Hierarchical Clustering of WWW Image Search Results Using Visual, Textual and Link Information

Deng Cai[1*] Xiaofei He[2*] Zhiwei Li[*] Wei-Ying Ma[*] and Ji-Rong Wen[*]

[*]Microsoft Research Asia
Beijing, China
{i-zli, wyma, jrwen}@microsoft.com

[1]Computer Science Dept.
UIUC, IL
cai_deng@yahoo.com

[2]Computer Science Dept.
University of Chicago
xiaofei@cs.uchicago.edu

ABSTRACT

We consider the problem of clustering Web image search results. Generally, the image search results returned by an image search engine contain multiple topics. Organizing the results into different semantic clusters facilitates users' browsing. In this paper, we propose a hierarchical clustering method using visual, textual and link analysis. 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. By using block-level link analysis techniques, an image graph can be constructed. We then apply spectral techniques to find a Euclidean embedding of the images which respects the graph structure. Thus for each image, we have three kinds of representations, i.e. visual feature based representation, textual feature based representation and graph based representation. Using spectral clustering techniques, we can cluster the search results into different semantic clusters. An image search example illustrates the potential of these techniques.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval – *Clustering, Query formulation, Retrieval Models.* I.4.10 [**Image Representation**]: Image Processing and Computer Vision – *Multidimensional, Statistical.*

General Terms

Algorithms, Management, Design, Theory

Keywords

Web Image Search, Vision Based Page Segmentation, Spectral Analysis, Image Clustering, Graph Model, Link Analysis, Search Result Organization

1. INTRODUCTION

Existing web image search engines such as Google [12] and AltaVista [1] return a large quantity of search results, ranked by their relevance to the given query. Web users have to go through the list and look for the desired ones. This is a time consuming task since the returned results always contain multiple topics and these topics are mixed together. Things become even worse when one topic is overwhelming but it is not what the user desires.

(a)

(b)

Figure 1. Top 8 returns of query "pluto" in Google's image search engine (a) and AltaVista's image search engine (b)

Let us take a look at a simple example. Figure 1 shows the image search results of the query of "pluto". Note that, the query used in this example is a hot query in image search according to the statistical result of Google image search engine [13]. Clearly, the search results of both image search engines contain two different topics: Pluto in the solar system and the dog named "Pluto" in Disney world. For this query, it is difficult to say which search engine performs better since we do not know what the user is really looking for. In fact, all these results are related to the query. However, in different situations, the results of Pluto about solar system may be noise to the user who is looking for dog Pluto.

A possible solution to this problem is to cluster search results into different groups with different topics. Many works have been done on web text search [19][28][29]. In this paper, we consider the problem of clustering image search results. In web image search, a good organization of the search results is as important as the search accuracy.

In traditional Content-Based Image Retrieval (CBIR) area, image clustering techniques are often used to design a convenient user interface [22], which helps to make more meaningful representations of search results [10]. However, as the images were usually represented by the low level visual features, it is hard to get good clustering result from semantic perspective.

WWW images have a lot of properties which are quite different from those images in small database such as Corel images and family album. Web images are associated with text and link information. In this paper, based on the Vision-based Page Segmentation (VIPS) [7], we consider the image and the block containing

MM'04, October 10–16, 2004, New York, New York, USA.

This work was done while Deng Cai and Xiaofei He were interns at Microsoft Research Asia.

that image as a whole. We propose a framework to represent the WWW images using three kinds of information, i.e. *visual information*, *textual information* and *link information*. And we propose a method to hierarchically cluster WWW image search results based on these image representations.

The rest of this paper is organized as follows. Section 2 relates a list of previous works to our work. In Section 3, we describe three kinds of representation of WWW images and how to extract these representations. Section 4 gives the clustering method using textual feature and graph based representation. Section 5 presents the method to cluster the images using low level visual feature. Some illustrative examples are provided in Section 6. Finally, we give concluding remarks in Section 7.

2. PREVIOUS WORKS

Image Searching and Clustering

Traditional image search [20] and clustering [10][14][22] techniques are content based. They are usually based on *small* and *static* (compared to the Internet) image databases, like family albums. It is still a hard problem to learn the semantic meaning of an image from low level visual features. This makes traditional image retrieval techniques not directly applicable to web image search and organization. Although there exists some systems using traditional CBIR techniques in WWW image search [11][24], they all have the problems of scalability and performance.

Almost all the commercial image search engines [1][12] use the text extracted from HTML pages to index the images. In such cases, the web image search problem is converted to a text search problem. Traditional text retrieval techniques, such as inverted indexing, TF-IDF weighting and cosine similarity measure, etc. can be used for comparing the images to the query keywords. Hyperlink is another kind of information useful for image search and organization in web context. Recently some researchers have used link information to improve image search [18] and image clustering [5].

So far the search problem is the primary focus of research. However, the problem of how to organize the search results is of the same importance [22].

Search Results Clustering

Search result clustering has a long history in information retrieval area [19][29]. Zamir and Etzioni [29] gave a good analysis on the special issues of clustering techniques on search result clustering. The main three issues are: first, the algorithm should take the document snippets instead of the whole documents as input, as the downloading of original documents on the Web is time-consuming; second, the clustering algorithm should be fast enough for online calculation; and third the generated clusters should have readable descriptions for quick browsing by users, etc. Vivisimo [28] is a real demonstration of such technique.

In our work, all the images and corresponding blocks and web pages have been pre-processed. This makes our system able to provide quick response to multiple users' queries simultaneously. By using spectral techniques, our clustering algorithm can be computed in polynomial time (even linear time). The search results are grouped into semantic categories, and for each category we select several representative images which give people a quick understanding of the topics.

Page Layout Analysis for Web Search

Most of previous web-based applications [4][17][18] regard web pages as information units. However, it is the case that a web page often contains multiple semantics. Thus, from the perspective of semantics, a web page should not be the smallest information unit.

To overcome the shortcoming of treating a web-page as a whole unit, many researchers consider segmenting a web-page into different parts [7][9]. Among them, the VIsion-based Page Segmentation (VIPS) algorithm [7] may be a promising one. VIPS aims to extract the semantic structure of a web page based on its visual presentation. Such semantic structure is a tree structure; each node in the tree corresponds to a block. Each node will be assigned a value (*Degree of Coherence*) to indicate how coherent of the content in the block based on visual perception. The VIPS algorithm makes full use of page layout feature. It first extracts all the suitable blocks from the html DOM tree, and then it finds the separators between these blocks. Here separators denote the horizontal or vertical lines in a web page that visually cross with no blocks. Based on these separators, the semantic tree of the web page is constructed. A web page can be represented as a set of blocks (leaf nodes of the semantic tree), for details, see [7]. Compared with DOM based methods, the segments obtained by VIPS are much more semantically aggregated. Noisy information, such as navigation, advertisement, and decoration can be easily removed because they are often placed in certain positions of a page. Contents with different topics are distinguished as separate blocks.

Many research show that VIPS can greatly improve the performance of web search [6][8][26].

3. THREE KINDS OF REPRESENTATIONS FOR WWW IMAGES

Based on the VIPS algorithm, a web-page can be divided into semantic blocks. For each image, there is a smallest block which contains that image. We call it *image block*. The image block contains information that might be useful for describing the image.

Figure 2. The interface of VIPS page segmentation system. The web images are contained in blocks, thus the texts and links in the block can be naturally used to represent the corresponding image.

Figure 2 gives a simple example. As can be seen, web images are contained in blocks, thus the texts and links in that block can be used to represent the corresponding image.

In summary, for each image, three kinds of representations can be derived, i.e. visual feature based representation, textual feature based representation and link graph based representation.

3.1 Visual Feature Based Representation

Image representation using low level features (e.g. color, texture, shapes) has attracted a lot of attentions in CBIR. The most widely used features include color features, such as color correlogram, color moments, color histogram, and texture features, such as Gabor wavelet feature [20]. As the color and texture features capture different aspects of images, their combination may be useful. Yu et.al [27] proposed a novel low-level feature, named **C**olor **T**exture **M**oments (CTM). It integrates the color and texture characteristics of an image in a compact form. Their experimental results showed good performance and more importantly, the dimension of this feature is only 48, much lower than that of many other features. As a result, we use the CTM feature in our system for visual representation of the WWW images.

CTM adopt local Fourier transform as a texture representation scheme and derive eight characteristic maps for describing different aspects of co-occurrence relations of image pixels in each channel of the (SVcosH, SVsinH, V) color space. Then CTM calculate the first and second moments of these maps as a representation of the natural color image pixel distribution, resulting in a 48-dimensional feature vector. CTM can also be regarded as a certain extension to color moments in eight aspects through eight orthogonal templates, see [27] for details. It is important to note that, in our system, the visual features of the images are extracted offline.

3.2 Textual Feature Based Representation

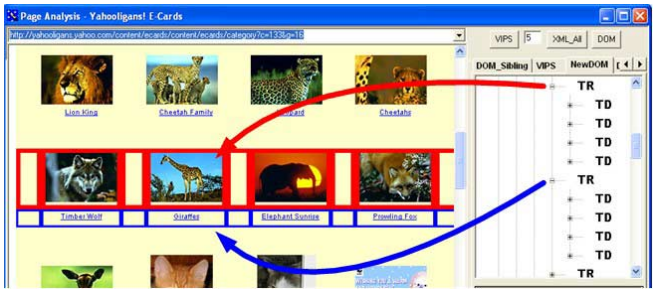
For web image search, the surrounding texts are usually very useful to reveal the semantic meaning of the image. Many commercial search engines [1][12] use surrounding text to index web images. Typically, the file name of the image file, the URL of the image, the image ALT (alternate text) in web page source and the title of the web page which contains that image will be very useful. Besides, the useful texts also include those surrounding text (text close to that image). Generally, there are three methods to extract the surrounding text: window based, DOM based and vision based.

The window based method treats html source as a text stream. For each image (IMG tag in the html source), it uses a fixed-length window to extract the text before and behind the image. The main advantage of this method is its fast speed. However the precision of the extracted text is a problem: HTML page has a 2-D structure, while the window based method only sees its 1-D structure. How to decide the window size is also an issue. It is hard to detect the complete semantic paragraph using some predefined window.

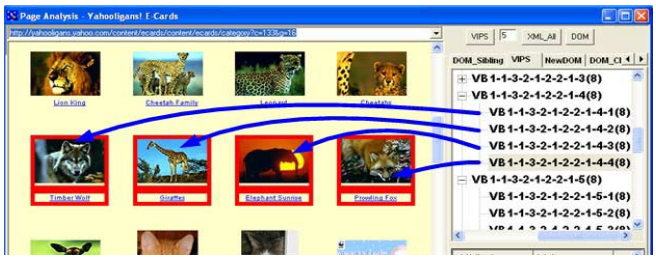
The DOM based method extracts the surrounding text from the HTML DOM[1] tree. In the HTML DOM tree, an image is always a leaf node. The DOM based method uses the text of the sibling nodes as the surrounding text of the image. However, because of the flexibility of HTML syntax, a lot of web pages do not obey the

```
<tr>
<td><img src="t1.jpg"></td><td><img src="t2.jpg"></td>
<td><img src="t3.jpg"></td><td><img src="t4.jpg"></td>
</tr>
<tr>
<td>Timber Wolf</td><td>Giraffes</td>
<td>Elephant Sunrise</td><td>Prowling Fox</td>
</tr>
```

a) Part of HTML source (only keep the backbone)



b) Partition on DOM tree (The red area and blue area are two different TR nodes)



b) Partition using VIPS (The image with their surrounding text are accurately identified)

Figure 3. An sample page. Part of html source; DOM partition result and VIPS partition result

W3C html specifications, which might cause mistakes in a DOM tree structure. Moreover, the DOM tree is initially introduced for presentation in the browser rather than description of the semantic structure of the web page. For example, even though two nodes in the DOM tree have the same parent, it might not be the case that the two nodes are more semantically related to each other than to other nodes (see figure 3 as an example). Furthermore, it is still a problem to identify how many texts should be included, i.e. how many sibling nodes or how many siblings of parent nodes should be included.

Recently, a VIsion-based Page Segmentation (VIPS) algorithm [7] was proposed to partition the web page as different parts. VIPS treats the web page from 2-D view, extract the semantic structure of a web page based on its visual presentation. Such semantic structure is a tree structure; each node in the tree corresponds to a block. Compared with DOM based methods, the segments obtained by VIPS are much more semantically aggregated. Moreover, each node in the VIPS tree will be assigned a value (*Degree of Coherence*) to indicate how coherent of the content in the block based on visual perception. Thus, it is easy to decide which block should be the right image block according to the DoC value. For more details about VIPS, see [7].

[1] http://www.w3c.org/DOM/

Figure 3 shows an example page[1]. Figure 3.a shows part of the html source (we only keep the backbone code). From this code, clearly, the window based method fails to identify the surrounding text for each image. Figure 3.b shows the DOM tree of the page, the four images and four textual parts are in different <TR> node, thus, it is still hard to correctly identify the surrounding text for each image from the DOM tree. Figure 3.c shows the VIPS result. Clearly each leaf node in the VIPS result is an image block. The surrounding texts are accurately identified for each image.

In our system, we use the text in the corresponding image block as the textual representation for each image. Besides surrounding texts, the file name of the image file, the URL of the image, the image ALT (alternate text) in web page source and the title of the web page which contains that image are also used as a part of textual representation. Text features are also extracted in the pre-processing stage, which does not cost on-line time.

In our system, these texts are also used for image indexing, which makes our system like a commercial image search engine which can give fast query responses to a possibly huge number of users.

3.3 Link Graph Based Representation

Hyperlink is another kind of useful information in web context. Most of previous web-based applications [4][17] regard web pages as information units, thus links are from page to page. Recently, a block level link analysis technique was introduced in web search and show promising result [6]. In block level link analysis framework, the links are from blocks to pages.

In this sub-section, we briefly describe how to construct an image graph whose weights defined on the edges reflect semantic relationships between images. More details can be found in [5][15]. We begin with some definitions. Let P, B, and I denote the set of all the web pages, all the blocks, and all the images, respectively. $P = \{p_1, p_2, \ldots, p_k\}$, where k is the number of web pages. $B = \{b_1, b_2, \ldots, b_n\}$, where n is the number of blocks. $I = \{I_1, I_2, \ldots, I_m\}$, where m is the total number of the web images. $b_i \in p_j$ means the block i is contained in the page j. Similarly, $I_i \in b_j$ means the image i is contained in the block j.

3.3.1 The Relationships between Page, Block and Image

The **page-to-block** relationships are obtained from page layout analysis. Let X denote the page-to-block matrix with dimension $k \times n$.

$$X_{ij} = \begin{cases} f_{P_i}(b_j) & if\ b_j \in p_i \\ 0 & otherwise \end{cases} \quad (1)$$

where f is a function which assigns to every block b in page p an importance value. f is empirically defined below,

$$f_p(b) = \alpha \frac{\text{size of block } b \text{ in page } p}{\text{dist. from the center of } b \text{ to the center of screen}} \quad (2)$$

where α is a normalization factor to make the sum of $f_p(b)$ to be 1.

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

The **block-to-page** relationships are obtained from link analysis. Let Z denote the block-to-page matrix with dimension $n \times k$. Z can be formally defined as follows:

$$Z_{ij} = \begin{cases} 1/s_i & if\ there\ is\ a\ link\ from\ block\ i\ to\ page\ j \\ 0 & otherwise \end{cases} \quad (3)$$

where s_i is the number of pages that block i links to.

Let Y denote the **block-to-image** matrix with dimension $n \times m$. Y can be simply defined below:

$$Y_{ij} = \begin{cases} 1/s_i & if\ I_j \in b_i \\ 0 & otherwise \end{cases} \quad (4)$$

where s_i is the number of images contained in the image block b_i.

3.3.2 Image Graph Construction

When we get the three relationship matrix of page, block and image, it is easy to construct the block graph from which the image graph can be further induced. Let W_B denote the weight matrix of the block graph. The definition of W_B is as follows

$$\begin{aligned} W_B(a,b) &= Prob(b \mid a) \\ &= \sum_{\gamma \in P} Prob(\gamma \mid a) Prob(b \mid \gamma) \\ &= Prob(\beta \mid a) Prob(b \mid \beta) \\ &= Z(a,\beta) X(\beta,b), \quad a,b \in B \end{aligned} \quad (5)$$

or

$$W_B = ZX \quad (6)$$

where W_B is a $n \times n$ matrix. n is the number of blocks.

Based on the block graph, the weight matrix of the image graph can be defined as follows:

$$W_I(i,j) = \sum_{i \in \alpha, j \in \beta} W_B(\alpha, \beta) \quad (7)$$

or

$$W_I = Y^T W_B Y \quad (8)$$

where W_I is a $m \times m$ matrix. m is the number of images.

3.3.3 Image Representation on Graph

Once we get the weight matrix of the image graph, it is easy to generate a vector representation for each image using spectral graph theory. The most well known method is Eigenmaps [3].

We first need to convert the W_I to a similarity matrix S such that $S = 1/2(W_I + W_I^T)$ which is symmetric. The eigenmaps can be obtained by solving the following eigenvalue problem:

$$L\mathbf{y} = \lambda D\mathbf{y} \quad (9)$$

where D is a diagonal matrix whose i^{th} element is the row (or column, since S is symmetric) sum of S, $D_{ii} = \sum_j S_{ij}$. $L = D - S$. L is generally called Laplace matrix, or graph Laplacian. The first k eigenvectors associated with the first k smallest eigenvalue of the equation (9) give each image a vector representation in the k-dimensional Euclidean space.

4. CLUSTERING USING TEXTUAL AND LINK INFORMATION

Most previous work on image clustering use visual features [10] [14][22]. They are usually based on *small* and *static* image databases (compared to the Internet). It is still a open problem to learn the semantic meaning of an image from low level visual features. This makes traditional image clustering techniques not directly applicable to web image search results.

Fortunately, in web image context, we have two other representations: textual based and link graph based. These two representations can reflect the semantic relationship of images.

Thus, in our system, the search result clustering algorithm is implemented as a two level clustering algorithm. The first level is clustering using textual and link representation of images. We can get some semantic categories. The second level is for each cluster result of the first level. We use low level visual feature to cluster each semantic category. Although we use the term "clustering" in the second level, yet our real goal is not to acquire different semantic clusters but to re-organize the images to make visually similar images be grouped together to facilitate user's browsing.

4.1 Clustering on Textual Feature

Once images are represented by textual information, image clustering becomes document clustering. Therefore, we use the cosine similarity which proved very effective in information retrieval community [2].

In our work, we need to automatically determine the number of clusters. Due to this concern, we use spectral clustering techniques which are adapted from [21][23]. In spectral clustering we can determine the number of clusters according to the gaps between two consecutive eigenvalues [23].

Suppose we have n points, $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$, which have unit norm. Our algorithm can be stated below:

1. Construct an $n \times n$ affinity matrix S, $S_{ij} = \mathbf{x}_i^T \mathbf{x}_j$ if $\mathbf{x}_i$ is among the h nearest neighbors of $\mathbf{x}_j$ or $\mathbf{x}_j$ is among the h nearest neighbors of $\mathbf{x}_i$, otherwise, $S_{ij} = 0$.
2. Define D to be diagonal matrix whose (i, i)-element is the sum of S's i-th row, and construct the matrix $L = D - S$.
3. Solve the generalized eigenvalue problem $L\mathbf{y} = \lambda D\mathbf{y}$. Let $(\mathbf{y}^0, \lambda^0)$, $(\mathbf{y}^1, \lambda^1)$, …, $(\mathbf{y}^{n-1}, \lambda^{n-1})$ be the solutions to the equation, and $\lambda^0 < \lambda^1 < \ldots < \lambda^{n-1}$.
4. Find the largest Eigengap. Eigengap is defined as the difference of the two consecutive eigenvalues ($\Delta^i = \lambda^i - \lambda^{i-1}$). Suppose Δ^k is the largest one. Using the first k eigenvector to form the matrix $Y = [\mathbf{y}^0\ \mathbf{y}^1\ \ldots \mathbf{y}^{k-1}] \in R^{n \times k}$ by stacking the eigenvectors in columns.
5. Treat each row of Y as a point in R^k, and cluster them into k clusters via **K**-means.

The textural representation of image always conveys the semantic meaning of the image, thus clustering using text feature usually reflects the semantic relationships.

The drawback of clustering using text feature is that sometimes the surrounding text is too little to accurately represent the image, thus some images may be mis-clustered. In the next subsection, we discuss the use of image link graph for clustering.

4.2 Clustering on Image Graph

When using the graph based representation of image, we first need to construct the image graph of the search results. Similar situations in web search and image search are discussed in HITS [17] and PicASHOW [18]. Both of these used the graph for ranking while here we use it for clustering.

Suppose we have a list of image search results. If we only consider the pages (like root set in HITS [17]) containing these images, in our image graph construction framework, these images might not have any relationship. Thus we need to expand the root set to a base set which consists of pages in the root set, pages that point to a page in the root set, and pages that are pointed to by a page in the root set. Finally, we get a set of pages and a set of images. According to Section 3.3, we can construct an image to image graph. Note that all the pages are preprocessed and the relationship of pages, blocks and images are already extracted. So the image to image graph construction can be very fast.

Once we obtain the image graph W_I, the clustering algorithm described in the previous section can naturally be applied. The only difference is that the affinity matrix S is described as $S = 1/2(W_I + W_I^T)$.

In real world applications, we find the image graph is formed by many disconnected sub-graph. Each sub-graph corresponds to a cluster. Thus there might be a large number of clusters. We consider the problem of combining those clusters with the same semantics in the next subsection.

4.3 Combining Textual and Link Information

Obviously, the textual and graph based representation can be combined to achieve a better result. The disadvantage of using textual only is that some images have few surrounding texts, and the disadvantage of using link graph only is that those pages talking about the same topic do not have link between them. Combining both information can help overcome the limitation of each method.

In our clustering algorithm, combining textual and link information is very natural. Note that in the first step of our clustering algorithm, we construct an affinity matrix S, which indeed reflects the relationship between images. Thus, in order to combine the textual and link information, we only need to combine the two affinity matrix S. We use the following modified definition:

$$S_{combine}(i, j) = \begin{cases} S_{textual}(i, j) & if\ S_{link}(i, j) = 0 \\ 1 & if\ S_{link}(i, j) > 0 \end{cases} \qquad (10)$$

5. CLUSTERING USING VISUAL FEATURE

In the previous section, we use the textual feature based representation and graph based representation to cluster the search result into different semantic clusters. However, in each semantic cluster, although the images are related to the same topic, the visual perception might not be good because of the different colors and shapes of these images. Previous work [22] indicates that arranging a set of images according to their visual similarity does indeed to be useful. Thus we consider performing clustering once more in each semantic cluster using low level visual feature.

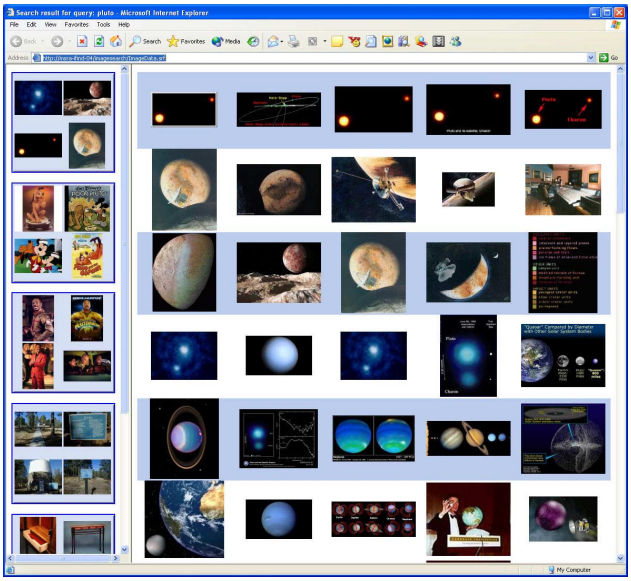


Figure 4. The interface of our system. The left frame shows the clusters and the right frame shows the images in the selected cluster. The query is "pluto".

The algorithm we used is simply the same as that in section 4.1, while we use L_1-distance to define the similarity instead of cosine similarity due to different properties of representation [25].

6. ILLUSTRATIVE EXAMPLE

In this section we use an example query to illustrate the potential of the techniques described in this paper. The purpose of this section is to provide people with an intuition on how our system works based on the techniques we described previously.

6.1 Data Preparation and System Overview

All the data used in our experiments are crawled from the Internet. Starting from Yahoo! Directory Photography Museums and Galleries

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

We crawled 26.5 millions web pages in total by breath first crawling. From these web pages, images are 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 quantity. We also removed those images whose width and height are both smaller than 60 pixels due to the same reason. Finally, we are left with 11.6 millions images.

For each web page, the VIPS page segmentation algorithm was applied to divide it into blocks. For each block, the hyper-links were extracted. For each image, the image blocks containing that image were identified and the surrounding texts were extracted within these image blocks and used to index these images. Thus, we built a real web image search system. More details about our system can refer to [15].

When the user submits a query, the system first computes the relevance score (based on the surrounding text) for every image and the images are ranked according to their relevance scores. Then we cluster the top N (in our system, N = 500) images and present them to the user.

Finally, the interface of our system will be a two frame web page, see Figure 4. The left frame shows the different clusters. Each cluster is represented as a 4-image thumbnail, and the 4 images are selected from the corresponding cluster using ImageRank [16]. These clusters are generated in the first level clustering in our framework using link and textual information. The right frame shows the images of the selected cluster. The images are re-arranged using the second level clustering based on the low level visual features.

6.2 Clustering Result

In this sub-section, some clustering results using different representations are provided to demonstrate the effectiveness of our system. In all the following experiments, we use the same query of "pluto" and we take the number of nearest neighbors to be 10 in our computations.

6.2.1 Clustering Using Visual Feature

Figure 5 is part of the clustering result using low level feature. From the perspectives of color and texture, the clustering results are quite good. Different clusters have different colors and textures. However, from semantic perspective, these clusters make little sense.

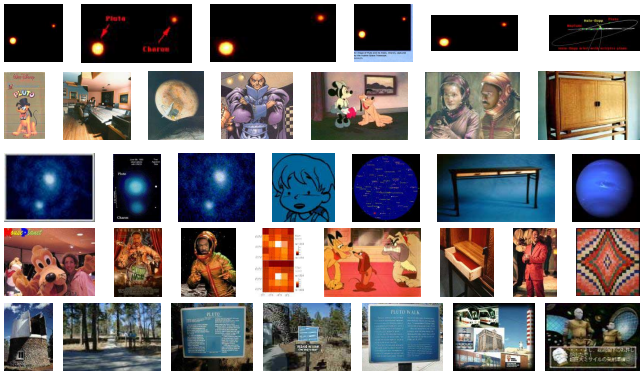


Figure 5. Five clusters of search results of query "pluto" using low level visual feature. Each row is a cluster.

6.2.2 Clustering Using Textual Feature

To determine the number of clusters, we use the Eigengap which is defined as the difference of the two consecutive eigenvalues in our algorithm. We use "Pluto" as our query. Figure 6 shows the Eigengap as a function of k. Clearly, there are two peaks. One is at k = 6, and the other is at k = 21. This indicates the search results can be grouped into 6 categories or 21 categories using the textual feature.

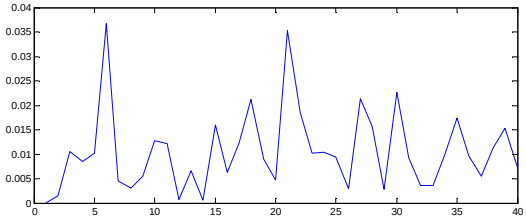


Figure 6. The Eigengap curve with k for the "pluto" case using textual representation

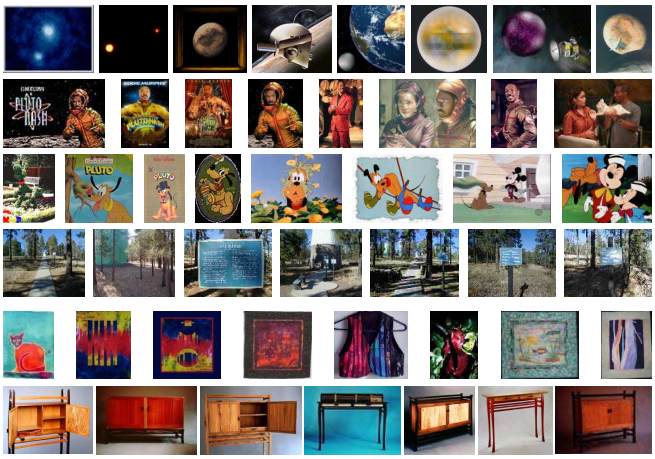


Figure 7. Six clusters of search results of query "pluto" using textual feature. Each row is a cluster

When we choose k = 6, the search results are grouped into six categories, as shown in figure 7. From the semantic prospective, these clusters are far better than figure 5 which used low level visual feature. Clearly, there are six semantic concepts in the results. The first category is about Pluto of solar system, having 157 images; the second category contained 46 images about a movie "The adventures of Pluto Nash: The man on the moon"; the third category is about the carton figure Pluto, having 70 images; the fourth category contained 110 images about a theme park of Pluto; the fifth category contained 28 images on site "http://pluto.njcc.com/~lfrankel/" and the sixth category contained 89 images on the site "http://pluto.njcc.com/~jhein/". These images are retrieved because the URL of these images contain the word of "Pluto".

When we try to cluster the search results into 21 clusters, some clusters above were further split into smaller clusters, even though the images are related to the same topic. This might be due to the use of textual feature only.

6.2.3 Clustering Using Graph Based Representation

As we mentioned in Section 4.2, clustering using only graph based representation always generate too many clusters. In "pluto" case, the top 500 results are clustered into 167 clusters. The max cluster number is 87, and there are 112 clusters with only one image. Figure 8 shows part of the clusters.

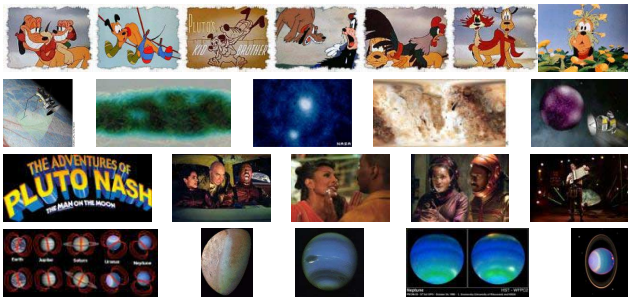


Figure 8. Five clusters of search results of query "pluto" using image link graph. Each row is a cluster

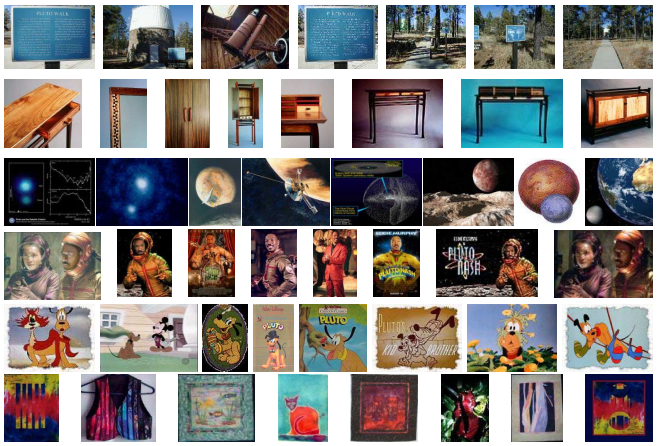


Figure 9. Six clusters of search results of query "pluto" using combination of textual feature and image link graph. Each row is a cluster

6.2.4 Combining Textual Feature and Link Graph

In this sub-section, we combined the textual and link features. Figure 10 shows the Eigengap as a function of k. Clearly, there is only one peak value at k = 6. Thus the combination of textual feature and image link graph actually reveal the semantic structure of the image set (image search results). Figure 9 shows some sample images from each semantic category.

For each semantic category, we can further re-organize them using visual feature. Figure 11 shows the re-organized semantic category "Pluto in solar system" by using the low level visual features. This makes it more comfortable for user's browsing.

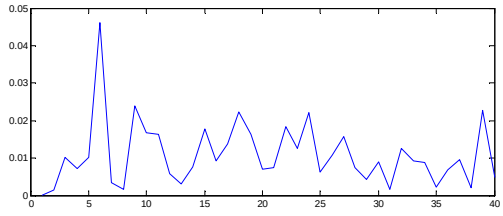


Figure 10. The Eigengap curve with k for the "pluto" case using textual and link combination

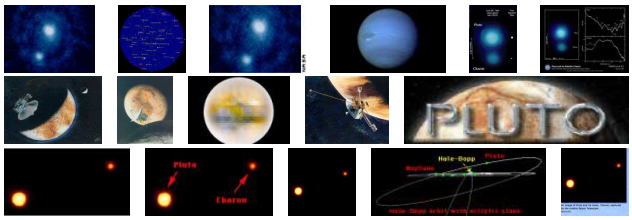


Figure 11. Reorganization result of the category "Pluto in solar system" using low level visual feature.

6.3 Discussion

How to determine the number of clusters is still an open problem. Many works on clustering assume the number of clusters is given [14][21]. While in image search result clustering, it is almost impossible to determine the number of clusters before clustering. In spectral clustering settings, we can use the difference of the consecutive eigenvlaues to determine the number of clusters while the

performance of this method greatly depends on the graph (affinity matrix S in our case) structure. Combining the textual and link based representations, we can actually reveal the semantic structure of the web images.

The example query in the previous section illustrates the potential of the techniques described in this paper. In order to develop a more detailed knowledge of the strengths and robustness of our techniques, a more thorough experimental evaluation of our system will be carried out in future work.

7. CONCLUSIONS

In this paper, we described a method to organize WWW image search results. Based on the web context, we proposed three representations for web image, i.e. representation based on visual feature, representation based on textual feature and representation induced from image link graph. Spectral techniques were applied to cluster the search results into different semantic categories. For each category, several images were selected as representative images according to their ImageRanks, which enables the user to quick understanding the main topics of the search results. The illustrative example show that the combination of textual feature based representation and graph based representation actually reflects the semantic relationships between web images. And the reorganization of each cluster based on visual features makes the clusters more comfortable to the users.

8. REFERENCES

[1] AltaVista image search, http://www.altavista.com/image/

[2] R. Baeza-Yates and B. Ribeiro-Neto, Modern Information Retrieval, Addison Wesley Longman 1999.

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

[4] S. Brin and L. Page, "The anatomy of a large-scale hypertextual (Web) search engine", In *The Seventh International World Wide Web Conference*, 1998.

[5] D. Cai, X. He, W.-Y. Ma, J.-R. Wen and H.-J. Zhang. "Organizing WWW Images Based on The Analysis of Page Layout and Web Link Structure", in *The 2004 IEEE International Conference on Multimedia and EXPO*, 2004.

[6] D. Cai, X. He, J.-R. Wen, and W.-Y. Ma, "Block-level Link Analysis", in *The 27th Annual International ACM SIGIR Conference on Information Retrieval*, 2004.

[7] D. Cai, S. Yu, J.-R. Wen, and W.-Y. Ma, "VIPS: a vision-based page segmentation algorithm", *Microsoft Technical Report*, MSR-TR-2003-79, 2003.

[8] D. Cai, S. Yu, J.-R. Wen, and W.-Y. Ma, "Block-based Web Search", *The 27th Annual International ACM SIGIR Conference on Information Retrieval*, 2004.

[9] S. Chakrabarti, "Integrating the Document Object Model with hyperlinks for enhanced topic distillation and information extraction", In the *10th International WWW Conference*, 2001.

[10] Y. Chen, J. Z. Wang, and R. Krovetz. "Content-based image retrieval by clustering". In Proceedings of the *5th ACM SIGMM international workshop on Multimedia information retrieval*, pages 193–200. ACM Press, 2003.

[11] C. Frankel, M. Swain, and V. Athitsos, "WebSeer: An image search engine for the world wide web", *TR-96-14*, Department of Computer Science, University of Chicago, 1996.

[12] Google image search engine, http://images.google.com/

[13] Google Zeitgeist - Search patterns, trends, and surprises according to Google, (2004) http://www.google.com/press/zeitgeist.html

[14] S. Gordon, H. Greenspan, and J. Goldberger. "Applying the information bottleneck principle to unsupervised clustering of discrete and continuous image representations". In *ICCV*, 2003.

[15] X. He, D. Cai, J.-R. Wen, W.-Y. Ma and H.-J. Zhang, "ImageSeer: Clustering and Searching WWW Images Using Link and Page Layout Analysis", *Microsoft Technical Report*, MSR-TR-2004-38, 2004.

[16] X. He, W.-Y. Ma, and H. J. Zhang, "ImageRank: spectral techniques for structural analysis of image database", *IEEE International Conference on Multimedia and Expo*, 2003.

[17] J. Kleinberg, "Authoritative sources in a hyperlinked environment", *Proc. 9th ACM-SIAM Symposium on Discrete Algorithms*, 1998.

[18] R. Lempel and A. Soffer, "PicASHOW: Pictorial authority search by hyperlinks on the web", *Proc. 10th Int. World Wide Web Conf.*, pp. 438-448, Hong Kong, China, 2001.

[19] A. V. Leouski and B. Croft, An Evaluation of Techniques for Clustering Search Results. *Technical Report IR-76*, Computer Science Dept., University of Massachusetts, 1996.

[20] B. S. Manjunath, W. -Y. Ma, "Texture Features for Browsing and Retrieval of Image Data", *IEEE Trans on PAMI*, Vol. 18, No. 8, pp. 837-842, 1996.

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

[22] K. Rodden, W. Basalaj, D. Sinclair, and K. R. Wood. Does organisation by similarity assist image browsing? In *Proceedings of Human Factors in Computing Systems*, 2001.

[23] J. Shi and J. Malik, "Normalized cuts and image segmentation", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8), pp. 888-905, 2000.

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

[25] M. Stricker and M. Orengo, "Similarity of color images", *Proc. Storage and Retrieval for Image and Video Databases,SPIE 2420*, pp. 381-392, 1995.

[26] S. Yu, D. Cai, J.-R. Wen, and W.-Y. Ma, "Improving pseudo-relevance feedback in web information retrieval using web page segmentation", *Proc. 12th World Wide Web Conference*, Budapest, Hungary, 2003.

[27] H. Yu, M. Li, H.-J. Zhang, and J. Feng. Color texture moments for content-based image retrieval. In *International Conference on Image Processing*, pages 24–28. 2002.

[28] Vivisimo clustering engine, (2004) http://vivisimo.com.

[29] O. Zamir and O. Etzioni, "Grouper: A Dynamic Clustering Interface to Web Search Results". In Proceedings of *the Eighth International World Wide Web Conferenc*，1999.