7–11 June 1999, Florence, Italy.

Relevance Feedback and Category Search in Image Databases

Christophe Meilhac and Chahab Nastar
INRIA BP 105 F-78153 Le Chesnay, France
Chahab.Nastar@inria.fr

Abstract

We present a sound framework for relevance feedback in content-based image retrieval. The modeling is based on non-parametric density estimation of relevant and non-relevant items and Bayesian inference. This theory has been successfully applied to benchmark image databases, quantitatively demonstrating its performance for target search, selective control of precision and recall in category search, and improvement of retrieval effectiveness. The paper is illustrated with several experiments and retrieval results on real-world data.

1 Introduction

Although the field of information retrieval has developed powerful relevance feedback techniques for document indexing for more than thirty years (e.g. the vector space model [12]), there have been few studies on improving user interactivity for retrieving images by content.

For instance, the pioneer QBIC system offers little interactivity: the user has to select image features such as color, shape, or texture [4]. Other systems require that the user provides a non-intuitive weighted combination.

The goal of relevance feedback is learning from user interaction. In image retrieval, this has first been studied by Picard and Minka [9, 5]: their system forms disjunctions between initial image feature groupings according to positive and negative examples of the user. Rui et al proposed a straightforward adaptation of the vector space model for images [11]. Other attempts have been made to automatically select the 'best' similarity metric based on user feedback [10, 13, 1].

Recently, a more structured view of content-based search was presented by increasing complexity [3]:

- **Target search**. Target search is about users trying to find specific target images in a database [2].
- **Category search**. This task concerns the search of one or more images from a category; subjective semantics of a category are responsible for the complexity of this problem.
- **Open-ended browsing**. Browsing is certainly the most useful and the most complex problem, since it includes all aspects of visual information management, in particular the retrieval problem.

In our previous work [7], we have addressed the category search problem by a forward model of density estimation. The main idea of [7] is to integrate both the positive (relevant) and the negative (non-relevant) examples of the user in a common parametric density estimation technique. Assuming independence, the estimation is performed over each feature component. The estimated density should be representative of as many relevant and as few non-relevant items as possible. A dedicated error count is minimized and ensures robustness to outliers.

In this paper, we view the category search in a sound theoretical framework. We use non-parametric density estimation and Bayesian inference for structuring all items in the database as relevant or non-relevant to the query (section 2). This allows for deciding the relevance of items in the database with respect to user interaction, and derive user strategies. Quantitative results on benchmark databases (section 3) prove the power of the method both for category and target search. Qualitative examples are also presented. We draw the conclusions in section 4.

2 A theory of relevance feedback

Let $X^i = (x^i_s)_{s\in[1,p]} \in \mathbb{R}^p$ the vector representation of the i-th image in the database. Note that in practice, X^i integrates several images signatures, spanning a large feature space that should capture various aspects of image content [7].

At each step, the user labels the presented images as relevant (R) or non-relevant (N) to their query. The system has then to estimate user intention of database categorization, i.e. for each image of the database, compute its relevance or non-relevance to the query.

We denote by $\mathcal{R}_l$ the set of images that the user has labeled as 'relevant' and by $\mathcal{N}_l$ the set of the images

that the user has labeled 'non-relevant'. $\mathcal{R}$ and $\mathcal{N}$ will be the set of 'true' images that the user would label if they labeled all the items in the database ($\mathcal{R} + \mathcal{N} = \mathcal{DTB}$).

For simplicity, we make two assumptions:

- the database may be partitioned by the user into relevant and non-relevant items.
- the feature components $(x_s^i)_{s\in[1,p]}$ are independent.

The first assumption is very weak: it simply means that the user is performing a category search. Database items are either acceptable or rejectable w.r.t to their model of perceptual similarity. The goal of the system is to guess the right partition of the database which reflects user's intention. The second assumption is stronger, since feature components are generally not independent. Still, their independence is a common assumption in particular in information retrieval theory [12]. Overall, these assumptions are very reasonable, given the constraint: we have to keep in mind that we want to keep the computations *real-time*. The assumptions enable us to achieve sophisticated modeling and sound analytic computations, as we will see hereafter.

2.1 Non-parametric density estimation

Our first goal is to estimate the densities $p(X|R)$ and $p(X|N)$ (feature vector X given its relevance or non-relevance to the query), given a sample of trials $\mathcal{R}_l$ and $\mathcal{N}_l$ provided by the user.

If we worked with a parametric method (e.g. Gaussian densities), we would simply have $p(X|R) \equiv N(\mu^R, \sigma^R)$, with $\mu^R = (\mu_s^R)_{s\in[1,p]}$ and $\sigma^R = (\sigma_s^R)_{s\in[1,p]}$, and a similar expression for $p(X|N)$. We would then have to estimate the parameter vector $\theta = (\mu^R, \sigma^R, \mu^N, \sigma^N)$. This task is classically performed by maximum likelihood.

But parametric densities (like the Gaussian) are limited, since they assume an a priori form of the density. In contrast, Parzen window estimation does not assume the form of the density to be known in advance. It rather uses the samples of the random variable for modeling its density. For the sake of simplicity, we use the Gaussian as the smoothing function of the Parzen estimator. Let:

$$g_\sigma(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{x^2}{2\sigma^2}} \tag{1}$$

The Parzen density $p(X|R)$ on each feature component s is then:

$$G_{\theta_s^{\mathcal{R}_l}}(x) = \sum_{r\in\mathcal{R}_l} g_{\sigma_s(r)}(x - \mu_s(r)) \tag{2}$$

where $\theta_s^{\mathcal{R}_l} = (\mu_s(r), \sigma_s(r))_{r\in\mathcal{R}_l}$ is the set of parameters of the relevant images for feature component s.

Similarly, the density $p(X|N)$ on each feature component s is:

$$G_{\theta_s^{\mathcal{N}_l}}(x) = \sum_{n\in\mathcal{N}_l} g_{\sigma_s(n)}(x - \mu_s(n)) \tag{3}$$

with $\theta_s^{\mathcal{N}_l} = (\mu_s(n), \sigma_s(n))_{n\in\mathcal{N}_l}$.

Assuming independence of the feature components, we have:

$$p(X|R) = \prod_{s=1}^{p} G_{\theta_s^{\mathcal{R}_l}}(x) \tag{4}$$

$$p(X|N) = \prod_{s=1}^{p} G_{\theta_s^{\mathcal{N}_l}}(x) \tag{5}$$

The parameter vector of the whole model over all feature components is: $\theta = (\theta_s^{\mathcal{R}_l}, \theta_s^{\mathcal{N}_l})_{s\in[1,p]}$.

2.2 Bayesian Inference

We now want to compute the probability of relevance versus probability of non-relevance for the items in the database. From Bayes decision rule, we can derive that the scalar:

$$J_\theta(X) = -\log(p(R|X)) + \log(p(N|X)) \tag{6}$$

allows us to decide if an image X in the database is relevant ("small" J_θ) or non-relevant ("large" J_θ). Thus J_θ is able to partition the database into relevant and non-relevant items.

Unfortunately J_θ cannot be directly computed. Applying Bayes rule:

$$p(R|X) = \frac{p(X|R)\,p(R)}{p(X)} \tag{7}$$

$$p(N|X) = \frac{p(X|N)\,p(N)}{p(X)} \tag{8}$$

and assuming that the priors $p(R)$, $p(N)$ and $p(X)$ are unknown but constant, we observe that the decision rule may also be derived by computing:

$$I_\theta(X) = -\log(p(X|R)) + \log(p(X|N)) \tag{9}$$

If we had used unimodal Gaussian densities, and assuming independence of feature components, I_θ would have had a simple expression:

$$I_\theta(X) = \frac{1}{2}\sum_{s=0}^{s=p}((\frac{x_s - \mu_s^R}{\sigma_s^R})^2 - (\frac{x_s - \mu_s^N}{\sigma_s^N})^2) \tag{10}$$

In case of Parzen window estimation, and again assuming independence, we have:

$$I_\theta(X) = -\sum_{s=0}^{s=p}(\log(G_{\theta_s^{\mathcal{R}_l}}(x_s)) + \log(G_{\theta_s^{\mathcal{N}_l}}(x_s))) \quad (11)$$

The computation of I_θ allows for partitioning items in the database as relevant ("small" I_θ) or non-relevant ("large" I_θ). Note that $\sigma_s(r)$ may be interpreted as a feature component weight that is automatically adjusted; it is also a measure of the discriminance of the feature component.

I_θ may also be seen as an axis of projection separating the two classes. This Bayesian (MAP) approach is in fact a generalized nonlinear extension of Linear Discriminant Analysis.

2.3 Parameter estimation

We note again that we have the real-time constraint (the user is waiting for the system to display images!). Therefore we cannot use a time-consuming optimization scheme for estimating the parameters.

With the Parzen density, one fast and effective way of estimating the parameters is to set:

$$\forall r \in \mathcal{R}_l \quad \mu_s(r) = x_s^r \quad (12)$$
$$\forall n \in \mathcal{N}_l \quad \mu_s(n) = x_s^n. \quad (13)$$

where $X^r = (x_s^r)_{s\in[1,p]}$ is an image labeled relevant by the user, and $X^n = (x_s^n)_{s\in[1,p]}$ is an image labeled non-relevant by the user. This means that every labeled image is the center of a Gaussian, and as an effect, each labeled image has an "influence zone" (neighboring images in feature space are probably in the same class).

For the standard deviation, we set:

$$\forall r \in \mathcal{R}_l \quad \sigma_s(r) = \frac{\sigma_s^{\mathcal{DTB}}}{\log(\#\mathcal{R}_l)} \quad (14)$$
$$\forall n \in \mathcal{N}_l \quad \sigma_s(n) = \frac{\sigma_s^{\mathcal{DTB}}}{\log(\#\mathcal{N}_l)} \quad (15)$$

where $\sigma_s^{\mathcal{DTB}}$ is the standard deviation of feature component s over the database. Having the standard deviation inversely proportional to the log of the number of labeled images in a class means that over the user interactions (labeling), the individual Gaussians of the Parzen estimator become narrower, whereas labeling a single image is not significant. Figure 1 shows a sketchy example with our estimation of the parameters.

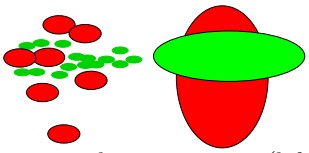

Figure 1: Parzen window estimation (left) versus Gaussian estimation (right) with the described estimation of the parameters.

2.4 Item relevance estimation

We now have all the tools for deciding the relevance or non-relevance of an image in the database through user interactions. Our decision rule is based on I_θ. The goal now is to find a threshold m which separates relevant and non-relevant items on the I_θ axis:

$$\text{find } m \text{ such that: } ((I_{\hat{\theta}}(X) < m)?(X \in \mathcal{R}) : (X \in \mathcal{N})) \quad (16)$$

where $\hat{\theta}$ is the estimator of the true parameters θ computed as in section 2.3. Let:

$$\begin{aligned}\mathcal{R}_{\hat{\theta}}(m) &= \{X \in \mathcal{DTB} \text{ such that } I_{\hat{\theta}}(X) < m\} \\ \mathcal{N}_{\hat{\theta}}(m) &= \{X \in \mathcal{DTB} \text{ such that } I_{\hat{\theta}}(X) > m\}\end{aligned} \quad (17)$$

Since we do not know if there exists such an m number, we compute:

$$\begin{aligned}m_1 &= \max(I_{\hat{\theta}}(X^r), r \in \mathcal{R}_l) \\ m_2 &= \min(I_{\hat{\theta}}(X^r), r \in \mathcal{N}_l)\end{aligned} \quad (18)$$

Figure 2 shows a graphic representation of these numbers.

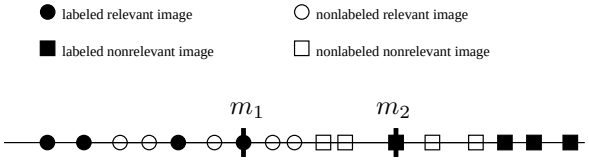

Figure 2: Definition of m_1 and m_2 from user-labeled items on the I_θ axis.

But does figure 2 hold in practice? By construction $\mathcal{R}_l \subset \mathcal{R}_{\hat{\theta}}(m_1)$. If the number m exists and if $\hat{\theta}$ is a good estimator of θ, then it is likely that $\mathcal{R}_{\hat{\theta}}(m_1) \subset \mathcal{R}$, and we have: $m_1 \leq m \leq m_2$. If $m_1 > m_2$, we will think that $\hat{\theta}$ does not correctly estimate θ, or that the modeling cannot discriminate between relevant and non-relevant items.

We now have a hysteresis threshold. The estimator $\hat{m} = m_1$ is most likely to minimize false matches (thus maximizing precision) while the estimator $\hat{m} = m_2$ will minimize misses (thus maximizing recall). These allows us to selectively control precision or recall while retrieving images (section 3.2).

In practice, the uncertain range is $\{X \in \mathcal{DTB}$ such that $m_1 < I_{\hat{\theta}}(X) < m_2\}$ but we note that over the iterations no item will be left in this range.

2.5 User strategies

Following the described framework, the system is now able to selectively retrieve several types of images. These are several types of strategies that are offered to the user.

2.5.1 Searching for most probable images

Mathematically, retrieving the most probable images means retrieving the set of images with smallest $I_{\hat{\theta}}(X)$ in increasing order (the most relevant image will minimize $I_{\hat{\theta}}(X)$).

This will most closely satisfy the user objectives, in other words, many images similar to the user model of perceptual similarity will be retrieved. The number of false matches are minimized with this strategy (good recall).

2.5.2 Disambiguation

Having the user feedback on 'sensitive' images will disambiguate the system modeling, allowing to clarify user's intention for the system. We have two alternatives:

1. Retrieve images with smallest $|I_{\hat{\theta}}(X) - m_1|$, where $I_{\hat{\theta}}(X) < m_2$ and $X \in \mathcal{DTB}$. This will allow to clarify false matches and minimize them after user interaction, eventually improving precision.
2. Retrieve images with smallest $|I_{\hat{\theta}}(X) - m_2|$, where $m_1 < I_{\hat{\theta}}(X)$ and $X \in \mathcal{DTB}$. In contrast, this is likely to first retrieve misses and then minimize them after user interaction, thus improving recall.

2.5.3 Best strategy

There is not a single best strategy that stands out, as the two above strategies are in essence cooperative. While the search for most probable image is a sophisticated version of 'find me more' that will minimize the risk of retrieving false matches, the disambiguation, on the other hand, will retrieve *on purpose* false matches and misses ('uncertain images') in order to infer a better prediction of user's intention. Therefore, the mixed strategy (alternating most probable search and disambiguation) is likely to be an efficient user action.

3 Experimental results

3.1 Target search

Our methodology is primarily aimed at classification by user interaction, and this task includes the 'target search' (see section 1) that we are experimenting in this section.

For retrieving a target image as quickly as possible, the natural strategy is 'the most probable' over the iterations. We compute the average number of iterations for retrieving the target image .

A classic quantitative evaluation is having a database of varying size and indicating the average number of iterations as a function of database size. Unlike [2] where the results are performed on simulated data, we use a real world benchmark database: the Columbia database of 1440 images consisting of 72 views of 20 objects, as used in [6] and many other studies.

We assume that the user wishes to retrieve all images of a single object. For deriving databases of varying size, we suppress images from the original Columbia database. We eliminate 1 image par class, yielding a new, smaller database on which we perform the target search. This reduction is performed recursively to provide several databases and infer the curve on figure 3. Note that the averaging is over all of the images in the database, which have each being assumed as the target image once. The figure demonstrates that our strategy for target search is 10 much better than chance.

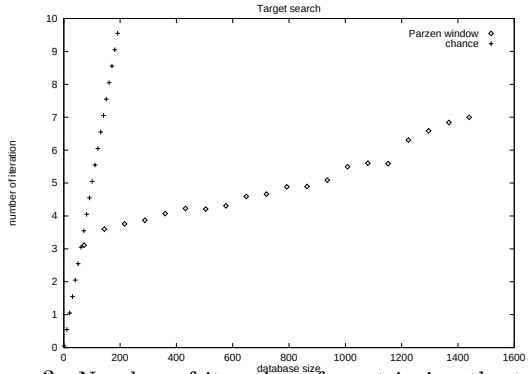

Figure 3: Number of iterations for retrieving the target image versus size of the Database. Results are from the Columbia database and prove that the method is 10 much better than chance.

Ideally, the target image should be part of a category which is reasonably represented (not a too rare category, and not a too dominant category). This can be clarified by considering two limit cases. Suppose that the whole database forms a single category. Consider now the case where the target image is a category by itself. In these extreme (and unrealistic) cases, our strategy is not better than chance, but in general, it is.

3.2 Control of precision and recall

Still considering the Columbia database, we first show quantitatively that precision and recall improve through user interactions. Remember that precision is the fraction of the retrieved images that are relevant, recall being the fraction of the relevant images that

are retrieved.

We consider several item relevance estimators for computing precision and recall (see section 2.4).

- $m = m_1$ maximizes precision, since it minimizes false matches. On the other hand, it minimizes recall, since many relevant items are missed.
- $m = m_2$ operates the other way round.

This is demonstrated experimentally on figure 4 on the Columbia database. From this figure we observe that:

- Both precision and recall improve over user interactions, and
- $m = m_1$ is the best estimator for maximizing precision, while $m = m_2$ is the best estimator for maximizing recall.

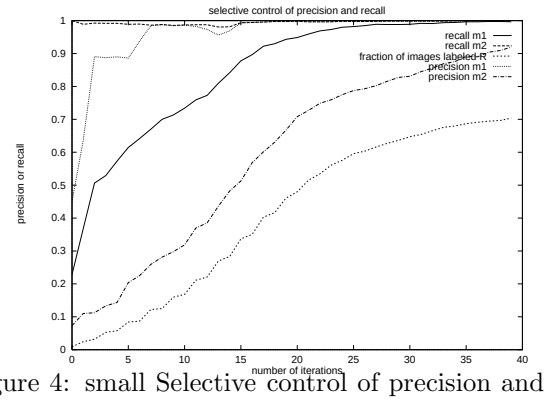

Figure 4: small Selective control of precision and recall. Note that both measures improve over user interactions.
s

3.3 Retrieval effectiveness

For computing the retrieval effectiveness (precision vs recall), we have to re-define these notions by taking into account the evolving notion of 'retrieved images'.

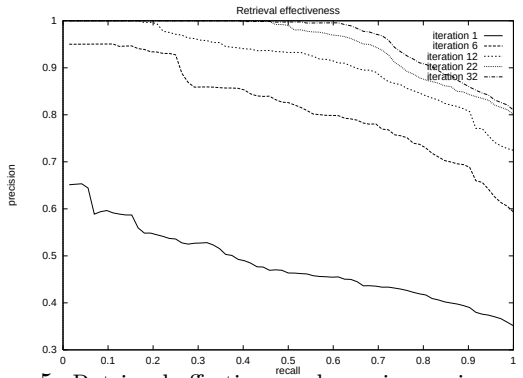

Figure 5: Retrieval effectiveness keeps improving over iterations. Results are from the Columbia database using the mixed user strategy.

Remembering that the retrieved images are defined by $\mathcal{R}_{\hat{\theta}}(m)$, the new definitions are naturally:

$$\texttt{precision}(m) = \frac{\#\mathcal{R}_{\hat{\theta}}(m) \cap \#\mathcal{R}}{\#\mathcal{R}_{\hat{\theta}}(m)} \quad (19)$$

$$\texttt{recall}(m) = \frac{\#\mathcal{R}_{\hat{\theta}}(m) \cap \#\mathcal{R}}{\#\mathcal{R}} \quad (20)$$

Computing $m(r), \forall r \in \mathcal{R}$ provides the retrieval effectiveness over user interactions. This is reported in figure 5. for the Columbia database. The mixed strategy is used (alternate most probable search and disambiguation). We observe that the retrieval effectiveness improves increasingly over user interactions.

3.4 Qualitative results

Note that all of our experimental results are derived from real-world databases indexed by classical image features (various color, texture, shape descriptors).

Figure 6: Correct classification of city scenes (the two images on top left have a green tag) versus 'rest of the database' with relevance feedback. Note that the 14 other displayed images are classified as non-relevant (red tags)

We illustrate the method on a classification example in a homebrew database of 3670 images with varied content. The database was built by merging the MIT *Vistex* database of textures, the *BTphoto* database of city and country scenes [14], a homebrew *paintings* database, and the *homeface* database of people in the lab. The user is looking for city scenes. The total number of city scenes in the database is about 50. The user has labeled about 10 city scenes (relevant images) and 10 non-relevant images. The system has then retrieved the images presented in figure 6. Note that on this figure the system has retrieved 2 city scenes and has classified them correctly (green tag on top left of the image) while the remaining images of the page are estimated to be non-relevant (red tag on top left of the image). Another example is the problem of multiple queries which is completely solved with

Figure 7: Multiple queries based on relevance feedback. Note that the two relevant objects (vase and cat) are obviously 'far apart' in feature space. This is no problem for our non-parametric density estimation.

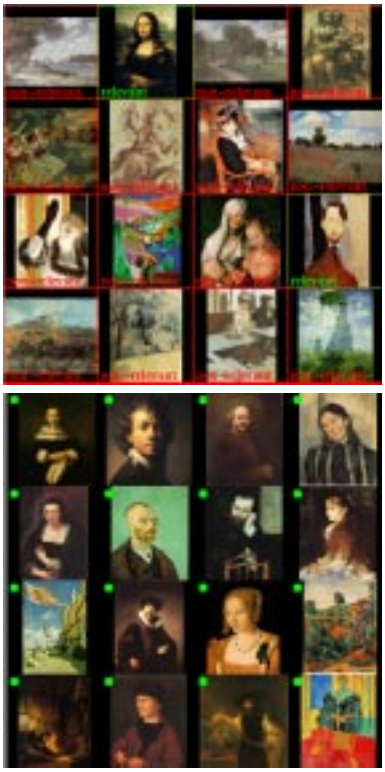

Figure 8: Subjective queries by our relevance feedback method: finding portraits in a generic painting database. Only a few false matches are observed, which are natural due to the complexity of the retrieval task.

our approach. Our non-parametric density estimation allows the two queries to be completely different in terms of content and feature space representation. An example is shown on figure 7 where the user is looking for vases and cats. A random shuffle through the database presents images with their tags.

Searching for all portraits in a painting database is a very difficult problem. Since it is a subjective task, there is no specific feature to describe portrait. Our feedback method allows to respond to such subjective queries, as shown on figure 8. Note that the database of size 3670 contains about 500 paintings among which only 50 are portraits.

4 Conclusion

We have presented a sound framework for relevance feedback. This framework is actually part of the `Surfimage` system [8] and has been tested on dozens of real-world databases by various users. The system is user-friendly and the user needs no expertise in image analysis. Relevance feedback is based on multimodal non-parametric estimation and Bayesian inference, allowing to derive the relevance of database items w.r.t. user perception of image similarity.

In practice, the method has been quantitatively shown to be performant on benchmark databases, in terms of target search, selective optimization of precision or recall in category search, and retrieval effectiveness. The method also allows for deriving heterogeneous image categories (e.g. sunset, faces and cars may be all be in the same relevant class, if that is the intention of the user).

References

[1] B. Bhanu, J. Peng, and S. Qing. Learning feature relevance and similarity metrics in image databases. In *IEEE workshop on Image and Video Libraries*, Santa Barbara, June 1998.

[2] I. Cox, M. Miller, T. Minka, and P. Yianilos. An optimized interaction strategy for bayesian relevance feedback. In *Computer Vision and Pattern Recognition (CVPR '98)*, Santa Barbara, June 1998.

[3] I. Cox, M. Miller, S. Omohundro, and P. Yianilos. Target testing and the pichunter bayesian multimedia retrieval system. In *Proceedings of Advanced Digital Libraries ADL'96 Forum*, Washington D.C., May 1996.

[4] M. Flickner et al. Query by image and video content: the qbic system. *IEEE Computer*, 28(9), 1995.

[5] T. Minka and R. Picard. Interactive learning using a society of models. *Pattern Recognition*, 30(4), 1997.

[6] H. Murase and S. Nayar. Visual learning and recognition of 3D objects from appearance. *Int. Journal of Computer Vision*, 14(1), 1995.

[7] C. Nastar, M. Mitschke, and C. Meilhac. Efficient query refinement for image retrieval. In *Computer Vision and Pattern Recognition (CVPR '98)*, Santa Barbara, June 1998.

[8] C. Nastar, M. Mitschke, C. Meilhac, and N. Boujemaa. Surfimage: a flexible content-based image retrieval system. In *ACM Multimedia'98*, Bristol, September 1998.

[9] R. Picard, T. Minka, and M. Szummer. Modeling subjectivity in image libraries. In *IEEE Int. Conf. on Image Proc.*, Lausanne, September 1996.

[10] Y. Rui, T. Huang, S. Mehrotra, and M. Ortega. Automatic matching tool selection using relevance feedback in MARS. In *Int. Conf. on Visual Inf. Systems*, San Diego, December 1997.

[11] Y. Rui, T. Huang, S. Mehrotra, and M. Ortega. A relevance feedback architecture for content-based multimedia information systems. In *Workshop on Content Based Access of Image and Video Libraries*, Porto Rico, June 1997.

[12] G. Salton. *Automatic Information Organization and Retrieval*. McGraw-Hill, New York, 1968.

[13] S. Sclaroff, L. Taycher, and M. La Cascia. Imagerover: a content-based image browser for the world wide web. In *Workshop on Content Based Access of Image and Video Libraries*, Porto Rico, June 1997.

[14] M. Szummer and R. W. Picard. Indoor-outdoor image classification. In *IEEE Int'l workshop on content-based access of Image and Video Databases*, January 1998.