Unifying Keywords and Visual Contents in Image Retrieval

Xiang Sean Zhou and Thomas S. Huang
University of Illinois at Urbana Champaign

We're interested in using keywords and visual content together in image retrieval. We used a seamless joint querying and relevance feedback scheme based on keywords and low-level visual content, incorporating keyword similarities. We developed an algorithm for a learned word similarity matrix and conducted experiments that validated our approach.

As an interdisciplinary research field, multimedia information retrieval addresses and explores both text-based indexing and processing issues and multidimensional signal/information processing techniques.

The performance of a content-based image retrieval (CBIR) system is inherently constrained by low-level features, and it can't give satisfactory retrieval results in many cases, especially when users' high-level concepts aren't easily expressed by low-level features. (Please see some of the other resources[1,2] we've cited for state-of-the-art attempts in associating low-level features with high-level semantic concepts and their limits.) Therefore, for real-world applications, CBIR systems should feature textual annotations to improve the retrieval performance.

In this article we explore the unification of keywords and feature contents for image retrieval. We propose a seamless joint querying and relevance feedback scheme based on both keywords and low-level visual contents incorporating keyword similarities. We propose an algorithm for the learning of the word similarity matrix (we use word, term, or concept similarity matrix interchangeably throughout this article) during user interaction, namely word association via relevance feedback (WARF). We can apply this learned similarity matrix, specific to the data set and the users, for keyword semantic grouping, thesaurus construction, and soft query expansion during intelligent image retrieval.

Integrating keywords and contents

In the area of CBIR, despite advances in feature selection algorithms and matching and retrieval techniques, current systems still have major difficulties relating low-level features to high-level semantics. From our extensive experiments on CBIR systems using features such as color, texture, structure, spatial layout, and relevance feedback from users, it's apparent that low-level contents often don't describe the high-level semantic concepts in users' minds. This is one of the major burdens in implementing a CBIR system for practical image retrieval applications. To overcome this burden, we should unify text-based retrieval with content-based retrieval. Actually, most of the online commercial image databases are annotated with keywords and descriptions or organized into categories. Some researchers have addressed this problem from various standpoints,[3–5] but no one has focused on the issues of online learning of term similarity matrix and word grouping for intelligent query expansion, which we emphasize in this article.

Keywords have more direct mapping toward high-level semantics than low-level visual features such as the color histogram or texture co-occurrence features, but the mapping isn't one-to-one because of the context-dependent interpretation of words and/or the use of synonyms. The system needs a thesaurus during the retrieval process, otherwise the keyword-based retrieval will be limited. This process depends on the consistency of the annotation, the consistency between the user and the annotation, and even the consistency of users at different times. We can use existing general-purpose thesauri in the system, but these thesauri may contain too much redundant information yet lack relevant information to the specific data set or to users' preferences. This brings us to the issue of automatic thesaurus construction from an image database.

In document processing literature, there's been extensive research on how to automatically construct thesauri, but all of these are based on the statistical analysis on term occurrences and co-occurrences in a particular document collection.[6,7] These techniques, relying mostly on the co-occurrences of related terms within one doc-

1070-986X/02/$17.00 © 2002 IEEE

ument, don't apply to image databases, where the annotation is usually concise. In other words, semantically similar terms don't occur in the annotation of the same image. For example, assuming you have a cat named "Socks," you may have annotated one photo as "Socks playing with its tail" and another as "Our cat on the sofa." For the system to figure out that Socks is a cat, one possible way is to look at annotations across images that we know are somehow related semantically. One way of obtaining semantically related images is to look at the relevant images fed back by users during an interactive image-retrieval process.[8]

The proposed method

We assume that some of the images in the database have textual annotations in terms of short phrases or keywords. These can come from pattern recognition,[1] automatic speech recognition, keywords spotting from text (such as surrounding HTML text on Web pages), manual annotation, and so forth.

We propose an algorithm for semantic grouping of keywords based on user relevance feedback during the retrieval process. The result facilitates the unification of keywords and contents in a flexible and meaningful way. During each user retrieval and feedback process, the algorithm will dynamically update the weights in a semantic network consisting of the keywords in the database. This algorithm runs automatically in the background with little computational overhead.

After users have queried the database, the output of the algorithm (such as the weights between pairs of terms) will correspond to either the similarity of the two terms or the estimated probability for users to request these two terms together in one query. By using Hopfield network or clique detection, we can further group terms into semantic classes, which can assist future retrieval processes.

In addition, since the algorithm extracts this knowledge from the user feedback, we can also regard the term *association information* as the users' search habits or preferences. Therefore, this real-time thesaurus construction algorithm based on user feedback will provide a practical way not only for grouping keywords semantically but also learning user preferences.

Background and assumptions

Research in document processing literature provided various methods for knowledge discovery. Many of these approaches represent the knowledge using a semantic net, where the nodes of the network represent different types of concepts and the weighted links among the nodes indicate the relevance among the concepts.[6,7] One form of weight computation is as follows:

$$\text{Weight}\ (T_j, T_k) = \frac{\sum_{i=1}^{n} d_{ijk}}{\sum_{i=1}^{n} d_{ij}} \qquad (1)$$

$$\text{Weight}\ (T_k, T_j) = \frac{\sum_{i=1}^{n} d_{ijk}}{\sum_{i=1}^{n} d_{ik}} \qquad (2)$$

where d_{ij} is a boolean variable with values 1 or 0, indicating whether term T_j is in document i; d_{ijk} indicates whether terms k and j are in document i. Most of the weight computation techniques build on the concepts of term, document, and inverse document frequency in a particular document collection.

In image databases, it's inadequate to directly adopt these co-occurrence-based estimation methods because of the lack of co-occurrence of semantically similar terms in the annotation of a single image. Therefore, we must rely on a group of images—such as the set of feedback images from users during the retrieval process—to estimate the relevance between keywords or terms for the automatic construction of thesauri. Nevertheless, we can regard the proposed algorithm as a natural extension of the pseudoclassification techniques in the text-processing domain[7] into the content-based image retrieval domain, with differences not only in the application domains but also in terms of how the relevance feedback is processed and whether it's a dynamic online process.

The proposed algorithm jointly considers the relevant and irrelevant images in a computationally efficient way (instead of in an iterative way,[7] which can be computationally expensive). In addition, the weight adaptation is in real time and dynamically follows users' retrieval preferences. It isn't a once-for-all process as in the case of pseudoclassification techniques in the document retrieval domain. The effectiveness of these techniques in the document retrieval domain is usually questionable once outside the special cases in which they're generated.[7] It takes time to read through a collection of documents, but images reveal their contents to users instantly.

To use relevance feedback to facilitate the thesaurus generation, we assume that the low-level features can represent the high-level semantics in a reasonable if not perfect way. That is, a subset of similar images in terms of semantics should appear in the top returns with certain (nonzero) probability. For low-level contents, the widely used features include color, texture,[9,10] shape, structure,[11] and so forth. In our experiments, we used a feature space of 37 dimensions with nine color moments, 10 wavelet moments,[9] and 18 edge-based structure features[11] (see Figure 1).

Joint querying and relevance feedback

To combine the use of low-level visual featuers with keywords, we convert keyword annotations for each image into a vector, with components v_{ij} indicating the appearance or probability of keyword j in image i. When $v_{ij} \in \{0,1\}$, we say it's a *hard vector representation*; when $v_{ij} \in [0,1]$, we say it's a *soft vector representation*.

Soft vector representation of keywords

We use a soft vector representation for keyword annotations. We assume that a keyword similarity matrix $\{S_{ij}, i, j = 1, \ldots, M\}$ is available, with a total of M keywords, where $S_{ii} = 1$, and $S_{ij} \leq 1$. This matrix can be symmetric or asymmetric, depending on the application and usage. Let l_{ij} be the jth component of an M-dimensional hard vector representing the keyword list for the ith image, such as $l_{ij} = 1$ if the $i$thimage has the jth keyword in its annotation, and zero otherwise. Then we define the soft vector representation for the ith image as

$$v_{ij} = \max_{k \in \{k \mid l_{ik} = 1\}} S_{kj} \qquad (3)$$

Equation 3 says that term j can represent image i with a probability no higher than its association with the most relevant term in i's annotation.

A subtly different choice could be

$$v_{ij} = \min \{\psi, \Sigma_{k\in\{k \mid l_{ik} = 1\}} S_{kj} \} \qquad (4)$$

where ψ is a value close to 1. The rationale here is that if several terms imply a new term j—although by weak association, but with a sum near (or over) 1—then the new term j is strongly implied in image i.

We could argue that an alternative way to model keyword relationships is to perform query expansion on the user query on the fly instead of keyword expansion for each image. We believe this approach isn't as reasonable as ours because, for example, the user queries for "car"—with the possible expansion of the keyword "car" into "Ford Taurus," "Toyota Camry," and so on—is enormous, while the database may only have "Taurus" in it. This is especially true for databases with an uncontrolled vocabulary.

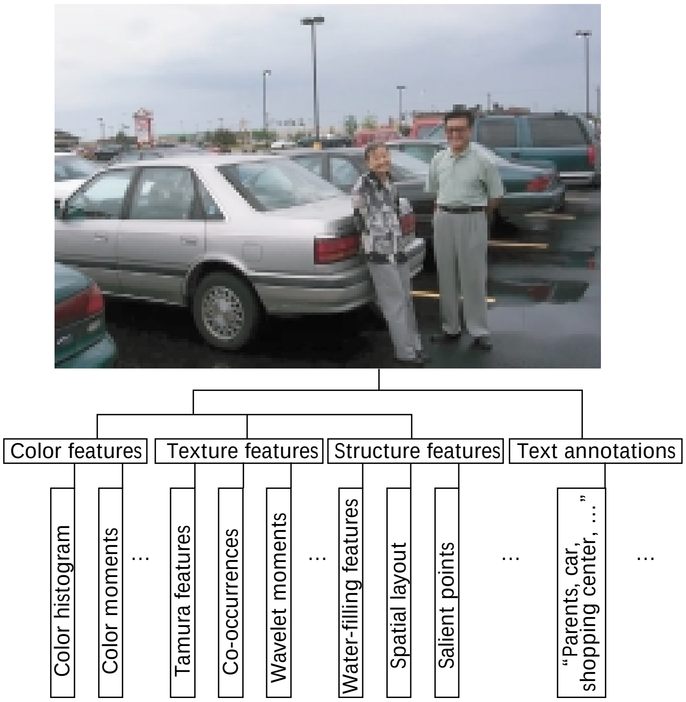


Figure 1. An image object with both content and keyword descriptors in our system.

This soft representation has the ability to model synonyms or a keyword set of hierarchical structures. For example, the system can link the current user-specified query keyword ("car") to a broader or a more specific one ("Ford") in the database automatically.

Another way of modeling relationships among keywords is to apply linear multidimensional scaling (MDS)[12] on the word similarity matrix to arrive at a low-dimensional space or use nonlinear techniques such as locally linear embedding to construct such a space.[13] In these approaches, a point represents each word, and it preserves their mutual distances as much as possible. These schemes have the advantage of compact feature representation with a small dimensional feature space, but they lose the semantic meaning of the axes. More importantly, they have poor scalability. For example, with new keywords the MDS procedure has to be repeated and the new lower-dimensional embed-

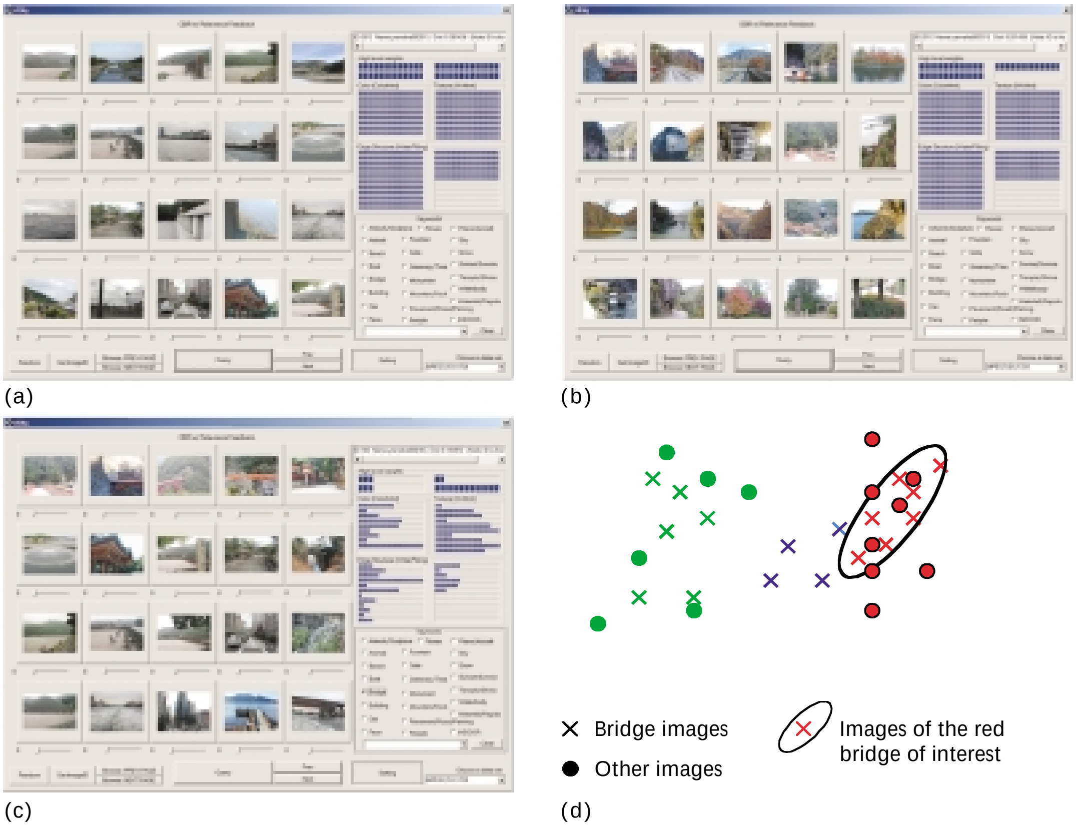


Figure 2. We assume users are looking for a specific red bridge, while the system only has the keyword "bridge." (a) Using the keyword "bridge," the system returns many different bridges, not necessarily what the user is looking for. (b) When the system uses a red bridge with only feature contents, we get images that are visually similar but semantic nonsense. (c) Combining the keyword and examples, the system returns all four occurrences of the red bridge in the database. (d) An illustration of the distribution of images in the feature space.

ding can differ from the previous one, even with a relatively small number of new data. However, in our scheme, the insertion of new words is a simple linear incremental process.

Joint global similarity search

For traditional database approaches, the keyword-based search gives a binary decision—and a content-based search becomes simply a similarity search within a subset of the database with certain keyword constrains. We propose a joint querying and relevance feedback scheme based on a joint global similarity search on both keywords and feature contents.

With the soft vector representation, we convert the keyword list of an image into a vector, which becomes comparable to a feature vector extracted directly from the image based on contents such as color, texture, and edge or structure (see Figure 1). The algorithm supports various query modes. For example, when users select the *j*th keyword, it sets the *j*th weight to be large while setting weights on other keyword components to be zeroes (that is, "don't care"). For feature content vectors, if users don't specify any example query images, the algorithm uses a random vector and sets the feature weights as ones (Figure 2a). The randomness provides users with a random browse option for wandering around different parts of the feature space. When users also select the *i*th image as a query in addition to the keyword selection, the query uses its content vector while it resets its keyword vector to reflect the users' keyword selections—this is the joint keyword and example query with one example, such as "give me 'bridge' images that look like this one" (Figure 2). When users select multiple examples, the algorithm applies the relevance feedback technique[14,15] to learn an optimal transformation in the augmented feature space for both keywords and visual contents so that it can further improve retrieval results.

This learning process also involves the keyword vector. So, for example, if all the positive examples have annotations of "Taurus," "Camry," and so on, with a proper concept similarity matrix (we discuss this matrix in the "Learning Semantic Relations between Keywords" section) the system can figure out that this user is looking for car images. Then it will return images with annotations related to cars, even though they can be visually different. We can't achieve this by relevance feedback in the visual feature subspace alone. In Figure 2c, the four bars under the title of "high-level weights" correspond to color, texture, structure, and keywords. The fourth bar (the longest) indicates that the system has learned from multiple examples that the annotation is the most important descriptor to use for this task.

In summary, the proposed scheme treats the keyword annotations in a soft way rather than the rigid treatment in a traditional text-driven database and unifies the keyword and content vectors in a transparent way for the user to facilitate joint querying and relevance feedback. Query expansion is a natural by-product of the relevance feedback process. The prerequisite for this scheme is that it needs a term similarity matrix.

Learning semantic relations between keywords

For an image database such as a personal digital photo album, users can add text annotations either by hand or by an automatic speech recognizer. Alternatively, for a dynamic image database on the World Wide Web, keywords can be extracted from the surrounding or related text. Then keyword-based retrieval is possible.

However, problems arise when different yet semantically similar keywords are assigned (by different people or by the same person but at different times) to similar images or when the user fails to use the exact wording as the one used for the images in the database. Obviously, we need a thesaurus to resolve term association problems.

One option is to use standalone thesauri. However, the major problem is that they usually contain too much redundant yet imprecise information. More importantly, data-dependent information, new knowledge, or user-specific knowledge in general doesn't exist in any general-purpose thesaurus. For example, an evolving Web-based news image database should provide a strong semantic link between "anthrax" and "terrorism." In this case, hand annotation or online learning is necessary because this is a piece of new information that isn't encoded in any existing thesauri.

Therefore, we propose automatic thesaurus construction—or more precisely, word association—based on user relevance feedback during the retrieval process. The assumption is that some images in the databases have keyword annotations. During the browsing or content-based retrieval process, the system can take relevance feedback from the user, which is essentially the set of images that users regard as relevant out of all the images. Based on this information and the annotations for both the relevant and irrelevant images the system retrieves, the system can apply an updating formula as described in the following section to increment the similarity or closeness among all the keywords assigned to the current images.

WARF

The *relevant set* of images generated from a query contains images that users want. If a term only appeared in the annotations for the images in the relevant set, it's called a *relevant term*. The number of occurrences of a relevant term *i* in the relevant set is called *relevant term frequency*, denoted as f_i. The number of co-occurrences of two relevant terms *i* and *j* in the same image is denoted by c_{ij}. The relevance of term *i* and *j*, S_{ij}, is then updated as

$$S_{ij} = S_{ij}, + \max(f_i, f_j) \times (\min(f_i, f_i) - c_{ij} \tag{5}$$

This formula for updating word association through relevance feedback, or the WARF formula, is executed after users provide feedback for retrieval results that have more than one relevant image. Note that the WARF formula implies that if two terms appeared in the annotations for one image, we can't get any information out of it. For example, a relevant image annotated as "car, house, tree ..." provides us with little information about how the concepts of car and tree relate to each other. It's a valid argument that the term co-occurrence sometimes provides us with useful information because certain things tend to appear together in one image, such as a beach and an ocean; while others are rarely together, such as an elephant and a polar bear. However,

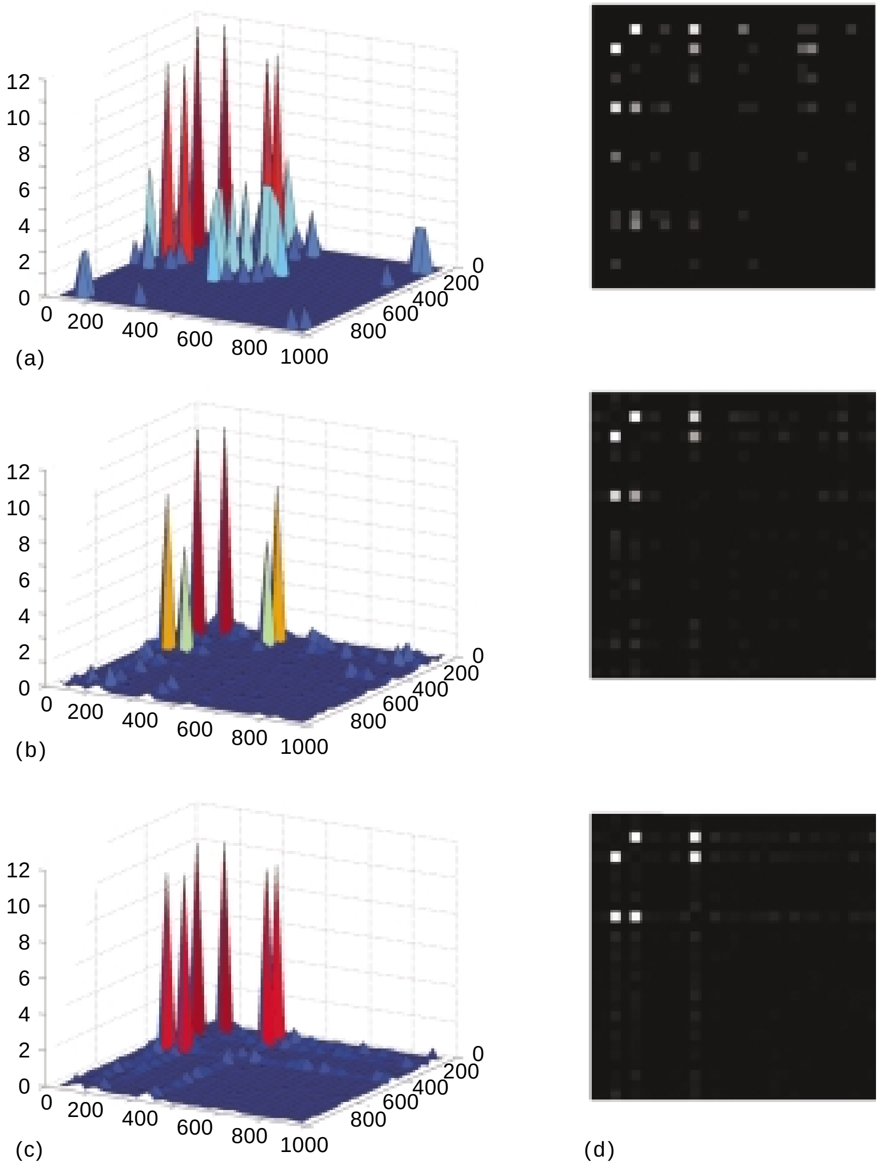


Figure 3. Concept similarity matrix with 30 words in the vocabulary, 5,000 images in the database, and up to three keywords per image. Concept similarity matrix after (a) five, (b) 20, and (c) 80 rounds of training. (d) The corresponding 2D views for the three cases.

we believe that such co-occurrence information is less important and consistent in the image annotation domain than in the document domain. In addition, this treatment can eliminate false updates, especially when the number of annotations per image is large relative to the vocabulary size.

For the increment term in Equation 5, the rationale for using multiplication instead of, say, addition is that the two terms are more likely to be relevant when they appear in an equal number of relevance images (such as five images for each term). Whereas, for the case where one term appears in nine images while the other term appears only in one image, the relationship between the two terms is more likely to be coincidental.

Simulated experiments and validation

Although we could perform a rigorous statistical analysis to validate the WARF formula, it could be very involved. Instead, we used a set of simulated experiments to show the WARF formula's effectiveness. In our user model we assumed that the simulated user is interested in a class of images that can be characterized by *k* words. If a displayed image has one of the *k* words in its annotation, the user model will mark it as relevant. We assumed that the database contained 5,000 images and each image was also annotated (randomly assigned by the machine) by up to *m* words from a vocabulary of *M* words. To imitate the relevance feedback process, the system randomly selected 20 images. If it detected more than one relevant image using the user model, the system counted it as a training session. Then it executed the WARF formula to increment the concept similarity matrix. The system performed multiple training sessions to reach a statistically valid estimation of the matrix.

We first tested the cases for M = 30 and m = 3. Figure 3 shows the concept similarity matrix in 3D and 2D views after different numbers of training sessions. For this test, users are interested in the concept "car" (row or column 3), "truck" (5), or "motorcycle" (11). The 3D meshes or images depict matrices of weights between all pairs of concepts. The element in the matrix m_{ij} is the relevance measure between concepts *i* and *j*. The peaks or bright dots indicate the higher weights between concepts 3 and 5, 3 and 11, and 5 and 11. We assume the relevance measure is symmetric, so the system estimates only half of the matrix, and it's added to its transpose to generate the symmetric matrices shown in Figure 3. The weight to the node itself (the diag-

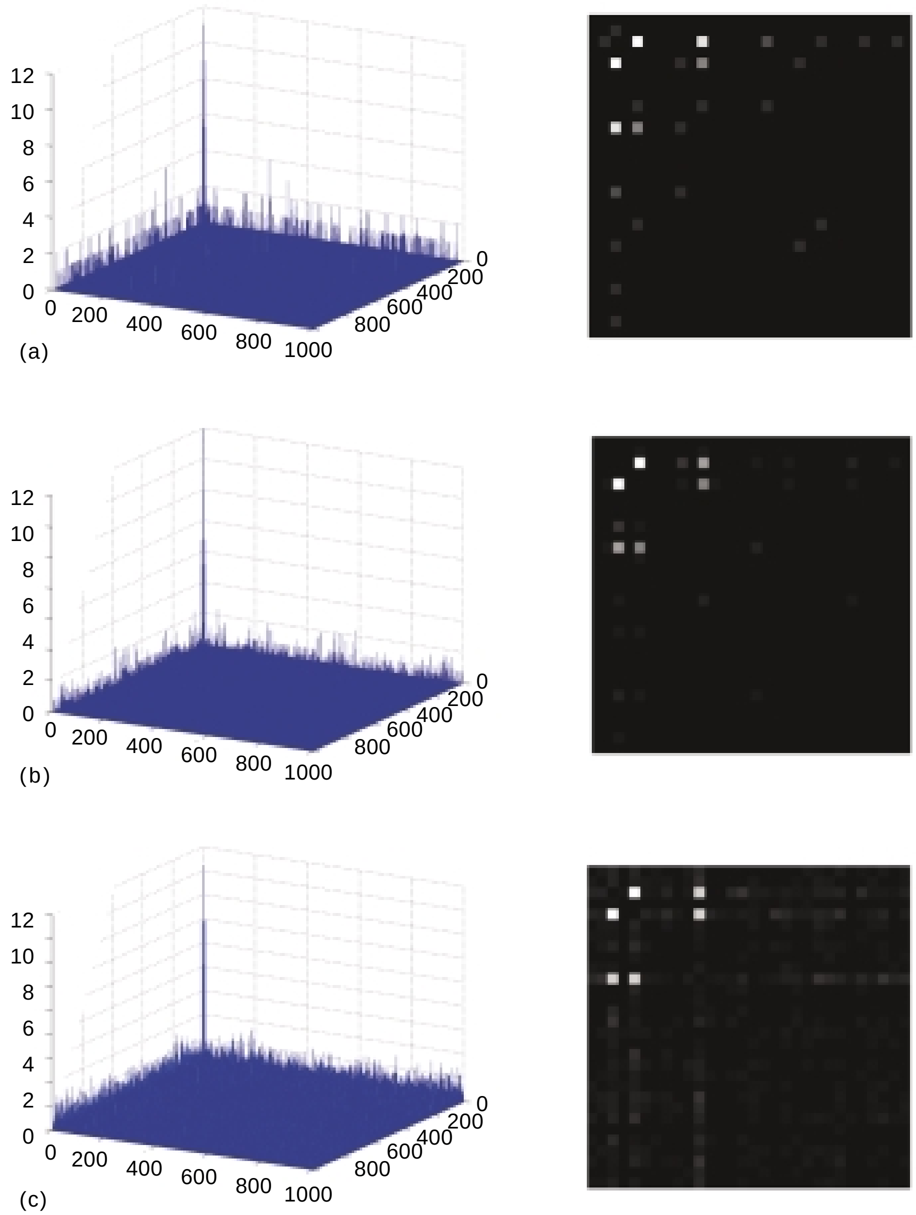


Figure 4. Concept similarity matrix with 1,000 words in the vocabulary and 5,000 images in the database. (a) Up to five keywords per image and 30 rounds of training, (b) up to five keywords per image and 100 rounds of training, and (c) up to 80 keywords per image and 30 rounds of training. The 2D views only show the first 30 rows of the first 30 columns for clarity.

onal elements) isn't updated for now. We can see that the similarity between concepts 3, 5, and 11 begins to emerge after only five training sessions, it clearly stands out within 20 sessions, and it stabilizes within 80 sessions.

To test the scalability of the proposed formula with respect to *M* and *m*, we further tested it using a vocabulary of 1,000 words and up to 80 keywords per image. Figure 4 shows the results. Noise is evident (but not damaging) when $m = 5$ and only 30 rounds of training is performed. Additional training sessions (100 rounds) apparently increased the signal-to-noise ratio—if we regard the ground truth word associations as the signals. The increase of *m* to 80—that is, up to 80 words per image—clearly didn't increase the possible confusion between the ground truth and noise. It seems that the signal and the noise grow

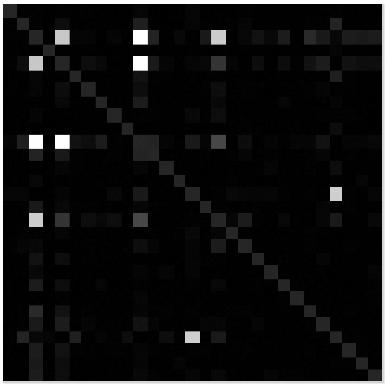


Figure 5. Concept similarity matrix for three users, or one user with three retrieval tasks (M = 30, m = 3).

Table 1. Hopfield activation results with limited iterations.

Concept	Active Nodes
1	3 5 11
2	3 5 11 17
3 (car)	3 5 11 17
4	3 5 11
5 (truck)	3 5 11 17
6	3 5 11
7	3 5 11 17
8	3 5 11 17
9	3 5 11
10	3 5 11
11 (motorcycle)	3 5 11 17
12	3 5 11 17
13	3 5 11
14	3 5 11
15 (tiger)	3 5 11 15 17 26
16	3 5 11
17 (van)	3 5 11 17
18	3 5 11
19	3 5 11 17
20	3 5 11
21	3 5 11
22	3 5 11 17
23	3 5 11
24	3 5 11 17
25	3 5 11 17
26 (lion)	3 5 11 15 17 26
27	3 5 11 17
28	3 5 11
29	3 5 11 17
30	3 5 11

at a comparable rate (note the scale on the vertical axis).

We also tested the case of multiple users or complicated user behaviors. This time there's a probability of one in three that users will mark "car" (3), "truck" (5), and "motorcycle" (11) images as relevant whenever they see them. There's also a one in three probability that users will mark "car" (3) and "van" (17) as they appear on the screen or that users are searching for "tiger" (15) and "lion" (26). Figure 5 shows the concept similarity matrix after 80 rounds of feedback. The bright dots clearly reveal the relevant concept pairs.

Semantic grouping of keywords

From the concept similarity matrix, we can either implement a Hopfield network or use a heuristic clique detection algorithm to obtain the semantic classes. We compare the two methods using the simulation result as the input data in Figure 5.

To use the parallel activation scheme of the Hopfield network, we treat the 30 concepts as the nodes in the network and assign the weights m_{ij} shown in Figure 5 as the synaptic weights between nodes. During the iteration, the output at node i at time $t + 1$ is

$$O_i(t+1) = \frac{1}{1+\exp\left[\frac{s_i - in_i}{s}\right]} \qquad (6)$$

where

$$in_i = \sum_{j=1}^{30} o_i m_{ij},$$

s_i (= 0.3) is a bias, and s (= 0.1) controls the Sigmoid function's shape. The convergence criterion is that the L_1 distance between two adjacent output vectors is less than a threshold (0.001 in our case).

Table 1 shows the results when we activate the Hopfield net by assigning 1 to each of the 30 nodes and iterating four times for each activation. Otherwise, if we iterate until convergence, the results will be the same for all nodes: {3 5 11 15 17 26}. The reason is that we use symmetric weighting, and the noisy estimation in the concept similarity matrix can spread any activation all over the network.

Table 1 shows that concepts 3, 5, 11, and 17 belong to one node class and 15 and 26 belong to another. Notice that the Hopfield network is suitable for a single-link classification system. That is, within one class, each term is relevant to at least one other term in the same class. In this example, "van" (17) is only relevant to "car" (3) but we classify "van" as a member of a bigger class. The drawback is obvious. If a user is only interested in "car" or "van," the system has a hard time isolating these two as separate classes (as the retrieval processes show).

On the other hand, a heuristic clique detection algorithm can perform complete-link classification where each term is relevant to all other terms in the same class. In fact, this is the definition of a clique in graph theory. This clique detection algorithm results in three classes: {3, 5, 11}, {3, 17}, and {15, 26}. With complete-link classification, when users search for "Cherokee," the system will have the intelligence to either learn from the context of the user actions whether the interest is really in {Cherokee, Jeep, Sport Utility Vehicle} or in {Cherokee, Indian}. At the very least, the system can then point out possible confusions and ask the user before returning many irrelevant images.

An intelligent retrieval system

With low-level features, textual annotations, and an automatically generated thesaurus, we can build a hybrid intelligent image-retrieval system to provide convenient retrieval for the user. With hybrid image objects integrating both low-level features and keyword annotations (Figure 1), the system can interact intelligently with the user. Some features of an intelligent system might include online learning of user preferences or semantic grouping of keywords, intelligent dialogue with users to understand the query and guide the retrieval process, an online feature selection (weighting) from relevance feedback, and so forth (see Figure 6).

Implementation issues

It's unreasonable to completely abandon the use of a static thesaurus. The right choice will be a

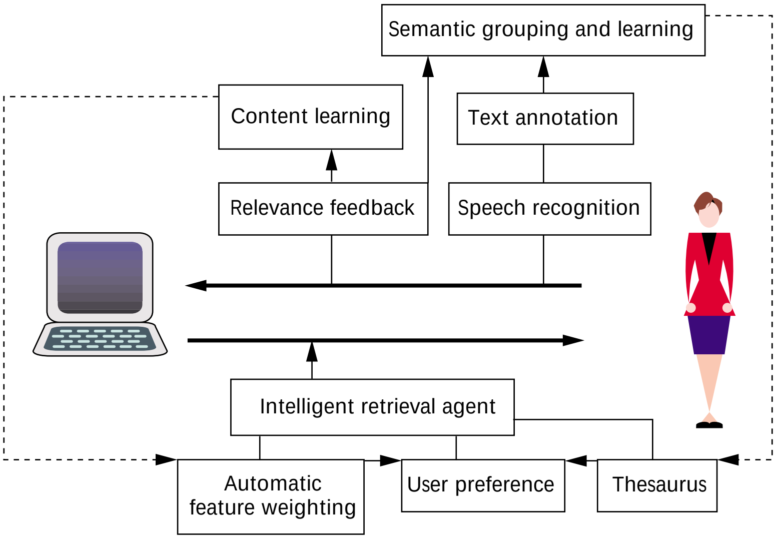


Figure 6. An intelligent retrieval system that can learn from the user interactions and understand the semantics of words and contents.

hybrid strategy—we could start with a reasonably good static thesaurus and add real-time learning results to it on the fly. This static thesaurus can be the result of any offline thesaurus construction algorithms based on a set of documents or an encyclopedia.

However, we need to normalize the concept similarity matrix generated from the WARF formula before we can use it to update the soft vectors of the image annotations. In general, we want the diagonal elements to be 1 and any off-diagonal elements to be less than 1. We propose the following normalization strategy: divide all the off-diagonal elements by a factor D defined as

$$D = \frac{S^{\max}}{1 - e^{-M/\mu}} \qquad (7)$$

where $S^{\max}$ is the maximal element of the matrix, M is the size of the vocabulary or the number of rows (or columns) of the matrix, and $\mu > 0$ is a scaling factor. After the normalization of the off-diagonal elements, the system sets all diagonal elements to 1.

This normalizing strategy reflects the strategy that when the vocabulary is small, the distance between any two terms will be larger while for a large vocabulary, the inclusion of synonyms is more likely. For example, if we set $\mu = 100$ for $M = 30$, the largest off-diagonal element will be normalized to 0.26. For $M = 1{,}000$, the largest off-diagonal element will be normalized to 0.99995, which means that the most similar pair of words out of 1,000 words is probably a synonym pair.

We can update the word similarity matrix independently for different users to facilitate user profiling and preference modeling. If we use one matrix for one database, it can serve as a knowledge discovery tool across users.

Example working scenarios

With the proposed framework, we expect our system to be capable of dealing with the following example scenarios:

- When a user searches for “Ford” and “Toyota,” the system should automatically infer that “car” is somewhat related to the user’s interest. This is sometimes referred to as automatic query expansion and it requires a thesaurus or a knowledge base as the underlying support. The system can learn this kind of general knowledge either online or offline using a static thesaurus.

- A user calls his cat “Socks” and annotates some of its photos with “Socks.” The user should be able to retrieve these photos when searching for “cat” or “pets.” The system must glean these kinds of domain- or user-specific synonyms or knowledge from the user’s interactions.

- When users search for images related to “American Indians” using an image annotated with “Cherokee,” the system shouldn’t retrieve an SUV. This is the case where polysemous words cause confusion and degradation in retrieval performance. Our word classification algorithm can detect the two classes—namely, “Cherokee” as in the “Cherokee Indian” and “Cherokee” as in the “Grand Cherokee, the SUV.” The system either can infer the correct class by looking at contextual information or, if this fails, ask users to clarify their intentions.

- Users can combine keywords and examples in any combination intuitively, and the relevance feedback module works in the joint keyword and content space. The system can learn whether the soft annotation vector is better at capturing users’ query concepts or the targeted images’ low-level features are more expressive.

Conclusions

Joint modeling of textual and visual information can be effective or beneficial only when high-level concepts and low-level visual features are somewhat dependent. Fortunately, current research in content analysis provides us with features that can facilitate high-level understanding of objects or semantics in many cases. However, a gap still exists between the two. Future research shall include the search for expressive low-level features, the use of intermediate features, semantics-guided segmentation and spatial relationship modeling, and the use of machine learning techniques to unify low-level features and keywords. MM

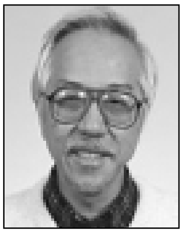
Acknowledgments

This work was supported in part by a US National Science Foundation grant CDA 96-24396.

References

1. M.R. Naphade et al., “Probabilistic Multimedia Objects Multijects: A Novel Approach to Indexing and Retrieval in Multimedia Systems,” *Int’l Conf. Image Proc.*, vol. 3, IEEE CS Press, Los Alamitos, Calif., Oct. 1998, pp. 536-540.
2. Y. Xu, E. Saber, and A.M. Tekalp, “Hierarchical Content Description and Object Formation by Learning,” *Proc. IEEE Workshop on Content-Based Access of Image and Video Libraries*, IEEE CS Press, Los Alamitos, Calif., June 1999, pp. 84-88.
3. K. Barnard and D. Forsyth, “Learning the Semantics of Words and Pictures,” *Proc. Int’l Conf. Computer Vision*, vol. 2, IEEE CS Press, Los Alamitos, Calif., 2001, pp. 408-415.
4. Y. Lu et al., “A Unified Framework for Semantics and Feature Based Relevance Feedback in Image Retrieval Systems,” *Proc. ACM Multimedia Conf. 2000*, ACM, New York, 2000.
5. S. Sclaroff et al., “Unifying Textual and Visual Cues for Content-Based Image Retrieval on the World Wide Web,” *Computer Vision and Image Understanding* (CVIU), vol. 75, nos. 1/2, 1999, pp. 86-98.
6. H. Chen et al., “A Parallel Computing Approach to Creating Engineering Concept Spaces for Semantic Retrieval: The Illinois Digital Library Initiative Project,” *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 18, no. 8, Aug. 1996, pp. 771-782.
7. G. Salton, *Automatic Text Processing*, Addison-Wesley, Reading, Mass., 1989.
8. X.S. Zhou and T.S. Huang, “Exploring the Nature and Variants of Relevance Feedback,” *Proc. IEEE Workshop Content-Based Access of Image and Video Libraries*, IEEE CS Press, Los Alamitos, Calif., 2001, pp. 94-101.
9. J.R. Smith and S.F. Chang, “Transform Features for Texture Classification and Discrimination in Large Image Databases,” *Proc. IEEE Int’l Conf. Image Processing*, IEEE CS Press, Los Alamitos, Calif., 1994, pp. 407-411.
10. B.S. Manjunath, “Gabor Wavelet Transform and Application to Problems in Computer Vision,” *Proc. 26th Asilomar Conf. Signals, Systems, and Computers*, 1992, pp. 796-800.
11. X.S. Zhou and T.S. Huang, “Edge-Based Structural Feature for Content-Based Image Retrieval,” *Pattern Recognition Letters*, vol. 22, no. 5, 2001, pp. 457-468.
12. T. Cox and M. Cox, *Multidimensional Scaling*, Chapman & Hall, London, 1994.
13. S. Roweis and L. Saul, “Nonlinear Dimensionality Reduction by Locally Linear Embedding,” *Science*, vol. 290, no. 5500, 22 Dec. 2000, pp. 2323-2326.
14. X.S. Zhou and T.S. Huang, “Small Sample Learning During Multimedia Retrieval Using BiasMap,” *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, vol. 1, IEEE CS Press, Los Alamitos, Calif., 2001, pp. 11-17.
15. X.S. Zhou and T.S. Huang, “Comparing Discriminate Transformations and SVM for Learning During Multimedia Retrieval,” *Proc. ACM Multimedia 2001*, ACM, New York, 2001, pp. 137-146.

Xiang Sean Zhou is currently with the Image Formation and Processing Group at the Beckman Institute for Advanced Science and Technology at the University of Illinois at Urbana Champaign, where he is a research assistant and PhD candidate. His research interests include computer vision, pattern recognition, machine learning, signal and image processing, and multimedia information retrieval. He received his bachelor degrees in electrical engineering and economics and management in 1993 from Tsinghua University, Beijing, China. He also studied economics there in a PhD program for two years before he joined the University of Cincinnati in 1996 and received his MS in electrical and computer engineering in 1998. In 2001, he was awarded the M.E. Van Valkenburg Fellowship Award for demonstrated excellence in research in the areas of circuits, systems, or computers.



Thomas S. Huang joined the University of Illinois at Urbana-Champaign in 1980, where he is now the William L. Everitt Distinguished Professor of Electrical and Computer Engineering. Dr. Huang's professional interests lie in the broad area of information technology, especially the transmission and processing of multidimensional signals. He received his BS in electrical engineering from National Taiwan University, Taipei, China, and his MS and ScD degrees in electrical engineering from the Massachusetts Institute of Technology, Cambridge, Massachusetts. He's a member of the National Academy of Engineering and a fellow of the International Association of Pattern Recognition, the IEEE, and the Optical Society of America. He has received a Guggenheim Fellowship, an A.V. Humboldt Foundation Senior U.S. Scientist Award, and a fellowship from the Japan Association for the Promotion of Science. In 2001, he received the IEEE Jack S. Kilby Medal.

Readers may contact the authors at the Beckman Institute for Advanced Science and Technology, University of Illinois at Urbana Champaign, Urbana, IL 61801, USA, email {xzhou2, huang}@ifp.uiuc.edu.