Adaptive and Efficient Image Retrieval with One-Class Support Vector Machines for Inter-Query Learning

IKER GONDRA and DOUGLAS R. HEISTERKAMP
Department of Computer Science
Oklahoma State University
Stillwater, Oklahoma 74078
U.S.A
{gondra, doug}@cs.okstate.edu http://www.cs.okstate.edu/~{gondra, doug}

Abstract: **-** We present an extension of previous work on improving the initial image retrieval set by exploiting both intra and inter-query learning. In most Content-Based Image Retrieval (CBIR) systems based on Relevance Feedback (RF), all prior experience is lost whenever a user generates a new query, thus inter-query information is not used. In previous work, a system was developed that learns One-class Support Vector Machines (1SVMs) from retrieval experience to represent the set memberships of users' query concepts. By doing a fuzzy classification of a query into the regions of support represented by the 1SVMs, past experience is merged with current intra-query learning. To satisfy a query, intra and inter-query knowledge are combined based on a fixed ratio. Experimental results confirmed that higher precision is obtained by using both current and historical information. However, a fixed ratio for combining intra and inter-query knowledge that is independent of past experience results in large performance drops at some data level. In this paper we extend this previous approach by incorporating Reinforcement Learning (RL) for adaptively changing the ratio of intra to inter-query knowledge that is used to satisfy a query. We also use M-trees as a flexible indexing structure for the efficient search of historical information and database images. Experimental results against a real data set show the proposed approach greatly reduces the large drops in precision that were observed in the original system.

Key-Words: **-** Content-Based Image Retrieval, Relevance Feedback, Support Vector Machine, Inter-Query Learning, Reinforcement Learning

1 Introduction

A Content-Based Image Retrieval (CBIR) system determines closeness between images by computing a distance between their feature vectors. The main problem encountered by CBIR systems is the gap between the high-level concept that the user is looking for and the low-level features extracted from the images [14]. Through user interaction during the search process, Relevance Feedback (RF) helps to alleviate this problem. Several approaches for improving the performance of RF have been proposed [13, 14]. In [13], a Probabilistic Feature Relevance Learning (PFRL) method that automatically captures feature relevance based on user's feedback is presented. It computes flexible retrieval metrics for producing neighborhoods that are elongated along less relevant feature dimensions and constricted along most influential ones. This technique has shown promise in a number of image database applications.

Recently, Support Vector Machines (SVM) [2] have been applied to CBIR systems with RF to significantly improve retrieval performance [3, 4, 9, 15, 17]. We can distinguish two different types of information provided by RF. The short-term learning obtained within a single query session is *intra-query* information. The long-term learning accumulated over the course of many query sessions is *inter-query* information. By accumulating knowledge from users, inter-query learning aims at enhancing future retrieval performance. Thus, both short and long-term learning are useful in CBIR. However, in most current systems, all prior experience from past queries is lost. That is, the system only takes into account the current query session without using any long-term learning.

A few approaches [6, 8, 11, 12, 18] perform inter-query learning (i.e., RF from past queries are used to improve the retrieval performance of the current query). In [12], the log files of the *Viper* system are used to perform feature relevance weighting. Both [11] and [18] perform a complete memorization of prior history and the correlation between past image labeling is merged with low-level features to rank images for retrieval. The model estimates the semantic correlation between two images based on their co-occurrence frequency (i.e., the number of query sessions in which both images were labeled relevant). Intuitively, the larger the co-occurrence frequency of two images is, the more likely that they are semantically similar. In [8] Latent Semantic Analysis (LSI) is used to provide a generalization of past experience. LSI is an important technique in information retrieval. It uses the context (document) of a word usage to uncover its hidden (i.e., latent) semantics. LSI creates a semantic space by performing a singular value decomposition on a term-by-document matrix. In [8], the images in a database are viewed as the fundamental vocabulary of the system. The RF from each query is considered as a document composed of many terms (images). Thus, assuming that the terms of a document have a latent semantic relationship, it is possible to use LSI to capture inter-query learning. In [18] the extra inter-query information is efficiently encoded as virtual features. The initial results from those approaches for inter-query learning show an enormous benefit in the initial and first retrieval iterations. Therefore, inter-query learning has a great potential for decreasing the amount of user feedback by reducing the number of interactions needed to satisfy a query.

This paper is an extension of previous work on improving the initial image retrieval set by exploiting both intra and inter-query learning. In [6], a system is presented that uses One-class Support Vector Machines (1SVM) to represent the set memberships of users' query concepts. By doing a fuzzy classification of a query into the regions of support represented by the 1SVMs, past experience is merged with current intra-query learning. To process a query, intra and inter-query knowledge are combined based on a fixed ratio. We extend this approach by incorporating an on-line Reinforcement Learning (RL) rule for the automatic adaptation of the ratio of intra and inter-query knowledge to be used based on the amount of accumulated historical information in a local region. M-trees [5] are also incorporated for the efficient search of both historical information and database images.

The rest of this paper is organized as follows. Section 2 gives a brief overview of the original image retrieval system with 1SVMs for representing regions of support presented in [6]. In Section 3, we give a brief description of the M-tree [5] data structure, which is used to perform an efficient search of both historical information and images in the database. Section 4 gives a brief introduction to RL and a description of the proposed learning rule for the automatic adaptation of the intra and inter-query knowledge ratio. In Section 5, we present the results of experiments conducted with the extended approach, which show significantly improved performance compared to the original system. Concluding remarks are given in Section 6.

2 Previous Work

The image retrieval system presented in [6] uses 1SVMs to model set membership knowledge about users' query concepts. The images marked as relevant by the user during a RF iteration are used as training data for a 1SVM. A 1SVM maps the relevant images into a nonlinearly transformed kernel-induced feature space and performs risk minimization by attempting to include most of those images into a hyper-sphere of minimum size. The use of kernels allows the 1SVM to deal with the non-linearity of the distribution of training images in an efficient manner, while at the same time, providing good generalization. The geometric view of a 1SVM allows a straightforward interpretation of the density of past interaction in a local area and thus allows the decision of exploiting past information only if enough past exploration of the local area has occurred.

In order to integrate prior experience (in the form of 1SVMs) with a user's current query, a fuzzy classification of the user's query into the existing concepts (i.e., regions of support) is performed. When a query $\mathbf{x} \in \Re^d$ is submitted, it is determined whether $\mathbf{x}$ falls into one of the existing 1SVMs. Possibilistic cluster analysis [10] is used to assign a degree of membership to each of the 1SVMs (i.e., to each cluster) according to the degree by which $\mathbf{x}$ can be ascribed to that particular concept. The following membership function, μ, is used to assign degrees of membership to the n hyper-spheres into which $\mathbf{x}$ falls [10].

$$\mu(\mathbf{x},\mathbf{a_i}) = 1/\Sigma(\|\Phi(\mathbf{x})- \mathbf{a_i}\|^2 / \|\Phi(\mathbf{x})- \mathbf{a_j}\|^2)$$

where $\mathbf{a_i}$ is the center of the i^{th} hyper-sphere and Φ: $\Re^d \rightarrow \Re^D$, with D > d. Therefore, the degree of membership of $\mathbf{x}$ into a 1SVM is based on the relative distances between $\mathbf{x}$ and the centers of all hyper-spheres into which $\mathbf{x}$ falls.

The approach that is used for selecting the set of images that is presented to the user (i.e., the retrieval set) is based on exploiting both intra and inter-query learning. To fully exploit the RF information provided by the current user (i.e., the intra-query knowledge), the user's feedback is used to train a hyper-sphere with center $\mathbf{a_{current}}$. A fixed value $0 \leq w \leq 1$, which signifies our confidence that $\mathbf{a_{current}}$

captures the user's query concept, is assigned to $\mu(\mathbf{x},\mathbf{a}_{\mathbf{current}})$. To combine this knowledge with accumulated experience, the $\mu(\mathbf{x},\mathbf{a}_{\mathbf{i}})$ of each hyper-sphere into which $\mathbf{x}$ falls is scaled to the range 0 to (1-w) and, in order to form the retrieval set, sample representative images from each hyper-sphere into which $\mathbf{x}$ falls (including $\mathbf{a}_{\mathbf{current}}$) are selected. The nearest neighbor images to a hyper-sphere's center (i.e., to a concept's prototype) are considered to be representative of that concept. The number of images that a particular concept (i.e., a particular hyper-sphere) contributes to the retrieval set is proportional to its $\mu(\mathbf{x},\mathbf{a}_{\mathbf{i}})$. Thus, the ratio of intra to inter-query knowledge that is used in processing $\mathbf{x}$ is w:(1-w). Figure 1 shows a block diagram of the original system.

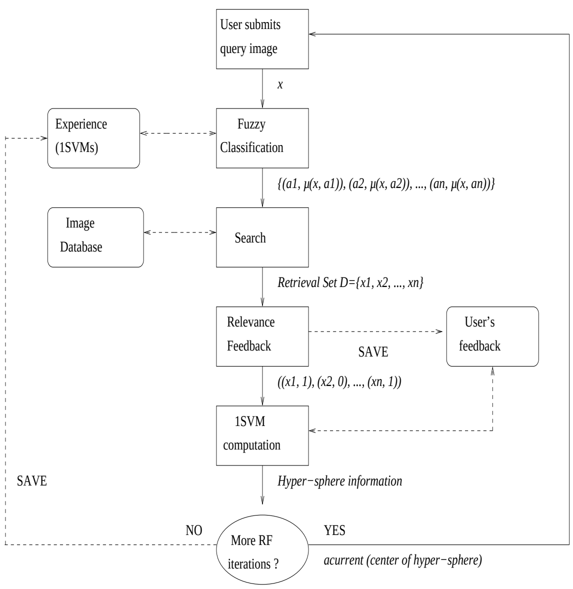

Figure 1: Original System Diagram

3 M-Tree

Many data structures (*B-tree*, for example) have been proposed for the efficient managing of one-dimensional data in traditional database systems. However, because of the rapid development of multimedia database systems during the past decade, the efficient manipulation of multi-dimensional data is vital [5]. In particular, there is an urgent need for indexing techniques that support the efficient execution of similarity queries. Therefore, a number of data storage and indexing techniques (such as the *R-tree* [7]) have been proposed. However, most of those techniques suffer from the *curse of dimensionality* [1], a phenomenon in which performance degrades as the number of dimensions increases. *Metric trees* are a general approach to the similarity indexing problem. In order to organize and partition the search space, they only consider relative distances between objects. They just require that the distance function is a *metric* (i.e., that it satisfies the symmetry, non negativity, and triangle inequality properties) [5]. An M-tree is a paged, balanced, and dynamic tree. It provides an efficient platform for the execution of multi-dimensional similarity queries using an arbitrary metric [5]. We use M-trees for the efficient search of both historical information and images in the database. The *image* M-tree (I-M-tree), contains all the images in the database and the *history* M-tree (H-M-tree) contains the learned 1SVMs (i.e., the historical hyper-spheres). Figure 2 shows a diagram of the proposed extended system.

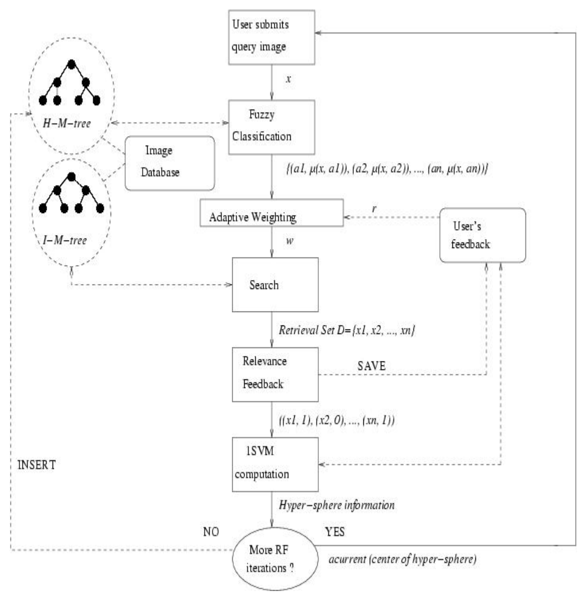

Figure 2: Extended System Diagram

4 Reinforcement Learning

In RL [16], an agent learns how to adapt to its environment by finding optimal actions for the current state. After taking an action, the agent receives a reward, which indicates the goodness of taking the action in that particular state. The agent's objective is to maximize the temporal discounted future reward (the value). This can be achieved by improving the strategy for selecting an action in each state (the policy). At each step of the learning task, the agent perceives its current state s_t and, following the policy π, selects an action a_t. After taking this action, the agent receives a reward r_t and is transferred to the next state s_{t+1}. The reward is used to update the value function that can be either a state-value function $V^{\pi}(s)$, which indicates how good it is for the agent to be in state s, or an action-value function $Q^{\pi}(s,a)$, which indicates how good it is to perform action a in state s. Informally, $V^{\pi}(s)$ is the expected total future reward when starting in state s and following the policy π thereafter, similarly for $Q^{\pi}(s,a)$. The goal is to find the optimal policy π^* (i.e., the optimal mapping from each state s and action a to the probability of taking action a when in state s). For each state and state-action pair, $V^*(s)$, and $Q^*(s,a)$ are the largest expected future reward respectively. Once we have $V^*(s)$ or $Q^*(s,a)$, it is easy to determine the optimal policy π^*. Given a complete knowledge of the environment dynamics, $V^*(s)$ or $Q^*(s,a)$ can be easily obtained by a Dynamic Programming (DP) method. Usually, however, we do not have a complete knowledge of the environment [16].

Monte Carlo (MC) methods can learn directly from experience without needing a model of the environment's dynamics. They estimate the value of a state or a state-action pair simply by averaging the rewards obtained after visits to that state or after taking an action in a state respectively. By the Law of Large Numbers, as the number of observed rewards increases, the average should converge to the expected value. The state-action update rule for a MC method is

$$Q(s,a) = average(Rewards(s, a))$$

where *Rewards(s, a)* is the set of all historical rewards obtained after taking action a in state s. As we can observe, the estimates for each state or state-action pair are independent. That is, unlike DP methods, the estimate of one state or state-action pair does not depend on the estimate of any other state or state-action pair [16].

4.1 Adaptive Weighting

Having a fixed ratio for the combination of intra and inter-query knowledge that is insensitive to historical information is not a good approach. Intuitively, initially we would like to rely heavily on current intra-query learning since, at the beginning, there is not much historical information. Similarly, we would like to increase the exploitation of inter-query knowledge as more queries are processed and experience accumulates. As it was observed in [6], using a fixed ratio for the combination of intra and inter-query knowledge results in a large drop in performance at some data level. In order to overcome this problem, we use RL to learn a rule for adapting this ratio.

The problem of determining the amount/weight of intra and inter-query information to be used for a particular query can be naturally expressed as a RL problem. Indeed, we have an agent operating in an environment - images in feature space. The agent interacts with the set of images at discrete time steps. At each time step, the state of the environment can be modeled by the density of historical information in different regions of the feature space (i.e., by the number and location of hyper-spheres), and by the location of the input query. The agent receives a representation of the environment state and selects an action - choose the ratio of intra to inter-query knowledge. At the next time step, the agent receives a numerical reward from the environment - the percentage of relevant images in the retrieval set (i.e., the precision). The goal of the agent is to include the largest number of relevant images in the retrieval set as quickly as possible. In order to achieve this goal, the agent should learn an adaptive retrieval strategy that is sensitive to the density and location of historical information.

In this paper, we approximate the state of the environment with a representation that is based only on quantitative information about historical data. That is, the state of the environment is based only on the amount of available historical information. We learn a rule that, for each query, adapts the intra to inter-query knowledge ratio based on the number of hyper-spheres into which the query falls. The RL method is used to learn the measured states (i.e., the number of hyper-spheres h that a particular query falls into); the actions are setting the amount of intra-query knowledge weight w to either one of n values in the range 0 to 1. The total number of different states is thus $h + 1$, including the case of a query that does not fall into any hyper-sphere. Therefore, the $Q(s, a)$ table to be learned consists of $(h + 1)n$ state-action pairs. The reward r is equal to the precision obtained after processing the query with the selected ratio. In future work, we will develop a more complex state representation that also encapsulates the particular region in feature space into which the query falls.

4.2 Learning the Adaptation Rule with a MC Method

The number of hyper-spheres into which a particular query falls (i.e., the state) is completely independent from the number of hyper-spheres that the previous query fell into (i.e., the previous state). That is, the policy should be memory-less in the sense that the next action to be taken depends only on the current state. In other words, we need to have a reactive policy that chooses an action based only on the current observation. Therefore, for our problem, it is natural to use a MC method in which the estimates for each state or action-state pair are independent. We use the following modified MC algorithm.

For all $s \in S$, $a \in A(s)$:
Initialize $Q(s,a)$ arbitrarily
$Rewards(s,a)$ = empty list
While more queries:

1. Generate an episode by processing next query
2. Observe s_t (number of hyper-spheres into which the query falls)
3. Choose a_t from s_t using policy derived from $Q(s_t, a_t)$ using an ε-greedy selection scheme
4. Observe r_t (precision)
5. Append r_t to $Rewards(s_t, a_t)$
6. $Q(s_t, a_t)$ = average($Rewards(s_t, a_t)$)

5 Experimental Results

We compare the performance of the extended approach against that of the original system in terms of retrieval performance. The retrieval performance is measured by precision, which is defined as *precision = number of relevant images retrieved / number of images retrieved*.

The *Letter* database consists of 20,000 character images, each represented by a 16-dimensional feature vector. There are 26 classes of the two capital letters O and Q. The images are based on 20 different fonts with randomly distorted letters. To determine the free parameters, a ten-fold cross-validation was performed. Each data set was divided into ten partitions. Each partition in turn was left out and the other nine were used to determine values for the free parameters. The left out partition was then used to test the algorithm. The values reported are the average of the ten tests. Both the original and extended approaches were then evaluated with different amounts of experience (data level) using a Gaussian kernel with width s = 10, and a misclassification penalty $C = 1/pn$, where n is the number of training images, with p = 0.001. The number of images in the retrieval set k = 20. For the original system, the fixed amount of intra-query learning $w \in \{0.1, 0.25, 0.50, 0.75, 0.95\}$. The MC exploration parameter ε = 0.1 in the modified approach. Figures 3 and 4 show the precision of the initial retrieval (i.e., with no RF iterations) with respect to different data levels for the original and modified systems, respectively. The data level is the number of hyper-spheres relative to the number of images in the database. From Figure 3, we can observe that using the original approach on low data levels results in initial retrieval performance worse than KNN retrieval, which is the approach to create the initial retrieval set taken by RF methods that do not use historical information. Figure 4 shows the large improvement obtained by introducing an adaptive ratio of intra to inter-query knowledge based on the amount of historical data in the region local to the query. In this case, the (comparatively much smaller) initial decrease in performance is due to the early stage of learning. As we can observe from those figures, precision in the initial retrieval set can be drastically improved by integrating inter-query learning. This results in a reduction on the number of RF iterations that are needed to satisfy a query. Thus, from the user's point of view, it is very beneficial since users cannot stand too many RF iterations.

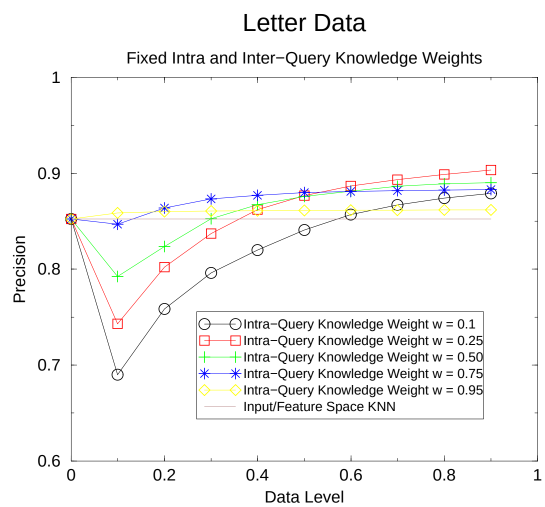

Figure 3: Initial Retrieval Set, Fixed Weighting

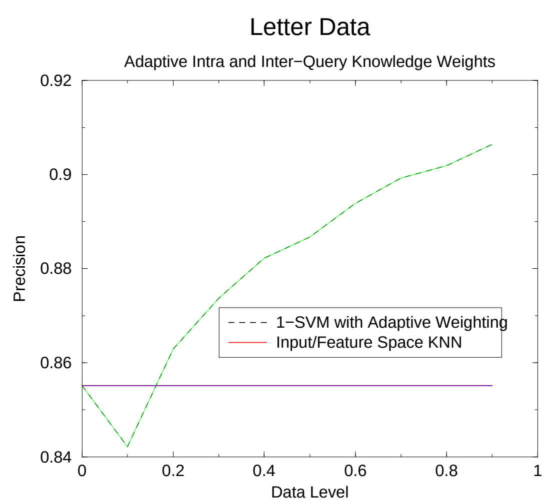

Figure 4: Initial Retrieval Set, Adaptive Weighting

6 Conclusions

This paper presented an extension of previous work on incorporating historical information into a Relevance Feedback (RF) system to improve image retrieval performance. In previous work, a system was developed that learns One-class Support Vector Machines (1SVM) from retrieval experience to represent the set memberships of users' query concepts. By doing a fuzzy classification of a query into the regions of support represented by the 1SVMs, past experience is merged with current intra-query learning. In this paper, we presented an extension of this system that incorporates Reinforcement Learning (RL) for adaptively changing the ratio of intra to inter-query knowledge that is used in processing a query. It also incorporates M-trees for the efficient search of both historical information and images in the database. Initial investigation suggests that an adaptive weighting scheme that is sensitive to the amount of historical information in a local region can overcome the large drops in performance that were observed in the original system. Our future research will focus on methods for combining or merging 1SVMs (i.e., users' concepts). This may be desirable when the amount of historical information is very large. We will also concentrate on developing a better approximation to the state of the environment for learning an adaptive weighting rule with RL.

References:

[1] R. Bellman. *Adaptive Control Processes:A Guided Tour*. Princeton University Press, Princeton, 1961.

[2] B. Boser, I. Guyon, and V. Vapnik. A training algorithm for optimal margin classifiers. In *Proceedings of the 5th Annual ACM Workshop on Computational Learning Theory*, pp. 144-152, July 1992.

[3] C. Campbell, N. Cristianini, and A. Smola. Query learning with large margin classifiers. In *Proceedings of the 17th International Conference on Machine Leaning*, pp. 111-118, June 2000.

[4] Y. Chen, X. Zhou, and T. S. Huang. One-class svm for learning in image retrieval. In *Proceedings of IEEE International Conference on Image Processing*, pp. 34-37, October 2001.

[5] P. Ciaccia, M. Patella, and P. Zezula. M-tree: an efficient access method for similarity search in metric spaces. In *Proceedings of the International Conference on Very Large Databases*, pp. 426-435, August 1997.

[6] I. Gondra, D. Heisterkamp, and J. Peng. Improving the initial image retrieval set by inter-query learning with one-class svms. In *Proceedings of the 3rd International Conference on Intelligent Systems Design and Applications*, pp. 393-402, August 2003.

[7] A. Guttman. R-trees: a dynamic index structure for spatial searching. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pp. 47-57, June 1984.

[8] D. Heisterkamp. Building a latent-semantic index of an image database from patterns of relevance feedback. In *Proceedings of the 16th International Conference on Pattern Recognition*, pp. 132-135, August 2002.

[9] P. Hong, Q. Tian, and T. Huang. Incorporate support vector machines to content-based image retrieval with relevance feedback.In *Proceedings of the IEEE International Conference on Image Processing*, pp. 750-753, September 2000.

[10] F. Hoppner, F. Klawonn, R. Kruse, and T. Runkler. *Fuzzy Cluster Analysis*. John Wiley & Sons, New Yok, 1999.

[11] M. Li, Z. Chen, and H. Zhang. Statistical correlation analysis in image retrieval. *Pattern Recognition*, 35(12):2687-2693, 2002.

[12] H. Muller, W. Muller, D. Squire, S. Marchand-Maillet, and T. Pun. Long-term learning from user behavior in content-based image retrieval. Technical Report 00.04, University of Geneva, Geneva, Switzerland, 2000.

[13] J. Peng, B. Bhanu, and S. Qing. Probabilistic feature relevance learning for content-based image retrieval. *Computer Vision and Image Understanding*, 75(1/2):150-164, 1999.

[14] Y. Rui and T. Huang. Relevance feedback: A power tool for interactive content-based image retrieval. *IEEE Transactions on Circuits and Systems for Video Technology*, 8(5):644-655, 1998.

[15] G. Schohn and D. Cohn. Less is more: Active learning with support vector machines. In *Proceedings of the 17th International Conference on Machine Learning*, pp. 441-448, June 2000.

[16] R. Sutton and A. Barto. *Reinforcement Learning – An Introduction*. MIT Press, Cambridge, 1998.

[17] S. Tong and E. Chang. Support vector machine active learning for image retrieval. In *Proceedings of the 9th ACM International Conference on Multimedia*, pp. 107-118, September 2001.

[18] P. Yin, B. Bhanu, K. Chang, and A. Dong. Improving retrieval performance by long-term relevance information. In *Proceedings of the 16th International Conference on Pattern Recognition*, pp. 533-536, August 2002.