LiveImage: Organizing Web Images by Relevant Concept

Shuo-Peng Liao*, Pu-Jen Cheng*, Ruey-Cheng Chen*, Lee-Feng Chien*,†
* Inst. of Information Science, Academia Sinica, Taiwan
† Dept. of Information Management, National Taiwan University, Taiwan
{deanliao, pjcheng, cobain, lfchien}@iis.sinica.edu.tw

Abstract

The images retrieved from conventional Web search engines are not normally well organized in a conceptually meaningful way. Users find it difficult to reformulate effective queries through browsing the retrieved images. This paper presents an approach that organizes the retrieved images with relevant concepts, which are generated automatically from a concept pool obtained from various online resource. A preliminary user study shows that LiveImage can enrich image-searching experience by organizing search result according to relevant concepts of a given query.

Keywords: Relevant Concept Acquisition, Image Search, Search Result Clustering, Query Classification, Query

1. Introduction

The increasing demand for high-performance Web image search engines is being driven by the rapid growth in the number of Web users and in the availability of image collections on the Web. The techniques used by most search engines to perform keyword-based search over Web images are similar to those used in textual information retrieval. The images whose textual information — including filenames, captions, HTML `<ALT>` tags, and surrounding texts — match the queries are ranked and returned back to users. However, keyword-based image search has not proved to be successful in every case, especially when users' queries are short.

Currently, popular image search engines, like *Google Images* (`http://images.google.com`), can find numerous images for a short query, e.g., "dog" or "buildings." However, the returned

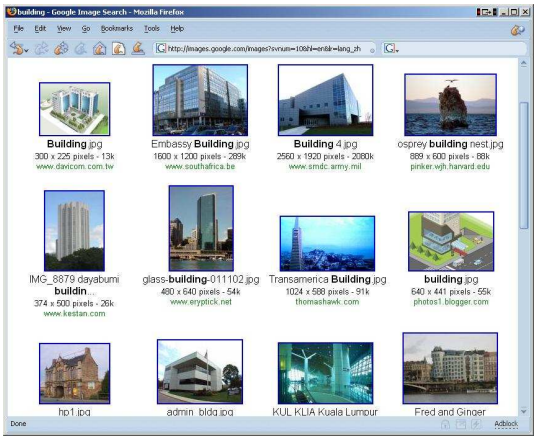


Figure 1: An example of images retrieved from Google Images

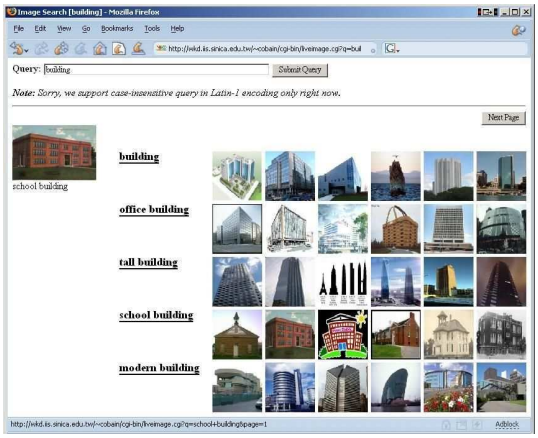
images are not well organized in a conceptually meaningful way; as a result, users may find it difficult to understand the structure of the concepts related to the query. Take a search result of "building" from Google Images as an example which expresses how the retrieved images from it response to the query. The result is shown in Figure 1. They are sorted by the degree of relevance to the query given. In such a short query or a query with many search results, unfortunately, the result ordered in such manner fails to help user find the images they really want. That is, users cannot see the ordering in top–N pages of Web images retrieved from Google Images. In this case, what the series of images means to user is equivalent to an unsorted image collection; thus forces users to browse plenty of disordered images to know if there are images they need.

Furthermore, according to Pu's work [8], a Web image search has more refined queries but also with higher numbers of failed queries. Therefore, assisting users refining their queries becomes an urgent demand for Web image search. However, popular image search engines cannot effectively help users refine their queries. For instance, suppose Alice searches images about building, but what she really want is a building with modern look. At the beginning, Alice does not have the semantic concept, "modern," in her mind. By browsing a lot of images retrieved, Alice can sharpen the concept she want to find. However, it exists a gap between the sharpen idea and the corresponding text query. That is, Alice may find it difficult to refine the query "building" to "modern building," which is what she really looks for. Unfortunately, those images which buildings look "modern" may not have annotations containing the word "modern," and this brings Alice more difficulties to refine the query.

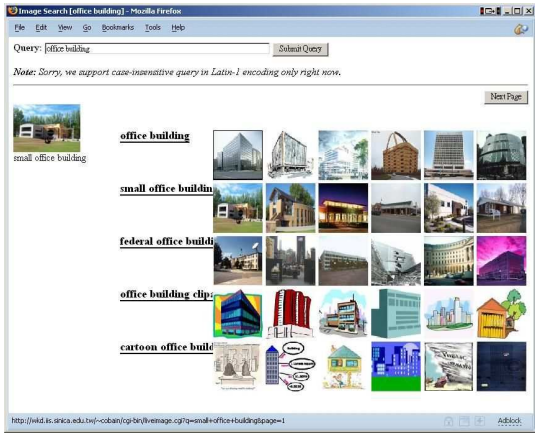
To assist users either efficiently finding desired images from retrieved result or effectively refining the query, this paper presents a new approach for them to find the images they want in a conceptually simple way that organizes retrieved images on the basis of auto-generated relevant concepts. *LiveImage* (`http://wkd.iis.sinica.edu.tw/LiveImage`), a prototype meta search engine for finding conceptually relevant images based on the approach is then implemented to observe how this approach could help user finding images. To contrast with traditional image-search results, the result of querying "building" using LiveImage is shown in Figure 2a. As we can see in this figure, the purpose of the proposed approach is to generate relevant concepts, such as "office building," "tall building," "school building," and "modern building," for the given query "building." The images listed in the retrieved results are organized according to their similarity to the relevant concepts, which makes it easier for users to locate images of interest with the corresponding concept names and relevant images. Besides, LiveImage facilitates users refining their queries by simply clicking the images represented the relevant concept they prefer. To illustrate, Figure 2b shows the images further retrieved by clicking the category of "office building." Some relevant concepts such as "small office building," "federal office building," "office building clips," and "cartoon office building," are generated for the clicked images with the associated relevant images.

To achieve our goal of assisting user searching images, what challenging most is to discover meaningful concepts relevant to a Web image query given. Collecting and organizing such relevant concepts manually is infeasible due to the dynamic nature of the Web environment. Therefore, we are interested in developing an automatic approach that organizes users' query terms into auto-generated concept categories. As users' queries are short and new queries appear all the time, the challenge is how to assign effective concept categories for users' queries and improve their search performance, even if the given queries have never appeared before. The proposed approach is a well-integrated set of innovative techniques, including relevant concept construction, query classification and relevant concept generation. LiveImage, a meta search engine based on the proposed approach is implemented and its pros/cons measured through a preliminary user study. The results show that the performance of Web image retrieval can be effectively improved with the proposed approach.

The rest of the paper is organized as follows. In Section 2, we give an overview of related works. In Section 3, we proposed a new direction of Web image search that facilitates users' searching ex-



(a) By querying "building"



(b) By clicking a image in the category of "office building" in Figure 2a

Figure 2: Images retrieved by LiveImage

perience as well as illustrated the framework of LiveImage that assists users in the way we proposed. Then, in Section 4, we conducted a preliminary user study to analyzing our approaches and system benefits. Finally, the discussion is listed in Section 5 and the conclusions are drawn in Section 6.

2. Related Works

Search results clustering for improving Web search has been investigated in numerous works recently. Most of them were focused on clustering search results for document searching [11, 5, 12]. They aimed to extract clusters from search-result snippets returned by text search engines. Recently, clustering search-result snippets is proven practical for online searching. However, clusters generated based on text-based search-result snippets are inadequate for browsing images as the ways to annotate images are often quite different from the ways to describe documents. Therefore, some works extended the technology to image retrieval. For example, Clough et al. [3] analyzed the co-occurrence between terms associated with image captions and hierarchically organize the terms into clusters. However, many terms in the captions, e.g., personal names and location names, can not well organized into a taxonomy.

Besides, for image-search result arrangement, lots of investigations have been explored to organize images into topic classes based on textural and/or visual information about Web images. Some works applied traditional content-based image retrieval (CBIR) techniques to cluster retrieved images. Park et al. [7] located images using a color feature and clustered the top-ranked images with a hierarchical agglomerative clustering method; then the clusters were ranked according to their similarity to the query-image. Huang et al. [4] proposed an automatic method for hierarchical classification of images via supervised learning. However, the CBIR-based methods face the scalability problem. Moreover, semantically-relevant images may not have similar visual features. Recent research results [9] revealed the arrangement of images by semantic similarity is more useful than by visual similarity.

Furthermore, some works took into account both of the textual and visual information for clustering images into topics. Mukerjea et al. [6] developed an image search engine, called AMORE, that grouped retrieved images by their URLs, which may give an indication of the types of the images it contains, their surrounding texts and their low-level image features such as color and object layout. Barnard and Forsyth [1] presented a statistical model for hierarchically modeling the

statistics of word and feature occurrence/co-occurrence. With the model, image collections were organized by simultaneously integrated semantic (textual) and visual information. Smith and Chang [10] performed a semiautomated classification of Web images into a topic hierarchy based on associated texts and URLs. These works exploit various characteristics of Web images to cluster visually-similar or semantically-relevant images. They require, however, much effort in image processing, link analysis and textual feature extraction, which is not the case in our approach. In this paper, we focus mainly on clustering retrieved images using users' queries obtained from search engine logs.

3. LiveImage: Finding Web Images by Relevant Concepts

3.1 A New Direction of Searching Images on the Web

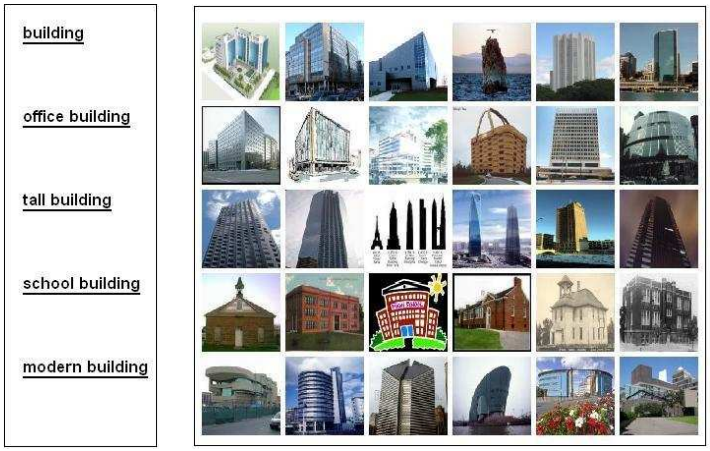
The rapid growth of the number of available images on the Web raises the demand of powerful image search engine. The mainstream image search engines, such as Google Images, can retrieve a lot of images to a given query. For a short and common query, however, millions of retrieved images are difficult and time-consuming for users to browse all of them. To search images for each query, though the corresponding images are sorted according to the degree of relevance to the query and users can focus on top–N images, such image ordering brings users no benefit because of its indistinguishable in the degree of relevance in most related images. To illustrate, we take the query "building" as an example: Google Images retrieved about three millions of images. Suppose we focus on only top–1% of images, however, more than 30,000 images are regarded as highly relevant and such numerous result images are impossible for users to go through.

Facing the result of such disordered images, the need of image organization becomes urgent and important to users searching the Web images. Rather than merely listing images by degree of relevance, we recommend that the retrieved images should be organized by the query's relevant concepts. Take "building" as an instance of a given query, we organize the images by grouping images with the same relevant concept together (in a row of LiveImage), which is shown in Figure 2a.

Here we define the term *concept* and *relevant concept*. For each query, it represents a *concept* users want to find, even for unique queries (specific person, place, or objects) because each retrieved image represents part of the form, property, class, etc., of the query so that all of them form a concept of the corresponding query. And a *relevant concept* is the subset of the concept that describes the form of a query (e.g., "building wallpaper" and "cartoon building"), the property of it (e.g., "tall building" and "modern building"), the class of it (e.g., "office building" and "school building"), and so on.

Now suppose we have relevant concepts for a given query in hand so that users can browse the search result by seeing top–N relevant concepts where each concept is represented by top–K images. Thought intuitive, organizing images in such manner can assist users to find out what they really want efficiently and effectively. The reasons why this approach works are illustrated as follows.

Firstly, users often have only rough idea/concept of what they want so that they have problem issuing a precise query. However, in another scenario, a precise query (query with many constraints) often leads to few or even empty search result, so users may issue a not-so-precise query at the first time. Then, in mainstream image search engines, users need to browse many pages to decide which images they prefer most. For example, Alice wants to find buildings with modern look. A lot of images conform to Alice's need, but they are spread out in many pages so that Alice must go through those pages to pick them out. By grouping images with the same relevant concept together, Alice can browse images that fit her need or interest her in a more condense way, thus she saves plenty of



(a) Text part (b) Retrieved image part

Figure 3: The search result of "building" by LiveImage

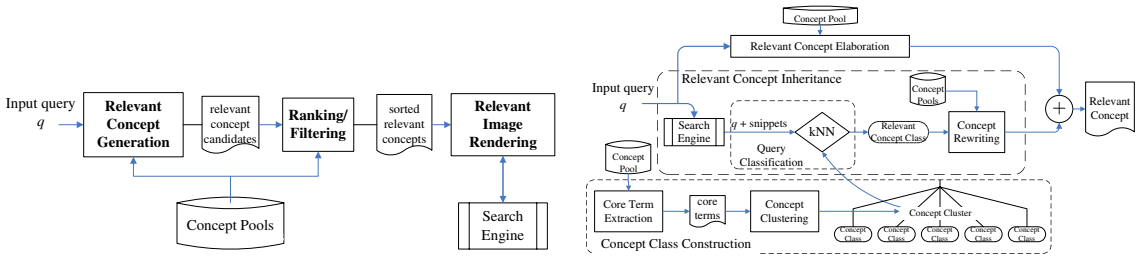
time on browsing disordered images.

Moreover, users can browse organized images by LiveImage faster than traditional ones. In traditional methods of representation, since images are not correlated with each other, it is time-consuming that users have to judge if a image suits their need. In proposed representation method, however, images which share the same concept in the same group are highly correlated with each other, users can browse the result faster. It is because by grouping images, the process that users decide preferred images turns into a batch mode. In this mode, the decision-making process becomes easier because a group of correlated images helps enhancing the concept belonging to itself. Also, users obtain preferred or discard unwanted images in batch manner, thus speeds up browsing as well.

One radical mystery must be disclosed here: besides showing images that share the same relevant concept in a group, the corresponding text annotation of that concept in LiveImage must be listed in parallel. Again, take the query "building" for example, we separate the result into text annotation part and retrieved image part, as shown in Figure 3a and in Figure 3b, respectively.

Suppose that LiveImage provides only the result images to users, as shown in Figure 3b, the relevant concept of images in each row may not come out at the first glance. With the supplementary aid of the annotated text, users can establish the relationship between a text and its corresponding images, as well as the correlation between images in a row. For example, in the last row of Figure 3b, users may think those buildings are quite special in shapes or avantgarde, but it is not easy for users to concrete the concept of those images into text, which refines the query "building" for those images. In contrast, as users see the text, "modern building," besides the images, probably all the users agree that those images have the concept of "modern building." With the text aside the retrieved image, LiveImage can assist users to realize the relative concepts from the search result as well as refine their queries. However, without the text aside, it not only weakens the effects of the organized images but also lacks the ability of assisting user refining their queries because it takes users extra effort to understand the underlying concept of images in each row.

On the contrary, suppose that LiveImage only provides the relevant concepts, just as what Figure 3a listed, users may find it less helpful because text itself cannot bring impressive information about what the images look like. Especially when users do not have specific target images in mind, only providing relevant concepts using text cannot assist them to select the preferred relevant concepts effectively. Though users can browse those relevant images by querying the relevant concepts one by one (i.e.,, by clicking the link of a relevant concept), the browsing efficiency as well as the impact of



(a) Overall architecture (b) Detailed view of *Relevant Concept Generation*

Figure 4: Framework of LiveImage

relevant images drop significantly. Moreover, users may not be such curious to click on the relevant concept link. What they would like is to browse the search result in a easy and impressive way, and this can be done by juxtaposing the texts of relevant concepts with their corresponding retrieved images.

In short, we highlighted the burgeoning demand of organizing search results of images in this section; thus a new direction of image search on the Web was proposed to provide a effective way to organize retrieved images. Our approach benefits users in the following way: 1) it provides a more efficient way browsing retrieved images by organizing them in a conceptual meaningful way; and 2) it gives users a hint of what they really want to find; i.e., to assist users refining their queries. We also pointed out the importance of juxtaposing the texts of relevant concepts with their corresponding retrieved images. Next, we illustrate the framework of LiveImage along with implementation details.

3.2 LiveImage: The Proposed Approach

Based on the direction proposed above, a meta search engine, LiveImage, was implemented. The architecture of LiveImage is shown in Figure 4a. Here we briefly explain the function of each stage (functional block) in LiveImage architecture.

First, in the *Relevant Concept Generation* stage, a large numbers of concepts are found/derived from the concept pool of the system and are treated as relevant concept candidates, where the *concept pool* might be built from query logs, search-result snippets, Web image annotations, etc., and can be seen as the knowledge base of this system.

Next, a *Ranking/Filtering* process is applied so that each candidate is assigned a score, called *interestingness*, to estimate the degree of users' interest and to prune candidates irrelevant to q.

Finally, we have the sorted relevant concepts, $R(q)$, of a given concept q after the Ranking/Filtering stage so that we can proceed *Relevant Image Rendering* process that performs image search q by showing its relevant images with concepts in $R(q)$.

We then elaborate the stages, which is organized as follows. In Section 3.3, we elaborate how to generate relevant concepts. Then we describe how to estimate a user's interestingness of a term and how to prune unlikely relevant concepts in Section 3.4. Finally, a rendering procedure is shown in Section 3.5.

3.3 Relevant Concept Generation

The function of Relevant Concept Generation task is to obtain a set of *relevant concept candidates*, $\tilde{R}(q)$ given an input query q. Since the problem we deal is finding relevant concepts, the input query

q can be seen as the input concept by nature (we use the term *query* and *concept* alternatively in this paper.) The proposed approach of relevant concept generation task is illustrated in Figure 4b.

First of all, we can see that there are three "concept pools" in the figure. The concept pool is a unified data structure (in the figure, three of them are identical and the copy of them are just for illustration use) consisting of a large numbers of concepts extracted mainly from query logs and it plays an important role in our framework. Since these concepts are originated from users' queries, it is reasonable to say that they are more meaningful to users than ordinary term sequences are. The concept pool is a set of concepts, denoted as $P = \{c_1, c_2, \ldots, c_{|P|}\}$, where $c_i \in 2^T$ for $i = 1, \ldots, |P|$ and T is set of terms $T = \{t_1, t_2, \ldots, t_{|T|}\}$. Our goal in the generating task is, given an input concept q, to obtain a set of relevant concept candidates, $\widetilde{R}(q)$, which can be obtained with two different strategies: by elaboration and by inheritance, proposed in Section 3.3.1 and Section 3.3.2, respectively.

The result is then obtained by union of both strategies. That is, $\tilde{R}(q) = \kappa_E(q) \cup \kappa_I(q)$, where $\kappa_E(q)$ and $\kappa_I(q)$ refer to the candidates generated by elaboration and by inheritance, respectively.

3.3.1 Generating by Elaboration

Given a concept q, we generate candidates by *elaboration* by looking up P to see if there are concepts elaborate q. The candidates generated by elaboration is denoted as a set $\kappa_E(q)$. The function $\kappa_E(q)$ is defined by

$$\kappa_E(q) = \{x | q \rightsquigarrow x, \forall x \in P\}, \tag{1}$$

where $q \rightsquigarrow x$ means x elaborates q.

The elaboration relation can be either implicit in semantic level, for example "building" $\rightsquigarrow$ "skyscraper;" or explicit in term composition level, such as "building" $\rightsquigarrow$ "office building." In this paper, we use only the latter method for simplicity reason. That is, $q \rightsquigarrow x$ holds iff q finds a match in x.

3.3.2 Generating by Inheritance

If we have all possible image queries of the world, we can just use elaboration method to generate relevant concept candidates. However, this is not always the truth. Due to the insufficient coverage of a query log, we propose an approach to compensate the sparseness nature of it. The motivation of the approach is: "Conceptually similar queries have similar usages of modifiers." The term *modifier* means a word/phrase used to modify or qualify a target word. For instance, let "building" be a target word, "tall" and "modern" are modifiers of "building".

Before going deeper, an example is taken to give a clear overview of the approach: The example is that we know that κ_E("building") $=$ {"office building", "tall building"...} from the query log. Now we have a new query $q =$"skyscraper", which is unseen in the log. First, we recognize that q is relevant to the concept "building" by search-result snippets; then we rewrite κ_E("building"); and finally $\tilde{R}(q) =$ {"office skyscraper", "tall skyscraper"...} is obtained. This approach is called *Relevant Concept Inheritance* because the relevant concept is "inherited" from the concept-class a query belongs to. The whole procedures of Relevant Concept Inheritance task with data flow is shown in Figure 4b.

In the first stage, we need to find out conceptually relevant terms. The intuitive way is to calculate the distance between a query and all terms in concept pool P. It looks fine at the first glance, but it is too time-consuming if P is large. So in the very first step, given a query set, we gather conceptually similar terms (concepts) into groups by adopting the clustering method proposed by Cheng [2]. The group has terms with similar concept are called *concept-class*, denoted as C. The

procedure that construct concept-class is called *Concept-Class Construction*, as shown in the bottom of Figure 4b.

Because we construct concept-classes to reduce the number of calculation, we need to determine which concept-class belongs to q. We use the *k–Nearest Neighbor* method to choose most relevant concept-class of q according to a query's search-result snippets. This task is called *Query Classification* and is illustrated in the center of Figure 4b.

After we classify q to concept-class C_q, named "Relevant Concept-Class of query q," we then analyze the relevant concepts of C_q, which can be formulated as $\bigcup_{c \in C_q} \kappa_E(c)$. We then extract the patterns that generating $\kappa_E(c)$ and then rewrite the patterns so that the relevant concept candidates $\tilde{R}(q)$ for a query q are obtained finally. To illustrate, after we classify $q =$"skyscraper" into relevant concept-class $C_q = \{$"building", "house", $\ldots\}$, we have to find out patterns that elaborate $c \in C_q$ to elements of $\kappa_E(c)$, such as "tall $*$", "modern $*$", etc. After that, we rewrite patterns using most intuitive way by simply substituting "$*$" to q. Finally we have the relevant concept candidates $\tilde{R}$("skyscraper") $= \{$"office skyscraper", "tall skyscraper", "school skyscraper", "modern skyscraper"$\ldots\}$.

3.4 Relevant Concept Ranking and Filtering

The major challenge of the ranking task is how to quantify the *interestingness* of a candidate. We assume that the *popularity*, namely, the number of occurrences of a candidate in the concept pool, reflects users' interests. This is reasonable, since our concept pool is constructed primarily from query logs. For an unseen candidate generated by inheritance only, however, its popularity is zero in the pool. To overcome this, we propose a measure called *potential popularity*. Consider the case where a candidate is generated by inherited pattern. In a concept-class, there may be more than one concepts derive that pattern; such concepts are called *voters*. For example, suppose "cute dog" and "cute cat" are in the same concept class, then both are voters of the pattern "cute $*$."

Our method is quite straightforward. Given an input concept q and a candidate r, the voters involved in the generation of r are defined by $V(q, r)$. Each individual voter v influences r according to its own popularity f_v, while the influence is proportion to the similarity measure between r and v, denoted as $Sim(v, r)$. So the potential popularity of r can be formulated as $\sum_{\forall v \in V(q,r)} f_v Sim(v, r)$. Then the interestingness of a candidate r with a concept q, denoted as $I(q, r)$, is calculated by combining popularity measure of r itself and potential popularity contributed by its voters $V(q, r)$.

The next problem we need to deal with is the removal of noise from the candidate set. Some heuristic rules can help us alleviate the problem. Presumably noise is of less interest to users and can be safely ignored by setting up a predefined threshold. However, although removing candidates with lower content-coverage is a good idea, it is costly in practice and risk the performance of our system by deleting concepts, such as people's names. To prevent this possibility, filtering out noise is performed in two steps: First we remove candidate r if $Sim(x, r) < \theta_S$, where θ_S is the predefined threshold. Then, we remove candidates with $I(q, r) < \theta_I$, where θ_I is the predefined threshold.

3.5 Rendering

The task can be divided into three steps: 1) sending a concept to the search engine; 2) extracting the value of the field "Number of Images" in the result page; and 3) extracting the URLs of images in the result page. Note that concepts with zero image returned are regarded as noise and are removed immediately. In practice, the rendering task is implemented as a standalone process to minimize the overhead of bringing up and down a network connection with the search engine.

Table 1: Query-focused Questions

Q1	Is the presentation scheme helpful in understanding the concepts related to the query?
Q2	Do images provide enough information to help you refine the query?
Q3	Do text descriptions of images provide enough information to help you refine the query?

Table 2: Performance-related Questions

G1	Are you satisfied with the response time?
G2	Is the search interface suitable for browsing Web images?
G3	Is the search interface suitable for locating specific Web images?
G4	Is the search interface easy to use?
G5	Is the retrieved images organized in a semantically meaningful way?
G6	Is the retrieved images organized in a visually comfortable way?
G7	Briefly comment on the search interface.

4. User Study for Accessing Pros and Cons of LiveImage

In this section, we conduct a user study to evaluate the performance of LiveImage under a real-world scenario: users attempt to find images without predetermined targets by interacting with the search engines. This survey focus on the comparison between two different types of result presentation schemes, Google Images and LiveImage, and how they assist users finding images interest them. All 23 subjects are asked to finish 30 predefined plus 10 free queries. In each query session, subjects are requested to search for Web images of their interest by using LiveImage and Google Images; then, they need to answer 3 query-focused questions, as shown in Table 1, for each system after finishing the search. The scores range from 1 to 5 (worst to best.) When they complete all the 30 query sessions, they give scores for each system in several performance-related aspects, as shown in Table 2.

We list the average scores for each query-focused question in Table 3, where each row represents a combination of a system and a group of queries, i.e. named-entities (-NE,) common queries (-Common,) or all the queries (-All.) The prefix "GI" and "LI" are abbreviations for Google Images and LiveImage, respectively. The results show that, in all three questions, the scores of "GI-All" are lower than those of "LI-All;" however, the difference between the system in the question Q2 is not as great as that in the question Q1 and Q3. This is probably because 1) it is easier for the users to understand the structure of the search-results from a categorized view and thus easier to refine their queries, 2) it takes less effort for users to refine their queries according to the text descriptions rather than image content, and 3) the text descriptions of images returned by LiveImage, primarily from query logs, contain less noise than the surrounding texts returned by Google Images.

In the first 4 rows, the performance of both systems on named-entities are lower than that on common queries. We also find that the difference between the scores of "GI-NE" and those of "GI-Common" is not obvious, while "LI-Common" outperforms "LI-NE" significantly. An interpretation for this phenomenon is that categorized search-results of a common query may greatly improve browsing experience because the images of a common query, such as "flower" or "cake", contain a great number of diverse relevant concepts from which users can easily distinguish.

Finally, the average scores for general questions G1 to G6 are listed In Table 3. Concerning the response time, Google Images gets a score of 4.7, while LiveImage gets only 3.35 in the question G1. Besides, Google Images is considered to be easier to use (4.3 in G4) and to have its search-results arranged in a visually–comfortable way (3.57 in G6.) On the contrary, LiveImage is more suitable for browsing task than Google Images is (4.09 in G2) and organizes the search-results based

Table 3: The user study results

	Q1	Q2	Q3
GI-NE	2.77	2.43	2.87
LI-NE	3.69	2.78	3.65
GI-Common	2.85	2.74	2.91
LI-Common	4.11	3.58	4.01
GI-All	2.83	2.63	2.89
LI-All	3.97	3.31	3.89

(a) The results of the query-focused questions

	G1	G2	G3	G4	G5	G6
GI	4.7	1.96	2.91	4.3	2.48	3.57
LI	3.35	4.09	2.35	3.96	4.04	3.48

(b) The results of the general questions

on their semantics (4.04 in G5.)

From the data, we can conclude that LiveImage assist users indeed and the search results are more satisfying. When users are clear and confident of what they want from the search, the results from LiveImage provides more related images and contains more information in both images and text descriptions than Google Images so that users can choose a classification in a shorter time. Furthermore, the presentation scheme of LiveImage assists users even progressively while users have only vague concepts or uncertain ideas to what they want because LiveImage offers users classified answers with both images and text descriptions so that users are given more information to find out and focus on the target images they want. On the contrary, we can see that the problem Google Images suffers in "GI-Common" is that once the users cannot search images in a specific term, Google Images can only order all the result images by how frequently they were searched before; however, this presentation scheme confuses and burdens users even worse. Therefore, LiveImage improves the search interface and reorganize the result images in classifications with text description assist users indeed.

5. Discussion

According to Pu's work [8], entertainment is the first motivation of image search interests; therefore, a system that can provide an impressive image-search result will enjoy the users. Though the contemporary technology of image search, CBIR, is able to group images with visual similarity, it suffers from the limitations of retrieval accuracy and scalability. On the contrary, our approach organizes the image-search results into topic classes by mining relevant concepts to the queries of the users from search engine logs. The clustered topics not only assist users browse the retrieved images but also attract their notice by exposing interesting ones with respect to the queries. Compare the image-search results of LiveImage with Vivísimo (`http://vivisimo.com`) as an example when these two systems present the results for the query "dog." LiveImage clusters the result images into many impressive categories, such as "dog licking," "mad dog," and "barking dog." However, Vivísimo generated thematic categories to the query, such as "pet," "cat," and "gallery." Therefore, LiveImage can achieve better performance in user's satisfaction to the query.

Besides, since longer queries to search for Web images are less possible for retrieved results comparing with searching for Web document, users are easily confused how to formulate their longer queries. The generated relevant concepts offered by LiveImage can assist users to refine users' queries and find other images with relevant concepts.

In addition, LiveImage is very efficient for online searching. The response time is in general in few seconds. This makes LiveImage practical in real image-search applications.

6. Conclusion

Organizing search results of images on the Web in a conceptual meaningful way both facilitates browsing efficiency of users and assists users sharpen their need. In this paper, we have proposed a new direction toward image search as well as an approach that organizes retrieved images with auto-generated relevant concepts. The perlimary user study demonstrates the feasibility and the effectiveness of the approach in improving Web image search.

References

[1] K. Barnard and D. Forsyth. Learning the semantics of words and pictures. *Proc. of the 8^{th} International Conference on Computer Vision*, 2:408–415, 2001.

[2] P.-J. Cheng and L.-F. Chien. Auto-generation of topic hierarchies for web images from users' perspectives. In *CIKM '03*, pages 544–547, 2003.

[3] P. Clough, H. Joho, and M. Sanderson. Automatically organizing images using concept hierarchies. In *Proc. of the SIGIR Workshop on Multimedia Information Retrieval*, 2005.

[4] J. Huang, S. R. Kumar, and R. Zabih. An automatic hierarchical image classification scheme. In *MULTIMEDIA '98: Proceedings of the sixth ACM international conference on Multimedia*, pages 219–228, 1998.

[5] K. Kummamuru, R. Lotlikar, S. Roy, K. Singal, and R. Krishnapuram. A hierarchical monothetic document clustering algorithm for summarization and browsing search results. In *WWW '04*, pages 658–665, 2004.

[6] S. Mukherjea, K. Hirata, and Y. Hara. AMORE: a world-wide web image retrieval engine. In *CHI '99: CHI '99 extended abstracts on Human factors in computing systems*, pages 17–18, 1999.

[7] G. Park, Y. Baek, and H.-K. Lee. Re-ranking algorithm using post-retrieval clustering for content-based image retrieval. *Inf. Process. Manage.*, 41(2):177–194, 2005.

[8] H.-T. Pu. A comparative analysis of web image and textual queries. *Online Information Review*, 29(5):457–467, May 2005.

[9] K. Rodden, W. Basalaj, D. Sinclair, and K. Wood. Does organisation by similarity assist image browsing? In *CHI '01: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 190–197, 2001.

[10] J. Smith and S.-F. Chang. WebSEEK: a content-based image and video search and catalog tool for the Web. In *Proc. of IEEE Multimedia*, 1997.

[11] O. Zamir and O. Etzioni. Web document clustering: a feasibility demonstration. In *SIGIR '98*, pages 46–54, 1998.

[12] H.-J. Zeng, Q.-C. He, Z. Chen, W.-Y. Ma, and J. Ma. Learning to cluster web search results. In *SIGIR '04*, pages 210–217, 2004.