Visual Diversification of Image Search Results

Reinier H. van Leuken
Universiteit Utrecht
Utrecht, the Netherlands
reinier@cs.uu.nl

Lluis Garcia
Yahoo! Research
Barcelona, Spain
lluis@yahoo-inc.com

Ximena Olivares
Universitat Pompeu Fabra
Barcelona, Spain
ximena.olivares@upf.edu

Roelof van Zwol
Yahoo! Research
Barcelona, Spain
roelof@yahoo-inc.com

ABSTRACT

Due to the reliance on the textual information associated with an image, image search engines on the Web lack the discriminative power to deliver visually diverse search results. The textual descriptions are key to retrieve relevant results for a given user query, but at the same time provide little information about the rich image content.

In this paper we investigate three methods for visual diversification of image search results. The methods deploy lightweight clustering techniques in combination with a dynamic weighting function of the visual features, to best capture the discriminative aspects of the resulting set of images that is retrieved. A representative image is selected from each cluster, which together form a diverse result set.

Based on a performance evaluation we find that the outcome of the methods closely resembles human perception of diversity, which was established in an extensive clustering experiment carried out by human assessors.

Categories and Subject Descriptors

H.3.1 [**Information Storage and Retrieval**]: Content Analysis and Indexing; H.3.5 [**Information Storage and Retrieval**]: Online Information Services

General Terms

Algorithms, Experimentation, Performance

Keywords

Visual diversity, Flickr, image clustering

1. INTRODUCTION

The common modality used for image search on the web is text, used both in indices of large search engines or in a more restricted environment such as social media sites like Flickr. Although not without its flaws, the assumption that a relevant image resides on a web page surrounded by text that matches the query is reasonable. Along the same lines, tags and textual descriptions of photos prove to be powerful ways to describe and retrieve images that are uploaded daily in massive quantities to dedicated sharing sites. The retrieval models deployed on the Web and by these photo sharing sites rely heavily on search paradigms developed within the field Information Retrieval. This way, image retrieval can benefit from years of research experience, and the better this textual metadata captures the content of the image, the better the retrieval performance will be.

It is also commonly acknowledged that a picture has to be seen to fully understand its meaning, significance, beauty, or context, simply because it conveys information that words can not capture, or at least not in any practical setting. This explains the large number of papers on content-based image retrieval (CBIR) that has been published since 1990, the breathtaking publication rates since 1997 [12], and the continuing interest in the field [4]. Moving on from simple low-level features to more discriminative descriptions, the field has come a long way in narrowing down the semantic gap by using high-level semantics [8]. Unfortunately, CBIR-methods using higher level semantics usually require extensive training, intricate object ontologies or expensive construction of a visual dictionary, and their performance remains unfit for use in large scale online applications such as the aforementioned search engines or websites. Consequently, retrieval models operating in the textual metadata domain are therefore deployed here.

In these applications, image search results are usually displayed in a ranked list. This ranking reflects the similarity of the image's metadata to the textual query, according to the textual retrieval model of choice. There may exist two problems with this ranking.

First, it may be lacking visual diversity. For instance, when a specific type or brand of car is issued as query, it may very well be that the top of this ranking displays many times the same picture that was released by the marketing division of the company. Similarly, pictures of a popular holiday destination tend to show the same touristic hot spot, often taken from the same angle and distance. This absence of visual diversity is due to the nature of the image annotation, which does not allow or motivate people to adequately describe the visual content of an image.

Second, the query may have several aspects to it that are not sufficiently covered by the ranking. Perhaps the user is interested in a particular aspect of the query, but doesn't know how to express this explicitly and issues a broader, more general query. It could also be that a query yields so many different results, that it's hard to get an overview of the collection of relevant images in the database.

WWW 2009, April 20–24, 2009, Madrid, Spain.
ACM 978-1-60558-487-4/09/04.

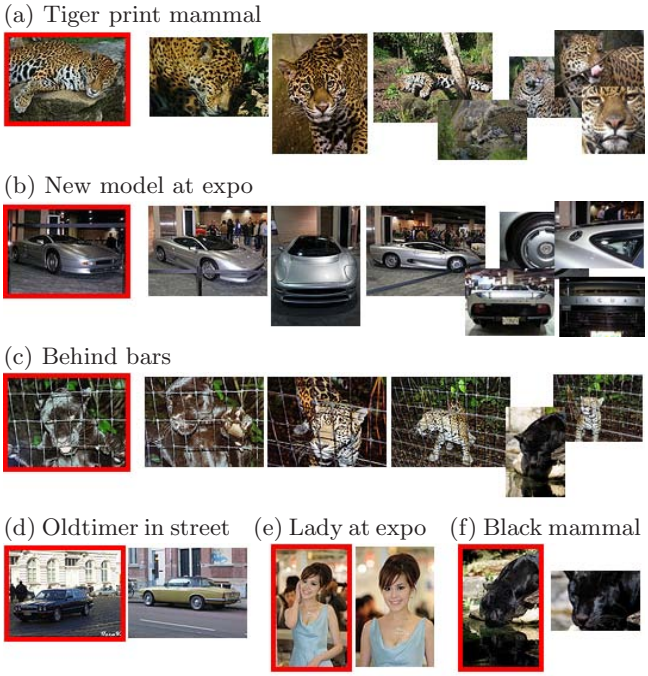

Figure 1: Example clustering: output of the reciprocal election algorithm for query *jaguar*. Cluster representatives are indicated by a red border.

We propose to create a visually diverse ranking of the image search results, through clustering of the images based on their visual characteristics. To organize the display of the image search results, a cluster representative is shown to the user. Depending on the interest of the user in one of the representatives, he can then explore the other images in that cluster. This approach guarantees that the user will be presented a visually diverse set of images.

An example clustering of one of our algorithms is given in Figure 1. The example uses the ambiguous query "*jaguar*". The image search result is not only ambiguous from a topical point of view (car, mammal), but also from a visual point of view. The algorithm separates mammals with a tiger print from black mammals and mammals behind bars. It also groups pictures from a new car model at an expo from cars in the street, and groups the accidentally found pictures of a lady at a car expo. The cluster representatives together form a diverse set of image search results.

1.1 Outline and contributions

In this paper we introduce new methods to diversify image search results. Given a user query, we first determine dynamically appropriate weights of visual features, to best capture the discriminative aspects of the resulting set of images that is retrieved. These weights are used in a dynamic ranking function that is deployed in a lightweight clustering technique to obtain a diverse ranking based on cluster representatives. We propose three clustering algorithms that are both effective and efficient, called folding, maxmin and reciprocal election. In the case of folding, the original ranking is respected by preferring higher ranked items as representatives over lower ranked items. Maxmin on the other hand discards this original ranking and aims for maximal visual diversity of the representatives. The key idea behind reciprocal election is to let each image cast votes for other images that it is best represented by: a strategy close to the intuition behind a clustering.

We have implemented the methods and performed a performance evaluation in a large scale user-study using 75 topics of both an ambiguous and non-ambiguous nature.

2. RELATED WORK

In context of the general task of this paper, we discuss the related work by first discussing the state of the art in image clustering, and then by focusing on related work in diversifying search results.

2.1 Image clustering

Most image clustering techniques are not dynamic, and therefore not suitable for clustering image search results. First off, we are only interested in unsupervised clustering techniques, which makes techniques such as presented in [7] unsuitable for our task. Furthermore, clustering techniques often partition the entire database to facilitate faster browsing and retrieval [6].

In [10] a method for extracting meaningful and representative clusters is presented that is based on a shared nearest neighbors (SNN) approach that treats both content-based features and textual descriptions (tags). They describe, discuss and evaluate the SNN method for image clustering and present some experimental results using the Flickr collections showing that our approach extracts representative information of an image set. Such techniques are often effective, but require a lot of processing power to produce a final clustering. When clustering image search results, the input varies depending on the user's query and it is essential that the clustering technique is not only effective, but the results can be efficiently computed.

In our case, we want the approach to be dynamic such that it best captures the particular context of the user's query. We'll therefore rely on a dynamic ranking strategy, which allows us to dynamically weight the importance of the visual dimensions such as color, shape an texture, in combination with lightweight clustering strategies. Although not incorporated in the current implementation, we can easily extend the dynamic ranking strategy to include a textual modality as is used in the aforementioned related work.

In Cai et al. [2] the problem of clustering Web image search results is studied, by organizing the results into different semantic clusters that facilitates users' browsing. They propose a hierarchical clustering method using visual, textual and link analysis that is mainly targeted at clustering the search results of ambiguous targets. In a related paper by Wang et al. [18], also from Microsoft, they evaluate a different approach, named IGroup, for semantic clustering of image search results, based on a textual analysis of the search results. Through a user study they report a significant improvement in terms efficiency, coverage, and satisfaction.

2.2 Diversity in search results

In our prior work, we have studied the diversification of image search results in two different contexts. In [19], we have presented a method for detecting and resolving the ambiguity of a query based on the textual features of the image collection. If a query has an ambiguous nature, this ambiguity should be reflected in the diversity of the result

set. Furthermore, in [16] we have presented how the topical (textual) diversity of image search results can be achieved through the choice of the right retrieval model. The focus in the current paper is on visual diversity of the search results. Our solution for the visual diversity builds upon the results of these two papers, as it takes as input the ranked list of images produced by the retrieval models for topical diversity.

In Zhang et al. [22] diversity of search results is examined in the context of Web search. They propose a novel ranking scheme named Affinity Ranking to re-rank search results by optimizing two metrics: diversity and information richness. More recently, Song et al. [13] also acknowledge the need for diversity in search results for image retrieval. They propose a re-ranking method based on topic richness analysis to enrich topic coverage in retrieval results, while maintaining acceptable retrieval performance.

Zeigler studied topic diversification to balance and diversify personalized recommendation lists in order to reflect the user's complete spectrum of interests [23]. Although their system is detrimental to average accuracy, they show that the method improves user satisfaction with recommendation lists, in particular for lists generated using the common item-based collaborative filtering algorithm. They introduced an intra-list similarity metric to assess the topical diversity of recommendation lists and the topic diversification approach for decreasing the intra-list similarity.

In a different setting, Yahia et al. [21] propose a method to return a set of answers that represent diverse results proportional to their frequency in the collection. Their algorithm operates on structured data, with explicitly defined relations, which differs from our setting, as we aim to diversify through visual content based on a dynamic ranking strategy, rather than using predetermined fractions.

3. IMAGE SIMILARITY

One of the key elements to any clustering algorithm or retrieval system, is a similarity measure between the objects. In content-based image retrieval or clustering, it is common to use several features simultaneously while calculating the similarity between images. These features represent different aspects of the image, such as color features, edge features, texture features, or alternatively concept detectors [11]. Each feature has its own representation (e.g. a scalar, a vector, a histogram) and a corresponding matching method (e.g. Euclidean distance, hamming metric).

The fusion of different modalities into a single ranking is not trivial. Various techniques have been proposed to effectively fuse multiple-modalities into a single ranking, using a simple linear weighting, principle component analysis [15], or by using a weighted schema for aggregating features based on document scores [20].

In this paper we introduce a dynamic ranking strategy that weights the importance of the different features based on the (normalized) variance of the similarities of all images in the results set. In our case, the similarity measure defined on the images has to reflect visual similarity, but the clustering algorithms presented in this paper work with any distance measure between two images.

In this section we first describe the dynamic feature weighting funtion that implements the ranking strategy, followed by a short description of the 6 well-known visual image features that we have adopted for our experiments.

3.1 Dynamic feature weighting

Based on the features described below, the similarity between two images can be expressed in 6 similarity values. These values, that may be of entire different range and distribution, need to be aggregated into one value for use in the clustering algorithm. Moreover, it is a priori unclear what the relative importance is of these features within the context of a specific set of image search results.

One assumption we make, is that the images retrieved by the textual retrieval model are topically relevant to the query. For each feature, we then calculate the variance over all image similarities within the set of image results. This variance is used as a weighting and normalizing factor at the same time. The image similarity according to a certain feature is divided by the variance of that feature in the result set. This brings image similarities according to different features in a similar range, and assigns a larger weight to features that are a good discriminator for the results that are presented to the user. The rationale is that when the variance of a certain feature is small, the images in the result set resemble each other in terms of that feature closely and thus it is a striking feature for this specific set.

More formally, the similarity between two images a and b is calculated as follows:

$$d(a,b) = \frac{1}{f}\sum_{i=0}^{f}\frac{1}{\sigma_i^2}d_i(a,b)$$

, where f is the total number of features, $d_i(a,b)$ is the similarity between a and b in terms of the i-th feature and σ_i^2 is the variance of all image similarities according to the i-th feature within this set of image search results.

3.2 Features

For our experiment we have extracted 6 visual features from each image to capture the different characteristics such as the color, shape and texture of an image. Below follows a short description.

Color histogram. A color histogram describes the global color distribution in an image. To compute the color histogram, we define a discretization of the RGB color space into 64 color bins. Each bin contains the number of pixels in the image that belong to that color range. Two color histograms are matched using the Bhatta Charrya Distance [1].

Color layout. Color layout is a resolution invariant compact descriptor of colors used for high-speed image retrieval [11]. Color layout captures the spatial distribution of the representative colors in an image. The image is divided into 64 blocks. For each block a representative color is obtained using the average of the pixel colors. Every color component ($YCbCr$) is transformed by a 8x8 DCT (discrete cosine transformation) obtaining a set of 64 coefficients, which are zigzag-scanned and the first coefficients are nonlinearly quantized.

Scalable color. Scalable color can be interpreted as a Haar-transform applied to a color histogram in the HSV color space [11]. First, the histogram (256 bins) values are extracted, normalized and nonlinearly mapped to a 4-bit integer representation. Afterwards, the

Haar transform is applied across the histograms bins to obtain a smaller descriptor allowing a more scalable representation. Two feature vectors are matched using a standard L_1-norm.

CEDD. The color and edge directivity descriptor (CEDD) incorporates both color and texture features in a histogram [3]. It is limited to 54 bytes per image making this descriptor suitable for large image databases. First, the image is split in a preset number of blocks; a color histogram is computed over the HSV color space. Several rules are applied to obtain for every block a 24-bins histogram (representing different colors). Then 5 filters are used to extract the texture information related to the edges presented in the image and classified in vertical, horizontal, 45-degree diagonal, 135-degree diagonal and non-directional edges. Two descriptors are matched using the Tanimoto coefficient.

Edge histogram. The edge histogram represents a local edge distribution of the image [11]. First, the image is divided in a 4x4 grid. Edge detection is performed to each block and the edges are grouped into 5 types: vertical, horizontal, 45 degrees diagonal, 135 degrees diagonal and non directional edges. The feature therefore consists of $16x5 = 80$ coefficients. For matching two feature vectors the standard L_1- norm is used.

Tamura Tamura et al. [14] identified properties of the images that play an important role to describe textures based on human visual perception. They defined six textural features (coarseness, contrast, directionality, line-likeness, regularity and roughness). We used 3 Tamura features to build a texture histogram: coarseness, contrast and directionality. The Tamura features are matched using the standard L_2-norm.

4. CLUSTERING ALGORITHMS

In this section we present the clustering algorithms, called *folding*, *maxmin* and *reciprocal election*. First, we introduce some notation. A set of image search results I contains n images. I can be stored either in a ranked list $L = L_1, L_2, \ldots, L_n$, sorted in decreasing degree of relevance to the query, or in a set $S = S_1, S_2, \ldots, S_n$, where there is no particular ordering. The input to a clustering algorithm can be either L or S, and its output is a clustering C: a partitioning of I. In C, all the images are divided over K clusters $C_1, C_2, \ldots, C_k$ such that $C_k \bigcap C_l = \emptyset$ for all $l, k \in K$ and $\bigcup_{k=1}^{K} C_k = I$. The number of images in cluster C_k is n_k, so $\sum_{k=1}^{K} n_k = n$, and in each cluster C_k one image is declared representative, called R_k. All the representatives together form the set R. Let C' be another clustering of I, with K' clusters $C'_1, C'_2, \ldots, C'_{k'}$. Note that K and K' may be very different.

The three clustering algorithms differ from each other in viewpoint. The folding algorithm appreciates the original ranking of the search results as returned by the textual retrieval model. Images higher in the ranking have a larger probability as being selected as a cluster representative. In one linear pass the representatives are selected, the clusters are then formed around them. The maxmin approach also performs representative selection prior to cluster formation, but discards the original ranking and finds representatives that are visually different from each other. Reciprocal election lets all the images cast votes for other images that they are best represented by. Strong voters are then assigned to their corresponding representatives, and taken off the list of candidates. This process is repeated as long as there exist unclustered images.

The number of clusters is never fixed, because it is impossible to predict a priori what a good value is. This should always be dynamically set for a specific clustering. We now present the algorithms in more detail.

4.1 Folding

In some cases, it is important to take the original ranking of the image search results into account while performing the clustering. For example, it might be that the query is very specific and only the top of the ranking is sufficiently topically relevant. It might also be that retrieval speed is valued over accuracy, and the uncertainty about topical relevance of the retrieved items decreases quickly while going through the ranking. Folding (see algorithm 1) is an approach that appreciates the original ranking, by assigning a larger probability of being a representative to higher ranked images.

The first step of the approach is to select the representative images, while traversing through the ranking L from top to bottom. The first image in the ranking, L_1 is always selected as representative. While going down the ranked list, each image is compared to the set of already selected representatives. When an image is sufficiently dissimilar to all the selected representatives in R, it is added to R. After this pass through the ranking, clusters are formed around each representative using a nearest neighbor rule: each image in L is assigned to the closest representative.

Key to this approach is the definition of *sufficiently dissimilar* while selecting the candidates. This parameter is set automatic and dynamic as well. It is defined as the mean distance all images in I have to the *average image*. The average image is a synthetic image that only exists in feature space and is constructed by aggregating per feature all the images into one canonical image. Since all features are histogram-like features, this aggregation step follows from their definition. It would be also possible to select a canonical image from I using a heuristic, e.g. the image with the smallest mean distance to all the other images.

Algorithm 1 Folding
Input: Ranked list L of I
Output: Clustering C

```
Let the image L_1 be the first representative R_1
for Each image L_i do
    if d(L_i, R_j) > ε(*) for all representatives R_j then
        add L_i to the set of representatives R
for Each image L_i ∉ R do
    Find representative R_j that is closest to L_i
    Assign L_i to the cluster of R_j
```

(*)ϵ is defined as the mean distance all images have to the *average image* in I

4.2 Maxmin

The maxmin approach (see algorithm 2) doesn't take the original ranking into account like folding does. It rather

tries to get as visually diverse representatives as possible. To achieve this, it uses a maxmin heuristic on the distances between cluster representatives. The algorithm takes as input I stored as unordered set S, so the first representative R_1 is selected at random. The second representative R_2 is the image of S with the largest distance to R_1. For each following representative, the image is selected that has the largest minimum distance to all the other selected representatives. This process is continued until this maximum minimal distance is smaller than ϵ, which is again defined as the mean distance all images have to the average image.

When the representatives are selected using this heuristic, cluster formation is again carried out using a nearest neighbor rule. Each image is assigned to the closest representative.

Algorithm 2 Maxmin
Input: Set S containing I
Output: Clustering C

Select the first representative R_1 randomly
while All pairwise distances in $R > \epsilon$ **do**
  for Each image $L_i \notin R$ **do**
    Let d_i be $\arg\min_{d(L_i,R_j),R_j \in R}$
  Add to R the image with $\arg\max_{d_i}$
for Each image $S_i \notin R$ **do**
  Find representative R_j that is closest to S_i
  Assign S_i to the cluster of R_j

4.3 Reciprocal election

In contrast to folding and maxmin, the reciprocal election approach interleaves the processes of representative selection and cluster formation. The key idea behind this approach (see algorithm 3) is that every image in I decides by which image (besides itself) it is best represented. They all cast votes for the other images, and all the votes an image receives determine its chances of being elected as representative. The process of voting is based on calculating reciprocal ranks in rankings of I. For each image S_i, the whole set of image search results I is ranked into L_i based on visual similarity to S_i. The image S_i then casts its highest vote for the image that appears on top of that ranking (excluding itself), its second highest vote to the number two of that list etcetera. Therefore, each image in L_i receives as a vote from S_i its reciprocal rank, i.e. $1/r$ where r is its rank in L_i.

When all the images have cast their votes, the image with the highest number of votes is selected as first representative R_1. Immediately, the cluster around R_1 is formed, by inserting those images that have R_1 in the top-m of their ranking. The rationale is that because R_1 appears so high in their ranking, they are sufficiently well represented by R_1. After cluster C_1 has been formed, its members and its representative are excluded from the list of candidate representatives, and the process is repeated until every image has been either selected as representative or assigned to a cluster.

5. EXPERIMENTAL SETUP AND RESULTS

Our test corpus consists of a pool of 75 topics that were randomly selected from the Flickr search logs. Based on the

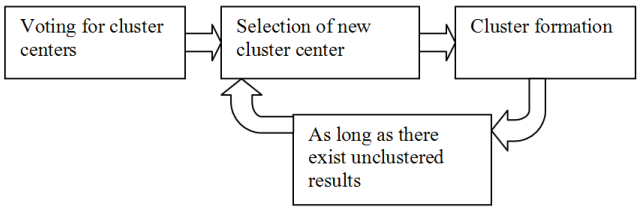

Figure 2: Overview of the reciprocal election algorithm

Algorithm 3 Reciprocal election
Input: Set S containing I, parameter m
Output: Clustering C

Initialize Votes map $V[0,\ldots,k] = 0,\ldots,0$
for Each image i in S **do**
  Rank S into L_i based on *visual* similarity to i
  for Each image j in L_i **do**
    $V[j] += 1/r$, where r is the rank of j in L_i
while V is not empty **do**
  Let R_i be the item with the highest score in V
  Remove R_i from V
  Initialize new cluster C with representative R_i
  for All items s in V **do**
    if R_i is in top-m of L_s **then**
      add s to cluster C
      remove s from V

method for resolving query ambiguity as presented in [19], we have divided our pool in 25 textually ambiguous queries, and 50 textually non-ambiguous queries. This allows us to measure the difference in performance of the visual clustering methods on both types of queries. For each query we have retrieved the top 50 results from a slice of 8.5 Million photos on Flickr.

To retrieve a list of 50 results for the non-ambiguous queries we have used a dual index relevance model that produces a focused result set. To obtain the top 50 results of the ambiguous queries, we have used a tags-only index relevance model that produces a balanced list of diverse results. The details of both retrieval models are described in [16]. The intuition behind these choices is simple. If the terms in a query are textually diverse, then we want to produce a diverse set of images that embodies many possible interpretations of the user's query. Consider for example the query "jaguar", which carries at least three different word-senses that are present in the Flickr collection: the mammal, the car, and the operating system. On the other hand, if a query is textually non-ambiguous, e.g it has a clear dominant sense, the precision can be improved by returning more focused results. The query "jaguar x-type" serves as an example for a non-ambiguous query. In both cases, the result sets produced contain visually diverse images on which we'll test our methods.

To evaluate the performance of the proposed algorithms, we compare their output to clusterings that were created by human assessors. The following sections present details on the establishment of the ground truth, the evaluation criteria and the experimental results.

5.1 Human assessments

To establish a ground truth, we divided the 75 topics over 8 independent, unbiased assessors and we asked them to cluster the images based on their visual characteristics. We implemented the following procedure.

1. Select a topic, and inspect the top 50 results during at least one minute. This allows the assessors to get an overall impression of the images in the result set, to get a rough idea of how many clusters will be needed, and of their level of inter cluster dissimilarity. At any point in the assessment, the assessor could switch to this overview.

2. Form image clusters by assigning each image to a cluster pool by entering the cluster id. In total the assessor could create 20 clusters, and he/she could undo the last assignment if needed to correct for errors. See Figure 3 for an example of this interface.

3. Once all images in the results were assigned to a cluster, the assessor was asked to label each cluster and to identify one image in each cluster that could serve as a cluster representative.

In total we obtained 200 topic clusterings created by the assessors, because each topic was assigned to multiple assessors. We are therefore able to calculate inter assessor variability, that provides us with an baseline during the performance evaluation of the algorithms.

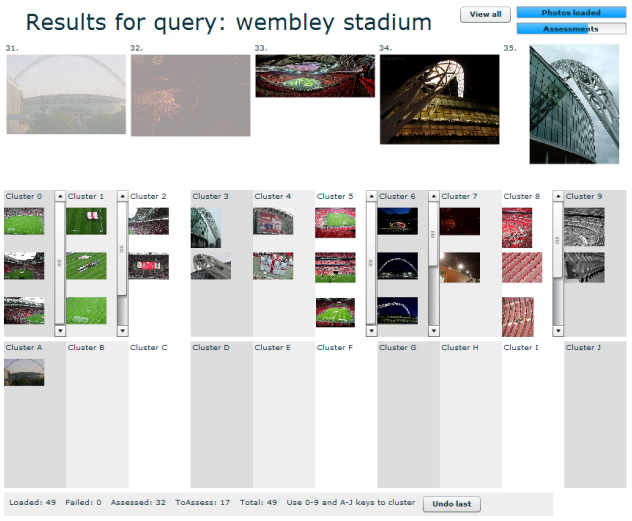

Figure 3: Example of the clustering interface used by the assessors.

5.2 Evaluation criteria

Comparing two clusterings of the same data set is an interesting problem itself, for which many different measures have been proposed. We adopt two clustering comparison measures that appreciate different properties. In this subsection we describe them briefly.

One popular category of comparison measures is based on counting pairs. Given a result set I and two clusterings C and C', all possible image pairs based on I are divided over the following four classes:

N_{11} :
image pairs in the same cluster both under C and C'

N_{00} :
image pairs in a different cluster both under C and C'

N_{10} :
image pairs in the same cluster under C but not under C'

N_{01} :
image pairs in the same cluster under C' but not under C

From now on, we will refer to the *cardinality* of these classes simply by its class name. These cardinalities are input to the comparison measures. The first clustering comparison measure we use is the Folwkes-Mallows index [5] that can be seen as the clustering equivalent of precision and recall. A high score of the Folwkes-Mallows index indicates that the two clusterings are similar. It is based on two asymmetric criteria proposed by Wallace [17]:

$$W_I(C,C') = \frac{N_{11}}{N_{11}+N_{01}}$$

$$W_{II}(C,C') = \frac{N_{11}}{N_{11}+N_{10}}$$

The Folwkes-Mallows index is the geometric mean of these two, making it a symmetric criterion:

$$FM(C,C') = \sqrt{W_I(C,C')W_{II}(C,C')}$$

Another class of comparison measures is based on a structure called the *contingency table* or *confusion matrix*. The contingency table of two clusterings is a $K \times K'$ matrix, where the kk'-th element is the number of points in the intersection of clusters C_k of C and $C'_{k'}$ of C'. Our second clustering comparison measure is the *variation of information* criterion, $VI(C,C')$, as introduced by Meilă [9]. Variation of information uses the contingency table, and is based on the concept of conditional entropy.

Given a clustering C, the probability that a randomly picked image belongs to cluster k with cardinality n_k, is

$$P(k) = \frac{n_k}{n}$$

This defines a random variable taking K values. The uncertainty about which cluster an image is belonging to is therefore equal to the entropy of this random variable

$$H(C) = -\sum_{k=1}^{K} P(k)\log P(k)$$

The mutual information $I(C,C')$, the information one clustering has about the other, can be defined similarly. First, the probability that a randomly picked image belongs to cluster k in C and to cluster k' in C', is

$$P(k,k') = \frac{|C_k \bigcap C'_{k'}|}{n}$$

Then, the mutual information $I(C,C')$ is defined as the sum of the corresponding entropies taken over all possible pairs of clusters:

$$I(C,C') = \sum_{k=1}^{K}\sum_{k'=1}^{K'} P(k,k')log\frac{P(k,k')}{P(k)P'(k')}$$

The mutual information coefficient can be seen intuitively as a reduction of uncertainty from one clustering to the other. For a random image, the uncertainty about its cluster in C' is measured by $H(C')$. Suppose now that it is given which cluster in C the image belongs to; how much does this reduce the uncertainty about C'? This reduction, averaged over all images, is equal to $I(C, C')$.

Finally, the variation of information is defined as the sum of the two conditional entropies $H(C, C')$ and $H(C', C)$. The first measures the amount of information about C that we loose, while the second measures the amount of information about C' that we gain, when going from clustering C to C'. It can be written as

$$VI(C, C') = [H(C) - I(C, C')] + [H(C') - I(C, C')]$$

The variation of information coefficient focuses on the relationship between a point and its cluster. It measures the difference in this relationship, averaged over all points, between the two clusterings, hence a low variation of information score indicates that two clusterings are similar.

5.3 Results

All 200 clusterings of the 75 topics that were obtained as a result of the human assessments are compared to the clusterings generated by the different techniques. Using the described comparison measures, variation of information and the Folwkes-Mallows index, performance is evaluated. In this section, results are presented for ambiguous topics separately, non ambiguous topics separately and all topics together.

Interassessor variability and random clustering

As a base line for the performance we use the inter assessor variability. A technique can not be expected to produce clusterings that resemble on average the human created clusterings better than the assessors agree among themselves. To put a bound on expected performance on the other end as well, we compare the human created clusterings with randomly generated clusterings. For this purpose, for each topic a random number (between 2 and 20) of clusters was generated. Every image was clustered randomly into one of the clusters, all random distributions were uniform. We expect that the performance of each of the three methods to lay within these two performance bounds.

Results on Fowlkes-Mallows Index

The best performing method according to the Fowlkes-Mallows index is folding, followed by reciprocal election and maxmin. Mean values and first and third quartiles are given in Figure 4 for both ambiguous and non ambiguous topics. The boxes show the average and the first and third quartiles for all comparisons, i.e. 50% of the 200 clustering comparisons fall within the box. The figure is showing the performance of reciprocal election, folding and maxmin; it is also showing the comparison results of a randomly generated clustering and the inter-assessor agreements according to the same comparison measure. Please note that a higher FM-index corresponds to better performance, as it indicates more agreement between the method and the assessors on point pairs that fall in the same cluster. Table 1 presents performance of the methods averaged over all topics, and

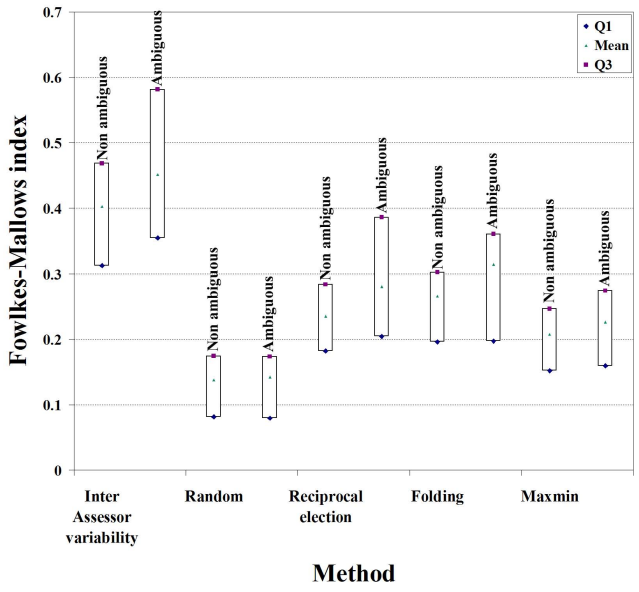

Figure 4: Performance of the three methods on the Fowlkes-Mallows index, compared to human assessments and the random baseline.

	Inter-assessor variability	Random	Reciprocal election	Folding	Maxmin
FM	0.419	0.139	0.250	0.282	0.214
VI	1.463	2.513	1.975	2.081	2.129

Table 1: Average performance over all topics and assessors.

with an FM-index of 0.282 folding outperforms again reciprocal election ($FM = 0.250$) and maxmin ($FM = 0.214$).

To test these claims for significance, we calculated p-values. The null-hypothesis that all methods perform equally well is rejected both times, with $p = 0.006$ for reciprocal election and $p = 2.3 \times 10^{-9}$ for maxmin. Moreover, Figure 5 shows per topic the FM-index for folding against the FM-index for reciprocal election and maxmin. For every topic under the equality line $y = x$, folding outperforms the other method. With respect to reciprocal election, folding outperforms 58% of the topics, and for maxmin this value is 73%.

The Fowlkes-Mallows index measures the degree of agreement on point pairs that fall in the same cluster under both clusterings. This measure is therefore rather sensitive to the number of clusters. The folding approach benefits from its strong mechanism to automatically and dynamically select a proper number of clusters.

Results on Variation of Information Metric

A different relative performance is given by the variation of information criterion. According to this measure, reciprocal election outperforms folding and maxmin. Mean, first and third quartile performance is given in Figure 6, while Table 1 presents the performance averaged over all topics. In this case, a lower variation of information indicates a better performance. It denotes that there is less change in cluster membership while going from one clustering to the other.

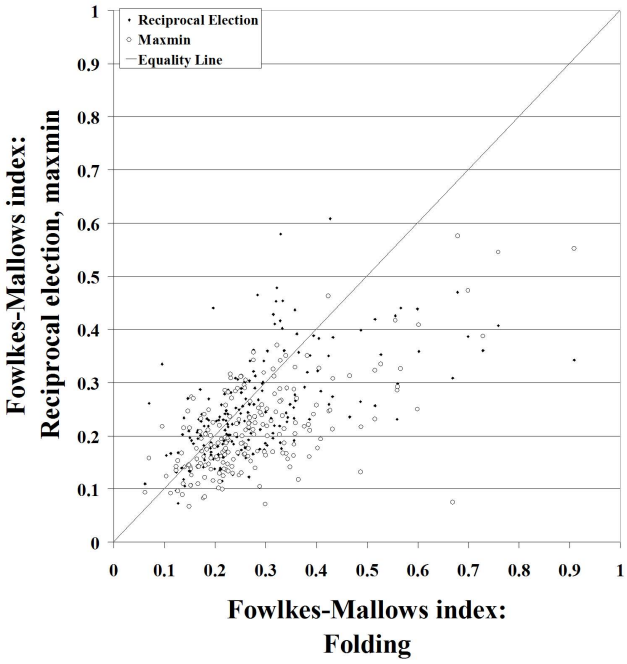

Figure 5: Performance evaluation per topic comparison. The plots show the performance of folding on the x-axis for each clustering comparison, with respect to the performance of the reciprocal election and maxmin on the y-axis. For every topic under the line, folding outperforms the corresponding other method.

Significance tests support the superiority of reciprocal election. The null hypothesis of all methods performing equally well is rejected with $p = 0.002$ for folding and with $p = 7.3 \times 10^{-7}$ for maxmin. Figure 7 presents relative performance comparisons per topic. It shows that the majority of folding and maxmin clusterings have a larger variation of information coefficient than reciprocal election, respectively 63% and 70%. In this figure, for every topic under the line reciprocal election achieves a better performance.

Rather than counting image pairs that fall in the same cluster under both clusterings, variation of information focuses on the relationship between an image and its cluster. It measures the difference in this relationship, averaged over all images, between the two clusterings. As this is a more general than counting successfully clustered image pairs, reciprocal election has a better overall performance. We conjecture that this is due to how the approach follows the intuition behind a cluster. Images in a cluster should all be well represented by that cluster, a notion that translates directly to how the reciprocal ranks are used as votes.

Ambiguous Topics vs. Non-ambiguous Topics

One more interesting result can be observed. Both Figure 6 and 4 clearly show that the assessors agree more on ambiguous topics than on non ambiguous topics. This is probably due to the fact that a more generally accepted clustering exists for topics that produce semantically different clusters. On non ambiguous topics the assessors may choose more different criteria to base their clustering on. This behavior is also visible in the performance of the methods; the performance on ambiguous topics is significantly better than on non ambiguous topics. This indicates the existence of clear visual dissimilarity between semantically different images.

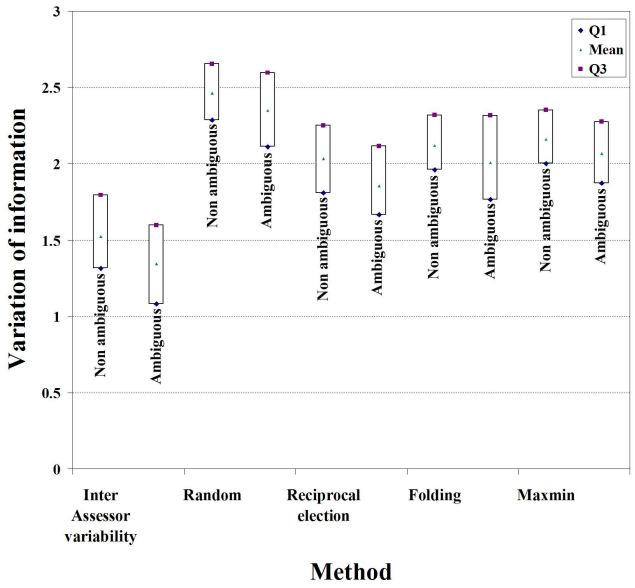

Figure 6: Performance of the three methods on the variation of information metric, compared to human assessments and the random baseline.

Parameter sensitivity of the algorithms

Both folding and maxmin are parameter free approaches. The number of clusters is determined automatically based on threshold ϵ, for which the appropriate value is calculated given a set of image search results. The image similarity measure or ranking function is also dynamic: weights for the visual features are established automatically for each set of image results.

Reciprocal election requires only parameter m, that determines the window size with which the ranked lists are inspected to decide upon cluster membership after a new representative has been found. We experimented with several values for this parameter m, and found more or less consistent performance for values between 3 and 8. The best performance was obtained using $m = 4$. With $5 <= m <= 8$, the variation of information increases slightly, but the method still outperforms the other approaches. Relative performance according to the Fowlkes-Mallows index did not change significantly either. With $m >= 9$, performance degrades quickly, because then the images are assigned to a representative too easily and clusters become too large.

6. CONCLUSION

Image search engines on the Web still rely heavily on textual metadata, causing a lack of visual diversity in image search results. Still, diversity is a highly desired feature of search results, no less in image search than in other search applications. In this paper, we present new methods to visually diversify image search results that deploy lightweight clustering techniques. These methods are effective, efficient

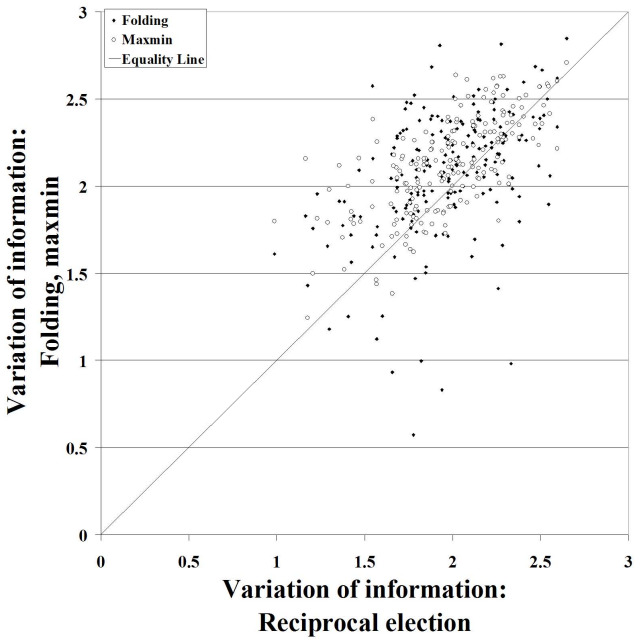

Figure 7: Performance evaluation per topic comparison. The plots show the performance of reciprocal election on the x-axis for each clustering comparison, with respect to the performance of the folding and maxmin on the y-axis. For every topic above the line, reciprocal election outperforms the corresponding other method.

and require no training nor parameter tuning. Given a user query, they adapt automatically to the set of image search results. The weights for visual features in a dynamic ranking function are computed on the fly to emphasize highly discriminant features for this set of results, and the number of clusters is adaptive as well.

The folding strategy respects the ranking order and picks the cluster representatives accordingly, while Reciprocal election aims to optimize the clustering and the (s)election of cluster representatives by a voting strategy where each image determines a list of candidate images that it would best represented by. After performing a large user-study to establish a ground truth and a baseline, we measure performance of all methods.

Folding shows a better performance according to the Folwkes-Mallows index, a performance measure that focuses on image pairs that can be formed with images from the same cluster. This indicates that the folding approach benefits from its strong mechanism to automatically and dynamically select a proper number of clusters. On the other hand, reciprocal election significantly outperforms the other methods in terms of variation of information, a more general performance measure. The selection of candidates and the decision on cluster membership both follow an intuitive notion behind a clustering. We conjecture that this is rewarded by means of a low variation of information, and therefore conclude that reciprocal election achieves the strongest overall performance.

As part of our future work, we plan to do an in depth evaluation of the feature weighting in the dynamic ranking function and to investigate the benefit of incorporating more features. Furthermore, we would like to relieve reciprocal election method from its parameter m and thereby making it parameter free as well, although it has proved to be insensitive to its setting to a certain extent. By performing an analysis of the distance distribution in the rankings, we will investigate means to set this parameter automatically as well. Finally, we will evaluate the quality of the cluster representatives and their suitability to serve as visually disambiguated query expansion in order to diversify the image search results beyond the scope of an initially returned set of images.

7. ACKNOWLEDGMENTS

The authors express their gratitude toward all the assessors that helped in the establishment of the ground truth.

8. REFERENCES

[1] A. Bhattacharyya. On a measure of divergence between two statistical populations defined by probability distribution. *Bull. Calcutta Math. Soc.*, 35:99–109, 1493.

[2] D. Cai, X. He, Z. Li, W.-Y. Ma, and J.-R. Wen. Hierarchical clustering of www image search results using visual, textual and link information. In *MULTIMEDIA '04: Proceedings of the 12th annual ACM international conference on Multimedia*, pages 952–959, New York, NY, USA, 2004. ACM.

[3] S. A. Chatzichristofis and Y. S. Boutalis. Cedd: Color and edge directivity descriptor: A compact descriptor for image indexing and retrieval. In A. Gasteratos, M. Vincze, and J. K. Tsotsos, editors, *ICVS*, volume 5008 of *Lecture Notes in Computer Science*, pages 312–322. Springer, 2008.

[4] R. Datta, D. Joshi, J. Li, and J. Z. Wang. Image retrieval: Ideas, influences, and trends of the new age. *ACM Computer Surveys*, 40(2):1–60, 2008.

[5] E. Fowlkes and C. Mallows. A method for comparing two hierarchical clusterings. *Journal of the American Statistical Association*, 78(383):553–569, 1983.

[6] J. Goldberger, S. Gordon, and H. Greenspan. Unsupervised image-set clustering using an information theoretic framework. *IEEE Transactions on Image Processing*, 15(2):449–458, 2006.

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

[8] Y. Liu, D. Zhang, G. Lu, and W.-Y. Ma. A survey of content-based image retrieval with high-level semantics. *Pattern Recognition*, 40(1):262–282, 2007.

[9] M. Meilă. Comparing clusterings : an information based distance. *Journal of Multivariate Analysis*, 98(5):873–895, 2007.

[10] P.-A. Moëllic, J.-E. Haugeard, and G. Pitel. Image clustering based on a shared nearest neighbors approach for tagged collections. In *CIVR '08: Proceedings of the 2008 international conference on Content-based image and video retrieval*, pages 269–278, New York, NY, USA, 2008. ACM.

[11] P. Salembier and T. Sikora. *Introduction to MPEG-7: Multimedia Content Description Interface*. John Wiley & Sons, Inc., New York, NY, USA, 2002.

[12] A. Smeulders, M. Worring, S. Santini, A. Gupta, and R. Jain. Content-based image retrieval at the end of the early years. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(12):1349–1380, Dec 2000.

[13] K. Song, Y. Tian, W. Gao, and T. Huang. Diversifying the image retrieval results. In *MULTIMEDIA '06: Proceedings of the 14th annual ACM international conference on Multimedia*, pages 707–710, New York, NY, USA, 2006. ACM.

[14] H. Tamura, S. Mori, and T. Yamawaki. Texture features corresponding to visual perception. *IEEE Transactions on Systems, Man and Cybernetics*, 8(6), 1978.

[15] R. van Zwol. Multimedia strategies for b3-sdr, based on principle component analysis. In *Advances in XML Information Retrieval*, Lecture Notes in Computer Science. Springer, 2006.

[16] R. van Zwol, V. Murdock, L. Garcia, , and G. Ramirez. Diversifying image search with user generated content. In *Proceedings of the International ACM Conference on Multimedia Information Retrieval*, 2008.

[17] D. Wallace. Comment. *Journal of the American Statistical Association*, 78(383):569–576, 1983.

[18] S. Wang, F. Jing, J. He, Q. Du, and L. Zhang. Igroup: presenting web image search results in semantic clusters. In *CHI '07: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 587–596, New York, NY, USA, 2007. ACM.

[19] K. Weinberger, M. Slaney, and R. van Zwol. Resolving tag ambiguity. In *Proceedings of the 16th International ACM Conference on Multimedia (MM 2008)*, Vancouver, Canada, November 2008.

[20] P. Wilkins, P. Ferguson, and A. F. Smeaton. Using score distributions for query-time fusion in multimediaretrieval. In *MIR '06: Proceedings of the 8th ACM international workshop on Multimedia information retrieval*, pages 51–60, New York, NY, USA, 2006. ACM.

[21] S. A. Yahia, P. Bhat, J. Shanmugasundaram, U. Srivastava, and E. Vee. Efficient online computation of diverse query results. In *Proceedings of VLDB*, 2007.

[22] B. Zhang, H. Li, Y. Liu, L. Ji, W. Xi, W. Fan, Z. Chen, and W.-Y. Ma. Improving web search results using affinity graph. In *SIGIR '05: Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 504–511, New York, NY, USA, 2005. ACM.

[23] C.-N. Ziegler, S. M. McNee, J. A. Konstan, and G. Lausen. Improving recommendation lists through topic diversification. In *WWW '05: Proceedings of the 14th international conference on World Wide Web*, pages 22–32, New York, NY, USA, 2005. ACM.