Octopus: Aggressive Search of Multi-Modality Data Using Multifaceted Knowledge Base

Jun Yang [1,2] Qing Li [1] Yueting Zhuang [2]

[1] Department of Computer Engineering and Information Technology, City University of Hong Kong, 83 Tat Chee Avenue, Kowloon, HKSAR, China 852-27889695

{itjyang, itqli}@cityu.edu.hk

[2] Department of Computer Science Zhejiang University Hangzhou, China, 310027 86-571-87951903

yzhuang@cs.zju.edu.cn

ABSTRACT

An important trend in Web information processing is the support of multimedia retrieval. However, the most prevailing paradigm for multimedia retrieval, content-based retrieval (CBR), is a rather conservative one whose performance depends on a set of specifically defined low-level features and a carefully chosen sample object. In this paper, an aggressive search mechanism called *Octopus* is proposed which addresses the retrieval of multi-modality data using multifaceted knowledge. In particular, *Octopus* promotes a novel scenario in which the user supplies seed objects of arbitrary modality as the hint of his information need, and receives a set of multi-modality objects satisfying his need. The foundation of *Octopus* is a multifaceted knowledge base constructed on a layered graph model (LGM), which describes the relevance between media objects from various perspectives. Link analysis based retrieval algorithm is proposed based on the LGM. A unique relevance feedback technique is developed to update the knowledge base by learning from user behaviors, and to enhance the retrieval performance in a progressive manner. A prototype implementing the proposed approach has been developed to demonstrate its feasibility and capability through illustrative examples.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval – *query formulation, relevance feedback, query models, search process.*

General Terms

Algorithms, Management, Design.

Keywords

Multi-modality, multimedia retrieval, multifaceted knowledge base, layered graph model, link analysis, relevance feedback.

1. INTRODUCTION

A close examination of content-based multimedia retrieval (CBR) systems reveals one of their common implications—the sample object used to formulate a query is virtually an eligible result of the query, usually the most relevant one. This observation leads to the following paradox. Suppose the user needs only one result, if he is able to find a good sample, he needs not bother to input it into the retrieval system, because the sample is exactly what he is looking for. If that is not the case, the users of CBR systems are still plagued by the task of finding representative samples to formulate effective queries. Quite often, the user has only a vague idea about the desired results in some details. On the other hand, even if the user has clear mind about what he would like to find, he may not be able to clarify it to the system due to the lack of a "right-to-target" sample object at hand.

The difficulty of finding good samples reveals a recognized problem in CBR systems—the lack of data semantics, which is of essential importance in judging the quality of retrieval results. However, what are used by most CBR systems are low-level features of media objects[1], such as color histogram for images, motion vectors for videos. Although these features reflect the data semantics to a certain degree, it is no doubt that they are inadequate to capture precisely the semantics of media objects. Providing good samples is a natural requirement of using low-level features: the system relies on the representative features of the sample to approximate the underlying semantics desired by the user. (There are also CBIR systems that use stretches or templates to formulate queries [5], which can be generally regarded as samples.) Moreover, since the low-level features are also media-specific, the sample object must be of the same modality as the desired results. The media objects retrieved by CBR systems are perceptually similar to (looks like or sounds like) the sample, but may not satisfy the requirement of the user who judges the relevance of an object at the semantic level.

Therefore, we regard the CBR systems as *conservative* systems, whose performance depends on a set of specifically defined features and carefully chosen sample object. Table 1 provides a summary of the CBR approaches vis-à-vis their drawbacks. In particular, to remedy these drawbacks, we propose a more *aggressive* mechanism—*Octopus*—for search of multi-modality data. It is characterized as aggressive based on the following two properties:

1. It exploits the knowledge on multiple aspects regarding the relevance between media objects. Based on such multifaceted knowledge, the retrieval results are not necessarily similar to the sample perceptually, but related to it in a more sophisticated and semantics-flavored manner.

[1] In this paper, a media object is an object of any modality, such as an image, video, text, etc. Meanwhile, if not indicated explicitly, we use "object" and "media object" interchangeably.

WWW 2002, May 7-11, 2002, Honolulu, Hawaii, USA.
ACM 1-58113-449-5/02/0005.

Table 1: CBR paradigm, drawbacks, and suggested remedies to multimedia retrieval

	CBR paradigm	Drawbacks	Octopus
Interaction	highly representative sample objects	difficulty of finding suitable samples	multi-modality objects serving as hints
Data index	low-level features	inadequate to capture semantics	multifaceted knowledge (user behaviors, structure, content)
Results	single-modality objects that are perceptually similar to the sample	no semantically relevant results	multi-modality, semantically related objects

2. It explores the relationships between media objects of different modalities, such that it becomes possible, for example, that an audio clip is retrieved from a sample image.

The objective of *Octopus* is to promote a novel scenario for multimedia retrieval: The user starts the search by supplying a set of seed objects as the hints of his information need, which can be of any modality (even different with the desired objects), and which are not necessarily the eligible results by themselves. From the seeds, the system figures out the user's need and returns a set of multi-modality objects that potentially satisfy his need. The user can give further hints by identifying the results (of any modality) that are close to his need, based on which the system improves the estimation of his need and refines the results accordingly. Therefore, the most prominent advantage of *Octopus* lies over traditional CBR systems in that externally it relieves the users from the task of providing highly representative samples, and internally it employs a broader range of knowledge to retrieve semantically relevant results.

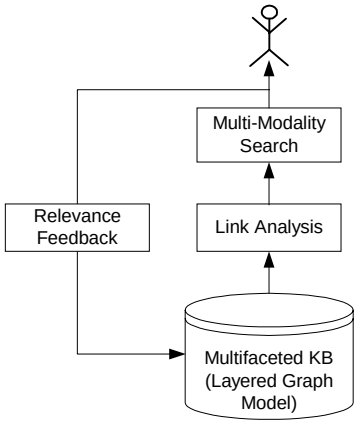


Figure 1: Overview of *Octopus* Mechanism

To support all the functionalities required by such a scenario, a suite of unique models, algorithms, and strategies are developed in *Octopus*. As shown in Figure 1, the foundation of the whole mechanism is a multifaceted knowledge base describing the relevance between multi-modality objects. The kernel of the knowledge base is a layered graph model (LGM), which characterizes the inter-object relevance estimated from three perspectives as (1) history of user behaviors, (2) structural relationships between media objects, and (3) content of media objects. Link structure analysis, an established technique in web-based applications, is adapted for the retrieval of multi-modality data based on LGM. The relevance feedback method used in *Octopus* can enrich the knowledge stored in LGM by learning from user-system interactions, such that *Octopus* has a hill-climbing nature (indicated by the loop in Figure 1) that allows its performance to be progressively enhanced based on the knowledge learned from previous queries and feedbacks.

We do not provide any quantitative performance evaluation in this paper, mainly due to the lack of benchmark for such multi-modality search. Actually, the main contribution of this paper is not on the performance improvement, but to bring out a novel retrieval scenario that is not even possible with previous retrieval approaches. Some characteristic queries and their results obtained using our prototype system are displayed to demonstrate the variety and flexibility of search in this scenario.

The rest of this paper is organized as follows. In Section 2, we present a formal description of the layered graph model as the core of the multifaceted knowledge base. The link analysis based algorithms for multi-modality data retrieval and relevance feedback are elaborated in Section 3. In Section 4, we introduce a prototype implementing the proposed approach and demonstrate its retrieval capability by illustrative examples. In Section 5, we discuss how our approach relates to the previous works on multimedia retrieval and link structure analysis. Finally we give the conclusion and suggest the future work in Section 6.

2. MULTIFACETED KNOWLEDGE BASE

In this section, we introduce a layered graph model as the core of the multifaceted knowledge base, along with a description of the knowledge acquisition process.

2.1 Layered Graph Model (LGM)

As the foundation of the retrieval functionality, the multifaceted knowledge base accommodates a broad range of knowledge regarding the relevance between media objects. In this paper, we use the term "media object" to refer to an object of various modalities, such as an image, a video clip, and a textual document. Some media objects can be regarded as composite objects that are composed from many "primitive" objects, e.g., a video clip is essentially a sequence of images.

In our approach, the relevance between two media objects can be evaluated mainly from three different perspectives: (1) Users' interpretation of the two objects, which can be deduced from user interactions, e.g., designating them as the positive examples of the same query. (2) Structural relationships between them, e.g., there is a hyperlink between them. (3) The similarity between two objects in terms of their content, which can be estimated based on their low-level features. To accommodate the knowledge on the three aspects, we develop a layered graph model (LGM) as the core of the multifaceted knowledge base, with each layer

modeling knowledge on one aspect. The formal definition of LGM is given as follows.

Definition. *The **layered graph model** (LGM) consists of three superimposed knowledge layers, which from top to bottom are **user layer**, **structure layer**, and **content layer**. A **knowledge layer** is an undirected graph G=(V, E), where V is a finite set of vertices and E is a finite set of edges. Each element in V corresponds to a media object $O_i \in O$, where O is the collection of media objects in the database. E is a ternary relation defined on V×V×R, where R represents real numbers. Each edge in E has the form of $<O_i, O_j, r>$, denoting a link between O_i and O_j with r as the weight of the link. The graph corresponds to a |V|×|V| **adjacency matrix**[2] M=[m_{ij}], where each element $m_{ij}=r$ if there is an edge $<O_i, O_j, r>$ between O_i and O_j, and $m_{ij}=0$ if there is no edge between them. Obviously, M is a symmetric matrix ($m_{ij}=m_{ji}$), and its elements on the diagonal are set to zero ($m_{ii}=0$). The vertices of the three layers correspond to the same set of media objects, while the links in each layer denote the relevance between two media objects defined from one of the three perspectives mentioned above.*

Figure 2 illustrates the LGM. Note that the order of the three layers is fixed, which reflects the degree of reliability of the inter-object relevance suggested by the links in each layer. The user layer is on the top, because user judgment is very reliable (not always reliable considering the subjective errors and biases) in suggesting the relevance between media objects. Structure link is also a strong indicator of the relevance between objects, but is not as reliable as user links. The content layer is at the bottom, since the similarity calculated based on low-level features does not have any well-defined mapping with object relevance perceived at semantic level.

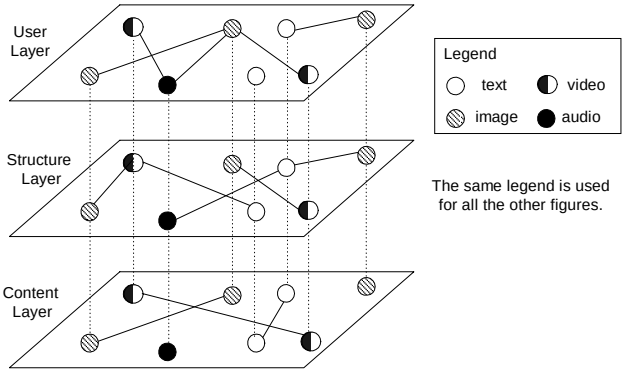


Figure 2: The layered graph model (LGM)

Different from the convention of storing the index of each object with itself, the LGM stores the knowledge as the links between media objects. An advantage of such link-based knowledge representation is that the retrieval can be restricted in a relatively small locality connected via links instead of in the whole database, and therefore it can effectively reduce the search space and afford more sophisticated retrieval algorithms. However, the graphic structure is also expensive in computation and storage, especially when the number of nodes and links get large.

2.2 Knowledge Acquisition

In the following, we describe the knowledge acquisition process on each knowledge layer, i.e., how to construct the three types of links in LGM.

- **User Layer**. User link reflects the user belief that two media objects are relevant in some sense, and the weight of a user link indicates the degree of confidence of such belief. A straightforward way of obtaining user links is to let the user create all the links manually, which is nevertheless a time-consuming and labor-intensive process. Alternatively, the links can be acquired implicitly by learning from user-system interactions in the retrieval process, specifically, relevance feedback. Consider a typical scenario in CBR systems: a user starts a query with object *A* as the sample object, and among the results returned by the system he designates objects *B* and *C* as relevant examples to the query. In this case, we may create new links between *A* and *B*, *A* and *C*, or even *B* and *C*. As the user interactions proceed, the coverage and the quality of user links are progressively improved. The advantage of this strategy lies in that it exploits the interactions of the entire population of users for knowledge acquisition, and thereby relieves the significant human labors. A detailed algorithm for the updates of user links using the above strategy is presented in Section 3.4.

- **Structure Layer**. Structure links can be interpreted as spatial neighborhood, hyperlink, or composition relationships between two objects, depending on the physical environment where the data are collected. For example, for a typical organization of web pages in Figure 3(a), we can create the structure links as shown in (b). The textual content of a page is regarded as a single text object. An image or a video clip is regarded as in the page either if it is embedded in the page or if it is pointed by a hyperlink on it. All the media objects within a page are interconnected by structure links (e.g., objects *A*, *B*, and *C* are connected to each other). A hyperlink is mapped to structure links from the source object to all the objects in the destination page (e.g., *A* is linked with *D* and *E*, while *E* is linked with *A*, *B*, and *C*). For simplicity, the weights of all structure links are set to 1. The same strategy for structure link construction can be applied to other forms of hypermedia (e.g., a digital encyclopedia). Further, it can be even adapted to non-hypermedia data collections (e.g., e-books), by interpreting the spatial vicinity as a hyperlink. Note that compared with the previous link analysis approaches, here we adopt a simplification of representing all the structure links as undirected links, in order to be consistent with user links and content links.

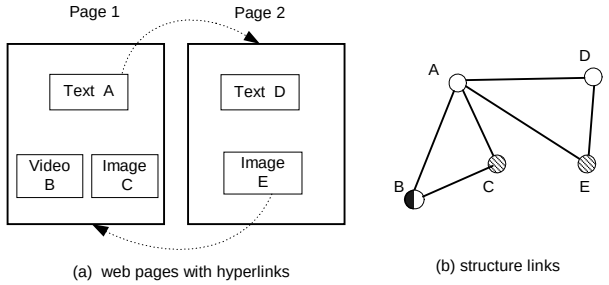


Figure 3: Structure links construction in a web environment

[2] The adjacency matrix defined here is slightly different from its mathematical definition, in which each component is a binary value indicating the existence of the corresponding edge.

- **Content Layer**. A content link reveals the similarity between the content of two objects, defined on primitive[3] and media-specific features, such as color histogram for images, motion vector for video clips, with a weight indicating the degree of similarity. Obviously, content links only exist between objects of the same modality, and if no restriction is imposed, they can exist between any pair of such objects, which are interconnected into several complete sub-graphs (one for each modality). However, since the content links with low similarity are unreliable and noisy, we apply a cut-off threshold on the link weights to remove the low-weighted links. In practice, when a new object is registered into the database, it is compared with all other objects of the same modality with it, and links are created between it and those that have a content similarity above the threshold with it.

3. LINK ANALYSIS BASED RETRIEVAL AND RELEVANCE FEEDBACK

As illustrated in Figure 4, the retrieval process of *Octopus* can be described as a circle: the desired objects are retrieved through the upper half-circle, and the user evaluations are collected and incorporated into the knowledge base though the lower half-circle, which initiates a new circle to refine the previously retrieved results based on the updated knowledge. Consequently, this process has a *hill-climbing* nature in the sense that the retrieval performance is enhanced incrementally as the loop is repeated.

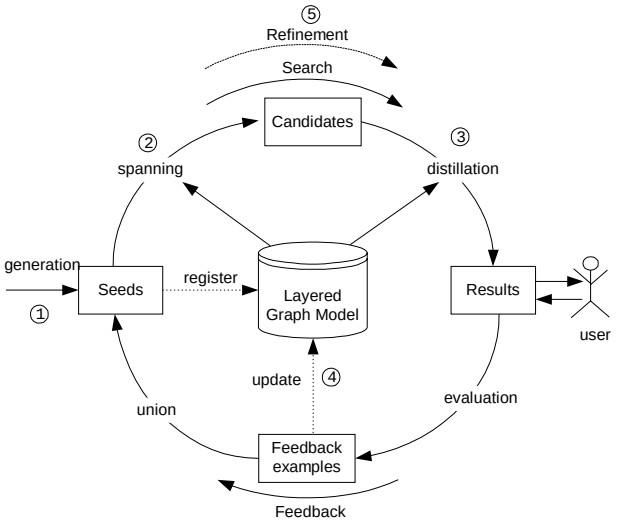


Figure 4: Overview of link analysis based retrieval algorithm

In this section, we describe the whole retrieval process in five steps (see Figure 4): (1) generating the seed objects as the hints of the user's information need, (2) spanning the seeds to a collection of candidate objects via the links in the LGM, (3) distilling the results by ranking the candidates based on link structure analysis, (4) updating the LGM by incorporating the user evaluations on the current results, and (5) refining the retrieval results based on the updated LGM and the user evaluations.

[3] We use the term "primitive" instead of "low-level", since the primitive feature for text object is keyword, which is not traditionally considered as low-level features.

3.1 Seed Generation

Seed objects play the similar role as query examples in the CBR paradigm—formulating user queries. Nevertheless, the differences between them are fundamental. On one hand, seed objects are not necessarily eligible results of the query, and therefore they need not to be highly representative; on the other hand, seed objects can be of any modality, which may not be the same as that of the desired objects.

The user generates the seed objects either by selecting them from the database or by introducing (creating) new objects. In the latter case, the new object is automatically registered into the database with its content links and structure links (if any) with existing objects created (see Section 2.2). Obviously, there are no user links connected to the new object before it is involved in any user interactions. Note that this query formulation paradigm naturally subsumes the query-by-example and query-by-keyword paradigms, since the seed can be a media object (e.g., an image) or a piece of text consisting of several keywords.

3.2 Candidates Spanning

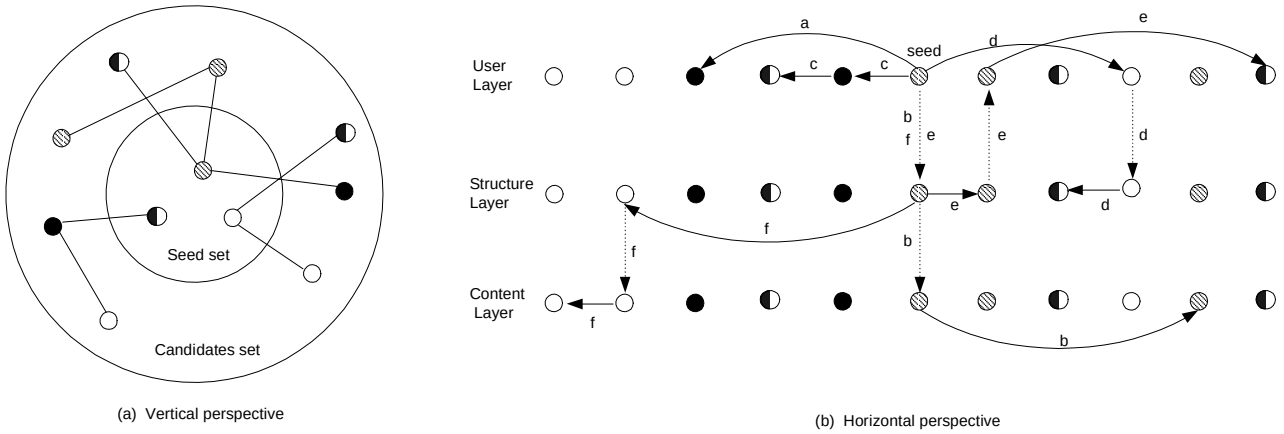
Since the seed objects provide the hints to the user's need, it is reasonable to assume that the desired objects are related to the seeds in a certain manner, specifically, through a path in the LGM. The path can be made up of links belonging to different layers in the LGM. Based on this assumption, we identify a collection of candidate objects by spanning from the seed objects through the links in the LGM. This operation equals to the construction of a small sub-graph around the seeds in the LGM. The candidate set *C* must satisfy the following two criteria:

(1) *C* must be rich in containing the objects that are highly relevant to the seed objects.

(2) *C* is relatively small, so that it can afford the computational cost of the distillation and feedback algorithms applied on it subsequently.

Both requirements favor the use of short paths in spanning, since short paths imply high relevance between the seeds and the candidates, and are less probable to produce large candidate set. Consequently, we place a threshold on the maximum length of the path (viz. number of links) between a seed and a candidate. The threshold is usually very small (e.g., 2 or 3), depending on the scale of the data collection and density of links. Only the objects that are reachable from the seeds through links less than the threshold are identified as the candidates for further processing.

However, even after this threshold is applied, the size of the candidate set is still very unpredictable, mainly because of the varying number of links each object has, especially the structure links and content links. Some web pages may have hundreds or even thousands of hyperlinks pointing to it (e.g., the official site of ACM), which may result in high density of structure links. Moreover, the number of content links is likely to be high when the corresponding object has many similar objects, and vice versa. Sometimes the number of candidates is so large that the subsequent processing is unaffordable and meaningless due to the low quality of the candidates.



(a) Vertical perspective

(b) Horizontal perspective

Figure 5: Candidates spanning

Consequently, we put a second threshold on the total number of candidates. If the candidates generated by spanning go beyond the threshold, the exceeding ones are discarded. But, what are the criteria to choose the appropriate victims? Put in other words, how to rank the candidates so that the most promising ones will not be discarded? In our algorithm, the ranking of candidates is determined by the shortest path through which the candidate is reached from the seed. In particular, two factors of the path are considered: the length of the path as well as the type of links that constitute that path. The first factor captures the intuition that the closer two objects are, the more relevant their relationship is. The second factor takes into account of the priorities of the three types of links. Consider two paths of the same length. If one path goes through the user layer while the other is at the content layer, it is natural to conclude that the two objects connected by the first path are more relevant than those by the second path. From this observation, we formulate the following three heuristic rules for ranking:

(1) A candidate c_1 reached through a path shorter than that of another candidate c_2 is ranked higher than c_2.

(2) If two candidates are reached through two paths of the same length, they are ranked according to the *lexicographic order*.

(3) The candidates whose relative order cannot be decided by (1) and (2) are ranked randomly.

Suppose we use *U*, *S*, and *C* to denote user link, structure link, and content link respectively (with the order *U* precedes *S* which, in turn, precedes *C*), and represent a path by the types of its links. Then, the rank of paths determined by the above heuristic rules is as follows:

U — S — C — UU — US — UC — SU — … — CS — CC —UUU — UUS…

Figure 5 gives a vertical view and a horizontal view of the candidates spanning process. From the horizontal view, one can see how the spanning goes through different paths and jumps between layers: path *a* is of pattern '*U*', path *b* is '*C*', path *c* is '*UU*', path *d* is '*US*', path *e* is '*SU*', and path *f* is '*SC*'. (The objects shown in a column represent the same object at different layers.)

The algorithm for candidates spanning together with the ranking is shown as follows:

Spanning (*S*, *l*, *t*)
S: the seed set
l: the maximum length of the path
t: the upper bound on the number of candidates
p: the string representing the pattern of a path
nextpath(*p*): subroutine that returns the path that lexicographically succeeds the path *p*, e.g., *nextpath*('*US*') = '*UC*'.
return: the candidate set

Set *C* to empty set
For *i*=1 to *l*
  Set *p* to the first path of length *i* in lexicographical order
  While *p* is not the last path of length *i* in lexicographic order
    Let *L* as the set of objects reachable from the objects in *S* through path *p*
    If |*C* ¡ È| < *t*, Then
      C := *C* ¡ È
    Else
      Randomly select (*t*-|*C*|) objects from *L* and add them into *C*
      Return *C*
    End If
    p := *nextpath*(*p*)
Next
Return *C*

Algorithm 1. Spanning from seed set to candidate set

Although the candidates are ranked by the heuristic rules, the ranking is rather tentative and rough. For example, it is very arguable to rank the candidates with path '*C*' higher than the candidates with path '*UU*'. Moreover, the weights of links are not considered in ranking these candidate objects. In the distillation process, this tentative ranking is discarded and all the candidates are re-sorted by analyzing the link structure using a more sophisticated algorithm.

3.3 Results Distillation

In this phase, the link structure of the sub-graph that corresponds to the candidate objects is analyzed, in order to determine the relevance of each candidate object to the query. Based on our basic premise that a link conveys relevance between two objects, we make a further assumption that a candidate object is more relevant to the query, if (1) it connects with a larger number of relevant candidates, or (2) it connects with relevant candidates

through links of higher weights, and (3) it connects to candidates that are more relevant to the query.

Since the LGM has three layers, the distillation is performed in two steps: firstly, the candidates are ranked by analyzing the link structure at each single layer, and then, the ranking of different layers are merged to give the final ranking. The single-layer ranking algorithm works iteratively. Suppose each candidate object O_i has a relevance score r_i, which is initialized to 1.0. In each round, we update r_i by setting it to the sum of the product of the link weight and the relevance scores of the objects linking with O_i and then normalizing it. Note that such an update nicely captures our assumption—the object with a large number of links, high-weighted links, and links with relevant objects will get a high relevance score. The process repeats until every r_i converges to a fixed value, which gives the final relevance scores of the corresponding object. The detailed algorithm is shown as follows:

Rank (C, s)
C: the candidate set
s={*"user", "structure", "content"*}: the knowledge layer
r=[r_i]: the relevance vector with each element r_i as the relevance score of object O_i in C
M=[m_{ij}]: the adjacency matrix of the sub-graph corresponding to C at the layer s
return: the relevance vector for C

Initialize all the elements of vector r to 1.0
While the vector r has not been converged
  For each object O_i in C
    $r_i := \sum_{j=1,\ldots,|C|} (r_j \cdot m_{ij})$
  Next
  Normalize R such that $\sum r_i^2 = 1$
Return r

Algorithm 2. Ranking candidates at a single layer

The above algorithm updates the vector r by repeating the operation $M \times r \rightarrow r$, until it converges. At that time, the elements of r give the final relevance score of each object to the query, according to which the candidates can be sorted. Many previous works on link analysis [13] [19] have proved the convergence of r (i.e., termination of the algorithm), and r is actually the *principal eigenvector* of the matrix M.

After applying the above ranking algorithm on each of the three layers in the LGM, we need to merge the three ranking lists into a uniform one. However, since the three layers deal with the knowledge on different aspects, it is nearly impossible to design a "fair" strategy for the combination of results. We suggest a heuristic strategy for this task by linearly combining the relevance scores (of a candidate) obtained from different layers to compute the overall relevance score, which is shown in Algorithm 3. Intuitively, the three weights used in the algorithm has the relation of $w^U > w^S > w^C$, which reflects the priorities of the three layers. The candidate objects are ranked according to their overall relevance scores generated by this algorithm before they are presented to the user.

Distillation(C)
C: the candidate set
r = [r_i]: the overall relevance vector with each element r_i being the overall relevance score of object O_i in C
w^U, w^S, w^C: the weight for the user layer, the structure layer, and the content layer
return: the overall relevance vector for C

r^U := **Rank** (C, "user")
r^S := **Rank** (C, "structure")
r^C := **Rank** (C, "content")
For each object O_i in C
  $r_i := w^U \cdot r_i^U + w^S \cdot r_i^S + w^C \cdot r_i^C$
Next
Return r

Algorithm 3. Ranking candidates by combing multiple layers

3.4 Knowledge Update

If the user is not fully satisfied with the results generated in the distillation phase, he can give further hints by labeling the current results as either relevant or irrelevant examples. Upon the acceptance of such user evaluations, the system initiates a two-stage process: firstly, it incorporates the knowledge deduced from user evaluations into the LGM; and then, it refines the previous results based on the updated LGM and the user evaluations. The first stage has a *long-term* influence since it updates the knowledge base, while the second stage focuses on *short-term* effect as the user satisfaction in the current retrieval session.

The user evaluations are incorporated into the LGM by updating the user links. The underlying principle of link update is rather intuitive: for a relevant example, we link it with every seed object, or increase the weight of the existing link between them; for irrelevant examples, we take the opposite action. The algorithm for user link update is presented as follows:

Update (S, F^+, F^-)
S: the original seed set
F^+: the set of relevant examples
F^-: the set of irrelevant examples
M_U =[m_{ij}]: the adjacency matrix of the user layer
s, t: positive real numbers

For each object O_i in S
  For each object O_j in P
    $m_{ij} := m_{ij} + s$
  Next
  For each object O_k in N
    $m_{ik} := m_{ik} - t$
    If $m_{ik} < 0$, then $m_{ik} := 0$
  Next
Return

Algorithm 4. Update knowledge base from user evaluations

Note that m_{ij} not only defines the weight of a link, but also governs the existence of the link. When m_{ij} is increased from zero to a positive value, a link between O_i and O_j is created; when m_{ij} is decreased to zero, the link is removed. The parameter t is usually set to a value larger than s, so that a link on which users have contradictory opinions will not receive a confidence weight.

By incorporating the up-to-date user evaluations into the LGM as user links, *Octopus* allows the future queries to benefit from these previously conducted user interactions, such that the retrieval performance can be progressively improved. Compared with the evolving user layer, the structure and content layer of the LGM are passive, which do not change after their initial construction.

3.5 Result Refinement

The objective of the refinement process is to refine the retrieval results based on the user's evaluation made on the previous results. As shown in Figure 6, the refinement process undergoes the similar three steps (seed generation, spanning, and distilling) at two levels (positive and negative) in parallel, with the results finally merged. Firstly, the original set of seed objects is combined with the relevant examples, resulting in a set of *positive seeds* S^+; meanwhile, the irrelevant examples are regarded as *negative seeds* S^-. Then, the positive and negative seeds are spanned into two groups of candidate objects, called *positive candidates* C^+ and *negative candidates* C^-, respectively. Finally, both groups of candidates are ranked using link analysis in the distillation process, and the results are merged to give the final ranking list. (By merge, we mean the integration of the relevance scores instead of combination of objects.)

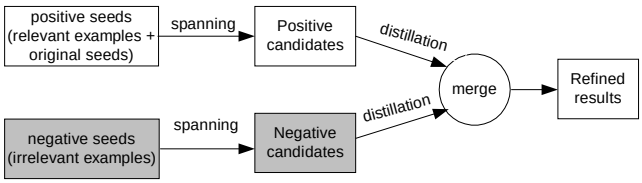


Figure 6: Result refinement process

The algorithm for the refinement process is presented below. The idea behind this algorithm is very intuitive: the refined results should be closely linked with the relevant examples and at the same time far away from the irrelevant ones. Again, the refined results are ranked according to the relevance vector returned as the outcome of this algorithm.

Refinement (S, F^+, F^-)
S: the seed set
F^+: the set of relevant examples
F^-: the set of irrelevant examples
return: the overall relevance vector for the refined results

$S^+ := (S \cup F^+) - F^-$
$S^- := F^-$
$C^+ :=$ **Spanning** (S^+, l^+, t^+)
$C^- :=$ **Spanning** (S^-, l^-, t^-)
$r^+ :=$ **Distillation** (C^+)
$r^- :=$ **Distillation** (C^-)
For each object O_i in C^+
    If O_i is in C^-, then
        $r(O_i) := r^+(O_i) - r^-(O_i)$
    Else
        $r(O_i) := r^+(O_i)$
    End If
Return r

Algorithm 5. Results refinement based on user evaluations

3.6 An Algorithmic Overview

So far we have completed the whole loop of retrieval process shown in Figure 4. We integrate all the aforementioned algorithms into the following "main routine" to present an algorithmic overview of the main flow of the *Octopus* mechanism.

Octopus (S):
S: the set of seed objects
sort(C, r): a subroutine that sorts the elements in set C according to vector r, which gives the relevance score of each element in C.
return: R, the set of ranked results

$C :=$ **Spanning** (S, l, t)
$r :=$ **Distillation** (C)
$R :=$ *sort*(C, r)
While the user is not satisfied with R
    Let F^+ and F^- be relevant and irrelevant examples of the current session
    Update (S, F^+, F^-)
    $r :=$ **Refinement** (S, F^+, F^-)
    $S := (S \cup F^+) - F^-$
    Let C^+ be the set of objects corresponding to r
    $R :=$ *sort* (r, C^+)

Algorithm 6. The main flow of *Octopus* mechanism

4. PROTOTYPING AND ILLUSTRATIVE EXAMPLES

A preliminary prototype system is implemented based on the *Octopus* mechanism. The modalities currently supported are text, image, and video; audio is left out simply because we do not have any audio processing algorithms at hand. The primitive features and similarity functions utilized for these media are shown in Table 2. To guarantee high efficiency, the maximum path length permitted for candidate spanning (see Algorithm 3) is set to 2, and the total number of candidates is restricted to 100.

Table 2: Primitive features and similarity metric used in the prototype system

	Primitive features	Similarity metric
Text	keywords (TF*IDF weighting)	cosine distance
Image	256-d HSV color histogram, 64-d LAB color coherence, 32-d Tamura directionality.	Euclidean distance
Video	shot boundary detection, using first frame of each shot as key-frame	key-frame similarity as shot similarity, average pair-wise shot similarity as video similarity

We do not conduct any quantitative evaluation on the retrieval performance mainly due to the lack of benchmark for such multi-modality search. There does not exist, for example, a criterion to evaluate the quality of some text and images retrieved by a video clip as the seed. Moreover, there are too many human factors involved in this cooperative mechanism, such as the selection of seeds and evaluation of results, which further complicate the task of performance evaluation. In fact, providing performance

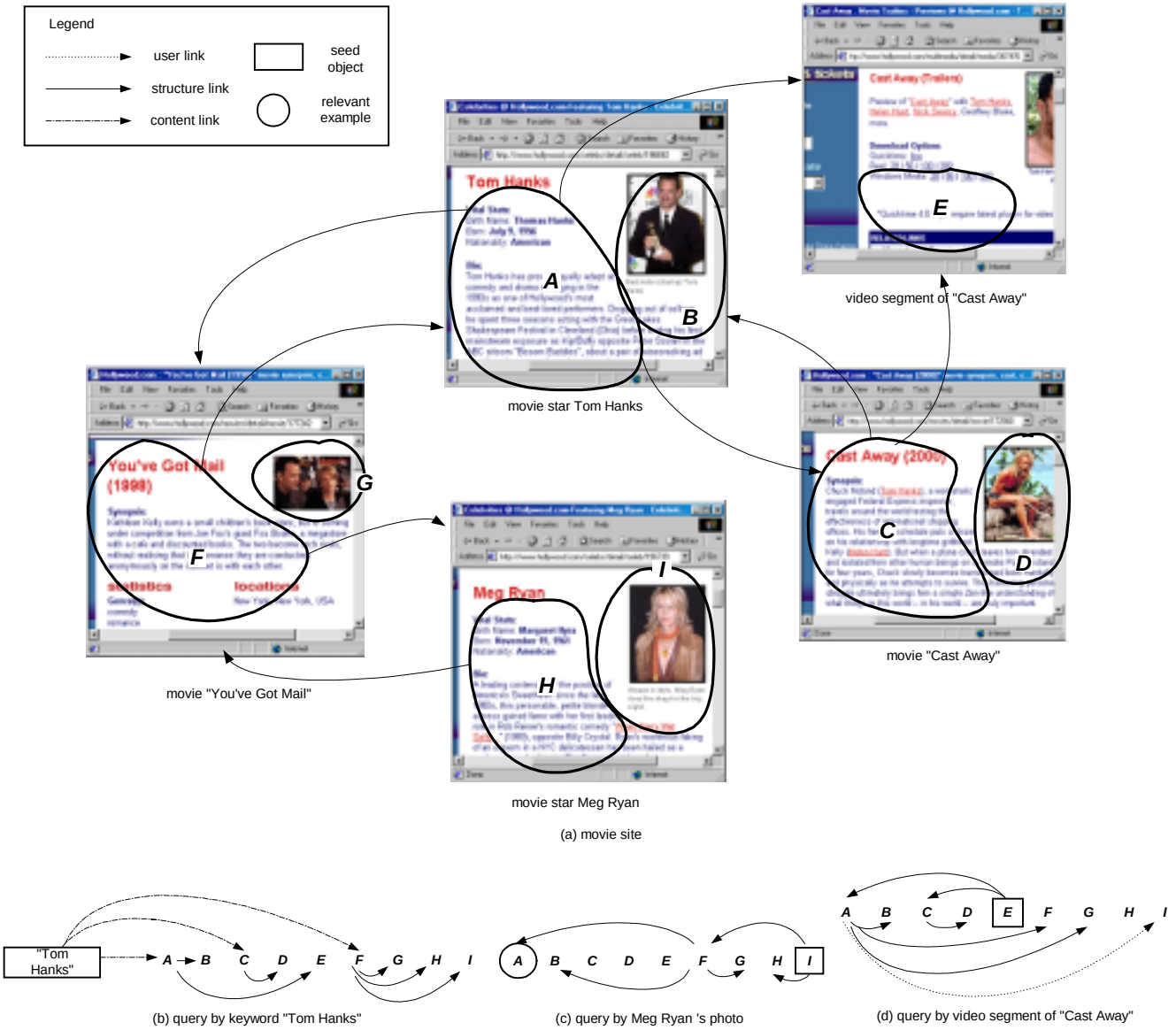


Figure 7: Illustrative examples

improvement over CBR approaches is not the main objective of *Octopus*; instead, its emphasis is on a novel scenario for multi-modality retrieval, which is not possible with previous approaches. Some characteristic queries and results are shown below to demonstrate the variety and flexibility of the retrieval in this new scenario.

Figure 7(a) shows some pages of a website about movies, whose content is rich in multimedia objects. There are two major types of pages in this site: page of movie stars such as Tom Hanks and Meg Ryan, as well as page for movies like "You've Got Mail" and "Cast Away". The star's page contains his/her photo and biography (text), while the movie's page has an introduction to the movie, along with a picture showing one of the movie scenes. The page of each star points to the pages of the movies in which he/she had played a role, e.g., the pages of Tom Hanks and Meg Ryan both point to the movie "You've Got Mail". Meanwhile, the page for a movie points to the pages of the stars who are in the cast. There is also a video clip of the movie "Cast Away" (object E) available in a separate page (it is not shown explicitly, but is pointed by a hyperlink on the page). All the hyperlinks are shown in Figure 7(a), based on which we can construct all the structure links using the strategy introduced in Section 2.2.

Figures 7(b)-(d) illustrate how *Octopus* works for three different types of queries. In the first case, the user input Tom Hanks' name as the query, intending to find some materials about him. Since the query is an isolated text object that does not previously exist in the LGM, it has neither user links nor structure links. Therefore, in the candidate spanning process, we firstly rely on the content links to find three text objects (introductions to "Cast Away" and "You've Got Mail", and his biography) in which Tom Hanks' name recurs several times. All the other objects are reached from these three text objects through structure links. So, although this query starts with a traditional "search-by-keyword" mode, it results in a rich collection of multimedia objects,

including his photo, the introductions to his movie, the movie scene and video clip, and even his partner Meg Ryan's materials.

In the second query (see Figure 7 (c)), the user chooses Meg Ryan's photo as the seed object. Following the structure links from it, we reach her biography, the materials about her movie "You've Got Mail", through which the Tom Hanks' page is also retrieved. Note that this search is opposite to what CBR systems usually do, i.e., using images to search text rather than searching images by text. We suppose that in feedback, the user labels Tom Hanks' biography as a relevant example, so that a user link is created between it and the Meg Ryan's photo.

The last query (see Figure 7(d)) is even more ambitious. Starting with a video clip, the user wants to find some related materials about the movie "Cast Away". As the results of traversing along structure links, the content on the page of "Cast Away" and Tom Hanks are returned. In addition, the user link created in the previous session leads us to Meg Ryan's photo via Tom Hanks' biography. (It makes sense since Meg Ryan and Tom Hanks had cooperated in many famous movies.)

5. RELATED WORK

In this section, we discuss the connection of our model with the previous works on multimedia retrieval and link analysis, and demonstrate in some cases, how our model can be reduced or transformed to other approaches.

5.1 Multimedia Retrieval

Previous works addressing multimedia retrieval can be classified into two groups: approaches on single-modality as well as on multi-modality integration.

- **Single-modality retrieval.** The retrieval approach in this group only deals with a single type of media, so that most content-based retrieval approaches (e.g., [4],[7],[12],[20],[21]) fall into this group. Among them, the QBIC system [7], MARS project [12], VisualSEEK system [20] focus on image retrieval, VideoQ system [4] is for video retrieval, and WebSEEK [21] system is a Web-oriented search engine that can retrieve both images and video clips. These approaches differ from each other in either the low-level features extracted from the data, as well as the distance functions used for similarity calculation. Despite the differences, all of them are similar in two fundamental aspects: (1) they all rely on low-level features; (2) they all use the query-by-example paradigm. Since the content layer of our LGM is built based on the similarity among objects on low-level features, our approach can be reduced to other CBR approaches if we consider only the content layer during the retrieval process, and rank the candidates according to the weight of their content links to the seed.

- **Multi-modality integration.** In the past few years, some works have investigated the integration of multi-modality data, usually between text and image, for better retrieval performance. For example, the *iFind* [17] system proposes a unified framework under which the semantic feature (text) and low-level features are combined for image retrieval, and the *2M2Net* [23] system extends this framework to the retrieval of video and audio. WebSEEK system [21] extracts keywords from the surrounding text of image and videos, which is used as their indexes in the retrieval process. Although these systems involve more than one media, different medias are not actually integrated but are on different levels. Usually, text is only used as the annotation (index) of other medias. In this regard, our mechanism enables an extremely high degree of multi-modality integration, since it allows the interaction among objects of any modality in any possible ways (via different types of links).

More recently, the MediaNet [1] and multimedia thesaurus (MMT) [22] are proposed, both of which seek to provide a multimedia representation of semantic concept—a concept described by various media objects including text, image, video, etc—and establish the relationships among these concepts. MediaNet extends the notion of relationships to include even perceptual relationships among media objects. Both approaches can be regarded as "concept-centric" approaches since they realize an organization of multi-modality objects around semantic concepts. From this view, our mechanism is "concept-less" since we make no attempt to identify explicitly the semantics of each object.

5.2 Link Analysis

There have been many successful previous works on link analysis, among which the most notable ones are the *PageRank* model and the notion of *hubs & authorities*. *PageRank* [3] is based on the random-walk model and is used to compute the probability that a Web surfer visits a certain page. The effectiveness of this model has been proved by its successful application in search engine Google [1]. In contrast, Kleinberg [13] suggested that each page has two scores: *authority* score, which describes how authoritative a page is to a certain topic, and *hub* score, which reflects how many authoritative pages it points to.

The link analysis technique has been successfully applied to a broad range of applications. The approaches of Bharat et al. [2], *PageRank* model [3], HITS [13] are used to search for most authoritative pages to a certain topic. The approach proposed by Rafiei et al. [19] identifies the topics of a designated page. Dean et al. [6] discusses how to find related pages to a certain page. There is also a group of works (e.g., Kumar et al. [14], Gibson et al. [8], Pirolli et al. [18]) that aim at inferring and analyzing web communities or other web structures from the hyperlinks. Henzinger et al. [10] suggested measuring link quality of a web page using the random-walking model. Very recently, Lempel et al. [15] proposes PicASHOW system, which employs link analysis to web-based image retrieval.

Since the link analysis approach in *Octopus* is geared towards the goal of multimedia retrieval, it differs from conventional link analysis approaches in the following aspects:

- **Application**: To our knowledge, *Octopus* is the first application of link analysis in the search of multi-modality data. (PicASHOW only deals with images.)

- **Link types**: Our multifaceted knowledge base accommodates three types of links, while most previous approaches focus on only hyperlinks, which is actually a special form of our structure link. This implies that our approach can be reduced to other approaches if only the structure layer is addressed in the retrieval process. Some link analysis approaches (e.g., [2],[19]) also take into account the content (text) similarity. However, they usually combine the content similarity with the analysis of hyperlinks, rather than building another separate layer for it as is the case in the LGM.

- **Link analysis algorithm**: In terms of algorithm, our link analysis algorithm is much closer to the *PageRank* model, since for each object we calculate only one score. However, we do not

use the random-walk model, since our LGM is fundamentally different from the world of hyperlinks in which the random-walk model makes sense. We do not adopt the *hubs* and *authorities* model because it is based on the observation that in the Web the relevance may propagate from one page to another via a totally irrelevant page through hyperlinks, which does not agree with our basic premise that relevance spreads between directly linked objects.

- **Link update**: Most previous works on link analysis suggest *static* approaches in that they only analyze the link structure. In contrast, our mechanism is *incremental* as it permits user links to be enriched and updated by learning from user behaviors. Undoubtedly, our approach is more preferable since it allows self-improvement of the retrieval performance.

6. CONCLUSION AND FUTURE WORK

In this paper, we have described the *Octopus* mechanism for aggressive search of multi-modality data based on a multifaceted knowledge base. Specifically, this mechanism applies link analysis techniques to search for multi-modality objects, the relevance between which is described by a layered graph model (LGM) as the core of the knowledge base. A unique relevance feedback technique is developed that can enhance the retrieval performance progressively by learning from user behaviors. The highlights of our mechanism are summarized as follows:

- At the interface level, *Octopus* provides users with great convenience and flexibility. For example, the seed objects can be of any modality and are not necessarily representative samples. The retrieval results are also of multiple modalities, which can meet the variety of user requirements.

- The LGM investigates a broad coverage of knowledge to evaluate the similarity between media objects. Therefore, the results retrieved based on it are more relevant (to the query) than those retrieved by the CBR systems, which rely on low-level features only.

- The knowledge base is enriched by learning from user behaviors, such that the retrieval performance can be enhanced in a hill-climbing manner.

- The LGM provides a solid and generic foundation for multimedia retrieval, which can be extended towards a number of directions. For example, a new type of media can be easily integrated into the model as long as its primitive features are specified. Moreover, a new class of knowledge (on the relevance between media objects) that is orthogonal with the existing knowledge can be introduced into the LGM as a new layer with only minor adjustment of the link analysis algorithms.

Due to the generality and extensibility of the LGM, many potential applications can be implemented based on it. We identify some of them as our future works:

- **Navigation**. The LGM provides abundant links through which the user can traverse from one object to its related objects. Therefore, it supports a natural navigation scenario: when a user is visiting (viewing) a media object, the system recommends him with the objects linked with it in the LGM, ranked according to the weights and types of links, from which he can select the next object to navigate.

- **Clustering**. Clustering multi-modality objects into semantically meaningful groups is also an important and challenging task, which requires a similarity function (between media objects) as well as a clustering method. Our LGM provides knowledgeable links, based on which different similarity functions can be easily formulated. Meanwhile, many clustering methods have been proposed, such as the simulated and deterministic annealing algorithm [11]. Moreover, our model inherently allows the clustering of multi-modality objects, rather than single-modality objects that most existing classification approaches deal with.

- **Personalized retrieval**. The user layer of the LGM characterizes the knowledge obtained from the behaviors of the whole population of users, and allows a query from a single user to benefit from such common knowledge. However, users also have personal interests and preferences that vary from one user to another. To provide personalized retrieval service, a mechanism need to be developed to model the user preferences and adapt the retrieval results towards such preferences. The 2-leveled "user profiling" mechanism proposed by us in [16] provides a viable solution in this regard.

7. ACKNOWLEDGMENTS

The work described in this paper was supported, substantially, by a grant from CityU (Project No. 7100196), partially by a grant from the Research Grants Council of the Hong Kong Special Administrative Region, China [Project No. CityU 1119/99E], and partially by a grant from the Doctorate Research Foundation of the State Education Commission of China.

8. REFERENCES

[1] Benitez, A. B., Smith, J. R. and Chang, S. F. "MediaNet: A Multimedia Information Network for Knowledge Representation". In Proc. of the SPIE 2000 Conference on Internet Multimedia Management Systems, vol.4210, 2000.

[2] Bharat, K. and Henzinger, M. R., "Improved Algorithm for Topic Distilling in Hyperlinked Environments". In Proc. of the 21st Int. ACM SIGIR Conf. on Research and Development in Information Retrieval, pp. 104-111, 1998.

[3] Brin, S. and Page, L., "The Anatomy of a Large-Scale Hypertextual Web Search Engine." In Proc. of the 7th Int. World Wide Web Conf, pp. 107-117, 1998.

[4] Chang, S. F., Chen, W., Meng, H. J., Sundaram, H. and Zhong, D., "VideoQ: An Automated Content Based Video Search System Using Visual Cues". In Proc. of ACM Multimedia, pp. 313-324, 1997.

[5] Chen, W. and Chang, S. F. "VISMAP: An Interactive Image/Video Retrieval System Using Visualization and Concept Maps", In Proc. of Int. Conf. on Image Processing (ICIP), Greece, October 2001.

[6] Dean, J. and Henzinger, M. R., "Finding Related Pages on the Web." In Proc. of the 8th Int. World Wide Web Conf. pp. 389-401, 1999.

[7] Flickner, M., Sawhney, H., Niblack, W. and Ashley, J., "Query by image and video content: The QBIC system." IEEE Computer, pp. 23-32, 1995.

[8] Gibson, D., Kleinberg, J. M., and Paghavan, P., "Inferring Web Communities from Link Topology." In Proc. of the 9th Conf. on Hypertext and Hypermedia, pp.225-234, 1998.

[9] Google Search Engine. http://www.google.com.

[10] Henzinger, M. R., Heydon, A., Mitzenmacher, M. and Najork, M., "Measuring Index Quality using Random Walks on the Web". In Proc. of the 8th Int. World Wide Web Conf. pp. 213-225, 1999.

[11] Hofmann, T. and Buhmann, J. M., "Pairwise Data Clustering by Deterministic Annealing", in IEEE Trans. on Pattern Analysis and Machine Intelligence, 19(1): 1-14, 1997.

[12] Huang, T. S., Mehrotra, S., and Ramchandran, K., "Multimedia analysis and retrieval system (MARS) project," In Proc of 33rd Annual Clinic on Library Application of Data Processing-Digital Image Access and Retrieval, 1996.

[13] Kleinberg, J. M., "Authoritative Sources in a Hyperlinked Environment." In Proc. of ACM-SIAM Symposium on Discrete Algorithms, pp. 668-677, 1998.

[14] Kumar, R., Raghavan, P., Pajagopalan, S., and Tomkins, A., "Trawling the Web for Emerging Cyber-communities". In Proc. of the 8th Int. World Wide Web Conf. pp. 403-415, 1999.

[15] Lempel, R. and Soffer, A., "PicASHOW: Pictorial Authority Search by Hyperlinks on the Web." In Proc. 10th Int. World Wide Web Conf., pp. 438-448, 2001.

[16] Li, Q., Yang, J., and Zhuang, Y. T., "Web-based Multimedia Retrieval: Balancing out between Common Knowledge and Personalized Views". In Proc. of 2nd Int. Conf. on Web Information System and Engineering, pp. 100-109, 2001.

[17] Lu, Y., Hu, C. H., Zhu, X. Q., Zhang, H. J. and Yang, Q. "A Unified Framework for Semantics and Feature Based Relevance Feedback in Image Retrieval Systems". In Proc. of ACM Multimedia, pp. 31- 38, 2000.

[18] Pirolli, P., Pitkow, J., and Rao, R., "Silk from a Sow's Ear: Extracting Usable Structure from the Web." In Proc. ACM SIGCHI Conf. on Human Factors in Computing Systems, pp. 383-390, 1997.

[19] Rafiei, D. and Mendelzon, A. O., "What is this Page Known for? Computing Web Page Reputations." In Proc. of Int. World Wide Web Conf. pp. 823-835, 2000.

[20] Smith, J. R. and Chang, S. F., "VisualSEEk: a fully automated content-based image query system," in Proc. of ACM Multimedia 96, pp. 87-98, 1996.

[21] Smith, J. R. and Chang, S. F., "Visually Searching the Web for Content." IEEE Multimedia Magazine, 4(3): 12-20, 1997.

[22] Tansley, R., "The Multimedia Thesaurus: An Aid for Multimedia Information Retrieval and Navigation", Master Thesis, Computer Science, University of Southampton, UK, 1998.

[23] Yang, J., Zhuang, Y. T., Li, Q., "Search for Multi-Modality Data in Digital Libraries", in Proc. of 2nd IEEE Pacific-Rim Conference on Multimedia, pp. 482-489, China, 2001.