# Clustering and Searching WWW Images Using Link and Page Layout Analysis

Xiaofei He
Yahoo! Research Labs
Deng Cai
Department of Computer Science, University of Illinois at Urbana-Champaign
Ji-Rong Wen
Microsoft Research Asia
Wei-Ying Ma
Microsoft Research Asia
and
Hong-Jiang Zhang
Microsoft Research Asia

Due to the rapid growth of the number of digital images on the Web, there is an increasing demand for effective and efficient method for organizing and retrieving the images available. This paper describes iFind, a system for clustering and searching WWW images. By using a vision-based page segmentation algorithm, a web page is partitioned into blocks, and the textual and link information of an image can be accurately extracted from the block containing that image. The textual information is used for image indexing. By extracting the page-to-block, block-to-image, block-to-page relationships through link structure and page layout analysis, we construct an image graph. Our method is less sensitive to noisy links than previous methods like PageRank, HITS and PicASHOW, and hence the image graph can better reflect the semantic relationship between images. Using the notion of Markov Chain, we can compute the limiting probability distributions of the images, i.e. ImageRanks, which characterize the importance of the images. The ImageRanks are combined with the relevance scores to produce the final ranking for image search. With the graph models, we can also use techniques from spectral graph theory for image clustering and embedding, or 2-D visualization. Some experimental results on 11.6 million images downloaded from the Web are provided in the paper.

Categories and Subject Descriptors: H.3.5 [**Information Storage and Retrieval**]: Online Information Services—*Web-based services*; H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval—*Clustering, Retrieval models, Search process*

General Terms: Algorithms, Management, Performance, Experimentation

Additional Key Words and Phrases: Web mining, image search, image clustering, link analysis

Author's address: Xiaofei He, Yahoo! Research Labs, 3333 Empire Avenue, Burbank, CA 91504, email: hex@yahoo-inc.com; Deng Cai, Department of Computer Science, University of Illinois at Urbana-Champaign, 201 N. Goodwin Avenue, Urbana, IL 61801, email: dengcai2@cs.uiuc.edu; Ji-Rong Wen, Wei-Ying Ma, and Hong-Jiang Zhang, Microsoft Research Asia, Beijing Sigma Center, No. 49, Zhichun Road, Beijing 100080, P.R.C, email: {jrwen, wyma, hjzhang}@microsoft.com.

## 1. INTRODUCTION

The emergence of the World Wide Web (WWW) has created many new opportunities but also challenges for organizing and searching a large volume of images available publicly. In this paper, we consider the problem of clustering and searching images on the web. The traditional image retrieval techniques based on content analysis, such as content-based image retrieval (CBIR) systems [Ma and Manjunath 1996; 1999; Rui et al. 1998; Smith and Chang 1996], are usually focused on small, static, and close-domain image data such as personal photo albums. When it comes to searching WWW images, these techniques may not be able to scale up to handle the large number of available images. Also, due to the diversity of the web, content-based features such as color, texture, and shape may not accurately reflect the semantic relationships between images. Moreover, different from traditional image retrieval and browsing, there is a lot of additional information on the web, such as surrounding texts, page layout and hyperlinks which is useful for organizing the images.

This paper introduces a web image search engine called iFind. Like text search engines, iFind does not have to access the original data to respond to a query; all analysis of the image and feature extraction are done off-line during the creation of the database. In this way, iFind can give fast query responses to a possibly huge number of users. The textual features we use to represent the image include image file title, image ALT (alternate text), image URL, page title, and page URL. However, textual information alone is insufficient for image search on the Web. For example, for a certain query, two different images may have the same relevance score to that query. To address this issue, one may use link analysis to assign an importance score to the images similar to what Google does for web page search [Brin and Page 1998].

PageRank [Brin and Page 1998] and HITS [Kleinberg 1999] are two of the most popular algorithms for link analysis. Based on them, PicASHOW [Lempel and Soffer 2001] was proposed for web image search. All these algorithms consider web pages as atomic units and are based on two assumptions: (a) the links convey human endorsement. If there exists a link from page A to page B and these two pages are authored by different people, the first author found the second page valuable. Thus the importance of a page can be propagated to those pages it links to. (b) pages that are co-cited by a certain page are likely related to the same topic. However, these two assumptions do not hold in many cases because a single web page often does not contain pure content but also things like navigation, decoration, interaction elements. It is also often the case that a single web page contains multiple topics. Thus, from the perspective of semantics, a web page should not be considered as the smallest unit. For example, the page at *http://news.yahoo.com/* contains multiple news topics, such as business, entertainment, sport, technology, etc. The hyperlinks contained in different semantic blocks usually point to the pages of different topics. Naturally, it is more reasonable to regard the semantic blocks as the smallest units of information.

In order to accurately model the semantic relationships between images on the Web, we propose to use a VIsion-based Page Segmentation (VIPS) algorithm [Cai et al. 2003b]. By using VIPS algorithm, each page can be segmented into a number

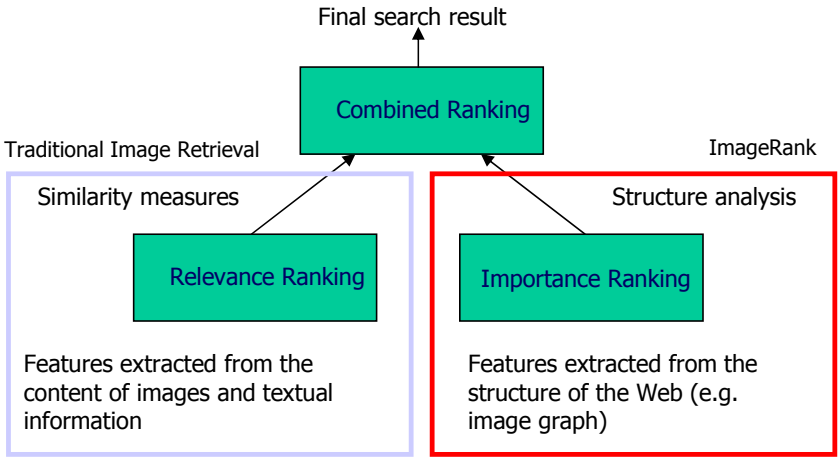


Fig. 1. Combined ranking for image search

of semantic blocks and each block contains semantically related information. Naturally, the hyperlink is considered from block to page rather than from page to page. For image search, we are interested in those blocks containing images (called *image blocks* hereafter). The surrounding texts are extracted within the image blocks rather than the whole page. Based on our block model, the Web structure for image search can be characterized into three relationships, i.e. *block-to-page* (link structure), *page-to-block* (page layout), and *block-to-image* (inclusion relation), which ultimately results in three graphs, i.e. page-to-page, block-to-block and image-to-image graphs. Unlike the graph model of PageRank which is constructed at the page level, our graph model is constructed at the block level. The above two assumptions become reasonable in block level link analysis.

With the graph models, we use techniques from spectral graph theory [Chung 1997; Guattery and Miller 2000; Mohar 1997] and Markov Chain theory for image ranking, clustering and embedding. Let us consider a random surfer on the graph. He jumps from image $i$ to image $j$ with some probability. Imagine the random surfer keeps jumping and after infinite jumps he finally stops at image $k$ with probability $\pi_k$. The vector $(\pi_1, \cdots, \pi_m)$ is then defined to be the stationary distribution of the Markov Chain induced from the graph. The asymptotic chance of visiting image $i$, that is, $\pi_i$, is then taken to be the "quality" or authoritativeness of image $i$. We call it *ImageRank* in this paper. This measurement is useful for ranking the image search result similar to what Google does for web page search [Brin and Page 1998]. That is, in addition to content and textual similarities, the image search could incorporate importance ranking to produce the final ranking (see Figure 1). It is also useful for browsing purpose. With the image graph model, the images can also be clustered into semantic classes by using spectral graph partitioning methods. Likewise, the image embedding problem can be formulated as graph embedding problem. The resulting vector representations of the images can be used for browsing or 2D visualization purposes.

Here, we summarize the novel contributions of our work.

(1) An unified framework for block level link analysis is presented.
(2) Based on our block level link analysis model, we propose a novel algorithm for clustering and searching WWW images.

The rest of this paper is organized as follows. Section 2 describes related works on searching WWW images. In Section 3, we briefly describe the VIPS page segmentation algorithm and its application to surrounding text extraction. In Section 4, we describe how to build the graph models. The ImageRank algorithm is introduced in Section 5. We present our methods for image clustering and embedding in Section 6. Some experimental evaluations are provided in Section 7. Finally, we provide some concluding remarks and suggestions for future work in Section 8.

## 2. PREVIOUS WORK

Image search is a long standing research problem. Previous work on web image search mainly falls into two categories, content-based [Frankel et al. 1996; Sclaroff et al. 1994; Smith and Chang 1997] and link-based [Lempel and Soffer 2001]. Note that, we do not distinguish text-based search from the above two categories simply because almost all practical search engines will use textual information.

For content-based web image search, the typical systems include WebSeer [Frankel et al. 1996], ImageRover [Sclaroff et al. 1994], WebSeek [Smith and Chang 1997], etc. All of them combine the textual information and visual information (color, texture, shape, etc.) for image indexing. However, content based computation is very expensive and the visual features might not be able to reflect the semantic relationship between images due to the diversity of the web. Therefore, content-based methods might not be practical for image search on the Internet.

The typical link-based image search system is PicASHOW [Lempel and Soffer 2001]. PicASHOW is based on web page search engine. The basic premise of PicASHOW is that a page $p$ displays (or link to) an image when the author of $p$ considers the image to be of value to the viewer of the page. Thus, PicASHOW first assembles a large collection of web pages relevant to the query, and then the images contained in those pages are ranked according to several link structure analyzing algorithms. We are especially interested in PicASHOW since our method can also be classified into this category.

It is worthwhile to highlight several aspects of the proposed approach here:

(1) In PicASHOW, there are two basic assumptions: (a) Images co-contained in **pages** are likely to be related to the same topic. (b) Images contained in pages that are co-cited by a certain **page** are likely related to the same topic. In fact, one can easily find many counter examples due to the fact that a page generally contains multiple different topics. The images contained in different blocks are likely related to different topics. The assumptions of our approach are the following: (a) Images co-contained in **blocks** are likely to be related to the same topic. (b) Images contained in pages that are co-cited by a certain **block** are likely related to the same topic.
(2) PicASHOW applies link analysis algorithms [Brin and Page 1998; Kleinberg 1999] to rank the images, which is computed on-line. The computation involved in our approach is off-line. The computed ImageRanks can be easily combined

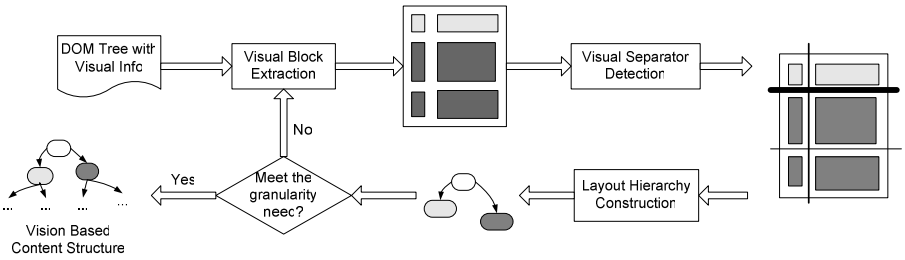


Fig. 2. Flowchart of the VIPS algorithm.

with the relevance scores to produce the final ranking. Thus, our approach can provide faster response to user's query.

(3) The framework of analysis presented in this paper contains three technical parts, i.e. page segmentation, graph model and spectral analysis. Within this framework, discovering semantic structure of the web image collections becomes possible. Specifically, this framework provides a ranking scheme, as well as a clustering and embedding scheme for the web images. All of them together provide an organization scheme for web images, which can be used for browsing purpose.

(4) Although our primary interest in this paper is in image, our framework of analysis actually provides a way of block-level link analysis. For example, within this framework, a block-level PageRank can be computed from the page-to-page graph which is induced from the block-level link structure analysis rather than traditional page-level analysis.

As a result of all these features, we expect the block based techniques to be a natural alternative to page based techniques in exploratory data analysis on the Web.

## 3. VISION-BASED PAGE SEGMENTATION

The VIsion-based Page Segmentation (VIPS) algorithm [Cai et al. 2003a; 2003b; Cai et al. 2004] aims to extract the semantic structure of a web page based on its visual presentation. Such semantic structure is represented as a tree; each node in the tree corresponds to a block. Each node will be assigned a value, *Degree of Coherence* (DOC), to indicate how coherent of the content in that block based on visual perception. DOC has the following properties:

—The greater the DOC value, the more consistent the content within the block.

—In the hierarchy tree, the DOC of the child is not smaller than that of its parent.

In VIPS algorithm, DOC values are integers ranging from 1 to 10.

The VIPS algorithm makes full use of page layout feature. The flowchart of the segmentation process is illustrated in Figure 2. First, by calling the analyzer embedded in the web browser, we obtain the DOM structure and visual information such as position, color, font size, font weight, etc. From the subtree within $\langle BODY \rangle$ in the DOM structure, we start the following iterative process to build the content structure.

(1) Visual Block Extraction: in the first step, we aim at finding all appropriate visual blocks contained in the current subtree. Normally, every child of the current node in the DOM structure can represent a visual block, like all the children of $\langle BODY \rangle$ in the DOM structure. However, some "big" nodes such as $\langle TABLE \rangle$ and $\langle P \rangle$ may act only for organization purpose and are not appropriate for representing a single visual block. Therefore, in these cases we should further divide the current node and replace it by its children. This process is iterated until all appropriate nodes are found to represent the visual blocks in the web page. For each node that represents a visual block, its DOC value is set according to its intra visual difference.

(2) Visual Separator Detection: After all blocks are extracted, they are put into a pool for separator detection. We define *visual separators* as horizontal or vertical lines in a web page that do not visually cross any blocks in the pool. We assign a weight to each separator according to distance, tag, font and color and select those with highest weights as the final separators.

(3) Content Structure Construction: once we obtain the separators, the visual blocks on the same sides of all the separators are merged and represented as a node in the content structure. The DOC value of each node in the content structure is defined in the same way as in Step 1. After that, each node is checked whether it meets the granularity requirement. For every node that fails, we go back to step 1 to construct the sub content structure within that node. If all the nodes meet the requirement, the iterative process is stopped and the final vision-based content structure is obtained.

We can pre-define a *Permitted Degree of Coherence* (PDOC) value to achieve different granularities of page segmentation for different applications. In this work, we empirically set it to 5. Meanwhile, we require that every image block must contain some text information. That is, a block which contains only images is not permitted. In Figure 3, the vision-based content structure of a sample page is illustrated. Visual blocks are detected as shown in Figure 3(a) and the content structure is shown in Figure 3(b). For more details about VIPS algorithm, please see [Cai et al. 2003b].

By using VIPS algorithm, noisy information, such as navigation, advertisement, and decoration can be easily detected because they are often placed in certain positions of a page. The user study in [Cai et al. 2003b] shows that VIPS can achieve 93% accuracy. Specifically, 1667 page segmentations have been judged by 5 users as "perfect", 1124 as "satisfactory", 184 as "fair", and 25 as "bad". The VIPS algorithm has been successfully applied to web information retrieval [Cai et al. 2004; Cai et al. 2004].

The VIPS algorithm can be naturally used for surrounding texts extraction. For each image, there is at least one (sometimes an image is cited repeatedly) image block that contains that image. Intuitively, the surrounding texts should be extracted within the image block. Figure 4 gives a simple example [1]. As can be seen, the surrounding texts are accurately identified. If two images are contained

[1]The URL of the presented web page is:
http://ecards.yahooligans.com/content/ecards/category?c=133&g=16

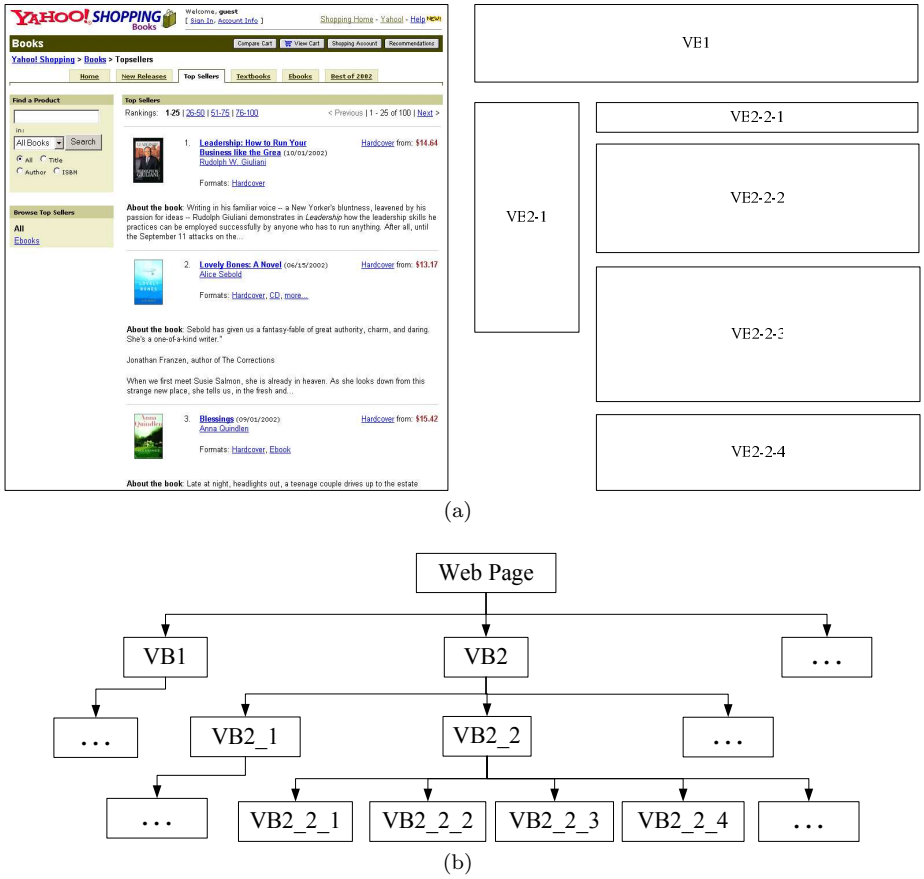


Fig. 3. Vision-based content structure of a sample web page. "VB" stands for visual block.

in the same block, they share the same surrounding texts. In our system, besides surrounding texts, we also use other textual information for image representation, such as image file title, image ALT (alternate text), image URL, page title, and page URL. Thus, even though two images are contained in the same block, they can have different text representations.

Once we obtain text representations of the images, the web image search problem becomes a text information retrieval problem. Thus, we can apply traditional text retrieval techniques, such as inverted indexing, TF-IDF weighting and cosine similarity measure, etc., for comparing the images to the query keywords. Unfortunately, due to the large amount of images available on the Web, many images end up having the same relevance score to the query, indicating that textual representation alone is insufficient for image ranking. Therefore, we need to use another kind of information, i.e. link structure, to compute importance ranking. This motivates us to consider the WWW images as a graph.

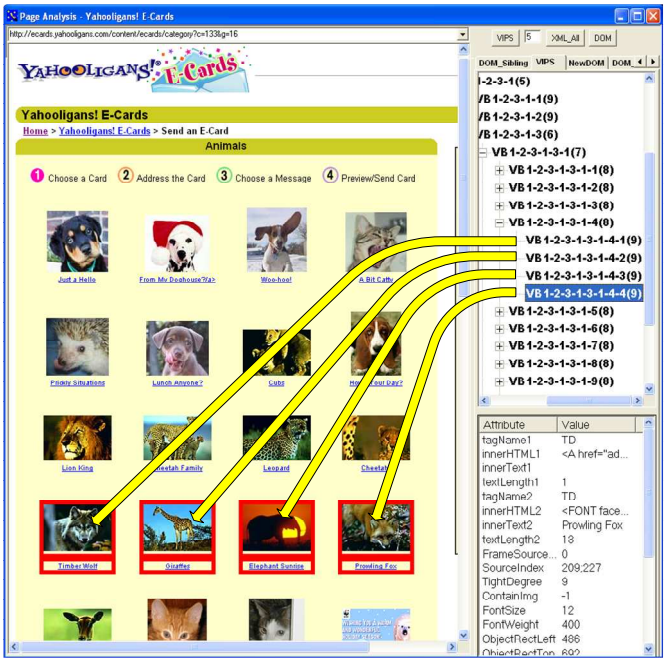


Fig. 4. The interface of our VIPS page segmentation system. The surrounding texts are extracted within the image blocks (with red frame).

## 4. GRAPH MODELS

The VIPS page segmentation algorithm does not only help to extract the meaningful surrounding texts, but also help to extract the useful links. In this section, we describe how to construct a block-to-block and image-to-image graphs. Like page-to-page graph model, the block-to-block model might be useful for many web based applications, such as web information retrieval and web page categorization, but in this paper our primary purpose is for image search and organization. Our graph model is induced from three kinds of relationships, i.e. **block-to-page** (link structure), **page-to-block** (page layout), and **block-to-image** (inclusion relation). We begin with some definitions. Let $\mathcal{P}$ denotes the set of all the web pages, $\mathcal{P} = \{p_1, p_2, \cdots, p_k\}$, where $k$ is the number of web pages. Let $\mathcal{B}$ denotes the set of all the blocks, $\mathcal{B} = \{b_1, b_2, \cdots, b_n\}$, where $n$ is the number of blocks. It is important to note that, for each block there is only one page that contains that block. Let $\mathcal{I} = \{I_1, I_2, \cdots, I_m\}$ denote the set of all the images on the web, where $m$ is the number of the web images. $b_i \in p_j$ means the block $b_i$ is contained in the page $p_j$. Likewise, $I_i \in b_j$ means the image $I_i$ is contained in the block $b_j$.

### 4.1 Block-Level Link Analysis

The block-to-page relationship is obtained from link analysis. Link analysis has proven to be very effective in web search [Brin and Page 1998; Kleinberg 1999]. However, a web page generally contains several semantic blocks. Different blocks are related to different topics. Therefore, it might be more reasonable to consider


Fig. 5. The useful links are within the image blocks, while the noisy links are outside the image blocks.

the hyperlinks from block to page, rather than from page to page. Let $Z$ denote the block-to-page matrix with dimension $n \times k$. $Z$ can be formally defined as follows:

$$Z_{ij} = \begin{cases} 1/s_i, & \text{if there is a link from block } i \text{ to page } j; \\ 0, & \text{otherwise.} \end{cases} \quad (1)$$

where $s_i$ is the number of pages that block $i$ links to. $Z_{ij}$ can also be viewed as a probability of jumping from block $i$ to page $j$. The block-to-page relationship gives a more accurate and robust representation of the link structures of the Web. For image search, those links outside the image blocks are regarded as noisy links, as shown in Figure 5. It would be important to note that, traditional link-based image search method like PicASHOW does not distinguish between the noisy links and useful links. Some detailed comparison between our link structure analyzing method and PicASHOW's method is given in the experimental section.

Figure 6 shows an example of block-to-page link structure. As can be seen, as the noisy links are eliminated the resulting link structure of images is much more accurate. The outlinks in image blocks have very high probability of pointing to those pages containing the images related to the same topics. The block-based link structure extracted by our method is much more meaningful than that extracted by previous methods, such as PageRank, HITS, and PicASHOW, which do not distinguish the useful links from noisy links.

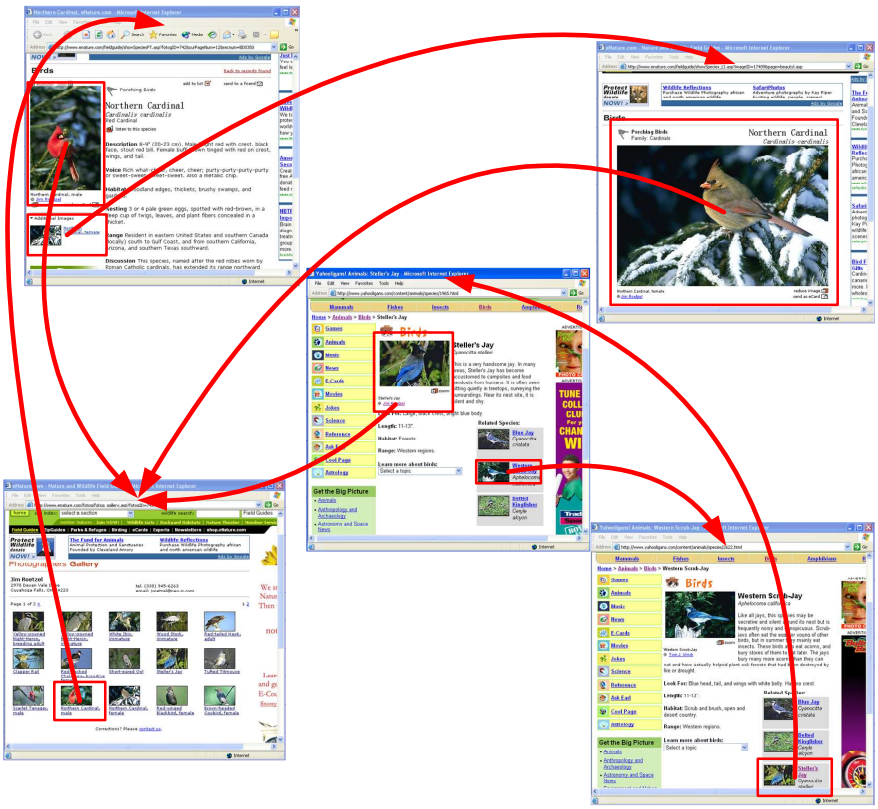

Fig. 6. The block-to-page link structure. The arrows denote the links from image block to web pages.

## 4.2 Page Layout Analysis

The page-to-block relationships are obtained from page layout analysis. Let $X$ denote the page-to-block matrix with dimension $k \times n$. As we have described above, each web page can be segmented into blocks. Thus, $X$ can be naturally defined as follows:

$$X_{ij} = \begin{cases} 1/s_i, & \text{if } b_j \in p_i; \\ 0, & \text{otherwise.} \end{cases} \tag{2}$$

where $s_i$ is the number of blocks contained in page $i$. The above formula assigns equal importance value to each block in a page. It is simple but less practical. Intuitively, some blocks with big size and centered position are probably more important than those blocks with small size and marginal position. This observation leads to the following formula,

$$X_{ij} = \begin{cases} f_{p_i}(b_j), & \text{if } b_j \in p_i; \\ 0, & \text{otherwise.} \end{cases} \tag{3}$$

where $f$ is a function which assigns to every block $b$ in page $p$ an importance value. Specifically, the bigger $f_p(b)$ is, the more important the block $b$ is. $f$ is empirically

defined below,

$$f_p(b) = \alpha \times \frac{\text{the size of block } b}{\text{the distance between the center of } b \text{ and the center of the screen}} \quad (4)$$

where $\alpha$ is a normalization factor to make the sum of $f_p(b)$ to be 1, i.e.

$$\sum_{b \in p} f_p(b) = 1$$

Note that, $f_p(b)$ can also be viewed as a probability that the user is focused on block $b$ when viewing page $p$.

The use of block importance function $f$ provides a way of distinguishing between image blocks and noisy blocks. Basically, there are two kinds of features which can be used to infer the block importance. One is the spatial feature (size and position) as what we use here, and the other is content feature. The content feature can include the number and size of the images contained in the block, the number of hyperlinks, the text length, etc. Our previous work [Song et al. 2004] has demonstrated that, spatial features are more useful than content features as to estimating the true block importance assigned by human. With spatial features alone, we can achieve 75% accuracy. And if we use both spatial and content features, we can achieve 79% accuracy. In our web image search system, we only use spatial features for simplicity.

It would be important to note that the block importance has nothing to with the text representations of the images. The block importance is only used in our graph model which produces ImageRank. That is, block importance is only useful for estimating the importance of the images.

### 4.3 Block Analysis

Let $Y$ denote the block-to-image matrix with dimension $n \times m$. For each image, there is at least one block that contains this image. Thus, $Y$ can be simply defined below:

$$Y_{ij} = \begin{cases} 1/s_i, & \text{if } I_j \in b_i; \\ 0, & \text{otherwise.} \end{cases} \quad (5)$$

where $s_i$ is the number of images contained in the image block $b_i$. Also, for those image blocks, the surrounding texts of the images are extracted to index the images, as described in Section 3.

### 4.4 Building Graph Models

In the preceding three subsections, we have constructed three affinity matrices, i.e. block-to-page, page-to-block, and block-to-image. Based on these three matrices, we can build three graph models, i.e. page graph $G_{\mathcal{P}}(V_{\mathcal{P}}, E_{\mathcal{P}}, W_{\mathcal{P}})$, block graph $G_{\mathcal{B}}(V_{\mathcal{B}}, E_{\mathcal{B}}, W_{\mathcal{B}})$, and image graph $G_{\mathcal{I}}(V_{\mathcal{I}}, E_{\mathcal{I}}, W_{\mathcal{I}})$. For each graph, $V$ is the set of the nodes (page, block, image, respectively), $E$ is the set of edges linking two nodes, $W$ is a weight matrix defined on the edges. We begin with the page graph.

4.4.1 *Page Graph.* When constructing a graph, we essentially define a weight matrix on the edges. $W_{\mathcal{P}}$ can be simply defined as follows. $W_{\mathcal{P}}(i, j)$ is 1 if page $i$ links to page $j$, and 0 otherwise. This definition is pretty simple yet has been widely

used as the first step to many applications, such as PageRank [Brin and Page 1998], HITS [Kleinberg 1999], etc. However, based on our previous discussions, different blocks in a page have different importance. Therefore, those links in blocks with high importance value should be more important than those in blocks with low importance value. In other words, a user might prefer to follow those links in important blocks. This consideration leads to the following definition of $W_{\mathcal{P}}$,

$$W_{\mathcal{P}}(\alpha, \beta) = \sum_{b \in \alpha} f_{\alpha}(b) Z_{b,\beta}, \alpha, \beta \in \mathcal{P} \tag{6}$$

The above equation can be rewritten in matrix form as follows:

$$W_{\mathcal{P}} = XZ \tag{7}$$

$X$ is a $k \times n$ page-to-block matrix and $Z$ is a $n \times k$ block-to-page matrix, thus $W_{\mathcal{P}}$ is a $k \times k$ page-to-page matrix.

Here we provide a simple analysis of our definition of $W_{\mathcal{P}}$ from the probabilistic viewpoint. Let's consider $W_{\mathcal{P}}(\alpha, \beta)$ as a probability $Prob(\beta|\alpha)$ of jumping from page $\alpha$ to page $\beta$. Since page $\alpha$ is composed of a set of blocks, we have

$$Prob(\beta|\alpha) = \sum_{b \in \alpha} Prob(\beta|b) Prob(b|\alpha)$$

where $Prob(\beta|b)$ is actually $Z_{b,\beta}$ and $Prob(b|\alpha)$ is the block importance $f_{\alpha}(b)$.

Finally, it would be interesting to see under what conditions our definition of $W_{\mathcal{P}}$ reduces to the ordinary definition (that is used in PageRank). It is easy to show that this occurs when the function $f(b)$ is defined as the number of links contained in block b.

4.4.2 *Block Graph.* The block graph is constructed over the blocks. Let us first consider a jump from block $a$ to block $b$. Suppose a user is looking at block $a$. In order to jump to block $b$, he first jumps to page $\beta$ which contains block $b$, and then he focuses his attention on block $b$. Thus, a natural definition of $W_{\mathcal{B}}$ is as follows,

$$\begin{aligned} W_{\mathcal{B}}(a, b) &= Prob(b|a) \\ &= \sum_{\theta \in \mathcal{P}} Prob(\theta|a) Prob(b|\theta) \\ &= Prob(\beta|a) Prob(b|\beta) \\ &= Z_{a,\beta} X_{\beta,b}, a, b \in \mathcal{B} \end{aligned} \tag{8}$$

The above equation can be rewritten in matrix form as follows:

$$W_{\mathcal{B}} = ZX \tag{9}$$

where $W_{\mathcal{B}}$ is an $n \times n$ matrix. By definition, $W_{\mathcal{B}}$ is clearly a probability transition matrix. However, there is still one limitation of this definition that it fails to reflect the relationships between the blocks in the same page. Two blocks are likely related to the same topic if they appear in the same page. This leads to a new definition,

$$W_{\mathcal{B}} = (1 - t)ZX + tD^{-1}U \tag{10}$$

where $t$ is a suitable constant. $D$ is a diagonal matrix, $D_{ii} = \sum_j U_{ij}$. $U_{ij}$ is zero if block $i$ and block $j$ are contained in two different web pages; otherwise, it is set

to the *DOC* (degree of coherence, please see [Cai et al. 2003a; 2003b] for details) value of the smallest block containing both block $i$ and block $j$. It is easy to check that the sum of each row of $D^{-1}U$ is 1. Thus, $W_{\mathcal{B}}$ can be viewed as a probability transition matrix such that $W_{\mathcal{B}}(a, b)$ is the probability of jumping from block $a$ to block $b$.

4.4.3 *Image Graph.* Once the block graph is obtained, the image graph can be constructed correspondingly by noticing the fact that every image is contained in at least one block. Let's consider the jump from image $i$ to image $j$. From image $i$ we first see block $a$ containing image $i$. Through the block graph defined above, we get a jump from block $a$ to block $b$ containing image $j$. Finally, we stop at image $j$. In this way, the weight matrix of the image graph can be naturally defined as follows:

$$W_{\mathcal{I}}(i, j) = \sum_{i \in a, j \in b} W_{\mathcal{B}}(a, b) \tag{11}$$

The above equation can be rewritten in matrix form as follows:

$$W_{\mathcal{I}} = Y^T W_{\mathcal{B}} Y \tag{12}$$

$W_{\mathcal{I}}$ is a $m \times m$ matrix. If two images $i$ and $j$ are in the same block, say $b$, then $W_{\mathcal{I}}(i, j) = W_{\mathcal{B}}(b, b) = 0$. However, the images in the same block are supposed to be semantically related. Thus, we get a new definition as follows:

$$W_{\mathcal{I}} = \theta D^{-1} Y^T Y + (1 - \theta) Y^T W_{\mathcal{B}} Y \tag{13}$$

where $\theta$ is a suitable constant and $D$ is a diagonal matrix, $D_{ii} = \sum_j (Y^T Y)_{ij}$. Like $W_{\mathcal{B}}$, $W_{\mathcal{I}}$ can be viewed as a probability transition matrix.

## 5. IMAGERANK

In this section, we will introduce the ImageRank algorithm using spectral techniques. By using spectral techniques on the graph models obtained previously, we can compute the importance value for each image.

*Definition* 5.1. Given a set of images, $\{\mathbf{x}_1, \cdots, \mathbf{x}_m\}$, we use ImageRank $\pi(\mathbf{x}_i)$ to denote the importance of image $\mathbf{x}_i$, i.e. the richness of the semantics contained in image $\mathbf{x}_i$. Without loss of generality, we let $\sum_i \pi(\mathbf{x}_i) = 1$.

### 5.1 The Algorithm

As described in the introduction section, one of the fundamental problems in web image search is ranking. Ranking by text information alone is insufficient since some images can have the same textual representation. In this section, we describe ImageRank which gives every image an importance value. We expect that the text information combined with the importance value will give a better ranking scheme than each alone.

Let $\mathcal{M}$ denotes the random walk naturally induced from the image graph. By our definition, the weight matrix $W$ of the graph is also the probability transition matrix of $\mathcal{M}$. Now let us consider a random surfer on the graph. He jumps from image $i$ to image $j$ with probability $W_{ij}$. There is also a possibility that the surfer does not follow the probability transition matrix induced from the web structure

but jumps to an image picked uniformly and at random from the collection. Thus, a more reasonable probability transition matrix $P$ can be defined as follows:

$$P = \epsilon W + (1 - \epsilon)U \tag{14}$$

where $\epsilon$ is a parameter, generally set to $0.1 \sim 0.2$. In our experiments, it is set to 0.15. $U$ is a transition matrix of uniform transition probabilities ($U_{ij} = 1/m$ for all $i$, $j$). In fact, the introduction of $U$ makes the graph connected and hence the stationary distribution of the random walk always exists.

Imagine the random surfer keeps jumping and after infinite jumps he finally stops at image $k$ with probability $\pi_k$. $\pi = (\pi_1, \cdots, \pi_m)$ is often called stationary distribution (or, limiting probability distribution). It can be computed by solving the following eigenvector problem:

$$P^T \pi = \pi \tag{15}$$

Clearly, $\pi$ is the eigenvector of $P^T$ corresponding to the eigenvalue 1. If an image has a high ImageRank, this means that this image has high probability to be viewed by the user. Therefore, the ImageRanks reflect the importance of the images to some extent.

## 6. CLUSTERING AND EMBEDDING WWW IMAGES

The following section is based on the standard spectral graph theory. See [Chung 1997; Guattery and Miller 2000; Mohar 1997] for a comprehensive reference. Spectral techniques use information contained in the eigenvectors and eigenvalues of a data affinity (i.e., item-item similarity) matrix to detect structure. Such an approach has proven effective on many tasks, including web search [Brin and Page 1998], image segmentation [Shi and Malik 2000], word class detection [Brew and Wade 2002], face recognition [He et al. 2005], etc.

Spectral graph embedding and clustering [Belkin and Niyogi 2001; Chung 1997; Guattery and Miller 2000; Mohar 1997; Shi and Malik 2000] are related to each other in the sense that both of them can be reduced to similar eigenvector problems. Spectral graph embedding can be viewed as the first step to spectral clustering. Here, by embedding we mean that each image can be endowed with a vector representation in Euclidean space such that the distance between two images reflects their similarity.

In Section 4, we have obtained a weight matrix of the image graph, $W_{\mathcal{I}}$. We first convert it into a similarity matrix $S = 1/2(W_{\mathcal{I}} + W_{\mathcal{I}}^T)$ which is symmetric. Note that content-based (color, texture, shape, etc.) similarity measure between images has been researched extensively in computer vision community in the past decades. The visual content of images might be helpful to define the optimal similarity measure. However, the extraction of visual features is computationally expensive, which makes it less practical to be used for web based applications.

Now, suppose $y_i$ is an one-dimensional representation of image $i$. The optimal representation $\mathbf{y} = (y_1, \cdots, y_m)$ can be obtained by solving the following objective function [Belkin and Niyogi 2001]:

$$\min_{\mathbf{y}} \sum_{ij} (y_i - y_j)^2 S_{ij} \tag{16}$$

The objective function with the choice of $S_{ij}$ incurs a heavy penalty if semantically related images are mapped far apart. Therefore, minimizing it is an attempt to ensure that if image $i$ and image $j$ are semantically related then $y_i$ and $y_j$ are close to each other. Let $D$ be a diagonal matrix whose $i$-th entry is the row (or column, since $S$ is symmetric) sum of $S$, $D_{ii} = \sum_j S_{ij}$. The objective function can be reduced to the following minimization problem [Belkin and Niyogi 2001]:

$$\min_{\mathbf{y}^T D\mathbf{y}=1} \mathbf{y}^T L\mathbf{y} \tag{17}$$

where the matrix $L$ is the so called graph Laplacian [Chung 1997], $L = D - S$. Thus, the optimal embedding is given by the minimum eigenvalue solutions to the following generalized eigenvector problem:

$$L\mathbf{y} = \lambda D\mathbf{y} \tag{18}$$

Let $(\mathbf{y}^0, \lambda^0), \cdots, (\mathbf{y}^{m-1}, \lambda^{m-1})$ be the solutions to (18), and $\lambda^0 < \cdots < \lambda^{m-1}$. It is easy to check that $\lambda^0 = 0$ and $\mathbf{y}^0 = (1, \cdots, 1)$. Therefore, we leave out the eigenvector $\mathbf{y}^0$ and use the next $k$ eigenvectors for embedding the images in a $k$-dimensional Euclidean space:

$$\text{image } j \leftarrow \left(\mathbf{y}^1(j), \cdots, \mathbf{y}^k(j)\right)$$

where $\mathbf{y}^i(j)$ denotes the $j$-th element of $\mathbf{y}^i$. In this way, we endow each image with a vector representation. Note that, since all the matrices involved in this computation are very sparse, the computation can be performed very fast.

Once we obtain vector representations of the images, clustering is straightforward. The simplest way is to use $\mathbf{y}^1$ (called Fiedler vector in spectral graph theory [Chung 1997]) to cut the image set into several clusters. For details, please see [Guattery and Miller 2000; Mohar 1997]. Another way is to use k-means clustering algorithm on the image vectors. Previous work has demonstrated that spectral embedding followed by k-means can produce good result [Ng et al. 2001].

## 7. EXPERIMENTAL EVALUATIONS

In this section, several illustrative examples and experimental evaluations are provided. We begin with an introduction of the iFind image search engine.

### 7.1 iFind: The System Overview

In the above three sections, we have systematically described our techniques for web image search and organization, i.e. vision-based page segmentation, link and page layout based graph models, and spectral analysis for image ranking, clustering and embedding.

In order to achieve fast response, the images are crawled from the web and their surrounding texts are extracted off-line. The surrounding texts are used to index the images. We use BM2500 in Okapi [Robertson and Walker 1999] as our relevance ranking function, which has been proved to be effective in information retrieval community. For the details about our implementation, please see [Wen et al. 2003]. Our system combines the textual relevance score and ImageRank as follows:

$$s(\mathbf{x}) = \alpha \times \pi(\mathbf{x}) + (1 - \alpha) \times r(\mathbf{x}, \mathbf{q}) \tag{19}$$

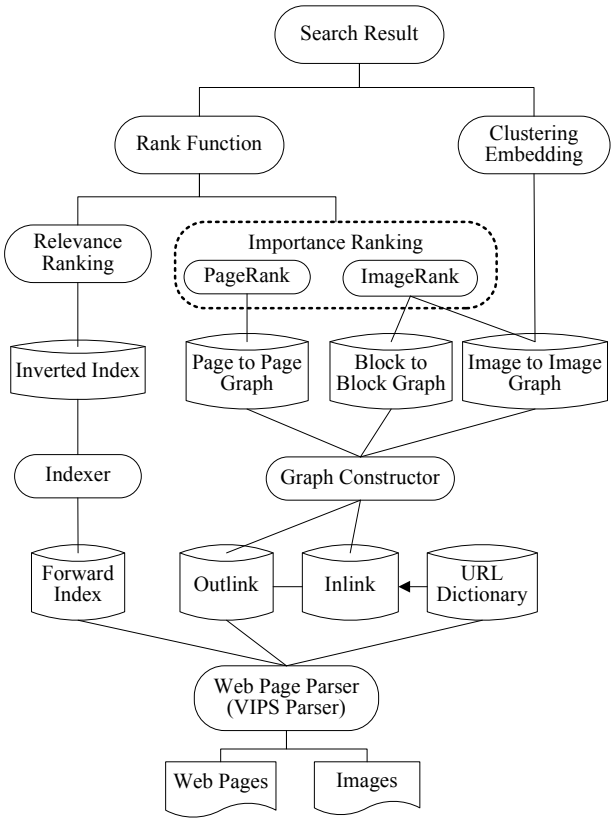


Fig. 7. Design of our WWW image search system.

where $r(\mathbf{x}, \mathbf{q})$ is the textual relevance score of image $\mathbf{x}$ to query $\mathbf{q}$, $\pi(\mathbf{x})$ is the ImageRank of $\mathbf{x}$, and $s(\mathbf{x})$ is the combined score. $\alpha$ is a parameter, generally set to $0.2\sim 0.3$. Here, $\pi(\mathbf{x})$ and $r(\mathbf{x}, \mathbf{q})$ have been normalized into the same scale. $\pi(\mathbf{x})$ can also be other link based ranks, such as PageRank.

When the user submits a query, the system first computes the relevance score for every image and the images are ranked according to their relevance scores. For the top $N$ images, we re-rank them according to the combined scores. The re-ranked top $N$ images are then presented to the user. Figure 7 shows the design of our system.

### 7.2 Comparison with PicASHOW: a Simple Example

Recall that PicASHOW has two basic assumptions: (a) Images which are co-contained in pages are likely to be related to the same topic. (b) Images which are contained in pages that are co-cited by a certain page are likely related to the same topic. Our system also has two assumptions listed in Section 2. Here, we give a simple example to compare PicASHOW's assumptions with our assumptions, which are the fundamental differences between our approach and PicASHOW from the perspective of link analysis.

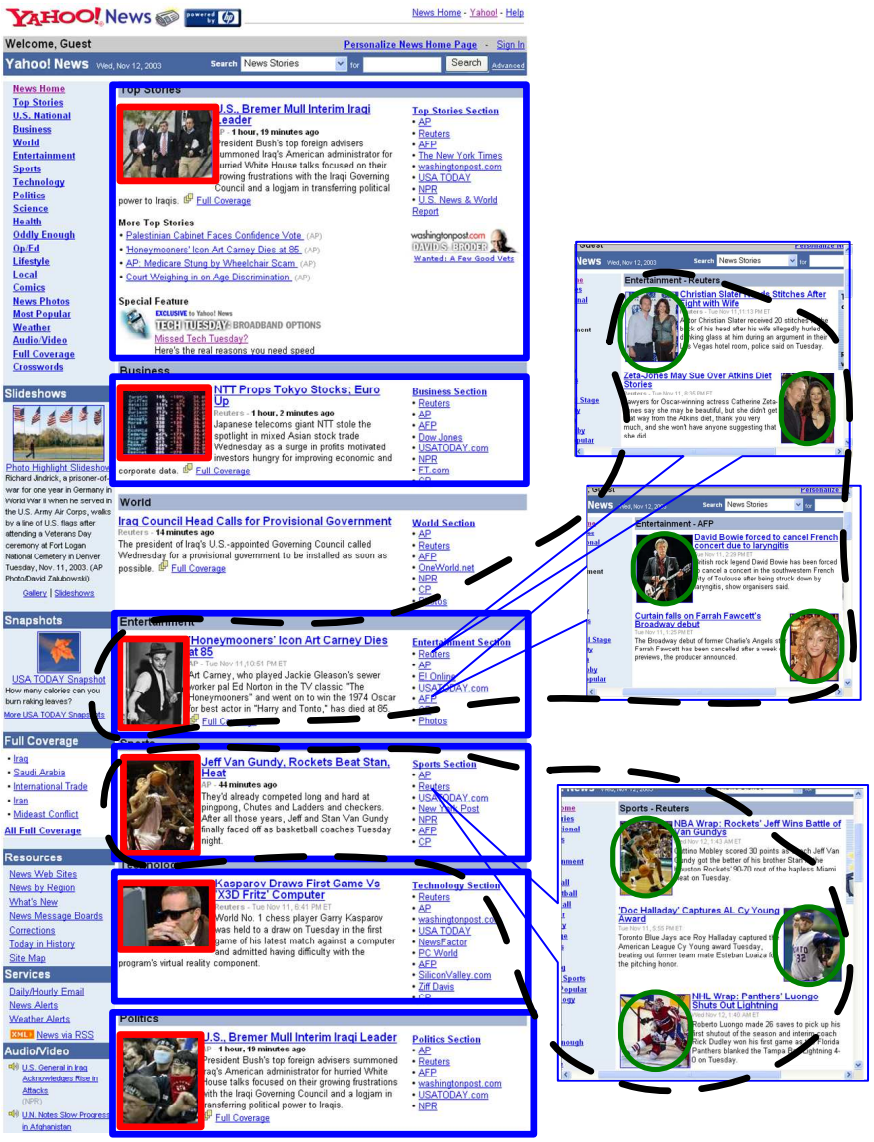

Fig. 8. The web page *http://news.yahoo.com* (left) is segmented into blocks. The right three pages are linked by the sport block and the entertainment block. The images in the dashed circle are related to the same topic.

Figure 8 shows the web page *http://news.yahoo.com* (left part) and three other pages (right part) it points to. The images in *http://news.yahoo.com* are with rectangular frame, and we call them "rectangular images" for the sake of simplicity. The images in other three pages are with circle frame, and we call them "circle images". Based on PicASHOW's assumption (a), all the rectangular images are related to the same topics since they are contained in the same page. However, it

is clear to see that they are related to different topics, i.e. business, entertainment, sport, technology, and politics, respectively. Also, PicASHOW's assumption (b) implies that the circle images are related to the same topics since they are co-cited by *http://news.yahoo.com*. Again, it can be seen that the circle images are related to two different topics, i.e. entertainment and sports.

If we view these web pages from the block level as suggested by our approach, we get different results. First, the web page *http://news.yahoo.com* is segmented into semantic blocks. Thus, the rectangular images are regarded as different semantic objects. The circle images are linked by two different blocks. Thus, they are classified into two different semantic classes (entertainment and sports), as shown in Figure 8. This example shows that our method can get more accurate information than PicASHOW.

### 7.3 Image Search on the Web

7.3.1 *Data Preparation.* All the image data we used in our search experiments were crawled from the Internet. Starting from the following website

*http://dir.yahoo.com/Arts/Visual_Arts/Photography/Museums_and_Galleries/*

We crawled 26.5 million web pages in total by breadth first search. From these web pages, 11.6 million images were extracted. We filtered those images whose ratio between width and height are greater than 5 or smaller than 1/5, since these kinds of images are probably of low quality. We also removed those images whose width and height are both smaller than 60 pixels due to the same reason.

For each web page, the VIPS page segmentation algorithm was applied to divide it into semantic blocks. For each block, the hyperlinks were extracted. For each image, the image blocks containing that image were identified and the surrounding texts were extracted from these image blocks.

7.3.2 *Search Results.* The importance of an image can be evaluated by its ImageRank, as described in Section 4. Or more simply, it can be evaluated by the PageRank of the web page containing this image. In this subsection, we compare two methods for web image search. In the first one, for each image we compute its relevance score to the query keyword. The relevance score and ImageRank are combined, and the combined scores are used for image ranking. In the second method, the relevance score and PageRank are combined for image ranking. Note that, PicASHOW is based on some specific web page search engine. It first collects a number of relevant web pages from certain web page search engine, then the images contained in these pages are ranked by using link analysis algorithms [Brin and Page 1998], [Kleinberg 1999]. The performance of PicASHOW relies heavily on the performance of the web page search engine. This is quite different from our system which is independent to the web page search engines.

In order to compare the ImageRank-based and PageRank-based methods, we chose 45 frequently used queries from the statistical result of Google image search engine [Google ]. For each query, we first used relevance scores to rank the images. Five volunteers were asked to evaluate the top 100 results returned by the system. In this way, we got a ground truth. In other words, according to the users' judgments,

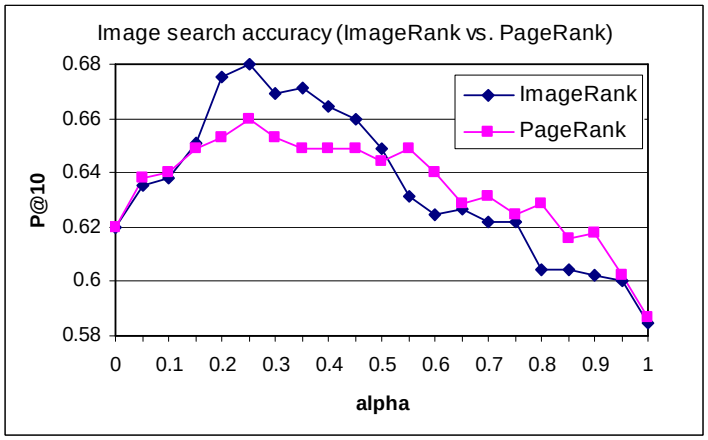


Fig. 9. Image search accuracy using ImageRank and PageRank. Both of them achieved their best results at $\alpha$=0.25.

each image in the top 100 returns is classified as relevant image or irrelevant image.

We used Eqn. (19) to combine the relevance score and importance score (PageRank or ImageRank) for each image. The combined scores were used for final ranking. For each query, we computed the search accuracy for the top 10 returns (or, Precision@10) using our ImageRank-based method, PageRank-based method, and the baseline method (using relevance score only). The average precision for each method was computed over the 45 queries. Also, the average precision was plotted as a function of $\alpha$ as Figure 9 shows. When $\alpha = 0$, only the relevance score is used for the *final* ranking, which is treated as the baseline method. When $\alpha = 1$, only the ImageRank (or PageRank) is used for *final* ranking. As can be seen, both PageRank method and ImageRank method achieved their best results at $\alpha = 0.25$, and our ImageRank method outperformed the PageRank method. The detailed image search results for each query is listed in Table 1. We did a $t$-test on the search accuracies of the baseline method and our ImageRank-based method. The $p$-value is 0.001952. We also did a $t$-test on the search accuracies of PageRank-based method and our method. The $p$-value is 0.04159. Both of these two tests show that the improvement of our proposed method over the other two methods is significant. It is worth noting that, in our experiments, ImageRank is used only for re-ranking results. In practice, one may also combine ImageRank and relevance score of every image in the database. This helps to bring in images that are not retrieved by the text retrieval system.

In this work, only 45 queries were used to test our system due to the limitation of labor. In order to test the stability of our system in statistical sense, more queries need to be tested. It is left for our future work.

### 7.4 Clustering and Embedding WWW Images

In this subsection, we evaluate our algorithm for clustering and embedding WWW images. The main difficulty is the lack of ground truth (class labels of the images).

Table I. Image search accuracy (P@10)

| Query | Baseline | PageRank | ImageRank |
|---|---|---|---|
| angel | 0.1 | 0.3 | 0.3 |
| monkey | 0.8 | 0.7 | 0.6 |
| basketball | 0.6 | 0.5 | 0.6 |
| moon | 0.1 | 0.4 | 0.4 |
| beach | 0.7 | 0.6 | 0.6 |
| picasso | 0.8 | 1 | 1 |
| bmw | 0.4 | 0.4 | 0.4 |
| pizza | 0.7 | 0.7 | 0.7 |
| chirac | 0.4 | 0.5 | 0.5 |
| pluto | 1 | 0.9 | 0.9 |
| christmas tree | 0.8 | 0.8 | 0.9 |
| pokemon | 0.6 | 0.7 | 0.8 |
| coca cola | 0.3 | 0.2 | 0.3 |
| porsche | 0.9 | 0.9 | 0.9 |
| coffee | 0.2 | 0.2 | 0.3 |
| pumpkin | 0.7 | 0.9 | 0.9 |
| earth | 0.8 | 0.9 | 0.9 |
| roses | 0.9 | 0.9 | 0.9 |
| ferrari | 0.2 | 0.6 | 0.5 |
| santa claus | 0.8 | 1 | 1 |
| fish | 0.6 | 0.6 | 0.6 |
| snoopy | 0.9 | 0.9 | 1 |
| flower | 0.8 | 0.7 | 0.8 |
| snowboard | 0.8 | 0.9 | 0.9 |
| garfield | 0.5 | 0.3 | 0.4 |
| space | 0.4 | 0.7 | 0.7 |
| george bush | 0.5 | 0.8 | 0.7 |
| spiderman | 1 | 1 | 1 |
| halloween | 0.5 | 0.6 | 0.7 |
| stars | 0.4 | 0.6 | 0.6 |
| harry potter | 0.7 | 0.5 | 0.8 |
| superman | 0.8 | 1 | 1 |
| hearts | 0.6 | 0.5 | 0.6 |
| tennis | 0.5 | 0.4 | 0.4 |
| horse | 0.6 | 0.6 | 0.5 |
| tiger | 0.7 | 0.6 | 0.5 |
| house | 0.5 | 0.7 | 0.6 |
| ufo | 0.1 | 0.1 | 0.2 |
| matisse | 0.8 | 0.7 | 0.7 |
| van gogh | 1 | 1 | 1 |
| matrix | 0.8 | 0.7 | 0.7 |
| water | 0.1 | 0.1 | 0.2 |
| mermaid | 0.7 | 0.6 | 0.6 |
| yoda | 1 | 1 | 1 |
| monet | 0.8 | 1 | 1 |
| **Average** | **0.62** | **0.66** | **0.68** |

In this reason, we crawled a relatively small data set yet the class labels could be obtained. The techniques of clustering and embedding of WWW images can be

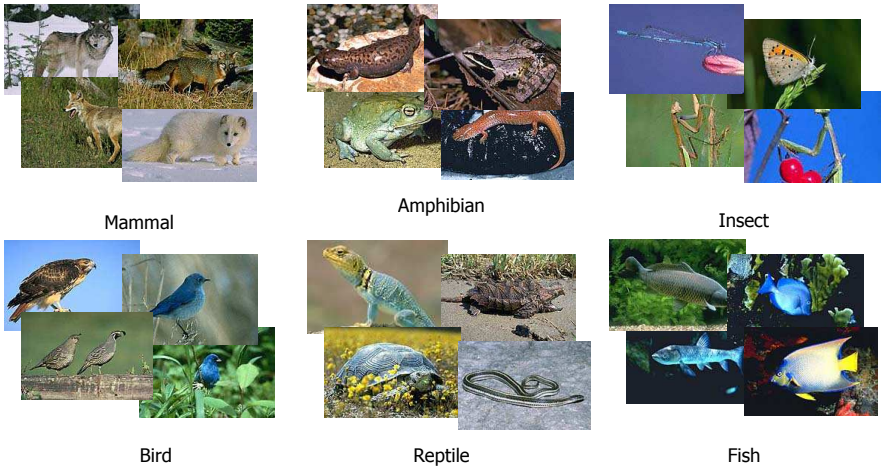


Fig. 10. Six image categories

used to re-organize the search results to facilitate the user's browsing.

7.4.1 *Data Preparation.* All the image data used in this experiment are crawled from the Web, starting from the following URL:

http://www.yahooligans.com/content/animals/

We crawled 1288 web pages in total. All the pages were restricted to be within this directory. From these web pages, 1710 JPG images were extracted. It can be clearly seen from the website that this data set is composed of six categories, i.e. mammal, fish, reptile, bird, amphibian and insect, as shown in Figure 10. Each category can be further divided into sub-categories since some spices are more related. In the following section, we show that the semantic structure of this image set can be accurately discovered using our block level link analysis method.

7.4.2 *Experimental Results.* We first constructed the image graph which reflected the semantic relationships between the images. With this image graph, the images were embedded in a 2-dimensional Euclidean space such that two semantically related images are close to each other. Figure 11 shows the embedding results. Each data point represents an image. Each color stands for a semantic class. Clearly, the image data set were accurately clustered into six categories.

If we use traditional link analysis methods that consider hyperlinks from page to page, the 2-D embedding result is shown in Figure 12. As can be seen, the six categories were mixed together and can be hardly separated. This comparison shows that our image graph model is much more powerful than traditional methods as to describing the intrinsic semantic relationships between WWW images.

Once the image data set is clustered into six semantic categories, we can look into each category in details. Again, using spectral techniques, each category can be visualized in a two-dimensional plane such that the spatial structure reflects its semantic relationship. As an example, we presented the 2-D visualization result of the mammal category in Figure 13 using the second and third eigenvectors. The entries of the first eigenvector are all 1. It contains no information, and hence

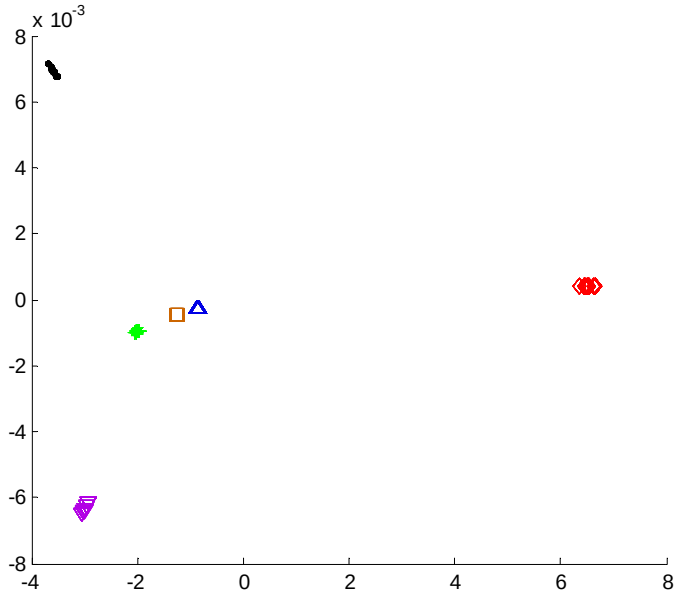


Fig. 11. 2-D embedding of the WWW images using our method. Each color represents a semantic category. Clearly, they are well separated.

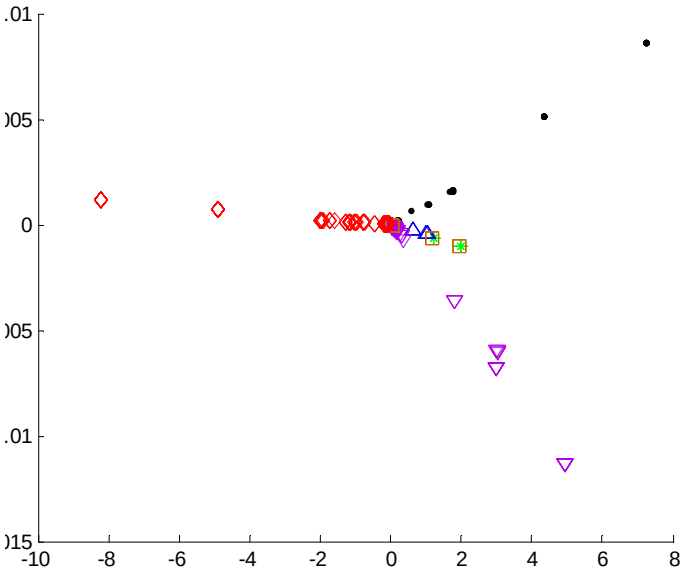


Fig. 12. 2-D embedding of the WWW images. The image graph was constructed based on traditional perspective that the hyperlinks are considered from pages to pages. The image graph was induced from the page-to-page and page-to-image relationships.

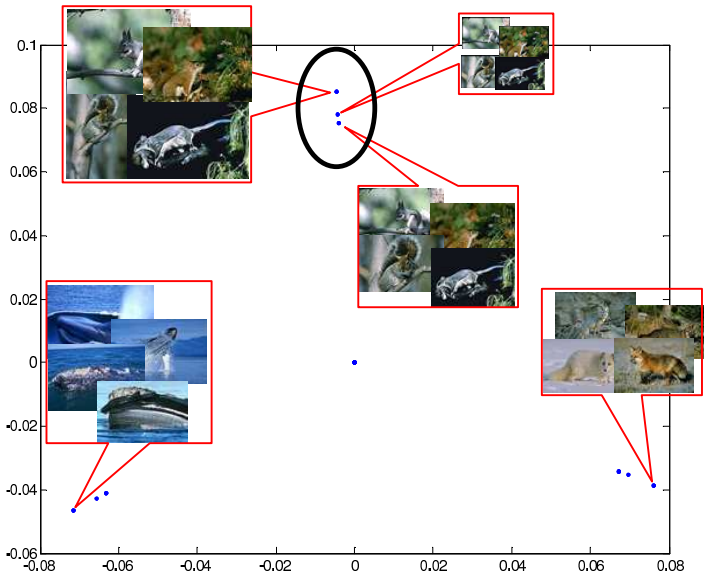


Fig. 13. 2-D visualization of the WWW images using the second and third eigenvectors. Every kind of animal has three images with different sizes, as shown in the circle. Each point represents four images which belong to the same sub-category.

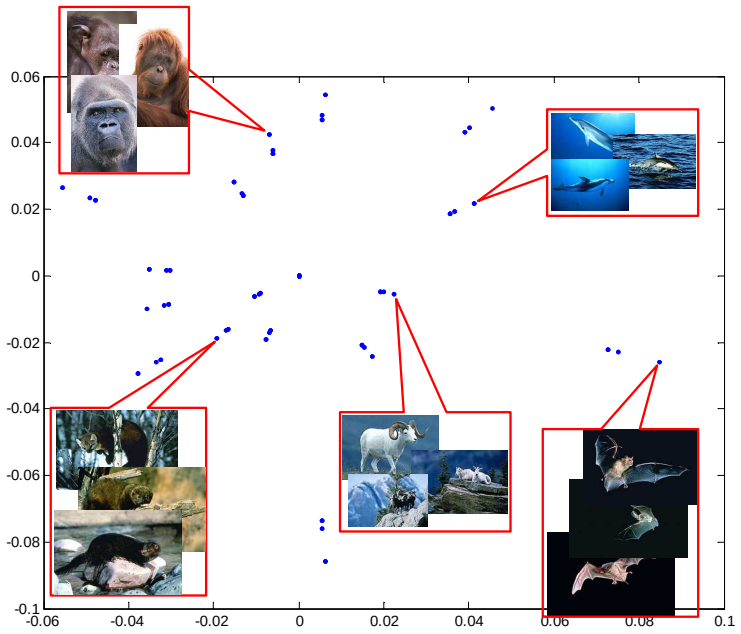


Fig. 14. 2-D visualization of the WWW images using the forth and fifth eigenvectors.

can not be used for image embedding. In this database, every kind of animal has three images with different sizes, as can be seen from the website. It is interesting to note that, in Figure 13, the three points close to each other exactly represent three different sizes. Also, each point represents 4 images which belong to the same sub-category. The 2-D visualization using the forth and fifth eigenvector is given in Figure 14.

## 8. CONCLUSIONS AND FUTURE WORK

In this paper, we described a WWW image search engine called iFind and focused on three fundamental problems, i.e. representation, similarity measure and ranking. For representation, two schemes were proposed. One is based on the textual representation obtained by surrounding text and image file title, etc. The other is based on a vector representation obtained from the image graph model such that if two images have strong link relationship, they are close to each other in the vector space. By constructing the image graph, the weights on the edges provide a similarity measure between images. We do not consider the visual similarity measure in this paper due to its computational complexity and the diversity of the web. By using the notion of random walk on the graph, we compute ImageRanks which is combined with the textual relevance scores for final ranking.

Several questions remain to be investigated in our future work:

(1) In this paper, the relationships between web pages, blocks and images are interpreted as set structure rather than tree structure. However, it might be more natural to interpret a web page as a tree and the images can be viewed as the leaf nodes of the tree. How to incorporate the tree structure into our graph models is an interesting direction to explore.

(2) Several graph models are constructed exclusively from the link structure and page layout. The textual information can also be incorporated into the graph models if computational complexity is not a concern.

(3) The system presented in this paper is a query-by-keywords system. We do not make use of the visual features (color, texture, shape, etc.) of the images. Visual features have been extensively explored in traditional Content-Based Image Retrieval systems. While the visual features might be less reliable than textual information due to the diversity of the web, in some cases they can be more desirable. For example, a painting image is hard to describe in simple text. It remains unclear how to incorporate content information of the WWW images into our search engine.

## REFERENCES

Belkin, M. and Niyogi, P. 2001. Laplacian eigenmaps and spectral techniques for embedding and clustering. In *Advances in Neural Information Processing Systems 14*. Vancouver, Canada.

Brew, C. and Wade, S. 2002. Spectral clustering for german verbs. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. Philadelphia, PA.

Brin, S. and Page, L. 1998. The anatomy of a large-scale hypertextual (web) search engine. In *Proceedings of the 7th ACM Conference on World Wide Web*. Brisbane, Australia.

Cai, D., He, X., Ma, W.-Y., Wen, J.-R., and Zhang, H.-J. 2004. Organizing www images based on the analysis of page layout and web link structure. In *IEEE International Conference on Multimedia and Expo*. Xi'an, China.

Cai, D., He, X., Wen, J.-R., and Ma, W.-Y. 2004. Block-level link analysis. In *Proceedings of the ACM SIGIR Conference on Information Retrieval*. Sheffield, UK.

Cai, D., Yu, S., Wen, J.-R., and Ma, W.-Y. 2003a. Extracting content structure for web pages based on visual representation. In *Proceedings of the 5th Asia Pacific Web Conference*. Xi'an, China.

Cai, D., Yu, S., Wen, J.-R., and Ma, W.-Y. 2003b. Vips: a vision-based page segmentation algorithm. Microsoft Technical Report, MSR-TR-2003-79.

Cai, D., Yu, S., Wen, J.-R., and Ma, W.-Y. 2004. Block-based web search. In *Proceedings of the ACM SIGIR Conference on Information Retrieval*. Sheffield, UK.

Chung, F. R. K. 1997. *Spectral Graph Theory*. Regional Conference Series in Mathematics, vol. 92.

Frankel, C., Swain, M., and Athitsos, V. 1996. Webseer: An image search engine for the world wide web. Technical Report, TR-96-14, Department of Computer Science, University of Chicago.

Google. Google zeitgeist - search patterns, trends, and surprises according to google. http://www.google.com/press/zeitgeist.html.

Guattery, S. and Miller, G. L. 2000. Graph embeddings and laplacian eigenvalues. *SIAM Journal on Matrix Analysis and Applications 21,* 3, 703–723.

He, X., Yan, S., Hu, Y., Niyogi, P., and Zhang, H.-J. 2005. Face recognition using laplacianfaces. *IEEE Trans. on Pattern Analysis and Machine Intelligence 27,* 3, 328–340.

Kleinberg, J. 1999. Authoritative sources in a hyperlinked environment. *Journal of the ACM 46,* 5, 604–622.

Lempel, R. and Soffer, A. 2001. Picashow: Pictorial authority search by hyperlinks on the web. In *Proceedings of the 10th ACM Conference on World Wide Web*. Hong Kong, China, 438–448.

Ma, W.-Y. and Manjunath, B. S. 1996. Texture features and learning similarity. In *IEEE Conference on Computer Vision and Pattern Recognition*. San Francisco, CA, 425–430.

Ma, W.-Y. and Manjunath, B. S. 1999. Netra: a toolbox for navigating large image databases. *Multimedia Systems 7,* 3, 184–189.

Mohar, B. 1997. Some applications of laplace eigenvalues of graphs. In *G. Hahn and G. Sabidussi, editors, Graph Symmetry: Algebraic Methods and Applications*.

Ng, A. Y., Jordan, M., and Weiss, Y. 2001. On spectral clustering: Analysis and an algorithm. In *Advances in Neural Information Processing Systems 14*. Vancouver, Canada.

Robertson, S. E. and Walker, S. 1999. Okapi/keenbow at trec-8. In *Eighth Text Retrieval Conference (TREC-8)*. 151–162.

Rui, Y., Huang, T. S., Ortega, M., and Mehrotra, S. 1998. Relevance feedback: a power tool for interactive content based image retrieval. *IEEE Trans. on Circuit and Systems for Video Technology 8,* 5, 644–655.

Sclaroff, S., Taycher, L., and Cascia, M. L. 1994. Imagerover: a content-based image browser for the world wide web. In *IEEE workshop on content-based access of image and video libraries*. San Juan, Puerto Rico.

Shi, J. and Malik, J. 2000. Normalized cuts and image segmentation. *IEEE Trans. on Pattern Analysis and Machine Intelligence 22,* 8, 888–905.

Smith, J. and Chang, S.-F. 1996. Visualseek: a fully automated content-based image query system. In *Proceedings of the ACM Conference on Multimedia*. New York.

Smith, J. and Chang, S.-F. 1997. Webseek, a content-based image and video search and catalog tool for the web. *IEEE Multimedia*.

Song, R., Liu, H., Wen, J.-R., and Ma, W.-Y. 2004. Learning block importance models for web pages. In *Proceedings of the 13th ACM Conference on World Wide Web*.

Wen, J.-R., Song, R., Cai, D., Zhu, K., Yu, S., Ye, S., and Ma, W.-Y. 2003. Microsoft research asia at the web track of trec 2003. In *Twelfth Text Retrieval Conference (TREC-12)*.