Real Time Google and Live Image Search Re-ranking

Jingyu Cui
Tsinghua University, Beijing, China.
cuijingyu@gmail.com

Fang Wen
Microsoft Research Asia, Beijing, China.
fangwen@microsoft.com

Xiaoou Tang
Chinese University of Hong Kong, Hong Kong, China.
xtang@ie.cuhk.edu.hk

ABSTRACT

Nowadays, web-scale image search engines (e.g. *Google Image Search*, *Microsoft Live Image Search*) rely almost purely on surrounding text features. This leads to ambiguous and noisy results. We propose to use adaptive visual similarity to re-rank the text-based search results. A query image is first categorized into one of several predefined intention categories, and a specific similarity measure is used inside each category to combine image features for re-ranking based on the query image. Extensive experiments demonstrate that using this algorithm to filter output of *Google Image Search* and *Microsoft Live Image Search* is a practical and effective way to dramatically improve the user experience. A real-time image search engine is developed for on-line image search with re-ranking: http://mmlab.ie.cuhk.edu.hk/intentsearch

Categories and Subject Descriptors

H5.2 [**Information interfaces and presentation**]: User Interfaces—*Graphical user interfaces (GUI), Prototyping*; H3.3 [**Information storage and retrieval**]: Information Search and Retrieval—*Information filtering*

General Terms: Algorithms

Keywords: Image search, Intention, Visual, Adaptive similarity

1. INTRODUCTION

Today's commercial Internet scale image search engines use only text information. Users type keywords in the hope of finding a certain type of images. The search engine returns thousands of images ranked by the text keywords extracted from the surrounding text. However, many of returned images are noisy, disorganized, or irrelevant. Even the state-of-the-art, such as *Google Image Search* [1] and *Microsoft Live Image Search* [2], use no visual information.

Using visual information to re-rank and improve text based image search results is a natural idea. Most of the existing works assume that there is one dominant cluster of images inside each image set returned by a keyword query, and treat images inside this cluster as "good" ones. Typical works include using each set of images returned by a keyword search to train a latent topic model [5], or emphasize images that occur frequently [9, 6]. Unfortunately, all these approaches require online training, so cannot be used for realtime online image search.

In addition, these approaches cannot handle ambiguity inside a keyword query, since the assumption that images returned by querying one keyword are all from one class does not hold, and the structure of the returned image set is much more complicated. For example, the query for "apple" can return images from 3 *main classes*(images that are semantically similar), such as Fruit apple, apple pie, and Apple digital products. Within each main class, there can be several distinct *sub classes* (images that are visually similar). Also, there are images that can be labeled as *noise* (irrelevant images) or *neglect* (hard to judge relevancy).

In this paper, we propose a framework and build a system to re-rank text based image search results in an interactive manner. After query by keyword, user can click on one image, indicating this is the *query image*. We then re-rank all the returned images according to their similarities with the query. The most challenging problem in this framework is how to define similarity. There are many features in vision and CBIR community, either low level ones such as color, texture, and shape, or higher level ones such as face. Using different features will produce different results, and there is no single feature that can work well for all images. How to integrate various visual features to make a decision about similarity between the query image and other images becomes the essential problem, especially for the diverse and open data set on the Internet. In CBIR community, the relevance feedback approaches [18] focus on how to find a better combination weight of features based on multiple labeled images provided during multiple user feedback sessions. The performance has been limited. In addition, most approaches require online training based on the feedback samples, thus are difficult to be used for realtime online applications.

In this paper, we propose a fast and effective online image search re-ranking algorithm based on one query image only without online training. The proposed *Adaptive Similarity* is motivated by the idea that a user always has a specific intention when submitting a query image. For example, if the user submits a picture with a big face in the middle, most probably he/she wants images with similar faces; if a scenery image is submitted, maybe using scene related features is more appropriate. The query image is firstly categorized into one of several predefined categories. Inside each category, we find a specific weight schema to combine the features adaptive to this kind of images. This correspondence between query image and its proper similarity measurement reflects user intention when using this image to query, so we named these categories *Intentions*. The specific weighting schema inside each intention category is obtained by minimizing the rank loss for all query images on a training set through the proposed method modified from RankBoost [8].

MM'08, October 26–31, 2008, Vancouver, British Columbia, Canada.

By using adaptive similarity measurement according to the query image, we improve overall retrieval performance.

2. SEARCH BY ADAPTIVE SIMILARITY

Given a set of images returned by text based image search engines, we try to leverage the power of visual features adaptively to re-rank the search results. Suppose we have F different features to characterize an image. The normalized similarity between image i and j on feature m is denoted as $s_i^m(j)$, which takes value in the range of $[0, 1]$. A vector $\boldsymbol{\alpha}_i$ is defined for each image i to express its specific "point of view" towards different features. The larger α_{im} is, the more important the mth feature will be for image i. Without losing generality, we further constrain $\boldsymbol{\alpha} \succeq 0$ and $\|\boldsymbol{\alpha}\|_1 = 1$, and get the adaptive similarity measurement at image i as a linear combination of its similarities on different features weighted by $\boldsymbol{\alpha}_i$: $s_i(\cdot) = \sum_{m=1}^{F} \alpha_{im} s_i^m(\cdot)$. Our purpose is to adjust the weight $\boldsymbol{\alpha}$ adaptively for each query image i.

2.1 Intention Categorization

Human can easily categorize images into high level semantic classes, such as scene, people, or object. It is observed in our experiment that images inside each of the classes are similar in terms of which kind of features can discriminate them best from other images. Inspired by this observation, we roughly summarize general images into typical intention categories as: 1. General Object. Images containing close-ups of general objects; 2. Object with Simple Background; 3. Scene. Scenery images; 4. Portrait. Images containing portrait of a single person; 5. People. Images with general people inside, and are not "Portrait". Note that we are proposing a new framework here. More intention categories can certainly be added in future work.

The problem of intention categorization is different from that of object categorization. In our case, images inside each intention category do not necessarily look the same. As a result, using common visual features and general classifiers is not a good idea. On the contrary, since human beings are good at high level categorization, we can make use of this ability by designing several specific "attributes" advised by human prior to be highly related to intention categorization. Using these attributes, we are actually mapping the images into a space in which intention categorization is relatively easy. As a second step, we use C4.5 decision tree [13] to handle the set of inhomogeneous attributes.

The attributes for intention categorization includes: 1. Face existence. Whether the image contains faces. Designed to be related to "Face", "Portrait". 2. Face number. Number of faces occurred in the image. Designed to be related to "Face", "Portrait". 3. Face size. The percentage of the image frame taken up by the face region. Designed to be related to "Portrait". 4. Face position. Coordinate of the face center relative to the center of the image. Designed to be related to "Portrait". 5. Directionality. Kurtosis of Edge Orientation Histogram (**EOH**, Section3). The bigger the Kurtosis is, the stronger the image shows directionality. Designed to be related to "Scene", "General object", and "Object with simple background". 6. Color Spatial Homogeneousness. Variance of values in different blocks of Color Spatialet (**CSpa**, Section3), describing whether color in the image is distributed spatially homogeneously. Designed to be related to "Scene". 7. Edge Energy. Total energy of edge map obtained from Canny Operator on the image. Designed to be related to "General object", "Object with simple background". 8. Edge Spatial Distribution. First divide the image into 3 by 3 regular blocks, then calculate the variance of Edge Energy in the 9 blocks. Describe whether edge energy is mainly distributed at the image center. Designed to be related to "Object with simple background".

With these attributes, we train a C4.5 decision tree on an image set with manually labeled intentions. The training process decides decision boundaries of the intention categories in the feature space defined by those attributes, and intention of a new input image is easily decided by applying the rules of the decision tree to it.

2.2 Intention Specific Feature Fusion

In each intention category, we pre-train an optimal weight $\boldsymbol{\alpha}$ to combine the visual features to achieve best re-rank performance in this specific intention based on the RankBoost Framework [8]. For a query image i, a real valued feedback function $\Phi(j, k)$ is defined to denote preference between image j and k. If image k should be ranked above image j, we let $\Phi(j, k) > 0$, and let $\Phi(j, k) = 0$ otherwise. Based on Φ, a distribution over all pairs of images (j, k) is defined as $D(j, k) = c\Phi(j, k)$, where c is a constant to ensure $\sum_{j,k} D(j, k) = 1$. Thus, the ranking loss using similarity measurement $s_i(\cdot)$ will be: $L_i = \Pr_{(j,k)\sim D_i}[s_i(k) \leq s_i(j)]$.

Optimizing this loss with respect to the variable $\boldsymbol{\alpha}_i$ in $s_i(\cdot)$ will produce optimal weight schema for image i. However, since we are looking for an optimal weight for a collection of images inside an intention category, we further add constraint that $\boldsymbol{\alpha}_i$ inside each category should be the same.

To solve the problem, we use the RankBoost framework in Algorithm 1, but there are 4 steps in the procedure that varies from the original approach to accommodate our scenario: Step 1, 3, 4, and 8. We describe them below.

Algorithm 1 Learning Feature Weight Inside Intention Category

1. **Input:** initial weight D_i for all possible query images i in the current intention category Q, similarity matrix on each feature $s_i^m(\cdot)$ for all i and m;
2. Initialize: Set $D_i^1 = D_i$ for any i. Set step $t = 1$;
while Algorithm not converged **do**
  for each $i \in Q$ **do**
    3. Select best feature and corresponding similarity $s_i^t(\cdot)$ for current re-ranking problem under weight D_i^t;
    4. Calculate real value α_i^t according to Equation 1;
    5. Adjust weight $D_i^{t+1}(j, k) \propto D_i^t(j, k) \exp\left\{\alpha_i^t [s_i(j) - s_i(k)]\right\}$;
    6. Normalize D_i^{t+1} with factor Z_i^t so that D_i^{t+1} will be a distribution;
    7. t++;
  end for
end while
8. **Output:** Final optimal similarity measure for current intention category: $s(\cdot) = \sum_{i,t} \alpha_i^t s_i^t(\cdot)$.

Step 1: Initialization. Our ranking problem is modeled as a bipartite problem. Given a query image i, we define 4 image sets: $S1_i$ includes images within the same sub-class as i; $S2_i$ includes images within the same main class as i, excluding those in $S1_i$; $S3_i$ includes images labeled as "neglect"; $S4_i$ includes images labeled as "noise". For any image j from $S1_i$ and image k from $S2_i \cup S4_i$, we set $\Phi(k, j) = 1$. For all other cases, we set $\Phi(k, j) = 0$. The weight matrix D_i is obtained according to Φ.

Step 3: Select Best Feature. In step 3 of Algorithm 1, we need to select one from a set of F features that can perform best under current weight D_i^t for query image i. This step is rather simple and efficient compared to general settings in RankBoost,

since we have already constrained our weak ranker to be one of the F similarity measurements $s_i^m(\cdot), m = 1, 2, ...F$. What we do is just evaluating the loss L_i^m according to $s_i^m(\cdot)$, and find $m = m^*$ that minimizes the loss. $s_i^{m^*}(\cdot)$ will be the best feature of current step.

Step 4: Select Ensemble Weight. It is proven in [8] that minimizing the normalization factor Z_i^t in each step is approximately equivalent to minimizing the upper bound of the rank loss. The optimization problem was further simplified to be minimizing $\widehat{Z} = \left(\frac{1-r}{2}\right)e^{\alpha} + \left(\frac{1+r}{2}\right)e^{-\alpha}$, which is the upper bound of Z, where $r = \sum_{j,k} D_i(j,k)\left[s_i(k) - s_i(j)\right]$. Instead of optimizing each L_i individually, we seek a $\boldsymbol{\alpha}$ that is good for all query images in a specific intention category. Thus, we penalize cases where $\boldsymbol{\alpha}_i$ is different for different i. So we add another smoothness term to $\widehat{Z} = \left(\frac{1-r}{2}\right)e^{\alpha} + \left(\frac{1+r}{2}\right)e^{-\alpha} + \frac{\lambda}{2}\left(e^{\alpha-\alpha'} + e^{\alpha'-\alpha}\right)$, where α' is the value of α obtained in the previous step, and λ is a hyperparameter to balance the new term and the old terms. In our experiments, we take $\lambda = 1$. Note that the third term takes minimum value λ if and only if $\alpha = \alpha'$. By imposing this new term, we are no longer optimizing individual L_i for each i, we are looking for a common α for all query images in current intention category, while trying to reduce all the losses L_i. Letting $\frac{\partial \widehat{Z}}{\partial \alpha} = 0$, we know that the new $\widehat{Z}$ is minimized when

$$\alpha = \frac{1}{2}\ln\left(\frac{1 + r + e^{\alpha'}}{1 - r + e^{-\alpha'}}\right) \tag{1}$$

Step 8: Obtain Final Weight for Feature Fusion. The final output of the new RankBoost algorithm is a linear combination of all the base rankers generated in each step. However, since there are actually F base rankers, the output is equivalent to a weighted combination of the F similarity measurements. So we finally obtain optimal weight in this intention category for the mth feature as

$$\alpha_m = \sum_{\forall t, i=m} \alpha_i^t \Big/ \sum_{\forall t, i} \alpha_i^t.$$

3. FEATURE DESIGN

In order to characterize images from different perspectives, such as color, shape, and texture, we adopt and design a set of features that are both effective in describing the content of the images, and efficient in their computational and storage complexity. It takes an average of 0.01ms to compute the similarity between two features on an Intel Pentium D 3.0GHz CPU. The total space to store all features for an image is $12KB$.

Attention Guided Color Signature. Color signature was first proposed in [14] to describe the color composition of an image. After k-Means clustering on pixel colors in LAB color space, the cluster centers and their relative proportions are taken as the signature. We propose Attention Guided Color Signature (**ASig**) as a color signature that accounts for varying importances of different parts of an image. We use an attention detector [10] to compute a saliency map for the image, then perform k-Means clustering weighted by this map.

Color Spatialet. We design a novel feature, Color Spatialet, to characterize the spatial distribution of colors in an image. An image is first divided into $n \times n$ patches by a regular grid. Within each patch, we calculate its main color as the largest cluster after k-Means clustering. The image is finally characterized by Color Spatialet (**CSpa**), a vector of n^2 color values. In our experiments, we take $n = 9$. We account for some spatial shifting and resizing of objects in the images when calculating the distance of two CSpas A and B, and $d(A,B) = \sum_{i=1}^{n}\sum_{j=1}^{n} \min\left[d(A_{i,j}, B_{i\pm1,j\pm1})\right]$, where $A_{i,j}$ denotes the main color of the (i,j)*th* block in the image.

Gist. Gist is proposed in [15] to characterize the holistic appearance of an image, and is proven to work well for scenery images.

Daubechies Wavelet. We use the 2nd order moments of wavelet coefficients in various frequency bands (**DWave**) to characterize texture properties in the image[16].

SIFT. We adopt 128-dimension **SIFT** [11] to describe regions around Harris interest points. A codebook of 450 words is obtained by hierarchical k-Means on a set of 1.5 million SIFT descriptors extracted from the training set. Descriptors are then quantized by this codebook.

Multi-Layer Rotation Invariant EOH. Edge Orientation Histogram (EOH) [7], which describes histogram of edge orientations, has long been used in vision applications. We incorporate rotation invariance when comparing two EOHs, resulting in a Multi-Layer Rotation Invariant EOH (**MRI-EOH**). To calculate the distance between two MRI-EOHs, we rotate one of them to best match the other, and take this distance as the distance between the two.

Histogram of Gradient (HoG). HoG [4] is the histogram of gradients within image blocks divided by a regular grid. HoG reflects the distribution of edges over different parts of an image, and is especially effective for images with strong long edges.

Facial Feature. Face existence and their appearances give clear semantic interpretations of the image. We apply face detection algorithm [17] to each image, and obtain the number of faces, face size and position as the facial feature (**Face**). The distance between two images is calculated as the summation of differences of face number, average face size, and average face position.

4. EXPERIMENTS

One test data is a collection of $451,352$ images associated with 483 keywords crawled from *Google Image Search* and *Microsoft Live Image Search*. Our database contains many concepts including object, scenery, people name, place name, etc., and covers a large range of keywords (listed at the demo site).

To facilitate evaluation of search performance, we invited several people who have no knowledge of the research project to categorize the images into *noise* (Irrelevant images), *neglect*(Hard to judge relevance), and *good* (Images not in *noise* and *neglect* category). *Good* images are further classified into *Sub Classes* (images that are visually similar) and *Main Classes* (images that are semantically similar). It is natural to treat images that are from the same *Sub Class* as the query image to be relevant, while other as "not so relevant".

We have 26,908 images from 30 keywords labeled, among which 46% are labeled as *good*, consisting of 128 sub classes and 70 main classes. Note that we do the visual similarity based ranking within each keyword related images, just as a real web image-search reranking system we described in Section 1.

Evaluation Criteria. We use the P_{20} and P_{40} (Proportion of images within the same sub class in top 20 and 40 returned ones) to evaluate performance, which is a natural measurement to focus on the first 1-2 pages of the search results on the web page.

Methods to Compare With. We compare the proposed algorithm with two baseline methods: 1. Select the feature which have the largest variance of similarity scores, based on the assumption that an effective feature should give good image much larger score than a mediocre one. 2. Use global weights to combine features.

Training and Testing Results. We divide the labeled benchmark database into two sets. One includes 9017 images from 10 keywords, and is used for SIFT codebook training (Sec. 3), inten-

Keywords	Precisions of each feature								Selected Feature	Global similarity	Adaptive similarity
	ASig	CSpa	Gist	DWave	SIFT	MRI-EOH	HoG	Face			
airplanes	0.333	0.320	0.327	**0.353**	0.165	0.305	0.348	0.152	**0.350**	**0.363**	**0.424**
beach	**0.633**	0.607	0.608	0.565	0.475	0.578	0.625	0.482	**0.620**	**0.687**	**0.727**
car	**0.430**	0.406	0.372	0.362	0.132	0.349	0.405	0.214	**0.378**	**0.406**	**0.492**
dolphin	**0.620**	0.583	0.491	0.573	0.319	0.446	0.467	0.272	**0.520**	**0.591**	**0.646**
guitar	0.207	0.213	0.446	0.400	0.265	0.339	**0.485**	0.103	**0.446**	**0.449**	**0.513**
paris	0.312	0.340	0.357	0.319	0.286	0.328	**0.373**	0.333	**0.3252**	**0.461**	**0.502**
rice	0.511	0.489	0.478	0.438	0.384	0.464	**0.579**	0.284	**0.499**	**0.591**	**0.656**

Figure 1: Performance comparison between using single features, selected feature, global similarity, and our proposed adaptive similarity. Different keywords requires different features to achieve overall best performance. Selecting "best" feature for each of the query image can approach the performance of using the real "best" feature. Using global similarity integrating all the possible features improves the results above, but our method to adaptively integrate features according to each image outperforms them all.

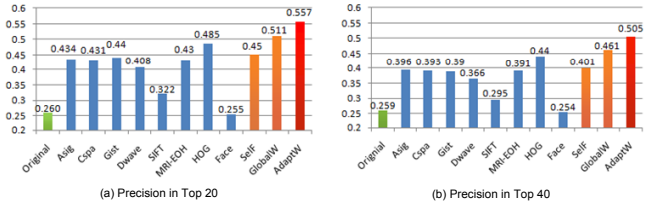

Figure 2: Average P_{20} (a) and P_{40} (b) comparison between original text based search (**Original**), using single features, using selected feature (**SelF**), using global weighted feature integration (**GlobalW**), and using the proposed adaptive integration (**AdaptW**).

tion classifier training, and feature combination weights training. Another includes 17,891 images from other 20 keywords, and is used for testing. For each intention category we manually labeled 200 images from the training set and trained a C4.5 decision tree. We apply it to classify all "good" images (3021 images) of the training set into 5 intention categories. Then optimal weight for each intention category is learnt using the algorithm described in Section 2.2.

Several quantitative comparison results are shown in Figure 1, and the comparison averaged over all test images is shown in Figure 2. We can see that by using a single query image we can significantly improve the text based image search result. Combining all features together using a global weight gives better results than selecting a best feature. However, for some keywords, both methods give worse result than the true best feature. The intention based method gives better results than all the features and the baseline methods. Comparing to the surrounding text based approach, our approach doubled the search accuracy. More importantly, we implemented the first real-time online image search engine based on text and query image. Please try the demo at `http://mmlab.ie.cuhk.edu.hk/intentsearch`, where we also share the data used in this paper. Figure 3 shows a few example search results. A novel collage based browsing tool is also presented in our demo paper [3].

5. CONCLUSION

In this paper, we propose a realtime re-ranking algorithm to enhance the performance of *Google Image Search* and *Microsoft Live Image Search*, by letting user select a query image from text search results. We use an intention categorization model to integrate a set of complementary features adaptive to the query image. We also build a large labeled database from Internet to share to the community. Using the developed technology, we implemented a real-time online image search engine, combining text and IntentSearch. In future work, we will look into combining our work with photo quality re-ranking method [12] for further improvement.

Figure 3: Comparison between original image search result and IntentSearch results. From top to bottom, the keywords for search are: bicycle, Tibet. The automatically inferred intentions are: Object with Simple Background, Scene. (Best viewed in color)

6. REFERENCES

[1] Google Image Search. `http://images.google.com`.

[2] `http://www.live.com/?\&scope=images`.

[3] J. Cui, F. Wen, and X. Tang. Intentsearch: Interactive on-line image search re-ranking. In *MULTIMEDIA '08: Proceedings of the 16th annual ACM international conference on Multimedia*, 2008.

[4] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR*, 2005.

[5] R. Fergus, L. Fei-Fei, P. Perona, and A. Zisserman. Learning object categories from google's image search. In *ICCV*, 2005.

[6] R. Fergus, P. Perona, and A. Zisserman. A visual category filter for google images. In *ECCV*, 2004.

[7] W. Freeman and M. Roth. Orientation histogram for hand gesture recognition. In *Int'l Workshop on Automatic Face- and Gesture-Recognition*, 1995.

[8] Y. Freund, R. Iyer, R. E. Schapire, and Y. Singer. An efficient boosting algorithm for combining preferences. *J. Mach. Learn. Res.*, 4:933–969, 2003.

[9] W. H. Hsu, L. S. Kennedy, and S.-F. Chang. Novel reranking methods for visual search. *IEEE Multimedia*, 2007.

[10] T. Liu, J. Sun, N.-N. Zheng, X. Tang, and H.-Y. Shum. Learning to detect a salient object. In *CVPR*, 2007.

[11] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vision*, 60(2):91–110, 2004.

[12] Y. Luo and X. Tang. Photo and video quality evaluation: Focusing on the subject. In ECCV, 2008.

[13] J. R. Quinlan. *C4.5: programs for machine learning*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1993.

[14] Y. Rubner, L. J. Guibas, and C. Tomasi. The earth mover's distance, multi-dimensional scaling, and color-based image retrieval. In *Proceedings of the ARPA Image Understanding Workshop*, 1997.

[15] A. Torralba, K. Murphy, W. Freeman, and M. Rubin. Context-based vision system for place and object recognition, 2003.

[16] M. Unser. Texture classification and segmentation using wavelet frames. *IEEE TIP*, 4:1549–1560, 1995.

[17] R. Xiao, H. Zhu, H. Sun, and X. Tang. Dynamic cascades for face detection. In *ICCV*, 2007.

[18] X. S. Zhou and T. S. Huang. Relevance feedback in image retrieval: A comprehensive review. *Multimedia Systems*, 8(6):536–544, 2003.