

A new CBIR approach based on relevance feedback and optimum-path forest classification

André Tavares da Silva
Unicamp, Brazil
atavares@dca.fee.unicamp.br

Alexandre Xavier
Falcão
Unicamp, Brazil
afalcao@ic.unicamp.br

Léo Pini Magalhães
Unicamp, Brazil
leopini@dca.fee.unicamp.br

ABSTRACT

Recently some CBIR approaches have shown the use of relevance feedback to train a pattern classifier to select relevant images for retrieval. This paper revisits this strategy by using an optimum-path forest (OPF) classifier. During relevance feedback iterations, the proposed method uses the OPF classifier to decide which database images are relevant or not. Images classified as relevant are sorted and presented to the user for a new iteration. Such images are ordered according to the normalized distance using relevant and irrelevant representative images, computed previously by the OPF classifier. Our experiments show that the proposed approach requires fewer iterations, being faster and more effective than methods based on SVM.

Keywords: CBIR, Relevance Feedback, OPF.

1 INTRODUCTION

Image collections have been created and used in several applications, such as digital libraries, medicine, and biodiversity information systems. Given the size of these collections, it is essential to provide efficient and effective means to retrieve images. Such a problem has raised the interest in putting together image processing, information retrieval, and database management to design content-based image retrieval (CBIR) systems for large image collections [2].

The visual content of an image in a CBIR system is often represented by a *feature vector*, which may encode color, texture, and/or shape measures. The image is then interpreted as a *point* in the feature space. A query in a CBIR system is usually done by range (returning all images whose distance to the query image is less than a given radius) or by similarity. We will focus on the second approach where a number of the closest images to the query point are retrieved from the database. Given that the meaning of those images may differ for distinct users since they are not completely represented by the feature vector, a *semantic gap* occurs between the user's expectation and the result of the query. Thus, *relevance feedback* techniques have been investigated to reduce the semantic gap by requiring more user interaction than simply the specification of a query image. These techniques usually involve three steps: (i) a small number of retrieved images is pre-

sented to the user; (ii) the user indicates which images are relevant; (iii) the system learns the user's opinion from this feedback in order to return more relevant images in the next iteration. This process may be repeated until the user is satisfied, but it is highly desirable to finish it in a few iterations.

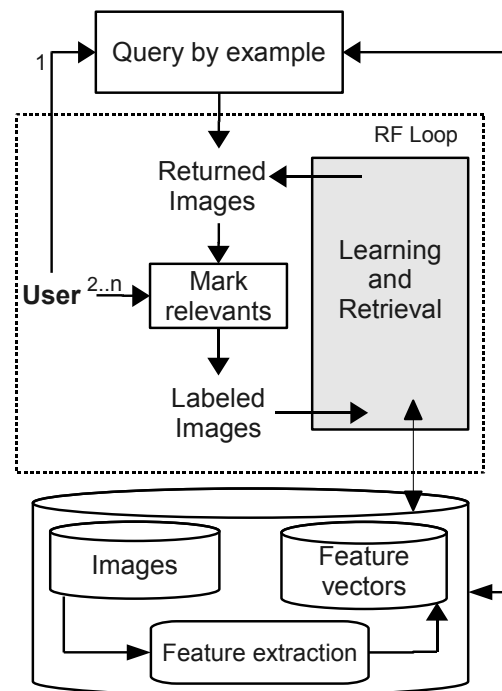


Figure 1: CBIR with Relevance Feedback.

Figure 1 shows an overview of the relevance feedback process. There are several studies on each stage of this pipeline, such as creating more robust local descriptors[15, 19] (i.e. feature vectors and distance functions to compare them) or providing scalability to

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

huge image databases[9, 20]. Our work focuses on the learning and retrieval process (gray box in Figure 1), especially in query classification and ranking.

Relevance feedback techniques were initially proposed for document retrieval, but have been successfully applied to CBIR systems[2, 13, 14, 17, 23]. Figure 2 illustrates three examples of simple relevance feedback techniques [7, 10]. In Figure 2a, the positive examples (relevant images) from a first iteration are used to move the next query point to their geometric center in the feature space. This idea stemmed from Rocchio’s formula [13] in document retrieval and it has been successfully exploited in CBIR systems, such as MARS [14] and MindReader [6]. Two other methods use the relevant images as next query points and, depending on the distance to this multi-point query set, different isosurfaces are formed in the feature space (Figures 2b and 2c). The method we present here is also a multi-point query but, differently from those approaches, we exploit relevant and irrelevant images as query points.

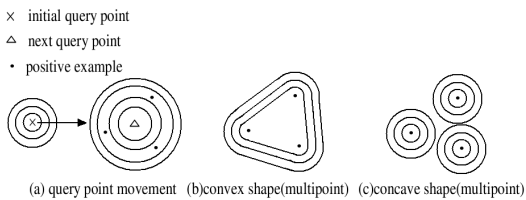


Figure 2: Simple relevance feedback techniques that change query shapes (i.e., isosurfaces with respect to the query points).

Approaches based on relevant and irrelevant images usually exploit active learning techniques to design a classifier that selects from the database the candidate relevant images for sorting by distance to the query point [1, 3, 16]. The method proposed by Tong and Chang. [16] uses Support Vector Machines (SVM) for image classification [22]. During the relevance feedback iterations, the method finds the optimum hyperplane that separates relevant and irrelevant images, presenting to the user the images closer to the hyperplane. This hyperplane is adjusted along the iterations and, after a last iteration, the method presents the images farther to the hyperplane, on its relevant side.

The method we propose here follows a similar strategy, using a faster and more effective classifier aiming to present the most relevant images in the database at each iteration, unlike SVM. For a given set of relevant and irrelevant images, the method designs an OPF (Optimum-Path Forest) classifier [12]. Only database images classified as relevant are sorted by distance and presented to the user in the next iteration. This distance is computed based on relevant and irrelevant prototypes (representative images), computed during the training

of OPF classifier. We show that this strategy is actually very effective reducing considerably the number of required iterations.

In addition to that, the *distance function* used to compare images has also influence on the retrieval process. Some methods use multiple pairs of feature vectors and distance functions, called *descriptors*, and compare two images by combining their distance based on each descriptor [11, 18]. In this case, the learning from relevance feedback may also change the way to combine descriptors [18]. Our method can exploit the same framework, but we will consider in this study only a single descriptor per image.

This paper is organized as follows. Section 2 presents the proposed algorithm based on OPF classifier and an example illustrates our relevance feedback process. The experiments and results using three heterogeneous image databases are described in Section 3. As baselines for comparison, we use the method of Tong and Chang [16] and the one illustrated in Figure 2c, which uses only relevant images for multi-point query. Section 4 states the conclusions and discusses our future work.

2 CBIR USING OPF CLASSIFIER

OPF is a classification method which represents each class of objects by one or more optimum-path trees rooted at given samples, called prototypes [12]. The training samples are nodes of a complete graph, whose arcs are weighted by the distance between the feature vectors of their nodes. In relevance feedback, we have two classes: relevant images chosen by the user and irrelevant ones. The prototypes computed by the OPF classifier are then used to sort the images according to the user’s selection.

Let \mathcal{L} be an image database. For every image $t \in \mathcal{L}$, we have a feature vector $\vec{v}(t) \in \mathfrak{R}^n$. That is, every image may be interpreted as a point in the feature space \mathfrak{R}^n . The distance $d(s, t)$ between two images s and t is the distance between their corresponding feature vectors. For an initial query point s , the proposed method returns the N closest images in \mathcal{L} to s (query by similarity). Due to the semantic gap, the closest images to s may not be the most relevant for a given user. By marking the relevant images among the returned ones, the user creates two sets: a set $\mathcal{I} \subset \mathcal{L}$ of irrelevant images and a set $\mathcal{R} \subset \mathcal{L}$ of relevant images. The method then uses sets \mathcal{R} and \mathcal{I} to compute two optimum-path forests (OPF), one for each class. Each database image $t \in \mathcal{L} \setminus \mathcal{I} \cup \mathcal{R}$ is then classified according to the root’s label of the forest (relevant/irrelevant) which offers to t the optimum path in the graph. Only the N closest images labeled as relevant will be returned in a set \mathcal{C} to the user in the next iteration. Relevant prototypes (\mathcal{A}) and irrelevant ones (\mathcal{B}), computed in the previous step, are then used to sort the images in \mathcal{C} for the next iteration.

The method computes the average distance $\bar{d}_{\mathcal{A}}(t, \mathcal{A})$ between each image $t \in \mathcal{C}$ and images in the set of relevant prototypes \mathcal{A} . It also computes the average distance $\bar{d}_{\mathcal{B}}(t, \mathcal{B})$ between t and images in the set of irrelevant prototypes \mathcal{B} . Finally, a distance $\bar{d}(t, \mathcal{A}, \mathcal{B})$ is computed as a normalized mean between relevant and irrelevant prototypes:

$$\bar{d}(t, \mathcal{A}, \mathcal{B}) = \frac{\bar{d}_{\mathcal{A}}(t, \mathcal{A})}{\bar{d}_{\mathcal{A}}(t, \mathcal{A}) + \bar{d}_{\mathcal{B}}(t, \mathcal{B})}.$$

Algorithm 1: Relevance Feedback Algorithm

Input: A query image s , a feature extraction function v , a distance function d , a desirable number N of relevant images, an image database \mathcal{Z} and a number T of iterations.

Output: An ordered list L of the N most relevant images in \mathcal{Z} .

Auxiliary: Sets $\mathcal{R} \subset \mathcal{Z}$ and $\mathcal{I} \subset \mathcal{Z}$ of relevant and irrelevant images, $\mathcal{A} \subset \mathcal{Z}$ and $\mathcal{B} \subset \mathcal{Z}$ of relevant and irrelevant prototypes, set $\mathcal{C} \subset \mathcal{Z}$ of images classified as relevant for the next iteration.

- 1 Compute the distance $d(s, t)$ for every image $t \in \mathcal{Z}$.
 - 2 Create an ordered list L of the N closest images t to s based on $d(s, t)$.
 - 3 Set $\mathcal{I} \leftarrow \emptyset$ and $\mathcal{R} \leftarrow \emptyset$.
 - 4 **for** each learning iteration $i = 1, 2, \dots, T$ **do**
 - 5 Set $\mathcal{C} \leftarrow \emptyset$.
 - 6 The user marks the relevant images in L , which are inserted into \mathcal{R} and the irrelevant ones are inserted into \mathcal{I} .
 - 7 **if** $|\mathcal{R}| < N$ **then**
 - 8 Compute OPF using sets \mathcal{I} and \mathcal{R} , resulting also \mathcal{A} and \mathcal{B} .
 - 9 **for** each image $t \in \mathcal{Z} \setminus \mathcal{I} \cup \mathcal{R}$ **do**
 - 10 **if** t is labeled as relevant by OPF **then**
 - 11 insert t into the set \mathcal{C} of images classified as relevant.
 - 12 **end**
 - 13 **end**
 - 14 **end**
 - 15 **else**
 - 16 Return the final ordered list L with the N most relevant images in \mathcal{R} , as defined by the user's selection.
 - 17 **end**
 - 18 Create an ordered list L with the N most relevant images in \mathcal{C} , in increasing order of $\bar{d}(t, \mathcal{A}, \mathcal{B})$.
 - 19 **end**
 - 20 Return the final ordered list L with the N most relevant images in \mathcal{R} , completing it with the $N - |\mathcal{R}|$ relevant images in \mathcal{C} in the increasing order of $\bar{d}(t, \mathcal{A}, \mathcal{B})$.
-

After classifying each image in $\mathcal{Z} \setminus \mathcal{I} \cup \mathcal{R}$, the method returns to the user a new set of N relevant images, which contains the lowest values of $\bar{d}(t, \mathcal{A}, \mathcal{B})$. This process is then repeated for a few iterations T and, finally, the system returns all relevant images obtained so far.

In order to illustrate the advantages of our relevance feedback approach as compared to a simple retrieval of the N closest images to s , we present an example of query image in Figure 3 from the image database Corel [21]. We use a color descriptor, called BIC, proposed by Stehling et al. [15]. The $N = 30$ closest images in that database are shown in Figure 16, where the relevant images are presented with a blue border. After $T = 3$ iterations (a reasonable number of iterations for practical situations), the system presents the $N = 30$ most relevant images found so far, as shown in Figure 17. It is important to note that the quality of this result may vary depending on the image descriptor.



Figure 3: A query image s .

3 EXPERIMENTS AND RESULTS

In order to evaluate our method, we use the BIC descriptor with the dLog distance function [15], and compare its effectiveness using precision-recall curves and two other approaches as baselines: the SVM-based method proposed by Tong and Chang [16] and the multi-point query with relevant images only, as illustrated in Figure 2c. The first, named here as SAL (SVM Active Learning), is also named SVM_{ACTIVE} or SVM_{AL} in the literature. It was chosen because it is based on a state-of-the-art technique for image classification. The second, named as QPM (Query Point Movement) [16], was selected to illustrate the importance of irrelevant images in the multi-point query set. Our approach is named here OPF_{AL} or simply OPF, because it is based on the OPF classifier.

As mentioned before, our work focuses in query classification and ranking. Indexing schemes to accelerate the search can be exploited in our method, as well as techniques for descriptor combination. But we consider in this study only a single descriptor in order to compare the proposed method against others.

The curves of precision-recall use the entire image database \mathcal{L} . Thus, lines 18 and 20 of our algorithm are replaced by: Create a list L with all relevant images in $\mathcal{C} \cup \mathcal{R}$, in their increasing order of $\bar{d}(t, \mathcal{A}, \mathcal{B})$, and compute the precision-recall curve for images in L .

The experiments used three heterogeneous image databases, representing different challenges for CBIR.

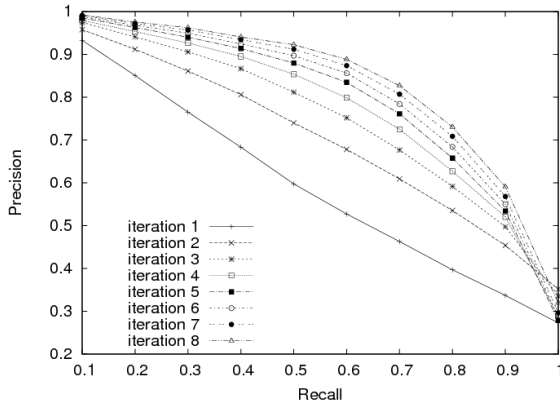


Figure 4: Mean precision-recall curves of OPF in Corel database, iterations 1 to 8.

- PASCAL [4, 5].

This database consists of 3,448 natural images, each one containing multiple regions of interest (subimages). Each region contains one object from a class of visual objects (bikes, boats, birds). The regions are labeled by their class performing a total of 23 classes with different number of images, varying from 72 to 446 subimages each.

- Corel [21].

This database is a collection with 200,000 images from the Corel GALLERY Magic-Stock Photo Library 2. We use a subset of 3,906 natural images, pre-classified into 85 classes. These classes have different number of images varying from 7 to 98 images each.

- ETH-80 [8].

This database is available in the project COGVIS, serving for both psychophysical and computational studies concerning object recognition and categorization. The project includes images of objects from 8 basic-level categories performing a total of 2,384 images, distributed uniformly among the classes.

For each image database, we simulate the user behavior by using each image as initial query point and marking the relevant points (images from the same class of the query) from 30 returned images at each iteration.

First, we present in Figures 4, 5 and 6 the mean precision-recall curves of OPF for the databases

Corel, ETH-80, and PASCAL, respectively, by varying the number of iterations from 1 to 8. These curves show that OPF improves its performance (the higher the precision-recall curve, the better is the method) with the number of iterations, as expected, but they also indicate the challenge degree of each database: PASCAL imposes more challenges than Corel which is more difficult than ETH-80.

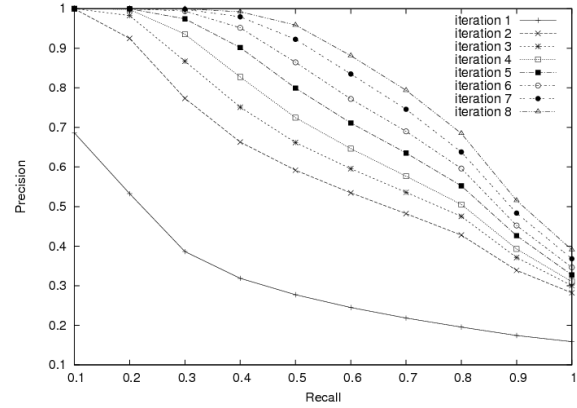


Figure 5: Mean precision-recall curves of OPF in ETH-80 database, iterations 1 to 8.

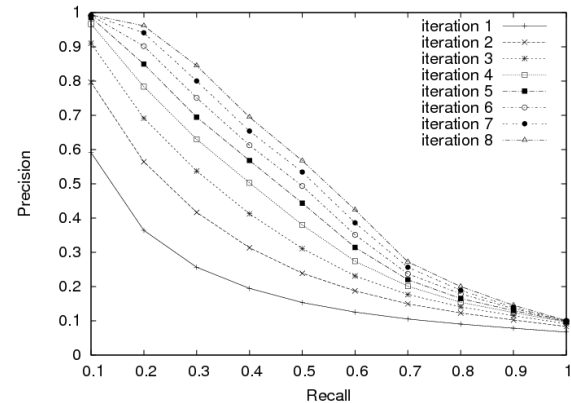


Figure 6: Mean precision-recall curves of OPF in PASCAL database, iterations 1 to 8.

Following, Figures 7 to 15 show the mean precision-recall curves of each method (OPF, QPM, and SAL) in each database (Corel, ETH-80, and PASCAL) for 3, 5 and 8 relevance feedback iterations. One may observe that OPF outperformed SAL and QPM in the most difficult databases, Corel and Pascal, and for all number of iterations. In the easiest case, ETH-80, the curves cross each other in some recall rates, but OPF is still better than the others up to 40% of recall for 3 and 5 iterations, and 50% of recall for 8 iterations. In addition to that, OPF is much faster than SAL and it learns quicker the simulated user's wish, providing effective results in fewer iterations. We consider 3 iterations as the ideal

number for practical situations. OPF has also outperformed QPM in all cases and this indicates the importance of using relevant and irrelevant points in multi-point query systems rather than only relevant points.

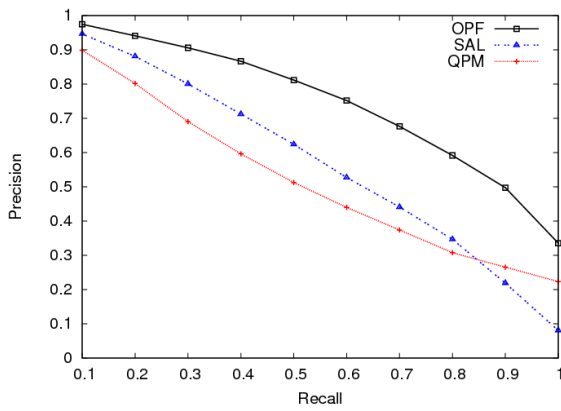


Figure 7: Mean precision-recall curves in Corel database, third iteration.

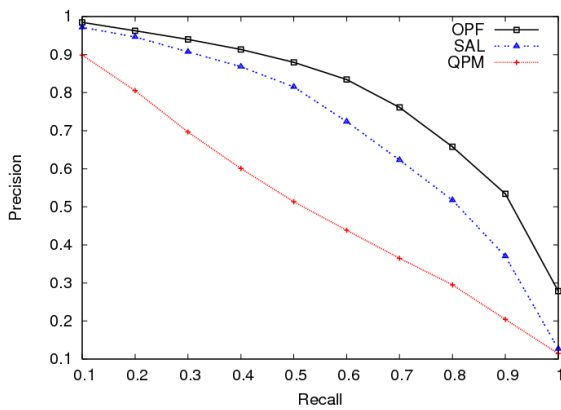


Figure 8: Mean precision-recall curves in Corel database, fifth iteration.

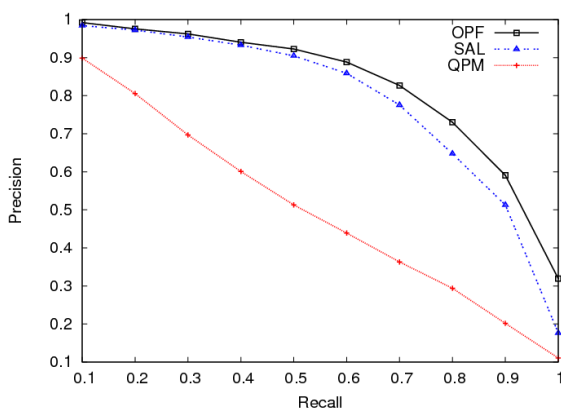


Figure 9: Mean precision-recall curves in Corel database, eighth iteration.

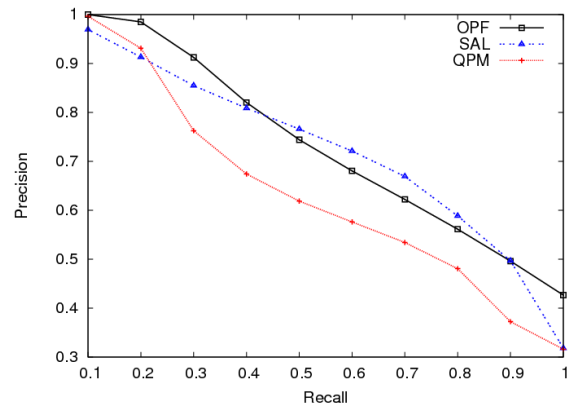


Figure 10: Mean precision-recall curves in ETH-80 database, third iteration.

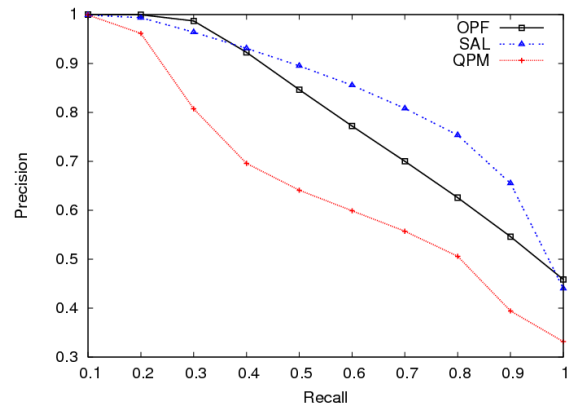


Figure 11: Mean precision-recall curves in ETH-80 database, fifth iteration.

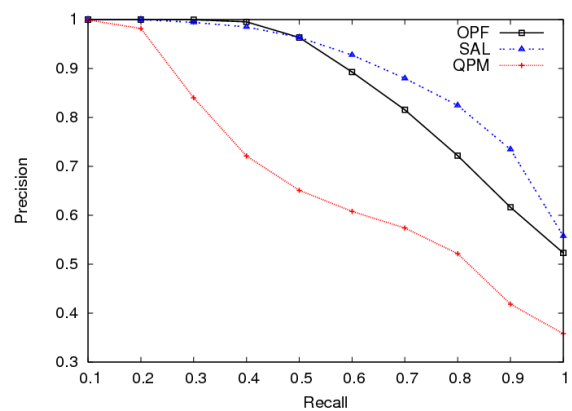


Figure 12: Mean precision-recall curves in ETH-80 database, eighth iteration.

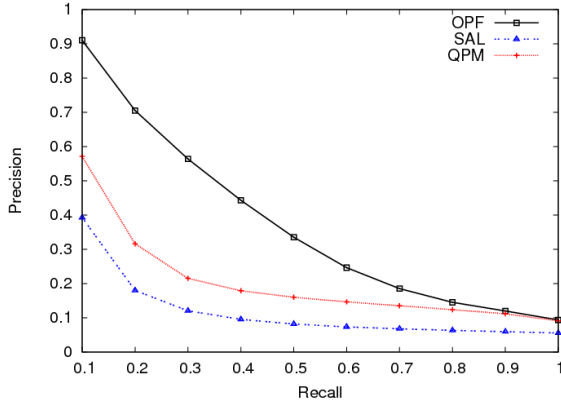


Figure 13: Mean precision-recall curves in PASCAL database, third iteration.

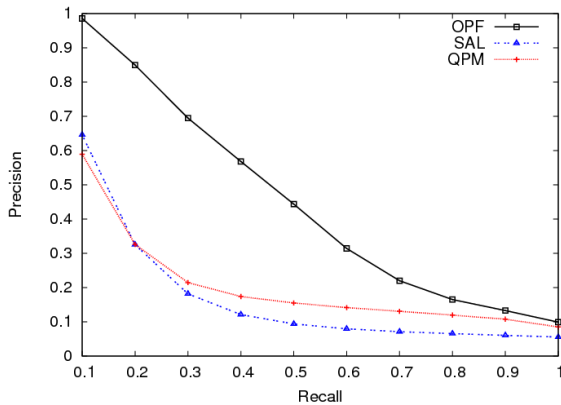


Figure 14: Mean precision-recall curves in PASCAL database, fifth iteration.

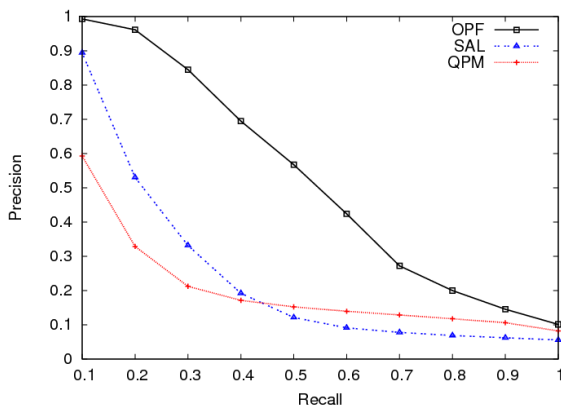


Figure 15: Mean precision-recall curves in PASCAL database, eighth iteration.

Tables 1 and 2 show the execution times of SAL and OPF, respectively. We present time values for all images of the databases for iterations 3, 5 and 8, used to compute the precision-recall curves of Figures 7 to 15.

Table 3 shows the average execution time for one iteration of SAL and OPF. Our approach could also be used with indexing schemes to further accelerate the search process and reduce the execution time. However, we present in this paper execution times without indexing structures. As the number of iterations grows, the runtime increases. In the Corel database for instance, our method takes 0.4 seconds to present images at the eighth iteration while SAL takes 10.2 seconds in the average. The tests were performed in a machine with Intel Pentium D processor at 3.4GHz and 1 GB RAM running the Linux operational system.

Table 1: Total execution time of SAL (minutes).

Database	Corel	ETH-80	PASCAL
3 iterations	1,116	420	903
5 iterations	2,170	702	1,743
8 iterations	5,330	1,120	4,483

Table 2: Total execution time of OPF (minutes).

Database	Corel	ETH-80	PASCAL
3 iterations	42.8	24.2	33.4
5 iterations	102.9	38.9	81.2
8 iterations	224.0	59.1	188.4

Table 3: Average execution time per query (seconds).

Database	Corel	ETH-80	PASCAL
SAL	5.71	3.53	1.96
OPF	0.22	0.20	0.19

Methods such as OPF and SAL classify candidate relevant images in the image database and sort them to select the N closest to the query point(s). One may ask about the relevant images misclassified as irrelevant. These images are lost by the system. Table 4 presents the percentage of images that were erroneously discarded by OPF for each database and for iterations 3, 5 and 8; the percentages of missed relevant images are insignificant. This result is even more important when we consider that the performance of CBIR systems, as mentioned before, usually increases with the number of descriptors and strategies to combine them [18], which is not being exploited in the present study.

Table 4: Percentages of relevant images missed by OPF due to classification in each database.

Database	Corel	ETH-80	PASCAL
3 iterations	0.36%	1.23%	3.37%
5 iterations	0.32%	1.21%	3.22%
8 iterations	0.27%	1.02%	3.06%

4 CONCLUSION AND FUTURE WORK

We presented a new relevance feedback technique for CBIR. This is the first time that the OPF classifier is being used and evaluated for small training sets, as required in learning by relevance feedback. Differently from the original method, we have separated the optimum-path forests of each class for classification. This constitutes a simple but very effective variant of the original method. We have also proposed a new order relation among the relevant images, which is based on the mean distances to the prototypes of the OPF classifier.

We have evaluated the method using a color descriptor, three heterogeneous image databases, two reference approaches, a few iterations, and query by similarity. The results indicated that the proposed method, named OPF, requires fewer iterations of relevance feedback. It outperformed the reference approaches in all databases and the number of missed relevant images due to classification was insignificant.

The new CBIR approach based on relevance feedback and optimum-path forest classification presented in this paper provides a solution in interactive time for practical applications. On average our method was twenty times faster than SAL.

Our future work involves the use of multiple descriptors and techniques to combine them. We intend to use other descriptors based on shape, texture and color and combine them by using techniques such as Bayesian framework, Genetic Programming [18] or other similar approaches. We also intend to investigate other image classifiers and to evaluate the methods for multiple users.

ACKNOWLEDGEMENTS

The first author thanks CNPq for financial support (140968/2007-5). The second author thanks CNPq (project ARPis, 481556/2009-5) and FAPESP (07/52015-0).

REFERENCES

- [1] M. Cord, J. Fournier, and S. Philipp-Foliguet. Exploration and search-by-similarity in cbir. In *SIBGRAPI '03: Proceedings of the Brazilian Symposium on Computer Graphics and Image Processing*, pages 175–182. IEEE Computer Society, 2003.
- [2] R. Datta, D. Joshi, J. Li, and J. Z. Wang. Image retrieval: Ideas, influences, and trends of the new age. *ACM Comput. Surv.*, 40(2):1–60, 2008.
- [3] L. Duan, W. Gao, W. Zeng, and D. Zhao. Adaptive relevance feedback based on bayesian inference for image retrieval. *Signal Process.*, 85(2):395–399, 2005.
- [4] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results. <http://www.pascal-network.org/challenges/VOC/voc2007/workshop/index.html>.
- [5] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2008 (VOC2008) Results. <http://www.pascal-network.org/challenges/VOC/voc2008/workshop/index.html>.
- [6] R. A. Johnson and D. W. Wichern. *Applied Multivariate Statistical Analysis*. Upper Saddle River, NJ: Prentice Hall, 2002.
- [7] D.-H. Kim and C.-W. Chung. Qcluster: Relevance feedback using adaptive clustering for content-based image retrieval. In *In Proc. of the ACM SIGMOD Int. Conf. on Management of Data*, pages 599–610, 2003.
- [8] B. Leibe and B. Schiele. Analyzing appearance and contour based methods for object categorization. In *In IEEE Conference on Computer Vision and Pattern Recognition (CVPR'03)*, pages 409–415, 2003.
- [9] H. Lejsek, F. Ásmundsson, B. Jónsson, and L. Amsaleg. An efficient disk-based index for approximative search in very large high-dimensional collections. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(5):869–883, 2008.
- [10] D. Liu, K. A. Hua, K. Vu, and N. Yu. Fast query point movement techniques for large cbir systems. *IEEE Trans. on Knowl. and Data Eng.*, 21(5):729–743, 2009.
- [11] R. Ohbuchi and Yushin Hata. Combining multiresolution shape descriptors for 3d model retrieval. In *Proc. WSCG 2006*, Plzen, Czech Republic, 2006.
- [12] J. P. Papa, A. X. Falcão, and C. T. N. Suzuki. Supervised pattern classification based on optimum-path forest. *International Journal of Imaging Systems and Technology*, 19(2):120–131, 2009.
- [13] J. J. Rocchio. *Relevance feedback in information retrieval*, pages 313–323. Prentice-Hall, Englewood Cliffs, NJ, USA, 1971.
- [14] Y. Rui, T. S. Huang, and S. Mehrotra. Content-based image retrieval with relevance feedback in mars. In *In Proc. IEEE Int. Conf. on Image Proc.*, pages 815–818, 1997.
- [15] R. O. Stehling, M. A. Nascimento, and A. X. Falcão. A compact and efficient image retrieval approach based on border/interior pixel classification. In *CIKM '02: Proceedings of the eleventh international conference on Information and knowledge management*, pages 102–109, New York, NY, USA, 2002. ACM.
- [16] S. Tong and E. Chang. Support vector machine active learning for image retrieval. In *MULTIMEDIA '01: Proceedings of the ninth ACM international conference on Multimedia*, pages 107–118, New York, NY, USA, 2001. ACM.
- [17] R. S. Torres and A. X. Falcão. Content-based image retrieval: Theory and applications. *Revista de Informática Teórica e Aplicada*, 13(2):161–185, 2006.
- [18] R.S. Torres, A.X. Falcão, M.A. Gonçalves, J.P. Papa, B. Zhang, W. Fan, and E.A. Fox. A genetic programming framework for content-based image retrieval. *Pattern Recognition*, 42(2):217–312, Feb 2009.
- [19] T. Tuytelaars and K. Mikolajczyk. Local invariant feature detectors: a survey. *Found. Trends. Comput. Graph. Vis.*, 3(3):177–280, 2008.
- [20] E. Valle, M. Cord, and S. Philipp-Foliguet. High-dimensional descriptor indexing for large multimedia databases. In *CIKM '08: Proceeding of the 17th ACM conference on Information and knowledge management*, pages 739–748, New York, NY, USA, 2008. ACM.
- [21] J. Z. Wang, J. Li, and G. Wiederhold. Simplicity: Semantics-sensitive integrated matching for picture libraries. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23:947–963, 2001.
- [22] L. Zheng and X. He. Classification techniques in pattern recognition. In *Proc. WSCG 2005*, London, United Kingdom, 2005.
- [23] X. S. Zhou and T. S. Huang. Relevance feedback in image retrieval: A comprehensive review. *Multimedia Systems*, 8(6):536–544, April 2003.

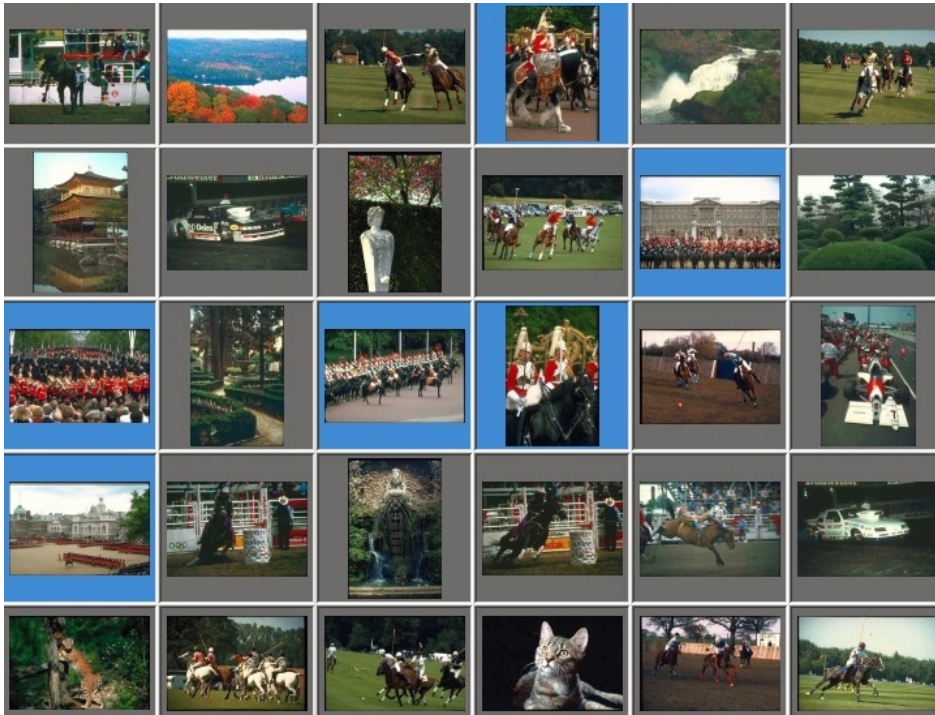


Figure 16: Closest images to s based on $d(s, t)$.



Figure 17: Result of OPF after three iterations.