages 13-20, UK, 1998.



Philippe Mulhem is a CNRS researcher in the Modeling and Multimedia Information Retrieval group of the CLIPS-IMAG laboratory, Grenoble, France. His research interests include formalization and experimentation of image, video, and multimedia document indexing and retrieval.Mulhem received a PhD and Habilitation to Manage Research from the Joseph Fourier University, Grenoble, France.

Emmanuel Debanne is currently a software engineer at Océ Print Logic Technologies, Créteil, France. Emmanuel Debanne worked on various softwares including web applications or print servers. In 2002, he received a diploma of "Ingénieur civil des Mines" from the "Ecole Nationale Supérieure des Mines de Nancy", France. The work reported here was achieved when he was working at the I2R in Singapore.