Grouping Web Image Search Result

Xin-Jing Wang[1,2] wxj01@mails.tsinghua.edu.cn
Wei-Ying Ma[1] wyma@microsoft.com
Qi-Cai He [1,3] heqicai@pku.edu.cn
Xing Li[2] xing@cernet.edu.cn

Microsoft Research Asia[1]
Department of Electronic Engineering, Tsinghua University, Beijing, China [2]
LMAM, Department of Information Science, School of Mathematical Sciences, Beijing, China [3]

ABSTRACT
In this paper, we propose a Web image search result organizing method to facilitate user browsing. We formalize this problem as a salient image region pattern extraction problem. Given the images returned by Web search engine, we first segment the images into homogeneous regions and quantize the environmental regions into image codewords. The salient codeword "phrases" are then extracted and ranked based on a regression model learned from human labeled training data. According to the salient "phrases", images are assigned to different clusters, with the one nearest to the centroid as the entry for the corresponding cluster. Satisfying experimental results show the effectiveness of our proposed method.

Categories and Subject Descriptors
H.5.3 [**Information Interfaces and Presentation]**: Group and Organization Interfaces – *Organizational design.* H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval – *Clustering.*

General Terms
Algorithms, Performance

Keywords
Search result organization, image clustering, image representation, regression analysis

1. INTRODUCTION
Existing image search engines such as Google and Yahoo! return unorganized list of images based on keyword matching. These images often have various perception properties. For example, Figure 1 shows a Google search result of query "prairie dog". There are cartoons, prairie dogs on green grassland and prairie dogs with yellow earth. Such unordered browsing structure makes it an effort for users to surfing before getting to their interested images.

A possible solution is to assign the images returned by a search engine into homogeneous groups and select one representative image for each group as the entry to guide users' browsing (as shown in Figure 2, which is a real example of our proposed approach for query "yellow-headed blackbird"). Similar ideas are addressed in text mining field where many effective and efficient clustering technologies are conducted for search result organization [8]. A real example is vivisimo search engine [6].

As to the authors' knowledge, few researches [3][4][5] have been done for online image search result organization. A main difference between database image clustering and Web image search result clustering is that images in the later one are assumed of the same concept.

In [3], a system called AMORE is constructed which supports user interaction, text-based image clustering, and color and composition-based image clustering. In [5], different snippets inferred from the query terms are used as the clues to cluster search results. In [4], the authors extract so-called invariant features and do k-means and LBG clustering based on them.

Figure 1. Google Search Results for Query "prairie dog"

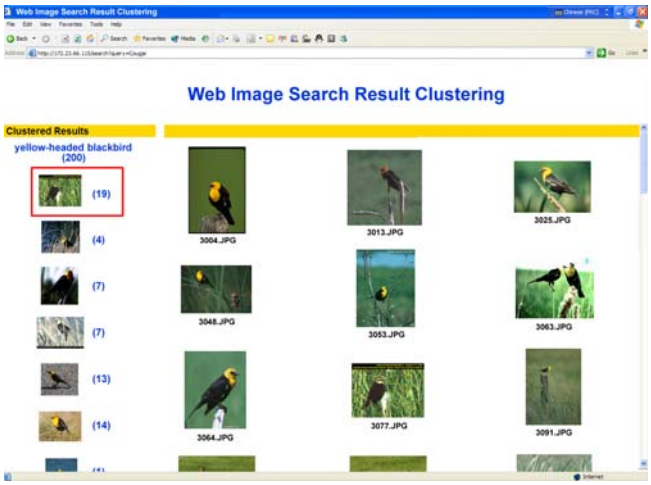

Figure 2. Web Image Result Clustering Interface. The Figure shows the Output of Query "Yellow-headed blackbird"

MM'04, October 10–16, 2004, New York, NY, The United States

In this paper, we assume each image is constructed by two kinds of regions: key region representing the main semantic content (e.g. the prairie dogs in Figure 1) and environmental region representing the context (e.g. earth, grasslands). Because the returned images by Web search engines are assumed to have the same semantic content, it is the context of one image that determines to which cluster it should be assigned. In our approach, we solve the organization problem by extracting salient "patterns" from image contexts. Each pattern names one cluster. An Image is then assigned to the cluster whose name best matches its context patterns. The method is reasonable in that, assume each image region is quantized to and represented by a pre-learned codeword list, intuitively, images with similar visual appearances will have similar codeword set, and the more codewords in common for two images, the more similar these images are. The key technology is to find those representative codeword set.

2. IMAGE PATTERN EXTRACTION

As mentioned above, we assume each image is constructed by key region and context regions. In this section, we present how we extract the context regions from one image and create candidate region patterns based on them.

2.1 Context Region Extraction

The extraction of context region is formulated as a two-step filtering process in our approach. First, the key regions are cut off using the attention model proposed by [7]. Then small regions are further filtered to both refine the result of key region filtering and avoid false-positives in salient pattern identification.

2.1.1 Key Regions Filtering

Figure 3 shows two examples of image saliency maps output by the attention model [7] (the 2nd image in each row). In this model, the attention value of each pixel is represented by a float number normalized to (0, 1). Because each image is segmented into homogeneous regions using JSEG algorithm [1], we define one region as a key region if its saliency, i.e. the average attention value of pixels enclosed by this region, exceeds a certain threshold λ_1 . Because an accurate threshold is difficult to select, λ_1 is set a higher value (currently 0.65) to avoid too much context pixels being falsely filtered as of the key regions.

Let R_{key} be the resulted key region. Let r_k denote the k^{th} region in an image and $s(r_k)$ denote its saliency, the key regions are given by:

$$R_{key} = \{r_k \mid s(r_k) > \lambda_1, 1 \le k \le N\} \quad (1)$$

where N is the number of regions contained in the image.

2.1.2 Small Regions Filtering

For the rest of the regions $\{r_k \mid 1 \le k \le N\} - R_{key}$, we calculate their average region size λ_2 and drop those regions whose size are under this threshold. This threshold is severe because it is the dominating regions which affect the human judgment of the content of one image.

In short, the final extracted context regions R_{con} is given by

$$R_{con} = \{r_k \mid s(r_k) \le \lambda_1,\ |r_k| \ge \lambda_2,\ 1 \le k \le N\} \quad (2)$$

Figure 3 shows the context regions extracted (the last image in each row). It can be seen that the main environmental information

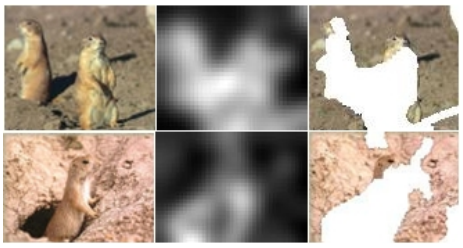

Figure 3. Examples of Saliency Map Given by [7]

are kept.

2.2 Candidate Phrase Generation

For every region of images output by the process in Section 2.1, we extract a set of low-level features (i.e. 36-bin color Correlograms) to represent it. We then quantize the regions into a set of codewords using k-means clustering. In this way, each image becomes a bag of unduplicated codewords, i.e. each codeword can appear no more than once in an image. This step is to avoid a semantic object being segmented into many regions by JSEG algorithm.

We then extract codeword phrases which are N-grams of codewords. Note that the N-grams in our approach are unordered because it is very difficult, if not impossible, to order the image regions. We extract all possible bi-grams and tri-grams plus the original uni-grams (i.e. the codewords) for each image. Hence for an image of three codewords, $C_3^1 + C_3^2 + C_3^3 = 7$ different N-grams are produced. In this way, a set of candidate codeword phrases is obtained.

3. CRITERIONS FOR SALIENT PHRASES

Given the set of candidate codeword phrases, we extract several properties to measure their saliency. The most salient phrases are then used to form image clusters.

Let w be a candidate phrase and $D(w)$ the set of images that contains w hereafter, we adopt five properties as the following:

Inverted Document Frequency (IDF)

Intuitively, if a phrase appears in most of the images, it might be less informative than those only appear in a few images. This is analogues to TF*IDF in text mining area. However, because one phrase can appear no more than once per image, only the IDF part is useful in our case:

$$IDF = \log \frac{N}{|D(w)|} \quad (3)$$

where N is the number of images.

Phrase Length (LEN)

The phrase length is the number of codewords in a phrase. Generally, a longer phrase will give more complete information about an image.

$$LEN = n \ \ (1 \le n \le 3) \quad (4)$$

Phrase Importance (IMP)

Intuitively, a region is important if it covers a large area of image. Let r be one region mapping to codeword c and imp_c is the

codeword importance, we define imp_c as the average region importance (i.e. its percentage to the size of image) given by:

$$imp_c = \frac{1}{|D(w)|}\sum_{r\in I_j, I_j\in D(w)}\frac{|r|}{|I_j|} \quad (5)$$

Where $|D(w)|$ is the size of $D(w)$ and I_j is the j^{th} image in $D(w)$. Note that it is possible that several regions are projected to a same codeword in one image.

The phrase importance is given by the average codeword importance which forms this phrase:

$$IMP = \frac{1}{LEN}\sum_{c\in w} imp_c \quad (6)$$

Cluster Compactness (CC)

As mentioned above, similar images will be mapped to similar set of codewords. Intuitively, if the set of images (i.e. $D(w)$) is compact, it means w is a good representation for these images. Because each image is now converted to a bag of codewords, we use the vector space model to express the images: $\mathbf{I}_i = (x_{i1}, x_{i2}, \ldots)$. Each component of this vector is a distinct codeword weighted by TF*IDF. The centroid of $D(w)$ is then given by

$$\mathbf{o} = \frac{1}{|D(w)|}\sum_{\mathbf{I}_i\in D(w)}\mathbf{I}_i \quad (7)$$

CC is calculated as the average cosine similarity between each image and the centroid.

$$CC = \frac{1}{|D(w)|}\sum_{\mathbf{I}_i\in D(w)}\cos(\mathbf{I}_i, \mathbf{o}) \quad (8)$$

Cluster Overlap Entropy (COE)

Intuitively, if w is a salient phrase, the overlap between $D(w)$ and other image sets will be small. We use Cluster Overlap Entropy to represent the distinctness of w, where we define $0\cdot\log 0 = 0$.

$$COE = -\sum_t \frac{|D(w)\cap D(t)|}{|D(w)|}\log\frac{|D(w)\cap D(t)|}{|D(w)|} \quad (9)$$

Where $D(t)$ denotes any image clusters other than $D(w)$. $|D(w)\cap D(t)|$ denotes the number of images overlapped in these two clusters.

4. IMAGE CLUSTERING

Given the above five properties, each phrase is represented by a vector of the five properties: $\mathbf{x} = (IDF, LEN, IMP, CC, COE)$, we then assign each phrase a real-valued score y (i.e. the salience score) using a pre-learned regression model. The higher the score, the more salient the phrase is.

We keep at most top 20 phrases as the final salient phrases to partition the entire image dataset. Note that images in the resulted clusters may overlap because there may be several dominating context regions in one image, e.g. a lighthouse image with background of sky and sea. For those images fail to be assigned to any of the top clusters, we put them to an "other" category to denote the exceptions. If one salient phrase is a snippet of another higher-ranked phrase, the images will then be assigned to the cluster named by the latter one.

We select the image which has the smallest Euclidean distance to the centroid as a cluster's entry.

5. EXPERIMENTS

We pass the queries listed in Table 1 to Google for the initial collections of search results. The former 200 results are crawled for each query.

Table 1. The Fifteen Queries Used in Evaluation

aircraft, antelope, black bear, bridge, butterfly fish, Egypt pyramid, gull, horse, house, lighthouse, mars, palm tree, plane, prairie dog, yellow-headed blackbird

5.1 Regression Model Training

We use 50 images of 10 queries each as the training dataset. For each query, we extract all possible N-grams as described in Section 2.2 and filter out those phrases with frequency no greater 3. For each phrase, we create its corresponding cluster as described in Section 4. We then ask 5 evaluators to select the good phrases according to the semantic uniformity of the images in one cluster. The final good phrases are selected by majority-voting. We then set $y = 1$ for good phrases and -1 for the others.

We use SVM^{light} [2] to do the support vector regression. Three different kernel functions are tried: linear kernel, RBF kernel and sigmoid tanh kernel. A three-fold cross validation is taken to evaluation the average performance of the three regression models and we found that best performance is obtained for the RBF kernel model with option "-g 0.2".

5.2 Evaluation Measure

We use the traditional cluster entropy in information theory to evaluate the clustering accuracy. The lower the entropy, the better the performance. Because the number of clusters given by our approach for each query is various, to make the result comparable, we average the total cluster entropy on the number of clusters produced. Specifically, given a cluster A and category labels of data objects inside it, the entropy of cluster A is defined as $H(A) = -\sum_j p_j \cdot \log p_j$, where p_j is the proportion of current cluster results appearing in the ground truth cluster j. Let C be the set of produced clusters, the total entropy is defined by

$$H = \frac{1}{|C|}\sum_{A_k\in C} H(A_k) \quad (10)$$

where A_k denotes the k^{th} cluster.

We use the traditional kmeans clustering method based on global images as the baseline method. The parameter k is set to be an overall optimal value 10. The features extracted are 36-bin color Correlograms.

5.3 Experimental Results

Figure 4 shows the performance of our proposed method vs. the baseline method. The rightmost column shows the average entropy on the 15 queries.

From Figure 4, it can be seen that our method exceeds the baseline method a great deal on most of the queries. A zero entropy is achieved on the query "palm tree" for both the methods which means a 100% clustering accuracy. This is because the visual properties of palm images are easier to be separated apart.

We also calculated the average maximum entropy, which is defined as the expected entropy of clusters where objects are

uniformly assigned to each cluster. It is 4.3219 for baseline method and 3.3959 for our method.

Figure 5 and Figure 6 show two clustering results (portion) of query "prairie dog" and "butterfly fish". Each row represents part of the corresponding cluster. It can be seen the images in each cluster possess a harmonious perceptual property.

6. DISCUSSION

1. In the situation that Google returned images are very noisy, to assume images search results are of the same concept and use only the context information to group images would be inferior. In this case, a possible approach is to first separate images into semantically uniform clusters according to their key regions and then apply the method proposed in this paper for each cluster. However, it is very interesting that current approach can already deal with the noisy search result problem to some extent. As shown in Figure 5 the last row and in Figure 6 the last two rows, the noisy images are grouped to the same clusters. A possible reason is that objects often appear with their distinct environments, e.g. tigers always live in forests (dark-yellow and green context regions) and seldom appear with ice-covered lands (snow-white context regions) as penguins do. The context regions often contain some latent semantic clues for their key regions. Thus by considering only the content regions, noisy images can be separated from the "good" images to a certain degree.

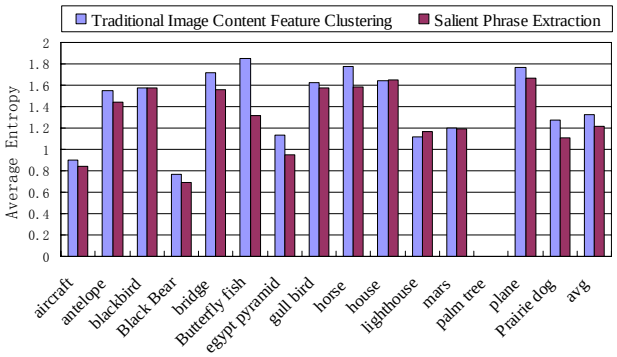

Figure 4. Clustering Result Evaluation on Queries in Table 1

Figure 5. Case Study 1 – "Prairie Dog"

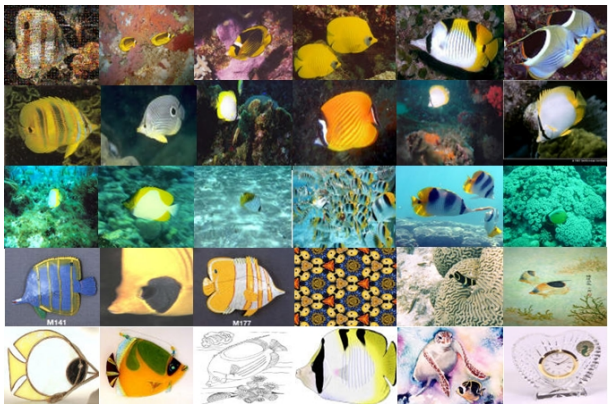

Figure 6. Case Study 2 – "Butterfly Fish"

2. A big superiority of our approach to the traditional image clustering methods lies in that the salient phrases extracted is exactly the name of the cluster. If each codeword is assigned a semantic meaning, then a textual description can be obtained for each cluster to better guide user browsing.

7. CONCLUSION

In this paper, we propose a Web image search result clustering method by extracting salient context patterns and group images according to these patterns. In the current approach, we assume all images returned by the search engine are of the concept hence base our candidate salient phrase extraction approach entirely on the context region. Although our method shows its capability in dealing with the "noisy" images, a more appealing way will be first group images according to their main semantic content and then perform the proposed method on each group respectively. A possible technique may be leveraging the key regions filtered in our current approach. We will research on this in our future work.

8. REFERENCES

[1] Deng, Y., and Manjunath, B. S., Unsupervised Segmentation of Color-Texture Regions in Images and Video. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 23(8), 2001, 800-810

[2] Joachims, T., Making Large-Scale SVM Learning Practical. Advances in Kernel Methods – Support Vector Learning. Scholkopf B. and Burges C. and Smola A. (ed.), MIT-Press, 1999

[3] Sougata, M., Kyoji, H., and Yoshinori, H. Using Clustering and Visualization for Refining the Results of a WWW Image Search Engine. Proc. of workshop on New paradigms in information visualization and manipulation, 1998, 39-35

[4] Thomas, D., Daniel, K., and Hermann, N. Clustering Visually Similar Images to Improve Image Search Engines. Informatiktage 2003 der Gesellschaft für Informatik, 2003

[5] Trystan, U., Rajehndra, N., and Nick, C. Visual Clustering of Image Search Results.citeseer.ist.psu.edu/upstill01visual.html

[6] Vivisimo Clustering Engine, http://vivisimo.com, 2004

[7] Yufei, M. and Hongjiang, Z. Contrast-based Image Attention Analysis by Using Fuzzy Growing. ACM Multimedia, 2003

[8] Hua-Jun, Z., Qi-Cai, H., Zheng, C., Wei-Ying, M. Learning to Cluster Search Results. SIGIR, 2004