A Few Steps Towards On-the-Fly Symbol Recognition with Relevance Feedback

Jan Rendek, Bart Lamiroy, Karl Tombre

▶ **To cite this version:**

Jan Rendek, Bart Lamiroy, Karl Tombre. A Few Steps Towards On-the-Fly Symbol Recognition with Relevance Feedback. 7th IAPR Workshop on Document Analysis Systems - DAS 2006, Feb 2006, Nelson/New Zealand, pp.604 - 615, 10.1007/11669487 . inria-00001048

HAL Id: inria-00001048
https://inria.hal.science/inria-00001048

Submitted on 24 Jan 2006

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

A Few Steps Towards On-the-Fly Symbol Recognition with Relevance Feedback*

Jan Rendek[1,2], Bart Lamiroy[1], and Karl Tombre[1]

[1] LORIA–INPL, B.P. 239, 54506 Vandœuvre-lès-Nancy CEDEX, France
[2] France Télécom R&D, Meylan CEDEX, France
Jan.Rendek@loria.fr, Bart.Lamiroy@loria.fr, Karl.Tombre@loria.fr

Abstract. This paper presents some first steps in building an interactive system which allows a user to efficiently browse a large set of scanned documents, without prior knowledge on the content of these documents, and retrieving symbols of interest to him personally, through a relevance feedback mechanism.

1 Introduction

The recognition of graphical symbols has been subject to much effort throughout the years. The methods used include template matching techniques, grammar-based matching techniques, recognition techniques based on structural features or dynamic programming, and a number of stuctural methods based on graph matching.

One of the reasons for which symbol recognition is in many cases a very difficult and ill-defined problem is the large number and variety of symbols to be recognized. Except in strongly context-dependent applications, it may often be impossible to provide a database of all possible symbols. It is also in many cases impossible to assume that symbol recognition can be performed on correctly segmented instances of symbols, as symbols are very often connected to other graphics and/or associated with text. The well-known paradox therefore appears: in order to correctly recognize the symbols, we should be able to segment the input data, but in order to correctly segment them, we need the symbols to be recognized!

The current state of the art makes the following assumption where recognition is concerned: symbols are subject to a number of deformations that need to be taken into account in order to obtain efficient recognition methods. These deformations may have various origins and result in different kinds of visual effects:

1. planar geometric transforms (rotation, translation, scaling) due to general document orientation or lack of a principal reading direction (*e.g.* complex assembly blueprints, annotated drawings, *etc.*),

* This work is partially funded by a CIFRE contract between France Télécom R&D and INPL-LORIA.

2. noise introduced by the physical image production process (speckling, blurs) or by subsequent treatments (rastering, binarization, scaling, numerical instabilities when rotating ...),
3. complex geometric transforms resulting from projecting 3D forms onto 2D images,
4. intra-class object variations when the recognition process is supposed to encompass a certain semantic class of "similar" items.

The classical scenario generally consists in identifying a finite set of *known symbols* in a set of documents. Depending on what type of documents are being considered, the recognition method will integrate the previously mentioned deformations to different degrees in order to produce an as efficient as possible recognition method. The key issue here is that, most of the time, all symbols to be recognized are previously known, and that there exists either a model, or a training set (from which a model can be built) for each symbol. Various efficient techniques have been developed, either using structural pattern description, or statistical pattern recognition techniques [1,2,3].

The problem arises when no model nor training set is available or can even be planned (typically in a very open and general application), or when the training set is too poor to derive a usable model. While it is clear that the first three points of the previously mentioned deformation models are rather general and can reasonably well be taken into account for a large range of situations (exception made for extreme scale changes or very distorted acquisition tools, and considering that 3D recognition usually falls into a separate category), it is the intra-class variability that usually calls for an adequately dimensioned training set or a sufficiently complex model. There are, however, situations where there is no *a priori* knowledge allowing for this variability to be captured. In this paper, we present an attempt in addressing a category of these problems.

2 On-the-fly symbol recognition: Proposed scenario

The scenario we work on is the following: we consider a large set of documents, possibly hand-written or hand-drawn, with very little domain knowledge about the kind of information they embed. A user, not necessarily a specialist, wants to be able to efficiently and interactively navigate in this set of documents, and find information relevant to his needs. These needs are unknown at the time of design of the document analysis system, and can widely vary over time. The scenario simply assumes that the user points out or sketches one or several instances of a symbol of interest to him, asking the system to retrieve all similar items from the set of documents. We insist on the fact that, for the scenario to remain open and generic, no *a priori* knowledge on the symbol, nor on the documents is provided.

Figure 1 presents a typical example from a document set we have used in our first experiments with this concept, and that we will present with more details in § 5. In the present case, we have a set of handwritten notes taken at various meetings, and the user happens to use some symbols such as arrows to indicate tasks to do (another user may use a completely different set of symbols

for the same purpose). We absolutely do not plan to try performing handwriting recognition on such notes but only to be able to easily retrieve in the document set those regions where an occurrence of the symbol appears. With the same scenario, we could easily imagine another user browsing through some technical documentation such as a car repair manual...

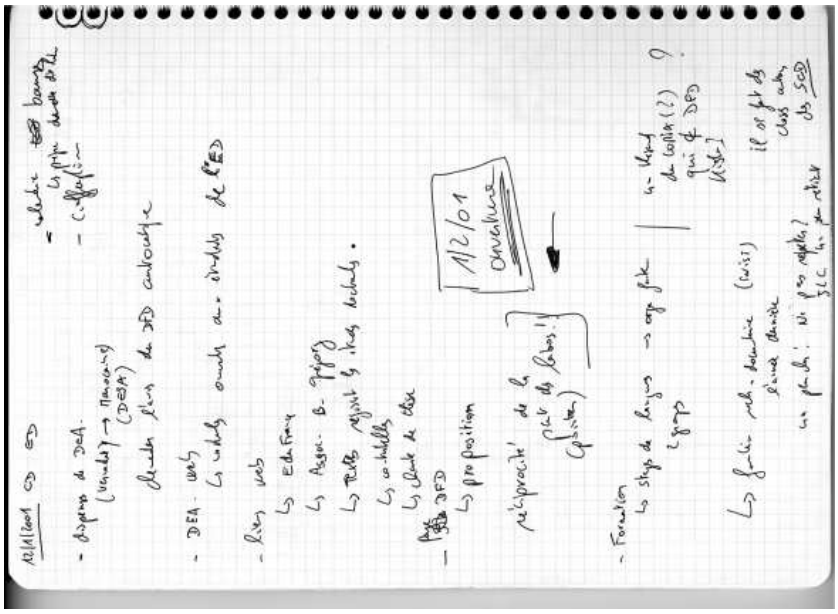

Fig. 1. An example of processed document.

The basic idea of our method is to perform a crude segmentation on the documents (as previously mentioned, it is impossible with this scenario to have an exact, reliable segmentation as we do not know in advance what to look for), to extract some features from the candidate symbol, and to define a similarity measure between these features and similar features computed on the regions provided by the segmentation. Unfortunately, finding a descriptor and a metric that maps the user concept of similarity into a distance measure between low level features is nearly impossible since we have no means of determining what intra-class deformation is acceptable to the user. Indeed, depending on the context of use, the same user, using the same documents, may have different retrieval goals, implicitly requiring different intra-class variability. Our idea is therefore to try our best with the provided data, and to interactively prompt the user for extra knowledge to improve the query results until a satisfactory level of symbol identification is achieved.

The user is by no means considered as a pattern recognition expert. He is only asked whether, according to his judgment, a given pattern matches his initial

query. Such approaches have initially been introduced in Content Based Image Retrieval [4,5]. In this paper, we present an attempt in applying such relevance feedback techniques to symbol recognition.

3 Relevance feedback

Relevance feedback has been an active research subject for the past few years. It has been successfully applied in Content Based Image Retrieval (CBIR) systems [6,7]. Its purpose is to involve the user in an interactive discussion loop, in order to adapt the similarity measure computed between two patterns to the user similarity concepts, based on their low-level representation.

The core scenario of a relevance feedback system can be summarized as follows. For a given query, the system retrieves an initial set of results, ranked according to a predefined similarity metric. The user provides judgment on the current retrieval, as to whether the proposed samples are correct or wrong, and possibly to what degree. The system learns from the feedback, and provides a new set of results which are then submitted to the user approval. The system loops until the user is satisfied with the result set.

Patterns are represented by a vector of measurements performed on them. With an ideal descriptor, all the samples belonging to the same class would form a cluster in the feature space. The pattern most similar to the query would correspond to the nearest neighbors of the query representation in the feature space. An ideal query would be located right at the center of the space.

Early relevance feedback systems were built using heuristic-based techniques derived from document retrieval [8]. The main idea was to estimate an *ideal query point*, from the given positive samples. Re-weighting the feature space or the metric parameters is also used to maximize the correlation between the user similarity concept and the low level image features. Further developments contributed to formalize the problem by optimization techniques to minimize the total distance between the positive samples to the query [9,10]. The principal findings were that the optimal query is obtained by averaging the positive samples, and that the Mahalanobis distance is the optimal weighted metric. MindReader [9] and Mars [11] CBIR systems apply these approaches with success.

Parallel work (though somewhat more recent) considers relevance feedback as a two-class classification problem and try to adapt known classification schemes to take into account the supplementary difficulties resulting from the small number of training samples, and the asymmetry in the data set.

Su [12] adapts a bayesian classifier based on the maximum likelihood, estimating the boundary of the relevant items from the positive samples and assigning penalties to unlabeled samples close to a negative one. Zhang [13] and Onada [14] use techniques based on Support Vector Machine, trying to iteratively determine the best hyperplane separating the positive and the negative samples in the projection space.

Other experiments have been carried out, involving decision trees [15,16], nearest-prototype [17] or Bayesian relevance feedback [18]. According to the experiments reported in the literature, all these adaptive classification based methods outperform the optimization based approaches. It is difficult to determine which is the best, as there is to our knowledge no review reporting on such a comparison.

4 Proposed prototype system

Our system demonstrates the usefulness of relevance feedback applied to on-the-fly symbol recognition. As described previously, we extract a candidate symbol from a set of handwritten documents and then proceed to finding all relevant representations of the same symbol in the set of documents.

This paper focuses on recognition and relevance feedback. It is clear that a number of preprocessing issues should not be underestimated. They are, however, out of the scope of this paper and we will only briefly summarize our choices.

4.1 Preprocessing

In our case, preprocessing involves two steps: document segmentation, and feature extraction.

Segmentation: documents are segmented into rectangular regions, each region potentially embedding a symbol. In the first instance of our prototype, we used a very simple and straightforward segmentation technique based on recursive X-Y tree decomposition [19,20]. While crude and simple, it provides good results on the kind of symbols we aim at in this paper. They consist of manual annotations, most often distinctly separated form the main text. Again, segmentation is not the problem we want to tackle in this experiment. Different application contexts certainly need adapted segmentation approaches. This point will undoubtedly be improved in a near future by implementing more efficient techniques such as connected components analysis [21] or the scale space approach proposed by Manmatha and Rothfeder [22]. Figure 2 shows the kind of segmentation results we currently obtain.

Feature extraction: for each region isolated in the previous step, a feature vector is extracted. In our first experiments, we used Zernike moments as the low level representation of the potential symbols. It is a well-known descriptor, with thoroughly studied performances [23], robust to the different deformation and distortion models cited previously cited , including those induced by hand sketching [24].

It is noteworthy to mention that our system is sufficiently modular to integrate other segmentation methods or different descriptors in a very straightforward manner. Those mentioned here have been implemented in order to prove the validity of our approach and in no way represent our ultimate choices.

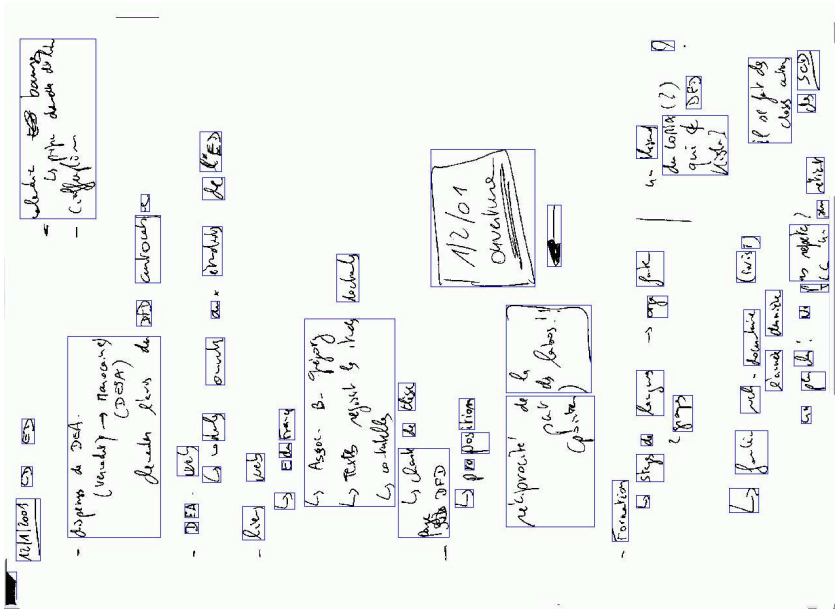

Fig. 2. Example of segmented document.

4.2 Query and relevance feedback

Initial step. The user selects a candidate symbol from the presegmented areas of the document set. From this initial query, the n best matching areas of all documents are retrieved, by computing the Euclidean distance from their representation to the query. A list of candidate areas, ordered in increasing distance from the query is presented to the user. Within this list, the user then selects which samples are relevant and which are irrelevant according to his own perception of similarity between his query and the presented candidates. Figure 3 exhibits a sample query, and the 20 best samples matching the query. This data is taken from the experiment described in section 5.

Feedback step. From the retrieved positive and the negative samples in the initial step, we compute a relevance measure for all the other remaining unlabeled samples.

The relevance estimation for an unlabeled sample is based on a method developed by Giacinto and Rolli [18] using a nearest neighbor rule. The relevance of a sample depends on its minimal distance to both positive and negative samples. The closest it is to a relevant sample, the highest its own relevance. On the contrary, the closest it is to a negative sample, the bigger is the penalty assigned to its relevance. The relevance R of a symbol s is computed as follows: let $\mathcal{N}$ be

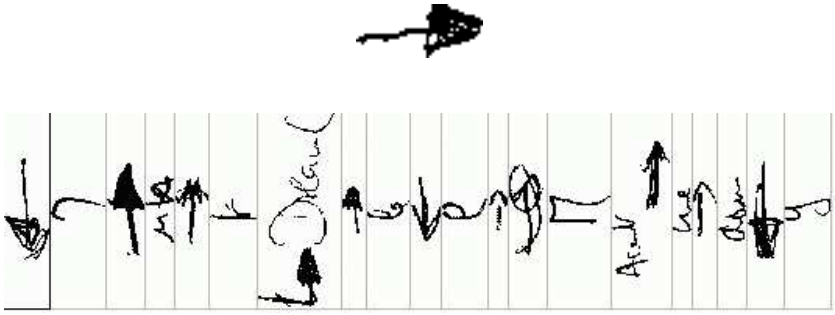

Fig. 3. Initial step: the query and the 20 best matches sorted using the euclidean metric.

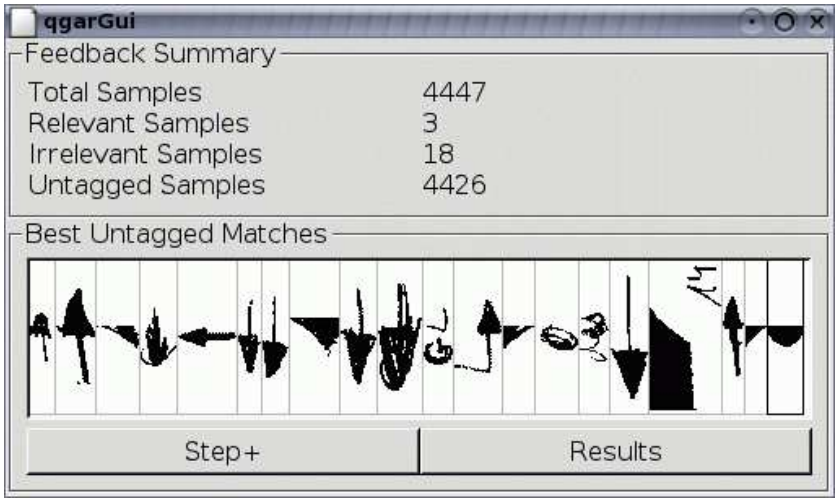

Fig. 4. Unlabeled best matches tagging after one feedback step.

the set of negatively labeled symbols and $\mathcal{P}$ the set of positively labeled symbols:

$$\mathsf{R}(s) = 1 - e^{-\frac{d_{\min}(\mathcal{N},s)}{d_{\min}(\mathcal{R},s)}} \tag{1}$$

where $d_{\min}(\mathcal{X}, x)$ is the minimum distance from a set $\mathcal{X}$ to a symbol x.

Iteration step. The areas are sorted by decreasing relevance and submitted to the user, who can then again mark positive and negative samples, and loop over the feedback step. Figure 4 shows the part of our system dedicated to this task.

The main quality of this method is that it locally estimates the relevance of a sample, and that it does not assume that all positive samples form a cluster in the feature space. The distribution of the relevant samples does not need to be Gaussian or have a boundary that can be modeled by a parametric shape. This presents a major advantage over other classification schemes that need these conditions to function properly. Furthermore, the nearest neighbor classifier (1-NN) is known to always yield correct results, provided it iterates over the entire sample set. Implementing a RF method based on the 1-NN rule is an excellent starting point and benchmark for subsequent evaluation of other methods.

5 First results

5.1 Experiment setup

To assess the performances of our approach, we built a concrete test case. We extracted 82 A4 pages from a notebook, and used our prototype to retrieve the plain arrows which the notebook owner uses to signal something to be reminded. The pages where scanned at 200 dpi, and segmented using the method previously described. This process yields 4447 regions, that where manually labeled in order to establish a ground truth suitable to our tests. Out of these 4447 regions, 52 were labeled as plain arrows. These regions are shown in Figure 5.

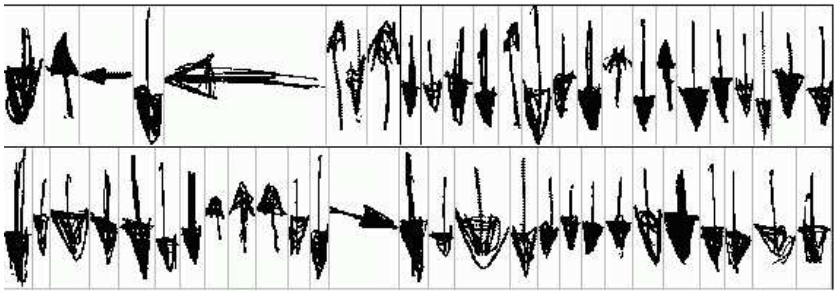

Fig. 5. The 52 positive samples of the corpus.

We use *Precision* and *Recall* to evaluate the performance of the proposed approach. *Precision* is defined as the number of retrieved relevant regions over

the total number retrieved regions. *Recall* is defined as the number of retrieved relevant regions over the total number of relevant regions (52 in our case).

In order to avoid spurious conclusions based on particular experiments, all measurements are taken on the whole set of arrow symbols. Each relevant symbol is used in its turn as initial query for our system, and 4 feedback iterations are performed, as described in section 4.2. We then plot average *Precision* and *Recall* curves computed over all 52 queries.

Furthermore, to asses the impact of user feedback, the above process is repeated with an increasing number of labeled symbols per feedback step. We performed experiments with 5, 10, 15, and 20 user selected regions per iteration. The *Precision–Recall* curves are shown in Figure 6.

5.2 Results and observations

A quick analysis of the results in Figure 6 naively implies that the higher the number of user labeled examples per iteration is, the better the final classification results. This, however, needs to be moderated.

The experiment does reveal that :

- User feedback drastically enhances system convergence towards a high recognition rates. Even with very few user-labeled symbols (*e*.g. 10) *P*recision goes up with 30% for a recall of 50%. In other terms, the rank of the median ranked symbol is 37 with 10 labeled symbols, while it is 68 without. Furthermore, in order to retrieve 80% of the 52 searched symbols (*i.e.* 42 items) within 4447 possible candidates, one needs to label 60 to 80 symbols, according to the number of symbols considered at each iteration (15 or 20, in our case).
- Initial convergence speed (*i*.e. enhancement of global recognition quality in the first stages of the relevance feedback loop) is fairly independent of the number of feedback iterations, but rather depends of the total number of labeled items. Figure 7 shows that global quality is equivalent for 4 iterations over 5 samples, 2 over 10 or 1 over 20, and similarly for 4 over 10 and 2 over 20.

6 Conclusion and future work

We have presented our basic choices in building a framework for quickly browsing a large set of scanned documents without any prior knowledge on their content. This framework allows a user to select a symbol of interest and interactively search for the most relevant symbols, using a relevance feedback mechanism to iteratively reach a satisfactory state. A first noticeable property is that the prototype we have built works and gives us good hope that we are on the right track.

Of course, we are perfectly aware that this represents only the first steps for a full, versatile system able to work on a number of document types. As previously mentioned, we have to provide a choice of segmentation and feature extraction

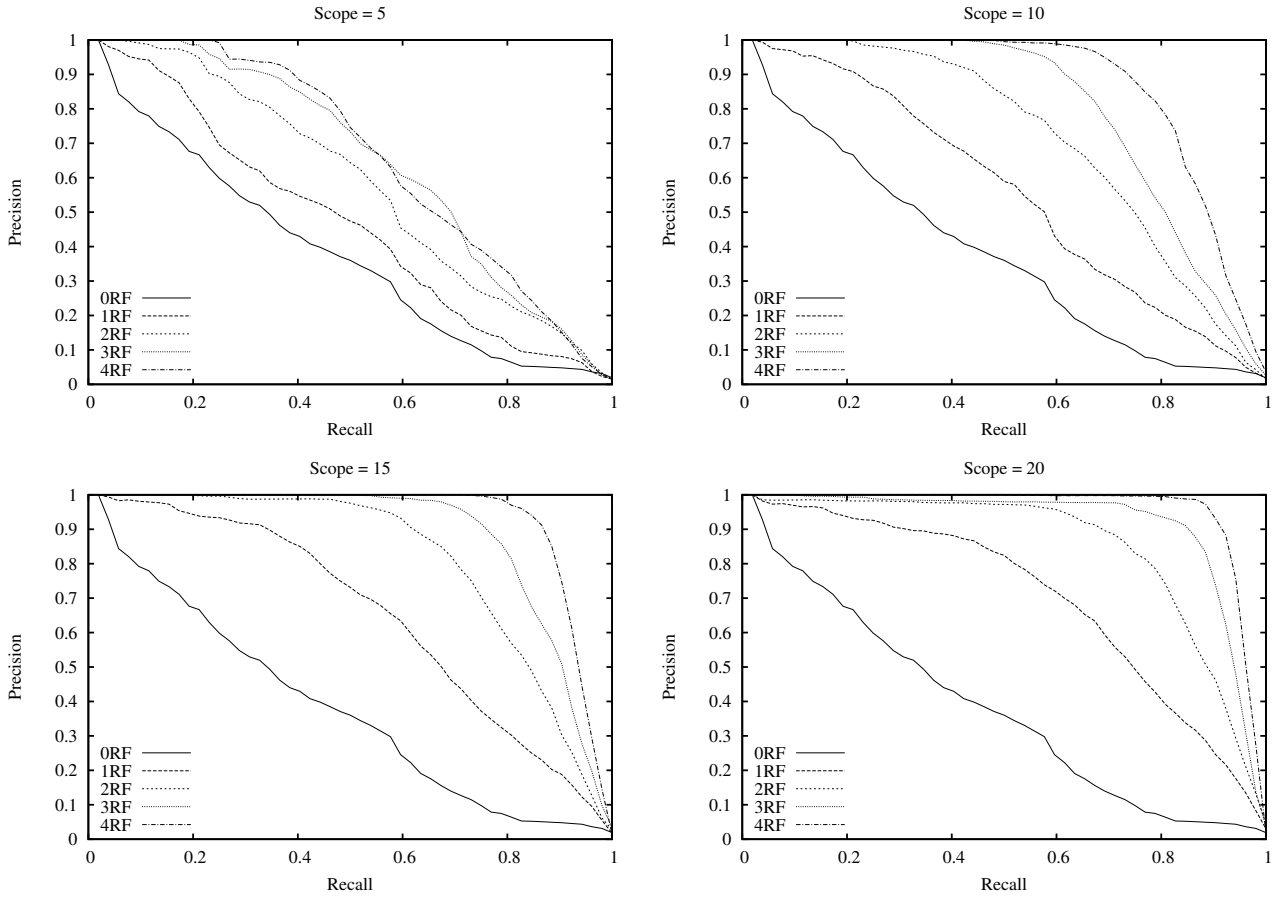

Fig. 6. Precision/Recall curves for 4 feedback steps

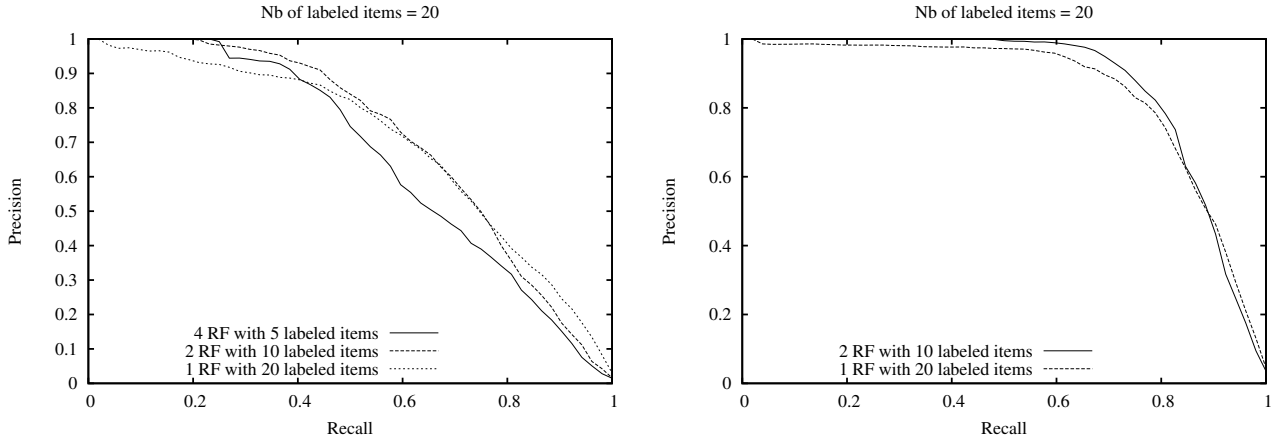

Fig. 7. Precision/Recall curves with identical number of tagged items

methods, able to robustly cope with various user needs. Indeed, Zernike moments, for instance, worked fine for the arrows used in the example presented in this paper, but are not necessarily the best decriptor in another context.

To extend our system towards the handling of bigger data sets, it will be necessary to work on how to efficiently store, index and access precomputed feature sets combined with various segmentations schemes, so as to provide a quick and efficient feedback to the user in searching and browsing mode.

On the classification front, the 1-NN classifier has also its drawbacks and we need to choose the adequate classifiers which allow user interaction and make it easy to decide when to stop, especially when user feedback starts to degrade the convergence of classification.

Finally, with respect to functionalities of such a system, our goal is that the user should be able to design her *personal dictionary* of symbols of interest, storing information of interest and restoring it for reuse from one session to the other.

At the present time, we are focusing on maximizing the knowledge extracted from the user-provided feedback. Instead of sorting the classification output by relevance, it might be a better idea to guide the user very quickly and efficiently to the "border zone" where his feedback will be most relevant for improving the classification in the next iteration. This may necessitate some "intelligent" analysis of the relevance curve. In the same order of idea, it would be useful to be able to find a correlation between the size of the database, the number of symbols to be found and the scope of the search, in order to minimize the number of iterations with the user and to optimize the convergence and the quality of the recall.

References

1. Chhabra, A.K.: Graphic Symbol Recognition: An Overview. In Tombre, K., Chhabra, A.K., eds.: Graphics Recognition—Algorithms and Systems. Volume 1389 of Lecture Notes in Computer Science. Springer-Verlag (1998) 68–79
2. Cordella, L.P., Vento, M.: Symbol recognition in documents: a collection of techniques? International Journal on Document Analysis and Recognition **3** (2000) 73–88
3. Lladós, J., Valveny, E., Sánchez, G., Martí, E.: Symbol Recognition: Current Advances and Perspectives. In Blostein, D., Kwon, Y.B., eds.: Graphics Recognition – Algorithms and Applications. Volume 2390 of Lecture Notes in Computer Science. Springer-Verlag (2002) 104–127
4. Zhang, L., Lin, F., Zhang, B.: Support vector machine learning for image retrieval. In: Proceedings of IEEE International Conference on Image Processing. (2001) 721–724
5. Zhou, X., Huang, T.S.: Relevance feedback in image retrieval: a comprehensive review. Multimedia Systems **8** (2003) 536–544
6. Smeulders, A.W.M., Worring, M., Santini, S., Gupta, A., Jain, R.: Content-Based Image Retrieval at the End of the Early Years. IEEE Transactions on PAMI **22** (2000) 1349–1380

7. Vasconcelos, N., Lippman, A.: Bayesian Relevance Feedback for Content-Based Image Retrieval. In: Proceedings of IEEE Workshop on Content-based Access of Image and Video Libraries. (2000) 63
8. Doermann, D.: The Indexing and Retrieval of Document Images: A Survey. Computer Vision and Image Understanding **70** (1998) 287–298
9. Ishikawa, Y., Subramanya, R., Faloutsos, C.: MindReader: Query databases through multiple examples. In: Very Large Databases. (1998)
10. Rui, Y., Huang, T.: Optimizing Learning in Image Retrieval. In: Computer Vision and Pattern Recognition. (2000) 1236
11. Rui, Y., Huang, T., Mehrotra, S.: Content-Based Image Retrieval with Relevance Feedback in MARS. In: Proceedings of IEEE International Conference on Image Processing. (1997) 815–818
12. Su, Z., Zhang, H., Ma, S.: Using Bayesian Classifier in Relevant Feedback of Image Retrieval. In: 12th IEEE International Conference on Tools with Artificial Intelligence (ICTAI'00). (2000)
13. Zhang, H.J., Chen, Z., Liu, W.Y., Li, M.: Relevance feedback in content-based image search. World Wide Web **2** (2003) 131–155
14. Onada, T., Murata, M., Yamada, S.: Relevance feedback document retrieval using support vector machines. In: Proceedings of International Joint Conference on Neural Networks (IJCNN-2003). (2003) 1757–1762
15. MacArthur, S.D., Brodley, C.E., Shyu, C.: Relevance Feedback Decision Trees in Content-Based Image Retrieval. In: IEEE Workshop on Content-based Access of Image and Video Libraries. (2000) 68
16. Wang, T., Rui, Y., Hu, S., Sun, J.: Adaptive Tree Similarity for Image Retrieval. Multimedia Systems **9** (2003) 131–143
17. Giacinto, G., Roli, F.: Nearest-prototype relevance feedback for content based image retrieval. In: Proceedings of the 17th International Conference on Pattern Recognition, Cambridge (UK). (2004) 989–992
18. Giacinto, G., Roli, F.: Bayesian relevance feedback for content-based image retrieval. Pattern Recognition **37** (2004) 1499–1508
19. Nagy, G., Seth, S.: Hierarchical Representation of Optically Scanned Documents. In: Proceedings of 7th International Conference on Pattern Recognition, Montréal (Canada). (1984) 347–349
20. Appiani, E., Cesarini, F., Colla, A.M., Diligenti, M., Gori, M., Marinai, S., Soda, G.: Automatic document classification and indexing in high-volume applications. International Journal on Document Analysis and Recognition **4** (2001) 69–83
21. Tombre, K., Tabbone, S., Pélissier, L., Lamiroy, B., Dosch, P.: Text/graphics separation revisited. In Lopresti, D., Hu, J., Kashi, R., eds.: Proceedings of the 5th IAPR International Workshop on Document Analysis Systems, Princeton, NJ (USA). Volume 2423 of Lecture Notes in Computer Science., Springer-Verlag (2002) 200–211
22. Manmatha, R., Rothfeder, J.L.: A Scale Space Approach for Automatically Segmenting Words from Historical Handwritten Documents. IEEE Transactions on PAMI **27** (2005) 1212–1225
23. Liao, S.X., Pawlak, M.: On the Accuracy of Zernike Moments for Image Analysis. IEEE Transactions on PAMI **20** (1998) 1358–1364
24. Hse, H., Newton, A.R.: Sketched Symbol Recognition using Zernike Moments. In: Proceedings of the 17th International Conference on Pattern Recognition, Cambridge (UK). (2004)