

Simone Santini  
Bio-Informatics Research Network  
Department of Neuroscience  
University of California, San Diego MC 0715  
9500 Gilman Drive  
La Jolla, CA 92093-0715  
Tel: (858) 822-0207  
Fax: (858) 822-0828  
email: [ssantini@ncmir.ucsd.edu](mailto:ssantini@ncmir.ucsd.edu)

Indexing Terms: Image search, multimodal search, image-text relation, relevance feedback, web search engines

# Multimodal Search in Collections of Images and Text

Simone Santini

University of California, San Diego

## Abstract

This paper presents a data model for images immersed in the world wide web and that derive their meaning from visual similarity, from the connection with the text of the pages that contains them, and from the link structure of the web.

I will model images on the web as a graph whose nodes are either text documents or images, and whose edges are links, labeled with measures of relevance of one document towards the other. The paper presents briefly the features used to characterize the text and the visual aspect of the images, and then goes on to present a data algebra suitable to navigate and query the database.

## 1 Introduction

Content Based Image Retrieval (CBIR), as the name implies, deals with the problem of accessing (presumably large) repositories of images based on their content. Now, the term “content”, per se, is quite ambiguous. For instance, an image may contain a certain number of blue pixels, but accessing an archive looking for images containing blue pixels would be, for most of us, utterly uninteresting. What the term really means to say is that images should be accessed by their *meaning* or, to be a bit more pedantic, by their referents on the mimetic plane [14].

This clarifies things a bit, but it also begins to highlight the troubles and the plight that content based image retrieval encountered in its development. For, what is exactly the meaning of a given image? Is it possible to define meaning in a way independent of the query context, as is done for traditional databases, or at least partially so, as it is done for information retrieval systems? Often one sees that the answer to these question is given, either directly or by implication, in terms of what I will call, for want of a better word, *naïve realism*: the meaning of an image—and therefore the basis for accessing that image—is given by the objects therein contained. So, if I see the picture of a red car, the meaning of the image is “there was a red car.” A less naïve position would admit that the

meaning of an image is not necessarily given by the objects that are contained in it, but by the *situation* that it depicts. In this case, the meaning of the previous image would be that “there was a red car running.” This position leaves open the possibility of drawing situational inferences. For instance, if the car is a sports car and we see a young couple inside, then a superficial knowledge of western customs might let us believe that the couple is running just for the fun of it, or that the male half of the couple is trying to impress the other half.

Roland Barthes [4] put the answer in a more precise form by stating that the noema of photography is *ça-a-été* (this-has-been). Barthes was studying a type of photographs for which his characterization was, under certain assumptions, valid but in general things are more complicated. If it is true that, as Barthes stated, the implicit message in every image is *ça-a-été*, it is also true that there is nothing in the image per se to specify the nature of the thing that really happened, since there are no syntactic differences between a documentary image and a staged scene.

Just like text, images are artifacts created to establish communication. There is no such a thing as a “natural” image: all images are artificial, at least to the extent to which the photographer acted as a filter, deciding what to include in the image and what to leave out, how to compose its subject, which relations between the subjects should be evidenced and which should be obfuscated. All these manipulations are made using a very sophisticated language which is in part iconic (registering, e.g., social preponderance through relative size, relative position, centrality, etc.), and in part symbolic (e.g. in the selection of certain clothes for the subjects).

Like all other forms of communication, images are immersed in a structure of power and solidarity between those who control the creation and transmission of the message and those who receive it. The meaning of an image is given as much by its content as by the social contract between the viewer and the producer. When this contract is violated, we conclude that the image lies. In this sense, for instance, a photograph of, say, the civil war in El Salvador doesn’t inform us just because of the things it depicts, but because of the social rôle of the documentary photographer as the registrar of a certain socially constructed truth. While, in a sense, it is true that the photography ensures that *ça-a-été*

(these people were in front of the camera, in that particular pose at the time of the shot), the true statement of the picture (a political statement, in this case), is only true because of the agreed rôle of the documentary photographer, and the trust invested in the publisher of the picture.

Talking about photographs, Barthes considered two planes of interactions with the viewer: the *studium*, and the *punctum*. The *studium* is the set of the cultural associations evoked by the photography. In the case of the civil war, this could include statements about the harsh reality of civil war, about the status of the military, or about foreign intervention. The *punctum* is the point, unique for each photograph, and not always present, that hits us with a high emotional intensity: the position of a hand, the face of a person, and so on. While the *punctum* is a function of the contents of a photograph<sup>1</sup>, the *studium* is a social and cultural construct that can't be given just by the contents of an image.

Eco[9] wrote that a sign is that which can be used to lie and, in this sense, an image is not a sign. An image is not a predicate, but something that can be predicated<sup>2</sup>. If I see a photograph of Jacques Chirac shaking hands with Georg Cantor on the first page of *Le Monde*, I am inclined to think of it as a lie. The lie, however is not in the picture itself, but in the picture taken in a particular context (the front page of the *Le Monde*). The conventions that regulate this context require that pictures on the first page of a newspaper represent “reality,” unless otherwise specified. That is, in this case, the context is stating the predicate *ça-a-été* of which the photograph is an object, and *the context* is lying, not the picture. Had the picture been in the book section of the same newspaper, used in the review of a book on “the political influence of higher mathematics,” it would have been a perfectly normal and “true” image.

Note that all these consideration don't take into account the contemporary sophisticated possibility of digital manipulation. Once this is taken into account, the question of truth is rolled back two centuries: before the invention of photography, it was taken for granted that every report was subjective and, as such, possibly non veridical. During the last century we

---

<sup>1</sup>It should be pointed out that [4] is interested mostly in the *punctum* aspect of meaning, and that the *ça-a-été* should be referred to this.

<sup>2</sup>This doesn't actually mean that an image is not a sign but, in the language of Peirce, that it is a rhematic indexical qualisign.

have lived a short interval in which we believed that there was a way—however imperfect—to record the truth about the world in an objective way. That age is now passed, and the question of interpretation can no longer rely on the message only, but must include an analysis of the messenger and of the receiver.

The necessity of external validation of content is related to the *contextual incompleteness* of pictures, the state of affairs by which pictures can't themselves be predicates, but become so with the help of a textual discourse [13]. Pictures are not predicates but entities which are predicated by some form of associated text with the help of some verbal shifter (like the famous *ceci* in Magritte's *ceci n'est pas une pipe*).

An image is therefore a predicate only in a certain textual context. I call *eidoneme* the elementary carrier of signification in the image world (much like the grapheme in the case of written communication). An important question (one might say *the* most important question for CBIR) is the nature of the textual component of an eidoneme. This question gives rise to three possible scenarios [21]:

1. The image is part of a coherent whole which includes text and a discourse that anchors this text to a meaning. The World Wide Web provides a perfect model of this situation. In this case the database operates in the territory between text and image, as a *trait d'union* between the two [26].
2. The text is implicit in the social discourse, which sets and delimits the possible interpretations of the images even before the database is designed. This is the typical case of limited domain databases, and here the image database can operate in its most traditional way, as a search engine based on automatically extracted content features.
3. The linguistic discourse is provided by the user; this is the case of strong feedback and tight loop interactive systems. The database in this case is a *writing tool* for the user, in which linguistic concepts are written and associated to the images in the repository.

This paper discusses the interaction between text and images in the first scenario above. Within the realm of a simple, direct relation between images and the accompanying text,

there are some important distinctions to be made. The simplest (and less interesting) relation is normative, and is represented by a label telling exactly what the image is supposed to mean. In this case, the label acts as a strong delimiter of the content of the image. Moreover, the label itself, taken in isolation, can't support signification without being immersed in a context. In other words: labeling images can only be done in a pre-defined domain which will immerse the image and the label in a context (case 2 above), and doesn't constitute a relation of the type in which I am interested here.

A more interesting circumstance occurs when the relation between text and images is more flexible and complex, such as in the case of images on the web embedded in the linguistic context of pages. This type of relation has been exploited in many systems, such as VisualSeek [26] and MAVIS2 [16]. These systems invoke first order relations between images and text, and images can be characterized to the extent that the text associated to them can.

This is a problem in cases in which certain images escape characterization by first order relations, either because supporting text couldn't be found, or because the text itself can't be interpreted. In this case it is possible to use *second order relations* induced by the similarity between images and by the first order relation between images and text. So, an image visually close to a cluster of images sharing a certain meaning will participate in that meaning.

## 2 A general model of Images on the Web

This section will introduce, in very general terms, the problem at hand. My present goal is not to derive a formal model of a database of images immersed in the web (I will do this in the next section), but simply to identify the qualitative structure of the problem, and to show that it generates two different types of query operations. This observation will set the stage for the following sections, in which the model will be formalized, the two types of operations will be defined first, and integrated into the same framework then.

I will consider a simplified version of the web, limited to the scope of this paper. In particular, I will assume that the web is a set of documents (pages) containing text and images (Figure 1), linked by *syntactic links* contained into the text of the page. This

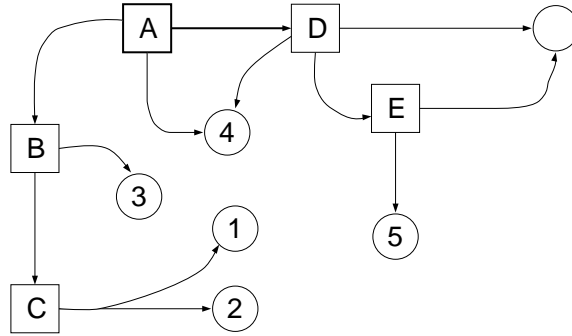


Figure 1: Documents and images on the web with the explicit connections represented by the links declared into the web pages. Squares represent web pages, and circles represent images.

assumption might seem a little drastic but, all in all, it is very hard to find a web site in which the non-text and non-image portion add significantly to the information content of the site, rather than being simple additions designed to attract viewers. Some obvious exceptions I can think of are sites in which multimedia data are stored and exchanged (e.g. music exchange sites with their vast collection of audio files) and sites in which a web page is a medium of artistic expression. In the case of media exchange sites, the analysis of the contents of video or music goes beyond the scope of this paper but, were it included, it would be relatively easy to extend the database to include such sites.

Art and graphic design sites are a different problem: in this case, a web page is not “about” some external reference, but it is, in itself, the object about which a discourse could unfold. In this case, one of the essential premises of my treatment—that is, that web pages are signs about an external mimetic plane—fails and, consequently, the considerations of this paper can’t be applied to web art or to all those circumstances in which web pages are at one time the subject and the object of discourse.

The model will assume that each link is labeled and that labels are tuples with a fixed and predefined signature.

In addition to the syntactic links, the model can determine the similarity between texts as well as between images. The similarity between texts is determined by a comparison of key terms taken from the body of the text, while the similarity between images is determined by a combination of terms from the text associated to the image and visual features extracted

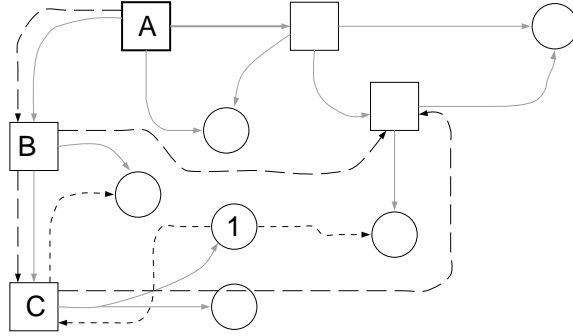


Figure 2: Semantic links created by a 2-nearest neighbors query involving documents A, B, C, and image 1.

from the image itself. These similarity measures can be used to answer  $k$ -nearest neighbors queries with respect to any document or set of documents in the database. A  $k$ -nearest neighbor query creates  $k$  *semantic links* between the query samples and the rest of the database: each query image is implicitly linked to the  $k$  images closest to it according to the similarity measure on which the query is based. In the previous example, a 2-nearest neighbors query involving images A, B, C, and image 1, could result in a situation like that of Figure 2. The links created by the query are superimposed as dashed lines to the syntactic links embedded in the documents. Coarse dashes indicate text to text links, while fine dashes indicate links involving images. The creation of links is not limited to a  $k$ -nearest neighbors query, as in this example, but can be seen as the outcome of every query on the database. Other types of queries that generate semantic links are the detection of a given keyword in a document, the detection of a given combination of keywords with a certain relevance value, and so on.

The creation of navigable links is a first example of integration between querying and navigation in a web database: queries can be used to create links that can later on be traversed. Alternatively, queries can be used to create a *navigation context* by delimiting the portion of the graph that satisfies the query: the result of a query is seen as a sub-graph of the web graph, and subsequent query and navigation operations are restricted to this sub-graph.

The creation of temporary semantic links, and the creation of a navigation context are the two modes of interaction between navigation and query that I will consider here. Note

that the abstract model that I am considering makes no a priori distinction between text documents and images: navigation and queries can include both, with the difference that images can be related to each other through commonality of associated text and by visual similarity, while only the former relation is defined for text documents.

### 3 A More Formal Model

In this section, it is my intention to define several models of the world wide web, progressively refining them in order to highlight the rôle of images in them and the relations between images and texts.

At a rather general level, the web is a graph  $W = (D, E, \phi)$ , where  $D$  is a set of nodes, or *documents*,  $E$  is the set of edges, and  $\phi$  is a function that associates to every edge  $e \in E$  an ordered pair of documents:  $\phi(e) = (d_1, d_2)$ , with  $d_1, d_2 \in D$ , that is<sup>3</sup>,  $\phi : E \rightarrow D \times D$ . In this case the link is said to go from document  $d_1$  to document  $d_2$ . A link can be either a syntactic link or a semantic link created by a query. Note that this implies that queries change the topology of the graph<sup>4</sup>.

Given a document  $d$ , its *downstream neighborhood* is the set

$$\nu(d) = \{d' \in D : \exists e \in E : \phi(e) = (d, d')\} \quad (1)$$

while its *upstream neighborhood* is the set

$$\nu^*(d) = \{d' \in D : \exists e \in E : \phi(e) = (d', d)\} \quad (2)$$

The two functions  $\nu, \nu^* : D \rightarrow 2^D$  are adjoint in the sense that

$$\begin{aligned} d &\in \nu^*(\nu(d)) \\ d &\in \nu(\nu^*(d)) \end{aligned} \quad (3)$$

---

<sup>3</sup>In many cases a graph is defined simply as  $W = (D, E)$ , where  $D$  is the set of nodes (documents, in this case), and  $E \subset D \times D$  is the set of edges, each edge being an ordered pair of documents. This definition is equivalent to that in this paper for *simple graphs*, but can't be generalized to *multigraphs*, since two edges between the same pair of documents would be undistinguishable. Since on the web there can be multiple links between the same two documents, it is more appropriate to start from the multigraph model as a basis.

<sup>4</sup>This is true, strictly speaking, only at the level of the data model. In the actual implementation, syntactic links and semantic links are created through different mechanisms and with different indexing policies. In particular, it is often possible to cache semantic links in central memory.

whenever  $\nu(d)$  and  $\nu^*(d)$ , respectively, are non empty. The functions  $\nu$  and  $\nu^*$  can be extended to a set of nodes  $N$  as  $\nu(N) = \bigcup_{d \in N} \nu(d)$  and  $\nu^*(N) = \bigcup_{d \in N} \nu^*(d)$ .

The *out-degree* of a node is the number of edges leaving that node, that is,  $o(d) = |\{e \in E : \pi_1(\phi(e)) = d\}|$ <sup>5</sup>. Similarly, the *in-degree* of a node  $d$  is the number of edges entering the node, that is  $i(d) = |\{e \in E : \pi_2(\phi(e)) = d\}|$ . The concepts of in-degree and out-degree can easily be generalized to sets of nodes.

I will admit from the outset the possibility that links be labeled, with labels drawn from a set  $\Lambda$  which can be finite or infinite. The labeling function  $l : D \times D \rightarrow \Lambda \cup \{\perp\}$  is defined as

$$l(e) = \begin{cases} \lambda \in \Lambda & \text{if } e \in E \\ \perp & \text{otherwise} \end{cases} \quad (4)$$

The web contains an appalling variety of types of documents or fragments thereof, and a more than cursory glance at any catalog of existing MIME types will convince anybody of this fact. In this context, however, I will purposely disregard most of these document types, and consider only the “simplified web” mentioned in the previous section, consisting exclusively of text documents containing images.

This refinement modifies the model in the sense that the graph is now a *colored* graph, with nodes of two types: *documents* and *images*. Consequently, edges are of different types, or colors: *document-to-document*, *document-to-image*, and *image-to-document*. I will assume that the graph contains no image-to-image edges, which seems to be largely the case for the web. The only exception to this absence is the case of thumbnails that point to a larger version of the same image. In this case, however, the thumbnail and its larger version are conceptually the same image placed in the same situation, and no semantic difference will exist between the two. These types of edges derive from well identified syntactic constructs in the html code that constitutes the web, as shown in Table 1.

Finally, I will consider the way in which images are modeled and how are they associated to the text of the pages. Typically, a fragment the page is considered and encoded suitably in order to provide a textual description of the image [23]. The text being analyzed in order

---

<sup>5</sup>Following the standard notation, for every pair  $(a, b)$ , the two projections  $\pi_1$  and  $\pi_2$  are defined as  $\pi_1(a, b) = a$  and  $\pi_2(a, b) = b$ .

Edge type	Syntactic HTML Construct
document-to-document	<code>&lt;a href="....."&gt;Link text&lt;/A&gt;</code>
document-to-image	<code>&lt;img src="....."/&gt;</code>
image-to-document	<code>&lt;a href="....."&gt;&lt;img src="....."/&gt;&lt;/A&gt;</code>

Table 1: Syntactic definition of the different types of edges

to describe an image includes the so-called *anchor text* that surrounds it, the title of the page containing the image, and its keywords (see below). The textual component of a web page, on the other hand, is indexed using the complete text of the page, regardless of the position of images. There are therefore two different circumstances in which a text-based index is extracted from a page: for the creation of a *document index*, associated to the whole page, and for the creation of an *image text index* for each image contained in the page. Each element in the index is a word, defined by a string, with an associated weight. Defining it as a data type for inclusion in the database, each element in the index is a datum of type *weighted word*, defined as:

$$\text{wword} : [\text{text} : \text{string}, w : \mathbb{R}] \quad (5)$$

Documents and images are also data types in the database. Documents are characterized by an unique identifier and a set of words describing their contents:

$$\text{doc} : [\text{id} : \mathbb{N}, \text{wds} : \{\text{wword}\}] \quad (6)$$

Images are also represented by a data type including a unique identifier, a text index, and a visual feature. At this time I am interested only in the text index portion of the description, so I will leave the visual features unspecified by making images a parametric data type. Moreover, images are a sub-type of document, since they also have an identifier and a set of words associated with them:

$$\text{img}(\mu) = \text{doc} + [\text{vis} : \mu] \quad (7)$$

Labels also can contain different types of information, depending on the application. The model allows a label to be an arbitrary database tuples, but requires that at least two fields be present: a *relevance measure* between the documents that they join, which is a real

value between 0 and 1, and a *status indicator*, which can assume the two values `syntactic` or `semantic`, depending on the nature of the link. So, as a data type, the set of labels  $\Lambda$  has, as parameters, whatever additional information is added, and is defined as

$$\Lambda(\mu) = [r : \mathbb{R}, \text{stat} : \{\text{syntactic}, \text{semantic}\}] + \mu \quad (8)$$

Relevance can be assigned either automatically at insertion time, as I will consider later, or updated explicitly using specific database operations.

The following definition therefore applies:

**Definition 3.1.** *A web image database is a graph  $W(\alpha, \mu) = (D(\alpha), E, \phi, l)$ , where:*

- $D(\alpha)$  is a set of documents of type  $D(\alpha) : \text{set}\{d : \text{doc|img}(\alpha)\}$
- $E$  is the set of edges of the graph.
- $\phi : E \rightarrow D \times D \cup \{\perp\}$  is the edge connection function
- $l : E \rightarrow \Lambda(\mu) \cup \{\perp\}$  is the labeling function, where  $\Lambda(\mu)$  is the type of the labels.

The data type of such a graph will be indicated as  $\Gamma(\alpha, \mu)$ .

This is the basic model on which the database operations are defined. These operations allow to navigate the model (traversing its links) and query the contents of the nodes. For the latter type of operation, the representation of images and text play an essential role, and the next sections will describe the text search mechanism, the visual search mechanism, and the interaction between the two.

## 4 Text Search Engine

Both images and documents are characterized (partly characterized, in the case of images) by a “doc” datum that is, essentially, by a set of weighted words. During the insertion of a page in the database, a suitable module analyzes the text of the page and the images therein contained in order to derive a description for them. Describing the content of an image using a series of words is in itself a very challenging task due to the great latitude with which users query image repositories [3, 10, 15].

The idea of extracting text from the web site that contains an image to obtain information about the content of that image is not new, and can be traced to Smith and Chang's *WebSeek* system [26]. *WebSeek* uses a *Web crawler* to find web pages containing images. Once the images are found, pieces of text like the name of the image, the name of the directory where the image is, and the title of the page that contained it are analyzed and, if proved meaningful, used as indices for the image. *Webseek* doesn't attempt any analysis of the image content, relying solely on the text associated to the image, while the representation that I am considering contains a textual part and a visual part, the latter being considered in the next section. The textual component of this model analyzes the whole text of the page containing a given image.

English text on the web differs from other forms of English text in several aspects: lexically, the distribution of words in web text is not the same as in standard English [2], and it privileges conceptual words over functional delimiters; structurally, in a web page links constitute natural points of aggregation around which the most significant parts of the text can be found.

Considering the case of document-to-image links, it has been observed that relevant information about an image embedded in a page can be found in the text surrounding the *IMG* link (including the text of the *ALT* tag for the image, if present). Such text is called the *anchor text* for that image, and its length is estimated to be 50 characters before and after the link text[6].

The images and the pages in which they are contained are gathered by a web traversal program ("web crawler") which, for each image, collects three sets of terms from three separate areas of the page: the anchor text of the *<IMG>* tag, the collection of keywords contained in the *<META>* tag of the page (if present), and the title of the page. These areas have different expected relevance for the characterization of the contents of the images contained in the page.

Following acquisition, the text is processed using standard information retrieval techniques; in particular, common English words, like "the," "from," and so on (*stop words*), are removed, and a stemming algorithm [12] is applied to remove suffixes and other word modifiers. The terms resulting from stemming are assigned a weight based on their relevance as

indices of a given image.

The three different areas of the page from which the index text is derived are assigned a priori “thematic” weights. In general, the anchor text is the most relevant for indexing, followed by the keywords and the title. I used the following three thematic weights:  $\nu_a = 3/5$  for the anchor text,  $\nu_k = \nu_t = 1/5$  for the keywords and the title. Each term is also assigned a weight measuring its potential indexing quality, determined using probabilistic weighting [12].

Let  $T_\mu$  be a term,  $df_\mu$  the *document frequency* of  $T_\mu$  (that is, the number of documents in which  $T_\mu$  appears as a term), and  $tf_{\mu k}$  the *term frequency* of  $T_\mu$  for image  $I_k$  (that is, the number of times  $T_\mu$  appears as a term in the index of image  $I_k$ .) The *basic term weight* for  $T_\mu$  in image  $I_k$ ,  $\beta_k(T_\mu)$  is then given by

$$\beta_k(T_\mu) = tf_{\mu k} \log \frac{N_I - df_\mu}{df_\mu} \quad (9)$$

where  $N_I$  is the number of images in the database. The formula considers a weight highly discriminative if it appears in relatively few documents in the collection (low value of  $df_\mu$ ), and many times in the image in question (high value of  $tf_{\mu k}$ ) [20].

The weight of a term is then multiplied by the thematic weight corresponding to the section in which the term was found. If, in the index of image  $I_k$ , the term  $T_\mu$  appears  $t_{\mu k}^a$  times in the anchor text,  $t_{\mu k}^k$  times among the keywords, and  $t_{\mu k}^t$  times in the title, then the associated weight is

$$\alpha_{\mu,k} = \alpha(T_\mu, I_k) = (t_{\mu k}^a \nu_a + t_{\mu k}^k \nu_k + t_{\mu k}^t \nu_t) \log \frac{N - df_\mu}{df_\mu}. \quad (10)$$

The weights of the various terms form a vector in a vector space in which every term in the corpus of document is an axis. The distance between such vectors represents the lexical dissimilarity between the corresponding images. Note that, although the vector space has hundreds of dimensions, the distances can be computed efficiently without any need to represent it explicitly.

Text documents are indexed in a similar weight. In this case, the three areas from which the index is extracted are the META tag keywords (weight 3/5), the title (weight 1/5) and the full text of the page (weight 1/5). The weights respond to the intuitive notion that

the META keywords, being selected for the purpose of indexation, are more informative than the general text. The equality between the weight of the text and that of the title is also posited a priori and then verified empirically.

## 5 Visual Representation

Important information about the content of images can be derived by dividing an image in regions homogeneous with respect to a certain criterion (e.g. homogeneous with respect to texture, color, or other point features of the image). This is the well known *segmentation* problem, for the satisfaction of which many algorithms have been proposed over a number of years. In this paper, I use the *edge flow* segmentation algorithm, developed by Ma and Manjunath, and available on B.S. Manjunath's web page [17, 18] on grounds of its robustness and its adaptability to work with many different point features.

The segmentation algorithm divides the image in a number of regions, located in certain spatial relations. At this point, the representation of the image requires the representation of the individual regions and the representation of the spatial relations between pairs of neighboring regions. Each region is described by a suitable feature structure, typically including a histogram of its color (or a suitable statistical characterization thereof), a texture description, a Fourier series description of its shape, and so on. I will not consider these descriptions in detail, but I will assume that the space  $X$  of region features is a vector space. Given a region  $r$  of an image, the *region descriptor*  $x_r$  is used to describe it.

The representation of an image is then a labeled directed graph, in which nodes represent regions and edges are labeled with a representation of the relative positions of the regions they join. That is:

**Definition 5.1.** *A visual feature of an image  $I$  is a directed graph  $(R, E, \phi, \rho, \lambda)$  where:*

1.  $R$  is the set of regions
2.  $\rho : R \rightarrow X$  is a function that assigns to each region  $r \in R$ ; its descriptor  $\rho(r) : X$ ;
3.  $\phi : E \rightarrow R \times R \cup \{\perp\}$  is the edge connection function;

4.  $\lambda : E \rightarrow \mathbb{C}$  assign a complex number to every edge (representing the spatial relation between the centroids of the regions joined by the edges),  $\lambda(\perp) = 0$ ;
5. for every edge  $e$  there is an adjoint edge  $e^*$  such that, if  $\phi(e) = (r_1, r_2)$ , then  $\phi(e^*) = (r_2, r_1)$ ;
6. if  $\lambda(e) = s \exp(i\theta)$ , then  $\lambda(e^*) = s \exp(-i(\theta + \pi))$ .

If  $\lambda(e) = s \exp(i\theta)$ , then:

- $s \in [0, 1]$ ;
- $s = f(d_{a,b})$ , where  $d$  is the distance between the centroids of the regions joined by  $e$  (i.e. between  $\pi_1(\phi(e))$  and  $\pi_2(\phi(e))$ ); and  $f(x) : \mathbb{R}^+ \rightarrow [0, 1]$  is a monotonically decreasing function with  $f(0) = 1$ ;
- $\theta$  is the angle between the horizontal and the line drawn from the centroid of  $\pi_1(\phi(e))$  to the centroid of  $\pi_2(\phi(e))$ ;

Giving an arbitrary order to the regions, the image can be represented by the region vector

$$\mathbf{R} = [x_{r_1}, x_{r_2}, \dots, x_{r_n}]^T = [x_1, x_2, \dots, x_n]^T \quad (11)$$

(where the second notation will be used when no ambiguity arises), and the connection matrix

$$\mathbf{C} = \begin{bmatrix} s_{11}e^{i\theta_{11}} & s_{12}e^{i\theta_{12}} & \dots & s_{1n}e^{i\theta_{1n}} \\ s_{21}e^{i\theta_{21}} & s_{22}e^{i\theta_{22}} & \dots & s_{2n}e^{i\theta_{2n}} \\ \vdots & & \ddots & \vdots \\ s_{n1}e^{i\theta_{n1}} & s_{n2}e^{i\theta_{n2}} & \dots & s_{nn}e^{i\theta_{nn}} \end{bmatrix} \quad (12)$$

Note that the matrix is anti-symmetric, that is,  $\mathbf{C}^T = -\mathbf{C}$ .

For instance, the image of Figure 3.a is represented by the graph of Figure 3.b which, choosing  $f(x) = 1/(x^2 + 1)$  and assuming the distance between node 1 and node 2 as the measure unit, is represented by the matrix

$$\begin{bmatrix} 1 & \frac{1}{2}e^{-\frac{\pi}{2}} & \frac{1}{2}e^{-\frac{\pi}{6}} & \frac{1}{2}e^{-\frac{5}{6}\pi} & 0 \\ \frac{1}{2}e^{\frac{\pi}{2}} & 1 & \frac{1}{2}e^{\frac{\pi}{6}} & \frac{1}{2}e^{\frac{5}{6}\pi} & \frac{4}{13}e^{-\frac{\pi}{2}} \\ \frac{1}{2}e^{-\frac{6}{6}\pi} & \frac{1}{2}e^{-\frac{5}{6}\pi} & 1 & 0 & 0 \\ \frac{1}{2}e^{\frac{\pi}{6}} & \frac{1}{2}e^{-\frac{\pi}{6}} & 0 & 1 & 0 \\ 0 & \frac{4}{13}e^{\frac{\pi}{2}} & 0 & 0 & 1 \end{bmatrix} \quad (13)$$

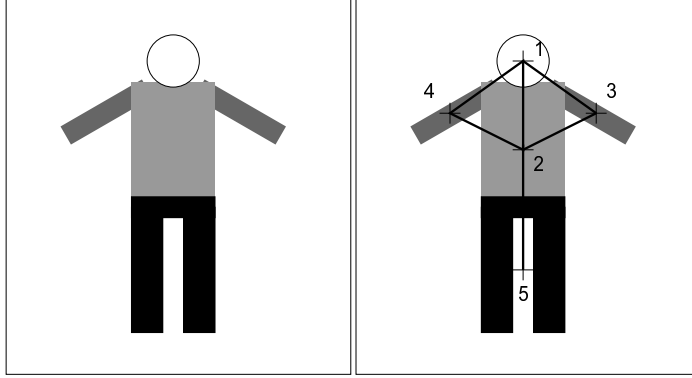


Figure 3: Graph representation of adjacent regions

In addition to this, each region representation  $x_r$  is collected into the matrix  $\mathbf{R}$ .

As usual in content based retrieval, the basic operation performed on these image representations is the determination of the similarity between two images, given a certain similarity function (which will, in many cases, depends on the query). The graph representation above introduces the complication that the result of the comparison of two images depends on the ordering of the nodes of the graphs. Checking all the possible ordering requires a time  $O(2^n)$ , where  $n$  is the number of nodes of the graph that is, the number of regions in the images. Since this number can be fairly high, it is necessary to look for alternative representations.

One such representation, invariant to the ordering of the nodes of the graph, is the *spectral representation* [7]. The spectral representation is usually applied to the adjacency matrix of a graph but, in this case, I will apply it to the matrix of the relative position.

In particular, given the matrix  $\mathbf{C} \in \mathbb{C}^{n \times n}$ , its singular value decomposition

$$\mathbf{C} = \mathbf{U}\mathbf{S}\mathbf{V}^* \quad (14)$$

is computed, where  $\mathbf{S} = \text{diag}\{\lambda_1, \dots, \lambda_n\}$  is the matrix containing the eigenvalues of  $\mathbf{C}$ . The eigenvalues of  $\mathbf{C}$  constitute an ordering-independent representation of the relative position graph.

This representation still depends on the number of nodes in the graph that is, on the number of regions in the image. In order to eliminate this dependency, I resort to the very

common technique of considering only the first  $k$  eigenvalues of  $\mathbf{C}$ . Let

$$S_{|k} = \begin{cases} [\lambda_1, \dots, \lambda_k]^T & \text{if } k \leq n \\ \left[ \lambda_1, \dots, \lambda_n, \overbrace{0, \dots, 0}^{k-n} \right]^T & \text{if } k > n \end{cases} \quad (15)$$

$$V_{|k} = \begin{cases} \begin{bmatrix} v_{11} & v_{12} & \dots & v_{1k} \\ v_{21} & v_{22} & \dots & v_{2k} \\ \vdots & & & \vdots \\ v_{n1} & v_{n2} & \dots & v_{nk} \end{bmatrix} & \text{if } k \leq n \\ \left[ \begin{array}{c|c} V & 0 \end{array} \right] & \text{if } k > n \end{cases} \quad (16)$$

and  $U_{|k}$  defined similarly.

The region descriptions are collected in a matrix  $\mathbf{R} \in \mathbb{R}^{n \times q}$ , in which each row contains a vector of dimension  $q$  with the feature representation of a region. This representation also depends on the ordering of the region and, therefore, it is unsuitable for image comparison for the same reason for which the adjacency matrix was. Now, however, we have available a transformation from the original, order-dependent, regions space to the spectral, order-independent, space, represented by the matrix  $V_{|k}^*$ . This transformation can be applied to the region matrix  $\mathbf{R}$ , obtaining the ordering independent region description

$$R_{|k} = V_{|k}^* \mathbf{R} \quad (17)$$

In the case of the image of Figure 3, for instance, the 3-representation of the graph is

$$S_{|3} = [2.04, 1.22, 1.13] \quad (18)$$

An image is therefore represented by the vector  $S_{|k}$  and by the matrix  $R_{|k}$ . A distance function between two images can be derived from any distance function between matrices. For instance, the weighted Euclidean distance between two images is given by

$$d(I, J; \alpha) = \alpha \|S_{|k}^{(I)} - S_{|k}^{(J)}\| + (1 - \alpha) \|R_{|k}^{(I)} - R_{|k}^{(J)}\| \quad (19)$$

## 6 Semantic Modification

Every image in the database contains two representations: a textual representation based on the terms extracted from the pages that link to the image, and a visual representation based on the image features introduced in the previous section. The visual characterization

of an image is given, as discussed in the previous section, by the  $k$  highest eigenvalues of the spectral representation of its structure,  $S_{|k}$ , and by the reduced, order-invariant region matrix  $R_{|k}$ . The distance in the space of the pairs  $(S_{|k}, R_{|k})$  is the *visual distance*  $d_v$ .

Let  $\{I_k, k = 1, \dots, N\}$  be the set of images, and  $\{T_\mu, \mu = 1, \dots, M\}$  be the set of terms in the database. Also, let the weight of the  $\mu^{\text{th}}$  term for the  $k^{\text{th}}$  image be  $\alpha_{\mu,k}$ , computed as in (10). An image can then be represented in the ‘‘term space’’ as

$$I_k = \sum_{\mu=1}^M \alpha_{\mu,k} T_\mu. \quad (20)$$

Similarly, a term can be represented in the image space as

$$T_\mu = \sum_{k=1}^N \alpha_{\mu,k} I_k. \quad (21)$$

In this model, the similarity between terms,  $\langle T_\mu, T_\lambda \rangle$ , is given by

$$s_l(T_\mu, T_\lambda) = \langle T_\mu, T_\lambda \rangle = \sum_{k=1}^N \alpha_{\mu,k} \alpha_{\lambda,k}, \quad (22)$$

and the lexical similarity between images,  $\langle I_k, I_h \rangle$  is given by

$$s_l(I_k, I_h) = \langle I_k, I_h \rangle = \sum_{\mu=1}^M \alpha_{\mu,k} \alpha_{\mu,h}. \quad (23)$$

Note that, if  $A = \{\alpha_{\mu,k}\}$  is the matrix of the terms weights, then the similarity between  $I_k$  and  $I_h$  is the element  $(k, h)$  of the  $N \times N$  matrix  $AA'$ , while the similarity between terms  $T_\mu$  and  $T_\lambda$  is the  $(\mu, \lambda)$  element of the  $M \times M$  matrix  $A'A$ .

The visual distance between images can be transformed in a visual similarity as  $s_v(I_k, I_h; \theta) = g(d_v(I_k, I_h; \theta))$ , where  $g$  is a positive function with  $g' < 0$  and  $g'' \geq 0$  [22]. The visual and lexical similarities can then be combined using a disjunction operator  $\vee$ . (This is a geometrical translation of the fact that two images are similar if they are visually similar *or* if they are associated to similar terms.) The result is the *total image similarity*:

$$s_t(I_h, I_k; \theta, A) = s_v(I_k, I_h; \theta) \vee (AA')_{hk}. \quad (24)$$

The operator  $\vee$  can be drawn to a class of disjunction operators known as s-norms [11]. For example, the *Hamacher sum* is defined as

$$x \vee y = \frac{x + y - 2xy}{1 - xy} \quad (25)$$

The total similarity between images, the lexical similarity between terms of two documents, and the similarity between the textual description of an image and that of a document form the basis of the similarity queries in the database. These similarities are not a datum, but depend on the context in which they are determined. As stated in the introduction, the relation between signifier and signified is not fixed and determined by the syntactic characteristics of the signifier: it is, in its most general incarnation, a cultural construct and, more immediately, a function of the context of the search at a given moment.

Context is determined by a complex set of relations, from which I single out three:

1. A *normative context*, independent of the query, can be determined by the specific application field of the database, its cultural norms, conventions, and habits.
2. The context can be inferred by the state in which the previous interactions have left the system. If previous interactions have restricted the query to a sub-graph, this reduction may have altered the relation between terms in a significant way. It might be the case, for instance, that unusual combinations of words have appeared, either in the same document or in linked documents: if, say, the word “tree” is consistently associated with the words “family” or “heraldic,” its meaning will be narrowed and clarified by this association.
3. The context can be given by the user himself through the use of context-generating interfaces, like that introduced in [24]. In these interfaces, the user modifies the results of a query by moving things around and placing them at a distance from one another reflective of their perceived mutual similarity.

The result of all these context-generating operations is the alteration of the desired cognitive similarity between images and terms. Let us consider, for the moment, the case in which the context defines a new similarity between images. This can be modeled as an  $N \times N$  matrix  $\Psi$  such that the element  $\psi_{ij}$  is the desired similarity between the images  $I_i$  and  $I_j$ . The matrix  $\Psi$  comes from user interaction: in a typical case, the elements  $\psi_{ij}$  are obtained by looking at the distance at which the user has placed icons representing the images in a configuration feedback interface [25]. The visual similarity measure  $s_v$  will be

changed using suitable optimization techniques[25] so as to minimize the error

$$\min_{\theta} \sum_{hk} [s_v(I_k, I_h; \theta) - \psi_{kh}]^2. \quad (26)$$

In addition, through the change in the similarity between images, the configuration  $\Psi$  will also change the similarity between terms: the weight matrix  $A$  will have to change in such a way that  $AA' = \Psi$ . Once this is done, one can determine the new term similarity matrix  $A'A$  and, from this, the new similarity between terms.

This results in a rather intractable quadratic problem, so an approximate solution is necessary. Instead of solving the quadratic problem, one can try to “move” the matrix a step in the right direction, that is, to find a matrix  $\tilde{A}$  such that

$$\tilde{A}\tilde{A}' = (1 - \gamma)AA' + \gamma\Psi \quad (27)$$

for some small constant  $\gamma$ . Since  $\gamma$  is small, the matrix  $\tilde{A}$  will not be too different from  $A$ . In particular, one can write  $\tilde{A} = A + E$ , where  $E = \{\epsilon_{ij}\}$  is composed of small elements. The previous equation can be expanded as

$$\tilde{A}\tilde{A}' - AA' = \gamma(\Psi - AA') \quad (28)$$

and

$$\begin{aligned} \tilde{A}\tilde{A}' &= (A + E)(A + E)' = AA' + AE' + E'A + EE' \\ &\approx AA' + AE' + E'A \end{aligned} \quad (29)$$

where the last approximation derives from the assumption that  $E$  is composed of small elements whose squares are negligible. The equation therefore becomes

$$AE' + E'A = \gamma(\Psi - AA'). \quad (30)$$

The right hand side is typically symmetric, since  $\Psi$  is, and therefore the left hand side is also symmetric, from which it follows that  $E'A' = AE'$ , and the equation becomes

$$AE' = \gamma(\Psi - AA'). \quad (31)$$

This is a system of  $N \times M$  linear equations in the unknown matrix  $E$ . The solution of this system will give a matrix  $E$  that represents the first “step” of the weight matrix  $A$  in the

direction of the similarities  $\Psi$ . One can then define

$$A \leftarrow A + E \tag{32}$$

and repeat the whole process. At the end, the result will be a matrix  $A$  that will allow the calculation of the new term-to-term similarities as influenced by the similarity between the images associated to those terms.

Note that the similarity matrix  $\Psi$  comes either from the result of a previous query, some normative statements, or the user interface; therefore in general there will be only a handful of non-zero entries. Instead of solving the whole system (which is intractably large), it will then be sufficient to consider a reduced matrix  $A$  consisting of the images whose similarity has been re-defined in the matrix  $\Psi$  and the terms associated to those images or, if this still result in a very large system, only of the most influential terms for those images can be considered (i.e. the terms with the highest weights).

If the context prescribes a new similarity between terms, rather than between images, the system will receive a matrix  $\Phi$  with the new inter-term similarity and will solve the system  $AA' = \Phi$  using a similar technique.

## 7 Database Query and Navigation

The model described in the previous sections can be synthesized in the following terms. The web is a graph  $\Gamma(\alpha, \mu)$ , where  $\alpha$  is the data type of the visual features of an image and  $\mu$  is the data type of the labels on the edges of the graph. The data type  $\alpha$  is, in the present example, that of the region graphs described in section 5, while the type  $\mu$  of the labels is unspecified, except for the prescription that it contain a measure of the similarity between the documents it connects.

This similarity is, in part, determined by the current status of the query. Images and pages have associated text indices whose semantic similarity is updated as described in section 6.

A database support for such a model should therefore be able to support two types of operations: on one hand, the database should support similarity queries and provide functions to manipulate similarity measures; on the other hand, it should provide operations

for query and navigation on an extended graph. The database model presented in the following section is an attempt to satisfy these requirements.

## 7.1 Multimedia Database Model

An image database is formed by one or more relations, or *tables*, of the form  $T(h : \mathbb{N}, x_1 : X_1, \dots, x_n : X_n)$ , where  $h$  is a unique image handle,  $X_1, \dots, X_n$  are feature data types, and  $x_i$  is the name of the  $i$ th field or *column* of the table. For the sake of simplicity, we will assume that every table contains only features. The *signature* of the table is the sequence of data types  $(\mathbb{N}, X_1, \dots, X_n)$ , and its *schema* is the sequence of names  $(h, x_1, \dots, x_n)$  with their associated data types. The elements of a schema are called the *explicit fields*, or simply the fields, of the table.

The  $k^{\text{th}}$  row of the table is indicated as  $T[k]$  and contains the handle and all the features relative to one of the images stored in the database.  $T[k].h$  is the handle of the image, and  $T[k].x_i$  is the value of its  $i^{\text{th}}$  feature descriptor. In addition to the explicit fields, each row of the table has a *score field*  $\varsigma$  (of type  $\mathbb{R}$ ) such that  $T[k].\varsigma$  is the distance between the image in the  $k^{\text{th}}$  row and the current query. The value of the field  $\varsigma$  is assigned by the scoring operator  $\Sigma$  introduced later on. If  $h_0$  is a handle, the notation  $T[h_0]$  will be used to indicate the (unique) row  $k$  for which  $T[k].h = h_0$ .

Many image queries are based on distance measures in feature spaces. Given a feature type  $X$ , let  $\mathfrak{D}(X)$  be the set of distance functions defined on  $X$ . All distance functions considered in this paper take values in the interval  $[0, 1]$  and are *curried* [27], that is, they are of type  $d : X \rightarrow X \rightarrow [0, 1]$ . Given an element  $x : X$ , and  $d \in \mathfrak{D}(X)$ , the function  $d(x) : X \rightarrow [0, 1]$  assigns to every element of  $X$  its distance from  $x$ . Such a function is called a *scoring function*, and the set of all scoring functions for a feature type  $X$  is indicated as  $\mathfrak{S}(X)$ . Each table  $T$  has associated a distance, indicated as  $T.d$ , such that, if the signature of  $T$  is  $(\mathbb{N}, X_1, \dots, X_n)$ , then  $T.d \in \mathfrak{D}(X_1 \times \dots \times X_n)$ . Each row of the table has associated a scoring function

$$T[k].d = T.d(T[k].x_1, \dots, T[k].x_n) \in \mathfrak{S}(X_1 \times \dots \times X_n) \quad (33)$$

that measures scores with respect to the image described by the row.

Moreover, a library of distance combination operators is defined. These are based on scoring combination operators [11, 8] of type  $\diamond : [0, 1] \times [0, 1] \rightarrow [0, 1]$ . Each one of these operators induces a distance combination as follows. Let  $d_1 \in \mathfrak{D}(X)$  and  $d_2 \in \mathfrak{D}(Y)$ , then the distance operator  $\diamond : \mathfrak{D}(X) \times \mathfrak{D}(Y) \rightarrow \mathfrak{D}(X \times Y)$  is the operator that makes the following diagram commute:

$$\begin{array}{ccc}
 \mathfrak{D}(X) \times \mathfrak{D}(Y) & \xrightarrow{\diamond} & \mathfrak{D}(X, Y) \\
 \text{eval} \times \text{eval} \downarrow & & \downarrow \text{eval} \\
 [0, 1] \times [0, 1] & \xrightarrow{\diamond} & [0, 1]
 \end{array} \tag{34}$$

That is, for  $x : X$  and  $y : Y$ , it is

$$(d_1 \diamond d_2)(x, y) = d_1(x) \diamond d_2(y) \tag{35}$$

The combination operators will be assumed to be symmetric and Lipschitz, that is

$$\begin{aligned}
 x \diamond y &= y \diamond x \\
 |x \diamond y - x \diamond z| &\leq L|y - z|
 \end{aligned} \tag{36}$$

for some constant  $L > 0$ .

**Operators.** Given a scoring function  $s$ , and a table  $T(\mathbb{N}, X_1, X_2, \dots, X_n)$  the *scoring operator*  $\Sigma_T(s)$  assigns a score to all the rows of  $T$  using the scoring function  $s$ . That is,  $\Sigma_T(s)$  is a table with the same signature as  $T$  and

$$(\Sigma_T(s))[k].s = s(T[k].x_1, \dots, T[k].x_n) \tag{37}$$

Given a table  $T(\mathbb{N}, X_1, X_2, \dots, X_n)$ , the  $k$  lowest distances operator  $\sigma_k^\#$  returns a table with the  $k$  rows of  $T$  with the lowest scores. The operators  $\sigma_k^\#$  and  $\Sigma_T$  are generally used together: the operator  $\sigma_k^\# \circ \Sigma_T$  is called the  $k$ -nearest neighbors operator for the scoring function  $s$ .

The operator  $\sigma_\rho^<$  returns all the rows of a table  $T$  with a score less than  $\rho$ . The operator  $\sigma_\rho^< \circ \Sigma_T$  is called the range query operator for the scoring function  $s$ .

The operator  $\sigma_P$  is the usual predicate selection operator on a table  $T$ . In the databases that we consider here,  $P$  has either the form  $h = h_0$ , where  $h_0 \in \mathbb{N}$  is a handle, or  $h \in H$ ,

where  $H$  is a set of handles. Note that the notation  $T[h_0]$  introduced above is a shorthand for  $\sigma_{h=h_0}(T)$  which, because of the uniqueness of the handles, always returns a single row.

Finally, the  $\diamond$ -join operator  $\bowtie_{\diamond}$  joins two tables  $T$  and  $Q$  on their handle field to form a new table  $W = T \bowtie_{\diamond} Q$ . If  $T = T(h : \mathbb{N}, x_1 : X_1, \dots, x_n : X_n)$  and  $Q = Q(h : \mathbb{N}, y_1 : Y_1, \dots, y_n : Y_n)$ , then

$$W = W(h : \mathbb{N}, x_1 : X_1, \dots, x_n : X_n, y_1 : Y_1, \dots, y_n : Y_n) \quad (38)$$

The row  $q$  such that  $W[q].h = h_0$  is obtained by collecting the features of the rows  $T[i]$  and  $Q[j]$  such that  $T[i].h = Q[j].h = h_0$ . The table  $W$  has a distance function

$$W.d = T.d \diamond Q.d \quad (39)$$

and, if the  $q^{\text{th}}$  row of  $W$  was obtained by joining the  $i^{\text{th}}$  row of  $T$  with the  $j^{\text{th}}$  row of  $Q$ , it has score

$$W[q].s = T[i].s \diamond Q[j].s \quad (40)$$

and scoring function

$$W[q].d = T[i].d \diamond Q[j].d \quad (41)$$

Note that the operators  $\diamond$  used in the two previous equations have different signatures: The operator in (40) is a score combination operator, while the operator in (41) is the corresponding scoring functions combination operator.

## 7.2 Graph Navigation

The second component of the data algebra consists of graph operations. All operations on the web graph are expression in the graph algebra, and are composed of *terms*. A term is either

- a variable, constant, or function symbol. The type of a function taking arguments of type  $T_1$  and producing results of type  $T_2$  will be represented as  $T_1 \rightarrow T_2$ ;
- a lambda abstraction

$$\lambda(x_1 : T_1, \dots, x_n : T_n).t : T \quad (42)$$

where  $x_1, \dots, x_n$  are variables,  $t$  is a term, and  $T, T_1, \dots, T_n$  are data types;

- an application

$$t_0(t_1 : T_1, \dots, t_n : T_n) : T \quad (43)$$

where  $t_1, \dots, t_n$  are terms and  $t_0$  is a function type.

I will use the standard convention of representing operators using the infix notation, so the sum function  $+(a, b)$  will be represented equivalently as  $a + b$ . Also, as a shortcut, given a set of nodes  $N$  from a graph, the expression  $GC(N)$  (GC for “Graph Completion”) will indicate the graph obtained taking the nodes in  $N$  and all the edges joining nodes in  $N$ , with their associated labels.

Given two documents  $d_1$  and  $d_2$ , a *path* between them is a list of edges  $e = [e_1, \dots, e_q]$  such that  $\pi_1(\phi(e_1)) = d_1$ ,  $\pi_2(\phi(e_q)) = d_2$ , and, for  $i = 2, \dots, q$ ,  $\pi_1(\phi(e_i)) = \pi_2(\phi(e_{i-1}))$ .

The *path similarity* along a path  $p$  from document  $d_1$  to document  $d_2$ , given the score composition operator  $\diamond$  is given by

$$\Pi_\diamond(p) = (\dots((l(p_1).r \diamond l(p_2).r) \diamond l(p_3).r) \dots \diamond l(p_q).r) \quad (44)$$

Let  $P(d_1, d_2)$  be the set of paths from  $d_1$  to  $d_2$ . The relevance similarity between  $d_1$  and  $d_2$  is the minimum over this set of the path similarities between  $d_1$  and  $d_2$ :

$$\Pi(\diamond)(d_1, d_2) = \min_{p \in P(d_1, d_2)} \Pi_\diamond(p) \quad (45)$$

The relevance transitive closure of a document, with relevance  $r$  is the graph

$$T(\diamond)(d, r) = GC(\{d' : \Pi(\diamond)(d, d') \geq r\}) \quad (46)$$

The downstream and upstream neighborhood functions (1) and (2) can be generalized to the  $k$ -downstream and  $k$ -downstream neighborhoods as

$$\nu(d, k) = \begin{cases} \nu(d) & \text{if } k = 0 \\ \nu(\nu(d, k - 1)) & \text{otherwise} \end{cases} \quad (47)$$

and similarly for  $\nu^*(d, k)$ .

### 7.2.1 Graph Operators

This section introduces the operators that the database defines in order to work on the graphs.

**Graph Creation** The following operators create and modify the web graph.

**graph** :  $\Gamma(\mathbf{doc}, \Lambda)$ . The function **graph**[*doc*,  $\Lambda$ ] creates an empty graph with nodes of type “doc” and edge labels of type  $\Lambda$ .

**append** :  $\Gamma(\mathbf{doc}, \Lambda) \times \Gamma(\mathbf{doc}, \Lambda) \times \mathbf{doc} \times \mathbf{doc} \times \Lambda \rightarrow \Gamma(\mathbf{doc}, \Lambda)$ . **append**( $G, H, n_1, n_2, \lambda$ ) with  $n_1 \in \mathbf{nodes}(G)$   $n_2 \in \mathbf{nodes}(H)$  builds a graph whose node set is the union of the node sets of  $G$  and  $H$ , and whose node set is the union of the node sets of  $G$  and  $H$  with an additional edge from the node  $n_1$  of  $G$  to the node  $n_2$  of  $H$ . This edge has label  $\lambda$ .

**insert** :  $\Gamma(\mathbf{doc}, \Lambda) \times \mathbf{doc} \rightarrow: \Gamma(\mathbf{doc}, \Lambda) \times$ . **Insert**( $G, n$ ) inserts the node  $n$  into the graph without connecting it to other nodes.

**insert** :  $\Gamma(\mathbf{doc}, \Lambda) \times \mathbf{doc} \times \mathbf{doc} \times \lambda \rightarrow: \Gamma(\mathbf{doc}, \Lambda) \times$ . **Insert**( $G, d_1, d_2, \lambda$ ) inserts an edge with lael  $\lambda$  between the documents  $d_1$  and  $d_2$ .

**delete** :  $\Gamma(\mathbf{doc}, \Lambda) \times \mathbf{doc} \rightarrow \Gamma(\mathbf{doc}, \Lambda)$ . **Delete**( $G, n$ ), with  $n \in \mathbf{nodes}(G)$  removes from the graph  $G$  the node  $n$  and all the edges connected to it.

**delete** :  $\Gamma(\mathbf{doc}, \Lambda) \times E(G) \rightarrow \Gamma(\mathbf{doc}, \Lambda)$ . **Delete**( $G, e$ ) removes the edge  $e$  from the graph  $G$ .

**nodes** :  $\Gamma(\mathbf{doc}, \Lambda) \rightarrow \mathit{set}\{\mathbf{doc}\}$ . **Nodes**( $G$ ) returns the set of nodes of the graph  $G$ .

**edges** :  $\Gamma(\mathbf{doc}, \Lambda) \rightarrow \mathit{set}\{E(G)\}$ . **Edges**( $G$ ) returns the set of edges of the graph  $G$ .

**type** :  $\mathbf{doc} \rightarrow \{\mathit{img}, \mathit{txt}\}$ . Given a document, determines whether it is a text document or an image.

$\sigma$  :  $\Gamma(\mathbf{doc}, \Lambda) \times (\mathbf{doc} \rightarrow \{\mathit{true}, \mathit{false}\}) \rightarrow \Gamma(\mathbf{doc}, \Lambda)$ .  $\sigma(G, \lambda x.Px)$  returns the graph formed by all the nodes that satisfy the predicate  $P$  and all the edges connecting them.

**union** :  $\Gamma(\mathbf{doc}, \Lambda) \times \Gamma(\mathbf{doc}, \Lambda) \rightarrow \Gamma(\mathbf{doc}, \Lambda)$ . The function **union**( $G_1, G_2$ ) (also expressed in infix form as  $G_1 \cup G_2$ ) returns the union of the two graphs  $G_1$  and  $G_2$ .

**intersection** :  $\Gamma(\mathbf{doc}, \Lambda) \times \Gamma(\mathbf{doc}, \Lambda) \rightarrow \Gamma(\mathbf{doc}, \Lambda)$ . The function **intersection**( $G_1, G_2$ ) (also expressed in infix form as  $G_1 \cap G_2$ ) returns the intersection of the two graphs  $G_1$  and  $G_2$ .

$\nu$  :  $\mathbf{doc} \times \mathbb{N} \rightarrow \Gamma(\mathbf{doc}, \Lambda)$  This is the  $k$ -downstream neighborhood introduced in the previous section.

$\nu^*$  :  $\mathbf{doc} \times \mathbb{N} \rightarrow \Gamma(\mathbf{doc}, \Lambda)$  This is the  $k$ -upstream neighborhood introduced in the previous section.

$TC$  :  $(\mathbb{R} \times \mathbb{R} \rightarrow \mathbb{R}) \rightarrow \mathbf{doc} \times \mathbb{R} \rightarrow \Gamma(\mathbf{doc}, \Lambda)$ . This is the score transitive closure operator introduced in the previous section. Note that, in the definition, the operator is *curried* [27] that is, applying it to a score combination operator  $\diamond$  one obtains a function  $T(\diamond) : \mathbf{doc} \times \mathbb{R} \rightarrow \Gamma(\mathbf{doc}, \Lambda)$  that computes the score-transitive closure with respect to that operator.

**similarity** :  $\{\mathbf{img}\} \times \mathbb{R}^{n \times n} \rightarrow \mathfrak{D}(\mathbf{img})$ . Given a set of images and a matrix of inter-image distances, the function **similarity** returns a similarity function adapted to the inter-image distances, as discussed in section 6.

**similarity** :  $\{\mathbf{doc}\} \times \mathbb{R}^{n \times n} \rightarrow \mathfrak{D}(\mathbf{doc})$ . Same as above, but for text documents.

### 7.3 Examples

The following examples illustrate the query process over a web image and text database. I will use a notation derived from ML[27] with the addition of a comprehension syntax[28, 5] to express the results of the query.

In this examples, I will make a rather informal use of similarity functions for visual features and for textual features. I will use the symbol  $s^v$  for visual similarity functions and the symbol  $s^t$  for textual similarity function. I will also assume that combination operators will be available as needed. How to manage these operators, and how to get them out of the database is a topic beyond the scope of this paper.

**Example 1.** Given a document  $d$  and an image  $i$ , find the image visually closest to  $i$  among those in the documents whose relevance for  $d$  is at least  $r$ , using the similarity

induced by a given configuration  $(I, \Psi)$  derived from the interface.

query  $q1(d, i, r, I, \Psi) \Rightarrow$

let

$$Q = \text{TC}(\diamond)(d, r)$$

$$I = \sigma(\text{nodes}(Q), \lambda x. (\text{type}(x) = \text{img}))$$

$$s = \text{similarity}(I, \Psi)(i)$$

in

$$\sigma_k^\#(\Sigma(s, I))$$

end

The first line builds, in  $Q$ , the graph of the documents whose relevance for the given one is at least  $r$ . This graph will contain, in general, both text documents and images. The following step breaks the structure of the graph, by taking only the nodes, and selects only those nodes that are of type “image.” The third statement builds a scoring function given by the similarity function induced by the interface.

Finally, the query statement takes the  $k$  nearest neighbors to the given image taken among the images of the set  $I$ .

**Example 2** Retrieve all the images from the pages pointing to document  $d$  that contain the word “mousetrap” with a weight of at least  $r$

query  $\text{mtrap}(d, r) \Rightarrow$

let

$$U = \sigma(\text{nodes}(\nu^*(d, 1)), \lambda x. (\text{type}(x) == \text{text}))$$

$$P(d) = \text{and}\{\text{true} | q \leftarrow d.\text{wds}, q.w > r, q.\text{text} = \text{“mousetrap”}\}$$

$$D = \{d | d \leftarrow U, P(d)\}$$

in

$$\{i | \text{type}(i) = \text{img}, i \leftarrow \nu(d), d \leftarrow D\}$$

end

The first step builds the set of documents that link to  $d$ , and extracts the set of nodes of type “text.” Note that the images themselves might contain the word “mousetrap,” but

the query explicitly requests all the images from the pages pointing to document  $d$  so one must start by considering the full text of the pages, that is, the documents of type “text.”

The second text defines a function that returns “true” if a document contains the word “mousetrap” with a weight of at least  $r$ . This is done using the comprehension syntax on the “and” monoid, whose details are beyond the scope of this paper[1].

The third step uses again the comprehension syntax to build the set of documents that satisfy the predicate  $P$ . Finally, the query statement extracts all the images from these documents. Note that in the web model used in this paper, images are linked to documents that is, to go from a document to an image therein contained, one must traverse a link. This is the reason why the statement contains the function  $\nu(d)$ .

**Example 3** Find all images that either more similar than  $q$  to image  $i$ , or that belong to documents whose relevance for the page containing  $i$  is at least  $r$ .

query  $\text{rel}(i, q, r) \Rightarrow$

let

$$I = \sigma_q^<(\Sigma(s_v(i)))$$

$$P = \sigma(\text{nodes}(\nu^*(i, 1)), \lambda x.(\text{type}(x) == \text{text}))$$

$$J = \{d \mid d \leftarrow TC(\diamond)(d'), \text{type}(d) == \text{text}, d' \leftarrow P\}$$

in

$$\{i \mid i \leftarrow \sigma(\nu(J, 1), \lambda x.(\text{type}(x) == \text{img})) \cup I\}$$

end

## 8 Analysis of the Assumptions

The system as a whole is a complex database, and can be “measured” along many directions. The most fundamental assumption I made in this paper, however, and that on which the whole model rests, is that there is some form of interaction between the textual information contained in the web page and the “visual semantics” induced by image similarity. Once this assumption is validated, the rest of the data model follows quite logically from it.

It is my intention in this section to present an empirical analysis of this assumption. I will use a somewhat indirect method, based on the following consideration: if the relation

posited between text descriptors and visual similarity exists, the text of a web page should provide information not only about the images contained in the page, but also on images visually similar to those contained in the page.

If, then, we take a database of images and attach a text description just to a portion of it, this should induce a better semantic information on the whole database. Suppose that in this database only a subset  $T$  of the images is associated to text, and that the user starts a query with some keywords. The database will retrieve the images with suitable text attached, plus some images without text attached but which are visually similar to the first. I am interested in three questions:

1. If the assumption is valid, the presence of the text in the images of  $T$  should increase the quality of the whole configuration that is returned. In particular, this effect should increase with the size of  $T$ .
2. Does the visual similarity come into play? That is, are significant images returned outside of the set of images associated with text?
3. Is the validity of the assumption related to a specific type of query, or is it valid across query types?

## 8.1 Method

The experiments were carried out using a database of 800 images harvested from the web. Four separate series of queries were considered in which the size of the labeled portion  $T$  was 16, 21, 28, and 40 images, corresponding to 2%, 2.6%, 3.5%, and 5% of the database. The evaluation of the quality of a query result is a complex issue that will not be considered in this paper. Rather, the 16 images closer to the query are collected and considered as the “answer” to the query, and the number of *satisfactory* images among them (see below for the determination of the satisfactory images) is used as a measure of quality. The number of satisfactory images divided by 16 gives the precision of the query answer [19].

The database was tested using seven queries covering a variety of situations. The queries asked for pictures of *airplanes*, *peaceful scenes*, pictures of *young women*, *sunny scenes*, pictures with *people*, *cartoons with animals*, pictures about *flight*, and pictures with a lot

of *action*. Each query  $q_i$  was first posed to the text database, obtaining a first set  $P_t(q_i)$  of 16 pictures. The pictures belonging to this set for which the similarity with the query was greater than a threshold  $\tau = 0.5$  were used as a visual query in the complete database  $D$ . The 16 top ranking images resulting from this query formed the set  $P_v(q_i)$ .

The total result is a collection of 14 sets of 16 images each:  $P_t(q_i)$  and  $P_v(q_i)$  for each query  $q_i$  ( $i = 1, \dots, 7$ ), the  $P_t$ 's being the results of the textual queries on the subset  $T$  of the database, the  $P_v$ 's being the results of the visual query on the whole database  $D$  initiated by the corresponding  $P_t$ 's. A group of 8 people was requested to evaluate the results. Each person was requested to evaluate 7 group of pictures corresponding to the 7 different queries. Before each presentation, the subject was told what the textual query was, and then a group of 16 pictures (either a  $P_t$  or a  $P_v$ ) was presented. The subjects were asked how many pictures in the set constituted a good answer to the query according to their own judgment. No subject was shown two groups of pictures relative to the same query, to avoid habituation effects, and no indication was given to the subjects of whether a particular group of pictures was a  $P_t$  or a  $P_v$  (as a matter of fact, the subjects were completely unaware of the goal and methodology of the experiment).

## 8.2 Results

The answer to the first question is synthesized in Fig. 4, which reports the average precision

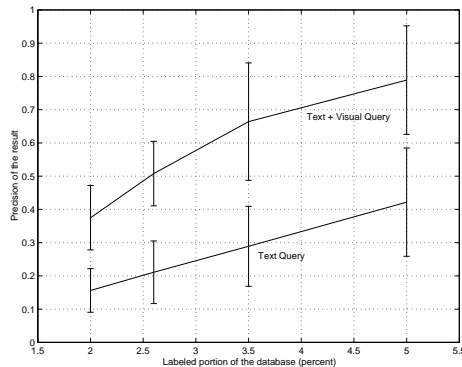


Figure 4: Precision of the visual query and the textual query as a function of the size of the labeled question.

across the seven queries for the textual query and for the visual query started by it as a function of the percentage of the database that has been labeled (that is, as a function of

$|T|/|D|$ ). As expected, the precision of the textual query increases roughly linearly with the size of the portion associated with text. For small values of  $|T|/|D|$  the precision of the visual answer seems to grow more rapidly than that of the textual answer until it reaches a point (around  $|T|/|D| = 0.03$ ) from which the two appear to grow at the same rate.

Fig. 5 shows the precision of the answer of the visual engine as a function of the precision

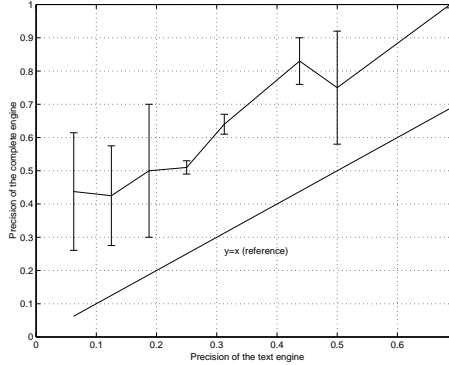


Figure 5: Precision of the result visual as a function of the precision of the answer of the textual engine.

of the answer of the textual engine. As expected, the precision of the visual engine is greater than that of the textual engine, indicating that indeed the visual query has explored regions of the image space beyond the labeled set  $T$ . The variance of the precision of the visual answer decreases with the increasing precision of the textual search, indicating that the more results the textual search provides, the more consistently the visual search can be started. It is hard to quantify this effect due to the possible presence of a disturbance not factored in the experiment.

As to the third question, at this time I can only offer some qualitative observation deriving from the execution of the experiments. Quite unsurprisingly, queries involving objects (an airplane, a young woman,...) has higher average precision and lower variance than queries involving more subjective concepts (a peaceful place, an action scene,...). This was expected, and is a consequence of the characteristics of symbolic and linguistic taxonomies, which are much better suited to describe the world in terms of objects than in terms of subjective experience. Visual search conducted with currently available technique, on the other hand, is ill suited for searching objects, while, in certain contexts [25] it is better

suited for searching “holistic” properties of images such as those giving rise to subjective impressions. This observation reveals a certain complementarity between textual and visual searches that the present method seems to be good at exploiting.

Note that this observation also hints at a disturbing factor in the interpretation of Fig. 5. Because of the characteristics of the textual search, the left portion of the figure, with high variances, is composed mostly of results of “subjective” queries, while the right portion contains mostly object oriented queries. The high variance in the left part of the figure might be due to inherent differences in subjective interpretations as well as characteristics of the search model.

## 9 Conclusions

This paper presented a data model and a query algebra for web image databases. The presence of the web, that is, of the pages in which the images are contained, and of the links between pages, provides a rich semantic structure from which a lot of information about images can be derived. While this can be sufficient for some query, the model posits (and the analysis appears to validate) that for some kinds of queries an interaction between the textual and visual components of the semantics might be necessary.

This model has been couched into a database theory suitable for navigation and query.

## References

- [1] A. Alcantara and B. Buckles. Supporting array types in monoid comprehensions. [citeseer.nj.nec.com/146257.html](http://citeseer.nj.nec.com/146257.html)
- [2] E. Amitay. Using common hypertext links to identify the best phrasal description of target web documents. In *Proceedings of the 21st ACM SIGIR Conference on Research and Development in Information Retrieval, Melbourne, Australia, 1998*.
- [3] L. Armitage and P. Enser. Analysis of user need in image archives. *Journal of Information Science*, 23(4):287–299, 1997.
- [4] R. Barthes. *La chambre claire : notes sùr la photographie*. Cahiers du cinéma, Paris, 1980.

- [5] P. Buneman, L. Libkin, D. Suciu, V. Tannen, and Limsoon Wong. Comprehension syntax. *SIGMOD Record (ACM Special Interest Group on Management of Data)*, 23(1):87–96, 1994.
- [6] S. Chakrabarti, B. Dom, P. Raghavan, S. Rajagopalan, D. Gibson, and J. Kleinberg. Automatic resource compilation by analyzing hyperlink structure and associated text. In *Proceedings of the Seventh International World Wide Web Conference, Brisbane, Australia*, 1998.
- [7] D. Cvetkovic, P. Rowlinson, and S. Simic. *Eigenspaces of graphs*. Cambridge University Press, 1997.
- [8] D. Dubois and H. Prade. A review of fuzzy set aggregation connectives. *Information Sciences*, 36:85–121, 1985.
- [9] U. Eco. *A Theory of Semiotics*. Indiana University Press, Bloomington, 1976.
- [10] P. Enser. Pictorial information retrieval. *Journal of Documentation*, 51(2):126–170, June 1995.
- [11] R. Fagin. Combining fuzzy information from multiple systems. In *Proceedings of the 15th ACM Symposium on Principles of Database Systems, Montreal*, 1996.
- [12] W. Frakes and R. Baeza-Yates, editors. *Information Retrieval, Data Structures and Algorithms*. Prentice Hall, Englewood Cliffs, 1992.
- [13] E. H. Gombrich. *Art and Illusion. A study in the psychology of pictorial representation*. Pantheon Books, 1965.
- [14] R. Hodge and R. Ian Vere. *Social Semiotics*. Polity, Cambridge, 1988.
- [15] C. Jorgensen. Attributes of images in describing tasks. *Information Processing and Management*, 34(2-3):161–174, 1998.
- [16] D. W. Joyce, P. H. Lewis, R. H. Tansley, M. R. Dobie, and W. Hall. Semiotics and agents for integrating and navigating through multimedia representations of concepts.

- In *Proceedings of SPIE Vol. 3972, Storage and Retrieval for Media Databases 2000*, 2000.
- [17] W. Y. Ma and B. S. Manjunath. Edge flow: A framework of boundary detection and image segmentation. In *Proceedings of CVPR '97 the IEEE International Conference on Computer Vision and Pattern Recognition*, 1997.
- [18] B. Manjunath. web page. <http://vision.ece.ucsb.edu/manjunath/>.
- [19] G. Salton. *Automatic Text Processing*. Addison-Wesley, 1988.
- [20] G. Salton and C. Buckley. Term weighting approaches in automatic text retrieval. *Information Processing and Management*, 24(5):513–523, 1988.
- [21] S. Santini. Semantic modalities in content-based retrieval. In *ICME 2000, IEEE International Conference on Multimedia and Expo New York, USA, July, 2000*.
- [22] S. Santini. *Exploratory Image Databases; Content Based Retrieval*. Academic Press, 2001.
- [23] S. Santini. A query paradigm to discover the relation between text and images. In *Proceedings of SPIE Vol. 4315, Storage and Retrieval for Media Databases 2000*, 2001.
- [24] S. Santini, Amarnath Gupta, and Ramesh Jain. Emergent semantics through interaction in image databases. *IEEE Transactions on Knowledge and Data Engineering*, 13(3):337–351, 2001.
- [25] S. Santini and Ramesh Jain. User interfaces for emergent semantics in image databases. In *Proceedings of the 8th IFIP Working Conference on Database Semantics (DS-8), Rotorua (New Zealand)*, January 1999.
- [26] J. Smith and S. F. Chang. Visually searching the WEB for content. *IEEE Multimedia*, 4(3):12–20, 1997.
- [27] J. Ullman. *Elements of ML Programming*. Prentice Hall, 1994.
- [28] P. L. Wadler. Comprehending monads. In *Proceedings of the 1990 ACM Conference on Lisp and Functional Programming, Nice, France*, 1990.