Two Step Relevance Feedback for Semantic Disambiguation in Image Retrieval

Daniel Heesch[1] and Stefan Rüger[2]

[1] Pixsta Research
London
United Kingdom
daniel.heesch@pixsta.com
[2] Knowledge Media Institute
The Open University
Milton Keynes
United Kingdom
s.rueger@open.ac.uk

Abstract. This paper presents a new approach to the problem of feature weighting for content based image retrieval. If a query image admits to multiple interpretations, user feedback on the set of returned images can be an effective tool to improve retrieval performance in subsequent rounds. For this to work, however, the first results set has to include representatives of the semantic facet of interest. We will argue that relevance feedback techniques that fix the distance metric for the first retrieval round are semantically biased and may fail to distil relevant semantic facets thus limiting the scope of relevance feedback. Our approach is based on the notion of the NN^k of a query image, defined as the set of images that are nearest neighbours of the query under *some* instantiation of a parametrised distance metric. Different neighbours may be viewed as representing different meanings of the query. By associating each NN^k with the parameters for which it was ranked closest to the query, the selection of relevant NN^k by a user provides us with parameters for the second retrieval round. We evaluate this two step relevance feedback technique on two collections and compare it to an alternative relevance feedback method and to an oracle for which the optimal parameter values are known.

1 Introduction

The question of how to infer semantic similarity between objects from their feature representations is central to content-based information retrieval. Not all features are equally well suited to pick up relevant properties of an object and often it is combinations of features that work best. How to combine evidence from different features depends on the particular object and is not known a priori. The problem is even more acute when objects are semantically rich and admit to a number of different interpretations, such as often holds for images.

M. Sebillo, G. Vitiello, and G. Schaefer (Eds.): VISUAL 2008, LNCS 5188, pp. 204–215, 2008.

Fig. 1. An image (leftmost) and the most similar images under different features

The optimal feature combination then also depends on the particular semantic facet the user is interested in.

As an illustration of semantic ambiguity or *polysemy* consider Figure 1. The three images to the right are visually closest to the image on the left under one of three different visual features (from left to right, texture, local colour distribution, global colour distribution). By ignoring colour content altogether the texture feature does not pick up the autumnal character of the query but captures well the distinct texture of leafy vegetation. The global colour feature is insensitive to how colours are distributed locally and ranks an image high if the overall proportions are similar as in a close-up view of fallen leaves. We may view the three images as representing different semantic facets of the query: "tree-like vegetation", "trees with autumn foliage", and "autumn colours". Which of these a user is interested in, and thus, which feature is to be given greater weight, cannot be decided on the basis of the query image alone. In such circumstances, some form of user feedback on the relevance of retrieved images becomes a methodological imperative. Indeed, relevance feedback features in many early research systems (e.g. [12], [14]), and has been researched extensively in more recent times (e.g. [5], [7], [10], [11], [15]). A large proportion of relevance feedback techniques are concerned with optimising parameters of a distance metric, often these are weights associated with feature-specific similarity scores. A widely overlooked issue is that of initialising feature weights. The standard practice of assigning uniform weights for the purpose of the first retrieval step renders the system semantically biased simply because any particular weight setting will favour some semantic facets over others. If users are interested in an aspect for which the initial setting is ill-suited, the scope for positive relevance feedback may be greatly compromised as few or no relevant items are brought to the surface in the first round. This limitation is bound to become more apparent as the image collections grow in size and users' information needs become more specific.

We explicitly address the problem of parameter initialisation by employing the idea of an image's NN^k [6]. This is the set of all images that are most similar to that image under *some* feature combination. Instead of retrieving with one weight set, we compute the top-ranked image under all possible weight vectors. The resulting set of images provide us with a pictorial representation of at least parts of the semantic spectrum of the query image. We forego high precision among the set of NN^k but thereby increase the chance that at least one of the returned images is relevant to a particular user. The set of NN^k now gives us a

powerful way to perform relevance feedback and to lift precision in the second step. The key is to associate each NN^k with the weight set under which it has been retrieved. By selecting NN^k, users thus implicitly select optimal distance metrics under which the chosen image is returned top. Following relevance feedback, images in the collection can be ranked according to this particular distance metric. The strength of this two step relevance feedback method therefore derives from two properties: it is semantically unbiased in the first step and it provides an effective way of associating images with optimal distance metrics to raise performance in the second step.

We tested the proposed method on two image collections of 8,200 and 32,000 images, respectively, and compared its performance with the relevance feedback method described in [12] as well as an oracle that knows the best distance metric for individual queries.

The paper is organised as follows. In Section 2 we describe related work. In Section 3 we define the set of NN^k and analyse some of their properties. Section 4 describes our relevance feedback technique and presents details of the evaluation. Section 5 reports results and Section 6 concludes the paper.

2 Related Work

In [1] relevance feedback is given on a set of images that are created on the fly by modifying segments of the query image in terms of shape, size and colour. Each modification can be viewed as representing a particular parameter setting under which the modified query is close to the original query. Positive relevance feedback is then utilised to update feature weights as in [13]. Although its motivation is different, the method is operationally similar to our approach. Instead of retrieving with a fixed parameter set, the user is presented with the spectrum of possible modifications. It is through interaction with these that parameters are initialised for the second step. The major limitations of the method are its dependence on good segmentation results and the small number of useful modifications that can be applied to the segments. When an image cannot meaningfully be segmented, users may find it difficult to judge the relevance of the modified images.

The idea of Bayesian sets [3, 8] shares the same motivation with ours but is otherwise very different. Based on several items exemplifying a certain concept, or, in our context, a certain information need, the system scores other items according to how well they fit into a set containing the exemplars. The requirement of submitting several items is key as the semantic facet can then be divined without recourse to relevance feedback.

3 NN^k and Their Properties

In [6], we define an image p to be an NN^k of image q if and only if there exists at least one convex combination of feature-specific distance functions $d_f(\cdot, q)$ for which p has minimal distance to q. Formally, p is an NN^k of q if and only if

$$\arg\min_i \left(\sum_{f=1}^{k} w_f d_f(i,q) \right) = p \tag{1}$$

for some $w = (w_1, w_2, \ldots, w_k)$ with $w_f \geq 0$ and $\sum w_f = 1$. k denotes the number of features. Note that we impose a convexity constraint on the weights without which *every* image would be an NN^k for *some* weight combination. Because weights are allowed to vary continuously, we cannot practically compute the NN^k for every weight set but need to discretise the space of all w that satisfy the convexity constraint and compute the weighted sum at a finite number of points. For $n+1$ grid points along each dimension of the weight space and k features, the total number of grid points can be shown to be $\binom{n+k-1}{n}$ [6].

3.1 Computational Speed-Up

We may reduce the computational cost inflicted at every grid point as follows. Recall that each vertex corresponds to a particular weight vector w. The nearest neighbour of the query for that weight vector is the image with the smallest distance to the query. As we recurse through the grid, most pairs of consecutive vertices differ in only two components of the weight vector. Instead of computing D as $\sum_{i=1}^{k} w_i d_i$ involving k multiplications and $k-1$ additions, we can exploit the fact that if two successive vertices only differ with respect to two dimensions, $k-2$ of the terms of the product have already been computed at the first vertex. In such cases we can obtain the distance at the new vertex by subtracting a correction term from the distance at the previous vertex. Let F be the set of indices whose weights differ between the two vertices. The update is then

$$D^j = D^{j-1} + \sum_{f:w_f \in F} (w_f^j - w_f^{j-1}) d_f,$$

where j indexes the step at which the vertex is visited during the recursion. For a collection of 8,200 images the performance benefits are quite appreciable and become greater as we increase the resolution of the grid. With eight features and six grid points per dimension, the update method speeds up performance by almost 40%.

3.2 Iso-NN^k Regions

We define an iso-NN^k region as the region in $\mathbb{R}^k$ that contains all the weight vectors for which the NN^k is the same. Since each point of the subspace is associated with exactly one NN^k, the set of NN^k partition the weight space. The boundaries of the partitions can in principle be determined arbitrarily close by increasing the resolution n. For $k = 3$, the set of permissible weight vectors lie in the two-dimensional triangular subspace of $\mathbb{R}^3$. Figure 2 shows the partitioning of the weight space into iso-NN^k regions for a particular query from the Corel collection, three features and varying resolution.

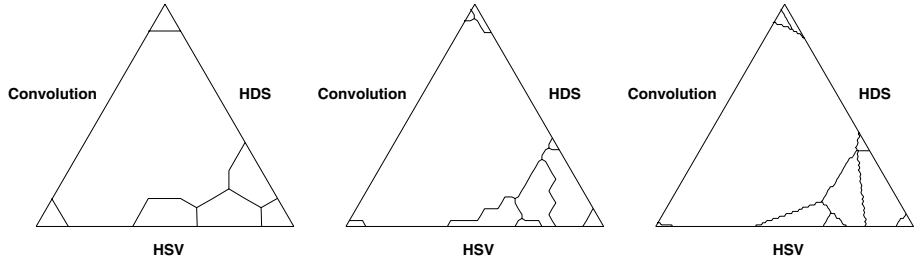

Fig. 2. Partitioning of the weight space for $n = 5, 10, 100$. Each region is associated with a particular image which is one of the NN^k.

3.3 Representative Weights

Let us define the *support* of an NN^k as the relative size of its iso-NNk region. The more feature combinations there are for which the image is ranked top, the greater its support. We may view the support $S(p, q)$ of an NN^k p as a measure of its relevance to the query q.

For the purpose of relevance feedback we wish to associate each NN^k with a characteristic weight set for which the image is closer to the query than is any other image. A natural way to define this is as the k-dimensional centre of mass of the iso-NN^k region. This can be approximated by the arithmetic average of all the weight vectors sampled within that region.

4 NN^k Relevance Feedback

4.1 Weight Estimation

Each NN^k is associated with the weight vector under which it is most similar to the query. It can be viewed as the optimal weight vector for the particular semantic facet represented by that NN^k. We should hope that more relevant images will surface if we retrieve more than just the top-ranked image (the selected NN^k) under that weight vector. The proposed relevance feedback method thus consists of two steps. In the first step, NN^k are computed for a particular query image using the techniques described in Section 3. In the second step, following relevance feedback, images are retrieved under the chosen weight set. If users select only one image from among the NN^k, the one ranked list induced by the corresponding weight set constitutes our final ranked list. If more than one image is selected we need to merge the ranked lists obtained from each of the weight vectors. Below we evaluate several different merging techniques.

The following sections investigate the effects of NN^k search on retrieval performance. As is customary for this purpose, we employ manually annotated image collections and automate the feedback process so that an evaluation can be carried out on a large scale.

4.2 Image Collections, Features and Performance Measures

Our evaluation is based on two collections. The first is a subset of the Corel 380,000 Photo Gallery with 31,997 images. The pre-assignment of images to categories by means of their position in the directory hierarchy greatly facilitates automatic evaluation as an image may be considered relevant to a query if it is from the same directory. By not allowing memberships to more than one category, however, the Corel classification fails to encode polysemy. To overcome this limitation we use the textual annotation that accompanies each image to form an alternative classification with overlapping classes. The joint vocabulary of the 31,997 images comprises 20,250 terms of which we only keep those that are associated with at least 20 and at most 100 images. Of the resulting 530 classes we discard those with too little visual coherence.

We also built a second collection of a more diverse kind with a richer and more consistent annotation than Corel. We downloaded 8,202 medium-resolution photographs from the Getty Image Archive (http://www.gettyimages.com) along with the annotations assigned by the Getty staff. The selection of photographs was obtained by submitting the query "photography, image, not composite, not enhancement, not 'studio setting', not people" to the Getty website with the additional search option to exclude illustrations. We thereby hoped to obtain a random selection of photographs that would exclude pictures with no photographic content, digitally composed or enhanced photos and any photos taken in a studio setting. Because the resulting dataset contains pictures from a number of different photo vendors we hope to reduce the chance of unrealistic correlations between images. The Getty collection contains 8,202 images from which classes are constructed as before. The joint vocabulary of 8,551 terms is reduced by retaining only those terms that are associated with at least 20 and at most 100 images. We again discard terms that we consider either too difficult for visual retrieval such as 'freshness' or too easy such as 'blue'. After these pruning steps the vocabulary has shrunk to 100 terms which we treat as class labels.

With images belonging to several classes, the same image may form more than one query. Because of the extensive reduction of the vocabulary, only a subset of the images are annotated with one of the remaining class labels and it is only these that we can employ as queries. For each collection we choose a random subset of 500 for evaluation purposes. A summary of various collection statistics is shown in Table 1.

Table 1. The two collections used for evaluation

	Corel	Getty
Number of images	31,997	8,202
Number of classes	191	100
Class size (avg)	49	44
Class size (range)	20–99	20–99
Avg polysemy	1.1	1.4
Max polysemy	5	7

A standard measure of retrieval performance in information retrieval is mean average precision. For the purpose of evaluating relevance feedback techniques, however, arguments based solely on mean average precision must be taken with caution. For when the set of relevant images is low, even a small shift of relevant images in the ranked list may result in substantial gains in average precision without necessarily leading to an increase in the number of relevant images displayed to the user. In addition to average precision we therefore measure performance in terms of precision at 50, Pr(50). We use eight texture and colour features including Tamura features, Gabor wavelets and local HSV colour histograms. More details of these can be found in [6].

4.3 Reference Performance

Because absolute performance tends to be very sensitive to the choice of features, the particular collection as well as the set of query images, systems are best evaluated in direct comparison with other methods under the same experimental conditions. In this paper, we benchmark NN^k search against three other methods: The first method weighs all features equally and provides us with a baseline. The second method, an oracle, weighs features according to an optimal weight vector that has been determined empirically for each individual query. No system that retrieves with only one weight vector can do better than this. The third method is an implementation of the weight update technique proposed by [12], a relevance feedback technique that aims to optimises a single weight vector. The three methods will be described in turn.

Baseline performance: The baseline performance in our experiments is obtained by assigning every feature the same weight. This is our best guess if we have no information about the relative performance of individual features, and is the initialisation strategy employed in the great majority of relevance feedback methods.

Oracle performance: We can define an *upper* performance bound by empirically determining the best weight vector for each individual query. This is an important benchmark because a large number of relevance feedback techniques are concerned with finding such a single weight vector. For such methods, the oracle provides the best possible performance. Our method could in principle exceed this bound because we may retrieve with as many weight vectors as there are NN^k selected by the user.

Owing to the discrete nature of ranks, an infinitesimal change in the weights may lead to a finite change in performance. As a result the functions mapping weights to average precision and Pr(50) are discontinuous. One way to optimise performance under such conditions is to provide a stochastic formulation of the objective function thereby rendering it differentiable, see for example [4]. Alternatively, we may carry out a grid search of the weight space and improve the grid-based estimate by a subsequent gradient ascent step. This is the method adopted here.

Figure 3 illustrates how retrieval performance varies for a particular query as we change the weight of a single feature. When performance is measured in

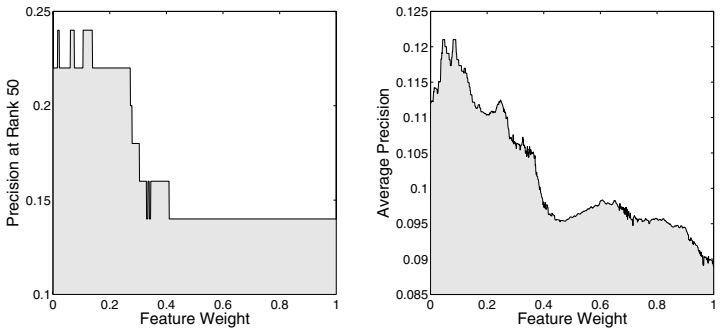

Fig. 3. Variation in retrieval performance as we vary one feature weight

terms of Pr(50) (left graph) the variation is visibly discontinuous. This is much less so for average precision (right graph). Indeed, we should be able to obtain an approximation of the slope of the average precision curve for the purpose of our gradient ascent step. Let j index the feature the weight of which we vary by a small amount h. An estimate of the jth component of the gradient is then given by

$$\frac{P(w) - P(w + \Delta w)}{h}, \tag{2}$$

where Δw is a vector whose jth element is h. Because the weights are sum-normalised, any increase in one feature weight must be accompanied by a corresponding decrease in the sum of all other weights. Strictly speaking, therefore, increasing the jth weight by h requires us to multiply all others by some common factor that ensures that they again sum to 1. We find the other components of Δw to be given by

$$\Delta w_i = \left(\frac{h}{w_j - 1}\right) w_i. \tag{3}$$

After identifying the weight and the direction (the sign of h) along which performance increases most, we perform a search along the line given by the set

$$\left\{ w \;\middle|\; w_i = 1 - \frac{t w_i^0}{1 - w_j^0},\ w_j = w_j^0 + t,\ t \in [-w_j^0, 1 - w_j^0] \right\}, \tag{4}$$

where j indicates the weight we vary, and w^0 denotes the starting weight. The w that maximises performance locally is found by moving along the line in small intervals, Δw, starting at w^0 until performance decreases. In Figure 3 the line search begins at $w_j^0 = 0$ and moves at intervals of 0.01 to the right. The line search terminates at the first peak $(0.05, 0.121)$ which also happens to be the global maximum in that direction. Subsequent iterations find the new direction along which performance can be increased further, and so on until performance decreases in every feature direction. In this case performance can be increased to above 0.14 with this method. Although we are not guaranteed to find the global

optimum, the final estimate generally comes very close to it as suggested by grid searches with very high resolution.

Weight update according to Rui: [12] derive an optimal solution for the feature weights w that minimises the summed distances between relevant images and the query with the distance of each relevant image being weighted by its relevance score v_i. Given N positive examples, the optimal w is found to satisfy

$$w_j \propto \left(\sum_{i=1}^{N} v_i d(p_{ij}, q_j) \right)^{-\frac{1}{2}}, \tag{5}$$

where we sum over weighted distances between the query and all relevant images under feature j. In our experiments we choose at random a maximum of ten relevant images and set their relevance scores to one. Note that since the aggregation formula for the feature-specific distances is linear in the feature weights, the proportionality constant in Equation 5 can be chosen arbitrarily. In our application of this method, relevance feedback is initially given on the results of a baseline run. We then allow for another two rounds of relevance feedback before measuring retrieval performance.

5 Results

The aim of the first step of the NN^k relevance feedback method is not to achieve high precision but rather to gather an eclectic set of images that represent a range of semantic facets of the query. Queries for which a uniform parameter initialisation happens to be appropriate are therefore expected to give better results under the baseline run than the first step of our method. Indeed, it turns out that the total number of relevant images returned from the first step of NN^k search is smaller than for a baseline search. However, by increasing the chance of containing at least one relevant image, our method may be expected to outperform the baseline search in the second step, the alternative update method by [12], and possibly even the performance attained by the oracle.

The main results are compiled in Table 2. It displays the absolute performance averaged over 500 queries for the baseline search, the optimised search, Rui's method, and NN^k search with three different merging methods. Round Robin (RR) merges the lists according to the highest rank achieved by an item in either of the lists. BordaFuse (BF) is another rank based method that has the effect of averaging over individual ranks [2]. CombSum (CS) ranks according to the average score.

Performance figures are very similar across collections. This may at first surprise as the Corel collection contains more than three times as many images as the Getty collection with roughly the same number of images per class. It appears, therefore, that the greater variability of images within Getty classes makes up for the smaller size and that it does indeed constitute a more challenging collection for retrieval.

Table 2. Retrieval Performance of NN^k search

	Corel		Getty	
	MAP (%)	Pr(50)(%)	MAP (%)	Pr(50) (%)
Baseline	2.29	1.58	2.95	1.64
Oracle	5.26	2.81	6.41	2.92
RUI	2.49	1.66	3.17	1.71
NN^k (RR)	4.45	2.12	5.76	2.28
NN^k (BF)	3.43	1.83	4.55	2.02
NN^k (CS)	3.79	1.95	4.91	2.14

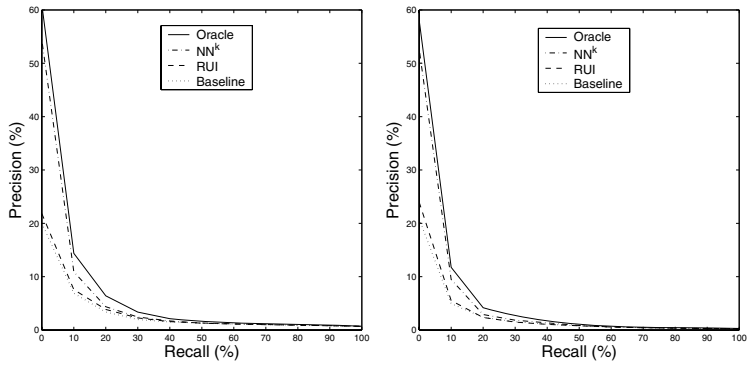

Fig. 4. Precision-recall graphs for 500 queries for Getty (left) and Corel (right)

From among the three merging techniques, the Round Robin method consistently achieves the greatest performance gains. The reason for the superior performance of Round Robin can be seen in the fact that, unlike Borda Fuse and CombSum, the final rank assigned by Round Robin to an object is only very weakly correlated with the average rank it occupies in the different lists to be merged. Averaging ranks or scores, which is the effect of CombSum and Borda Fuse, appears sensible in situations in which we do not want extreme individual ranks to dominate the final ranking. That they are less helpful in the context of image retrieval may result from the observation that relevance classes tend to be composed of several groups of visually similar images with little visual coherence between groups [9]. If the selected NN^k happen to come from different groups, we expect the different weight sets to be particularly good at retrieving more relevant images from the respective groups. Relevant images, therefore, are expected to be ranked high for some weight sets and low for others. Round Robin ensures that such images still receive a high overall rank.

Irrespective of the fusion technique, the performance of our NN^k-based method lies markedly above the baseline and reaches close to the performance achieved under an optimised single weight set (oracle). The differences to the baseline is statistically significant, but so is its difference to the optimal performance (for both we find $p < 0.001$ under a paired t-test).

As further confirmation of the previous observation, we note that our method does considerably better than the update method by [12]. In fact, the weight update according to Equation 5 does not seem to lift performance much above the baseline and, indeed, with $p = 0.066$ the difference in Pr(50) is not significant under a paired t-test. This is perhaps to be expected of semantically biased relevance feedback methods when evaluated on realistic image collections. Visual confirmation of these claims comes from the precision against recall curves obtained by averaging over the 500 queries (Figure 4).

6 Conclusions

We have proposed a new relevance feedback technique for content-based image retrieval that addresses the problem of parameter initialisation and weight optimisation for image retrieval. We propose a solution in which parameters are not initialised at all for the first retrieval step. Instead of retrieving with some fixed parameter setting, we compute the images that are most similar to the query under *at least one* parameter setting. These images represent pictorially the range of semantic facets users may have had in mind when posing the query. In addition to covering a wider range of possible image interpretations, and thus being more likely to capture the semantic facet that matters to the user, this first step makes it possible to associate each of the images with a characteristic weight set, which is the average weight set under which that image is closest to the query. By selecting relevant images from among the retrieved set, users thus implicitly select weight sets which can subsequently be used for retrieval. The method has been tested on two image collections for both of which retrieval performance gets close to the optimum performance and consistently outperforms an alternative relevance feedback method.

References

[1] Aggarwal, G., Ashwin, T., Ghosal, S.: An image retrieval system with automatic query modification. IEEE Trans Multimedia, 4(2), 201–213 (2002)
[2] Aslam, J., Montague, M.: Models for metasearch. In: Proc. Int'l ACM SIGIR, pp. 276–284 (2001)
[3] Ghahramani, Z., Heller, K.: Bayesian sets. In: Proc. NIPS (2005)
[4] Goldberger, J., Roweis, S., Hinton, G., Salakhutdinov, R.: Neighbourhood component analysis. In: Proc. NIPS (2005)
[5] Gosselin, P., Cord, M.: Active learning methods for interactive image retrieval. IEEE Trans. Image Processing (to appear, 2008)
[6] Heesch, D.: The NN^k technique for image searching and browsing. PhD thesis, Imperial College London (2005)
[7] Heesch, D., Rüger, S.: Interaction models and relevance feedback in content-based image retrieval. In: Zhang, Y.-J. (ed.) Semantic-Based Visual Information Retrieval, pp. 160–186. Idea-Group (2006)
[8] Heller, K., Ghahramani, Z.: A simple Bayesian framework for content-based image retrieval. In: Proc. IEEE Conf. Computer Vision and Pattern Recognition, vol. 2, pp. 2110–2117 (2006)

[9] Kim, D.-H., Chung, C.-W.: Qcluster: relevance feedback using adaptive clustering for content-based image retrieval. In: Proc. ACM SIGMOD Int'l Conf. Management of Data, pp. 599–610 (2003)
[10] Nguyen, G., Worring, M., Smeulders, A.: Interactive search by direct manipulation of dissimilarity space. IEEE Trans. Multimedia (to appear, 2008)
[11] Qin, T., Zhang, X.-D., Liu, T.-Y., Wang, D.-S., Ma, W.-Y., Zhang, H.-J.: An active feedback framework for image retrieval. Pattern Recognition Letters 29, 637–646 (2008)
[12] Rui, Y., Huang, T.: Optimizing learning in image retrieval. In: Proc. IEEE Conf. Computer Vision and Pattern Recognition, pp. 236–243 (2000)
[13] Rui, Y., Huang, T., Ortega, M., Mehrotra, S.: Relevance feedback: A power tool for interactive content-based image retrieval. IEEE Trans. Circuits, Systems and Video Technology 8(5), 644–655 (1998)
[14] Sclaroff, S., Taycher, L., La Cascia, M.: ImageRover: A content-based image browser for the WWW. Technical report, Boston University (1997)
[15] Urban, J., Jose, J.: Evidence combination for multi-point query learning in content-based image retrieval. In: Proc. IEEE Int'l Symposium Multimedia Software Engineering, pp. 583–586 (2004)