

Orthogonal Complement Component Analysis for Positive Samples in SVM Based Relevance Feedback Image Retrieval

Dacheng Tao and Xiaoou Tang
Department of Information Engineering
The Chinese University of Hong Kong
{dctao2, xtang}@ie.cuhk.edu.hk

Abstract

Relevance feedback (RF) is an important tool to improve the performance of content-based image retrieval system. Support vector machine (SVM) based RF is popular because it can generalize better than most other classifiers. However, directly using SVM in RF may not be appropriate, since SVM treats the positive and negative feedbacks equally. Given the different properties of positive samples and negative samples in RF, they should be treated differently. Considering this, we propose an orthogonal complement components analysis (OCCA) combined with SVM in this paper. We then generalize the OCCA to Hilbert space and define the kernel empirical OCCA (KEOCCA). Through experiments on a Corel Photo database with 17,800 images, we demonstrate that the proposed method can significantly improve the performance of conventional SVM-based RF.

1. Introduction

Content-based image retrieval (CBIR) system [1] tries to retrieve images semantically relevant to user's query from an image database based on automatically extracted visual features. However, the gap [2] between the low-level visual feature and the high-level semantic concepts of the image often leads to poor results.

To bridge the gap and to scale the performance, the interactions between the user and the search engine are required. The user labels the previous retrieved images as semantically relevant or irrelevant and the computer uses the information to refine the retrieval results. The technique is generally named as relevance feedback (RF) [2-4]. RF is widely used as an important method to scale the performance in CBIR systems.

MARS [4] introduced both a query movement and re-weighting techniques to estimate user's sentiment. MindReader [3] formulated a minimization problem on parameters estimation process. PicHunter [5] proposed a stochastic comparison search as its RF algorithm. Zhou and Huang [6-7] formulated the RF as an optimal learning problem. Jing modeled the RF as a multi-class

problem [8]. Friedman tried to learn local feature relevance to combine the best ones for k-nearest-neighbor search [9]. Recently, support vector machine (SVM), a small sample learning algorithm, was introduced to RF procedure [10-13] because of its generalization ability. However, directly using SVM to RF may not be suitable because SVM handles the positive and negative feedbacks equally. In order to improve the performance of SVM RF, we propose an orthogonal complement components analysis to put more emphasis on positive samples and also simplify the SVM hyper-plane. Experiments on a Corel database show significant improvement of the RF performance by the new approach.

2. Orthogonal Complement Components Analysis for SVM

2.1. Analysis

From the statistical learning theory [14], we know that the following inequality (1) holds with probability of at least $1 - \delta$ for any $n > h$.

$$\begin{cases} R[f] \leq R_{emp}[f] + G(h, n, \delta) \\ G(h, n, \delta) = \sqrt{\frac{h \left(\ln \left(\frac{2n}{h} \right) + 1 \right) - \ln \left(\frac{\delta}{4} \right)}{n}} \end{cases} \quad (1)$$

where h denotes the Vapnik-Chervonenkis (VC) dimension of the classifier function set, n is the size of the training set, and R_{emp} describes the empirical risk. For all $\delta > 0$ and $f \in F$ the inequality bounds the risk. The inequality (1) gives us a way to estimate the error on future data based only on the training error and the VC dimension of the classifier function set.

It is well known that the smaller the risk value $R[f]$, the better the performance of the classifier. From (1), we can see that the risk depends on the empirical risk R_{emp} and $G(h, n, \delta)$. Based on the representation of $G(h, n, \delta)$, we know that $G(h, n, \delta)$ is a strictly monotonically increasing function of h for given n and δ . h is determined by the support

vectors when training data number is smaller than feature dimension. In addition, the VC dimension h is almost an increasing function of the number of support vectors. Consequently, SVM's performance depends mostly on the empirical risk, the number of the support vectors, and δ . Since δ cannot be controlled manually, we can restrict R_{emp} and the number of support vectors to achieve a good performance.

In CBIR RF, it is easy to achieve zero empirical risk R_{emp} by enough number of the support vectors. However, a large number of support vectors enlarge the VC dimension of SVM classifier h . Therefore, we want to restrict both h and R_{emp} . To solve the problem, an intuitive way is to search a subspace to reduce the training set. There are two possibilities: 1. project all positive feedbacks onto their center and then project all negative feedbacks onto the subspace; 2. project all negative feedbacks onto their center and then project all positive feedbacks onto the subspace. For CBIR, the first method is much more reasonable than the second one because all positive feedbacks are similar to the query image. Meanwhile, in the projection step, the optimal hyper-plane of SVM classifier can be deformed by any increasing positive feedbacks and SVM classifier will not be sensitive to any negative feedbacks. Therefore, more emphasis is put on positive samples. In addition, the resulting SVM hyper-plane will be simpler around the projection center. Following this observation, we propose an orthogonal complement components analysis to improve SVM.

2.2. Orthogonal Complement Components Analysis Support Vector Machine

Orthogonal Complement Components Analysis SVM can be mainly implemented in three steps. The first step is to project all positive feedback samples onto their center, the second step is to project all negative feedback samples onto the subspace, and the last step is to construct a SVM classifier in the subspace.

For a set of positive feedback samples $\{\mathbf{x}_i^+, 1 \leq i \leq P\}$, where $\mathbf{x}_i^+ \in R^M$, M is the dimension of the feature space and P is the number of the positive feedbacks, Karhunen-Leove transformation (KLT) can be used to extract the principal subspace and its orthogonal complement. The principle components describe the variation of the positive feedbacks' distribution while the orthogonal complement components describe the in-variation of the positive feedbacks' distribution. The basis functions for the KLT are obtained by solving the eigenvalue problem:

$$\begin{bmatrix} \Lambda & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} = [\Phi \quad \Phi^\perp]^T \Xi [\Phi \quad \Phi^\perp], \quad (2)$$

where Ξ is the covariance matrix of the positive feedbacks, Φ is the principle subspace of Ξ , Φ^\perp is the orthogonal complement subspace of Φ in Ξ , Λ is the corresponding diagonal matrix of eigenvalues of Φ , and the eigenvalues of Φ^\perp are 0. The unitary matrix Φ^\perp defines a coordinate transform, which decorrelates the data, makes explicit the invariant subspaces of the matrix operator Ξ , and ensures that all positive feedbacks are mapped to their center. By KLT, we can obtain the orthogonal complement feature vector $\mathbf{y}^+ = (\Phi^\perp)^T (\mathbf{x}^+ - \bar{\mathbf{x}}^+)$, where $\bar{\mathbf{x}}^+ = \frac{1}{P} \sum_{i=1}^P \mathbf{x}_i^+$ is the center of the positive feedbacks, \mathbf{x}^+ is the data matrix constructed by all positive feedbacks, and \mathbf{y}^+ is the projected data matrix of positive feedbacks (It is clear that all columns of \mathbf{y}^+ are equal). We call the transformation as orthogonal complement component analysis (OCCA), just like principal component analysis (PCA). OCCA preserves the invariant direction of the data distribution.

Table 1. The algorithm of OCCA SVM.

-
1. Calculate the covariance matrix Ξ of the positive feedbacks.
 2. Calculate the orthogonal complement components Φ^\perp of Ξ according to $[\Phi^\perp]^T \Xi \Phi^\perp = 0$.
 3. Project all positive feedbacks \mathbf{x}^+ onto their center \mathbf{y}^+ .
 4. Project all negative feedback samples \mathbf{x}^- onto the orthogonal complement subspace, $\mathbf{y}^- = (\Phi^\perp)^T (\mathbf{x}^- - \bar{\mathbf{x}}^+)$.
 5. Project the remaining images \mathbf{x} in the database onto the orthogonal complement subspace, $\mathbf{y} = (\Phi^\perp)^T (\mathbf{x} - \bar{\mathbf{x}}^+)$.
 6. Train a standard SVM classifier on $\mathbf{z} = [\mathbf{y}^+, \mathbf{y}^-]$.
 7. Resort the remaining projected images \mathbf{y} using the output of SVM $f(\mathbf{y}) = \sum_{i=1}^{N_i} \alpha_i y_i K(\mathbf{z}_i, \mathbf{y}) + b$.
-

After projecting all positive feedbacks onto their center, we can project all negative feedbacks onto the subspace according to $\mathbf{y}^- = (\Phi^\perp)^T (\mathbf{x}^- - \bar{\mathbf{x}}^+)$, where \mathbf{x}^- is the data matrix constructed by all negative feedbacks and \mathbf{y}^- is the projected data matrix of the negative feedbacks.

Then all the images \mathbf{x} in the database are also projected onto the subspace through $\mathbf{y} = (\Phi^\perp)^T (\mathbf{x} - \bar{\mathbf{x}}^+)$,

where \mathbf{y} is the projected data matrix of the original data matrix \mathbf{x} .

The standard SVM classification algorithm is executed on $\mathbf{z} = [\mathbf{y}^+, \mathbf{y}^-]$, where $|\mathbf{z}| = 1 + N$ and N is the number of the negative feedbacks. Finally, we can measure the dissimilarity through the output of SVM $f(\mathbf{y}) = \sum_{i=1}^{N_s} \alpha_i y_i K(\mathbf{z}_i, \mathbf{y}) + b$, where N_s is the number of the support vectors. The outline of the proposed algorithm is shown in Table 1.

2.3. Orthogonal Complement Components Analysis SVM in the Kernel Space

In last Section, we derived the linear space OCCA. We know that a single Gaussian distribution often accurately describes the distribution of samples in the input feature space when the positive feedbacks are similar objects under the same conditions (e.g. similar view angle, similar illumination, etc.). However, this is not the case for CBIR. Therefore considering all positive feedbacks forming a single Gaussian is not reasonable. Meanwhile, the dimension of the orthogonal complement components decreases with the increasing of the positive feedbacks. Consequently, the performance of the system will be degraded by the noise. Therefore, generalizing the algorithm to its kernel version (KEOCCA SVM) will be helpful.

To complete the KEOCCA SVM, the kernel version of KLT is required. The principal components can be extracted by kernel principal component analysis (KPCA) [16], because all eigenvectors with nonzero eigenvalues must be in the span of the mapped data. However, we cannot obtain all the orthogonal complement components of the positive feedbacks in this way. A feasible solution is to extract a subset of the orthogonal complement components. It means we can think that parts of the orthogonal complement space of positive feedbacks are spanned by the positive and negative feedbacks in the Hilbert space. Note that the orthogonal complement space of the positive feedbacks cannot be spanned by all images in the database, because many of the images in the database which are query relevant but not positive feedbacks, and we can only obtain the covariance matrix of the positive feedbacks. Hence the orthogonal complement components of the positive feedbacks constructed by all feedbacks are called the kernel empirical orthogonal complement components (KEOCC), while the transformation is called kernel empirical orthogonal complement component analysis (KEOCCA).

Similar to SVM and KPCA, we first map the data \mathbf{x} to $\psi(\mathbf{x})$ in Hilbert space, and then the kernel trick

$K(\mathbf{x}_i, \mathbf{x}_j) = \psi^T(\mathbf{x}_i) \cdot \psi(\mathbf{x}_j)$ is utilized to obtain the solution. We first calculate the covariance matrix of the positive feedbacks in the Hilbert space according to,

$$\Xi = \sum_{i=1}^P (\psi(\mathbf{x}_i^+) - \bar{\psi}(\mathbf{x}^+)) (\psi(\mathbf{x}_i^+) - \bar{\psi}(\mathbf{x}^+))^T, \quad (3)$$

where $\bar{\psi}(\mathbf{x}^+) = \frac{1}{P} \sum_{i=1}^P \psi(\mathbf{x}_i^+)$ is the center of the positive feedbacks in the Hilbert space. According to the previous analysis of the orthogonal complement components in the Hilbert space, we know that $\tilde{\Phi}^\perp \in \text{span}\{\psi(\mathbf{x}_1^+), \psi(\mathbf{x}_2^+), \dots, \psi(\mathbf{x}_p^+), \psi(\mathbf{x}_1^-), \dots, \psi(\mathbf{x}_N^-)\}$ (because we cannot obtain the complete orthogonal complement space, we mark the empirical orthogonal complement components as $\tilde{\Phi}^\perp$.) Therefore the basis function for the KEOCCA can be solved by the eigenvalue problem,

$$\mathbf{0} = (\tilde{\Phi}^\perp)^T \Xi \tilde{\Phi}^\perp,$$

where $\tilde{\Phi}^\perp = \sum_{i=1}^P \xi_i \psi(\mathbf{x}_i^+) + \sum_{i=1}^{N+P} \xi_i \psi(\mathbf{x}_{i-P}^-)$.

Through the kernel trick, the eigenvalue problem can be solved by using the kernel matrix \mathbf{K} ,

$$(\tilde{\Phi}^\perp)^T \Xi \tilde{\Phi}^\perp = \xi^T \sum_{i=1}^P \left(\mathbf{K}_{(:,i)} - \frac{1}{P} \sum_{k=1}^P \mathbf{K}_{(:,k)} \right) \left(\mathbf{K}_{(:,i)} - \frac{1}{P} \sum_{k=1}^P \mathbf{K}_{(:,k)} \right)^T \xi. \quad (4)$$

Kernel matrix is defined as,

$$\mathbf{K} = \begin{bmatrix} \mathbf{K}_{(:,1)} & \dots & \mathbf{K}_{(:,i)} & \dots & \mathbf{K}_{(:,P+N)} \end{bmatrix} = \begin{bmatrix} K(\mathbf{x}_1^+, \mathbf{x}_1^+) & \dots & K(\mathbf{x}_1^+, \mathbf{x}_p^+) & K(\mathbf{x}_1^+, \mathbf{x}_1^-) & \dots & K(\mathbf{x}_1^+, \mathbf{x}_N^-) \\ \dots & \dots & \dots & \dots & \dots & \dots \\ K(\mathbf{x}_p^+, \mathbf{x}_1^+) & \dots & K(\mathbf{x}_p^+, \mathbf{x}_p^+) & K(\mathbf{x}_p^+, \mathbf{x}_1^-) & \dots & K(\mathbf{x}_p^+, \mathbf{x}_N^-) \\ K(\mathbf{x}_1^-, \mathbf{x}_1^+) & \dots & K(\mathbf{x}_1^-, \mathbf{x}_p^+) & K(\mathbf{x}_1^-, \mathbf{x}_1^-) & \dots & K(\mathbf{x}_1^-, \mathbf{x}_N^-) \\ \dots & \dots & \dots & \dots & \dots & \dots \\ K(\mathbf{x}_N^-, \mathbf{x}_1^+) & \dots & K(\mathbf{x}_N^-, \mathbf{x}_p^+) & K(\mathbf{x}_N^-, \mathbf{x}_1^-) & \dots & K(\mathbf{x}_N^-, \mathbf{x}_N^-) \end{bmatrix}. \quad (5)$$

Therefore, we can obtain the KEOCC according to ξ ,

$$\text{which makes } \mathbf{0} = (\tilde{\Phi}^\perp)^T \Xi \tilde{\Phi}^\perp.$$

Similar to OCCA SVM, we project the positive feedbacks, negative feedbacks, and all images in the database onto the KEOCC spanned space by $\mathbf{y} = (\tilde{\Phi}^\perp)^T (\psi(\mathbf{x}) - \bar{\psi}(\mathbf{x}^+))$. In KEOCC, the positive feedback, negative feedback, and image in the database are represented by \mathbf{y}^+ , \mathbf{y}^- , and \mathbf{y} respectively.

Using $\mathbf{z} = [\mathbf{y}^+, \mathbf{y}^-]$, the standard SVM classification algorithm is trained. Finally, we can measure the dissimilarity through the output of SVM according to $f(\mathbf{y}) = \sum_{i=1}^{N_s} \alpha_i y_i K(\mathbf{z}_i, \mathbf{y}) + b$, where N_s is the number of the support vectors. The algorithm is shown in Table 2.

Table 2. The algorithm of KEOCCA SVM.

1. Calculate the kernel matrix \mathbf{K} .
2. Calculate the kernel empirical orthogonal complement components $\tilde{\Phi}^\perp$ of the kernel covariance matrix Ξ of the positive feedbacks by $\mathbf{0} = (\tilde{\Phi}^\perp)^T \Xi \tilde{\Phi}^\perp$.
3. Project all positive feedbacks \mathbf{x}^+ onto their center \mathbf{y}^+ according to $\mathbf{y}^+ = (\tilde{\Phi}^\perp)^T (\psi(\mathbf{x}^+) - \bar{\psi}(\mathbf{x}^+))$.
4. Project all negative feedback samples \mathbf{x}^- onto the empirical kernel orthogonal complement subspace according to $\mathbf{y}^- = (\tilde{\Phi}^\perp)^T (\psi(\mathbf{x}^-) - \bar{\psi}(\mathbf{x}^+))$.
5. Project the remaining images \mathbf{x} in the database to the subspace according to $\mathbf{y} = (\tilde{\Phi}^\perp)^T (\psi(\mathbf{x}) - \bar{\psi}(\mathbf{x}^+))$.
6. Train a standard SVM classifier on $\mathbf{z} = [\mathbf{y}^+, \mathbf{y}^-]$.
7. Resort the projected remaining images \mathbf{y} using the output of SVM $f(\mathbf{y}) = \sum_{i=1}^{N_s} \alpha_i y_i K(\mathbf{z}_i, \mathbf{y}) + b$.

3. Image Retrieval System

In CBIR, we assume that the user expects the most possible retrieval results after each RF iterations, i.e. the search engine is required to feedback the most semantically relevant images according to the previous feedback samples. Meanwhile, the user is impatient, who will never label a large number of images in each RF iteration and only does a few numbers of iterations [17]. For image retrieval, the images are represented by color [18], texture [19], and shape [20]. Color information is the most important features for image retrieval because color is robust with respect to scaling, orientation, perspective, and occlusion of images [18]. Texture information is also an important cue for image retrieval. Previous studies on texture have shown that texture information based on structure and orientation fits the model of human perception well. Shape information is another type of important clues that fit the perception of human, and many image retrieval systems use the feature. In this paper, we select the color histogram [18], Gabor texture [19], and edge direction histogram [20] to represent images.

Figure 1 shows the user interface of our image retrieval system. Here query by example is used. To scale the performance, we focus on the RF algorithms. First, user selects a query image from the thumbnail gallery and clicks the “Set as Query” button. Then user clicks the “Retrieval” button, and the images in the gallery are resorted. Next, user provide the feedback by clicking on the “thumb up” or “thumb down” button in terms of his judgment of the relevance of the retrieved image. Finally, user clicks the “Retrieval” button to

resort the images in the gallery. The last two steps can be done iteratively to obtain a satisfactory performance.

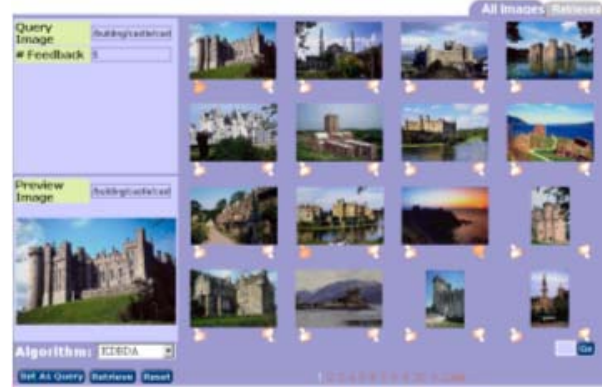


Figure 1. The user interface of the system.

4. Experimental Results

The experiments were divided into three parts. Accuracy, which is the ratio of the number of relevant images retrieved to the top N retrieved images, is used to evaluate the retrieval performance. For algorithms, i.e. SVM [10], OCCA SVM, KEOCCA SVM, we choose the Gaussian kernel:

$$K(\mathbf{x}, \mathbf{y}) = e^{-\rho \|\mathbf{x} - \mathbf{y}\|_2^2}, \quad \rho = 1. \quad (6)$$

The first evaluation experiment was executed on a small size database, which includes 1,600 wildlife images with 16 different types of wildlife animals from Corel. We use all 1,600 images as queries. During RF iterations, the first 5 query relevant and irrelevant images were selected as positive and negative feedbacks from the top 48 retrieved images in the previous iteration, respectively. In the first experiment, we want to compare the performance between these proposed algorithms and the traditional SVM based RF algorithms. In this experiment, we did RF 4 times. Figure 2 shows the experimental results. We can see that the proposed KEOCCA SVM can significantly outperform SVM.

Most recent CBIR evaluation experiments were executed on large-scale image database. In this experiment, we compare the new algorithm KEOCCA SVM with SVM in a subset of Corel Photo Gallery [1], which includes 17,800 images with 90 concepts. The computer randomly selected 300 queries. For each query image, 9 RF iterations were executed. The experimental results are shown in Figure 3. From the figure, we can see that the proposed method KEOCCA SVM performs much better than the original SVM.

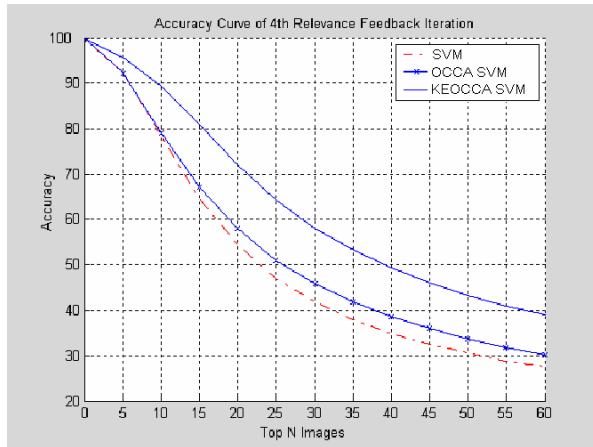


Figure 2. Evaluation experiment on small database.

At last, we also did some real-world experiments. We randomly select some images as the queries. For each query, we did RF iteration 4 times. For each RF iteration, we select some query relevant and irrelevant images as positive and negative feedbacks from the first three screen shots, respectively. The number of the positive and negative feedbacks is less than 10. Meanwhile, they are not the top retrieved images. We chose them according to the sentiments. Figure 4 shows the experimental results. The top-left image of each figure is the query. We can see that the proposed algorithm KEOCCA SVM can work well in practical applications.

5. Conclusion

To improve the performance of content-based image retrieval (CBIR), relevance feedback (RF) plays an essential role. Recently, Support Vector Machine (SVM) has been used in RF. The advantage of SVM is that it can generalize better than many other classifiers. To improve SVM based-RF we propose the orthogonal complement component analysis (OCCA) combined with the SVM. We then generalize the OCCA to Hilbert space and define the kernel empirical OCCA (KEOCCA). Finally, we combine the KEOCCA with SVM. Through experiments on Corel Photo Galley with 17,800 images, we show that our new method can outperform the original SVM-based RF significantly.

6. Acknowledgement

The work described in this paper was fully supported by a grant from the Research Grants Council of the Hong Kong SAR. (Project no. AoE/E-01/99).

7. References

- [1] J.Z. Wang, J. Li, G. Wiederhold, "SIMPLiCity: Semantics-Sensitive Integrated Matching for Picture Libraries," *IEEE Trans. on PAMI*, vol. 23, no. 9, pp. 947-963, Sept. 2001.
- [2] Y. Rui, T. S. Huang, and S. Mehrotra. "Content-based Image Retrieval with Relevance Feedback in MARS," In *Proc. IEEE ICIP*, 1997.
- [3] Y. Ishikawa, R. Subramanya, and C. Faloutsos. "Mindreader: Querying Databases through Multiple Examples," In *VLDB* 1998, pp 433-438.
- [4] Y. Rui, T. S. Huang, M. Ortega, and S. Mehrotra. "Relevance Feedback: A Power Tool in Interactive Content-based Image Retrieval," *IEEE Trans. on CSVT*, Sept. 1998.
- [5] I.J. Cox, L. Miller, P. Minka, V. Paphomas, and P. Yianilos, "The Bayesian Image Retrieval System, PicHunter: Theory, Implementation and Psychophysical Experiments," *IEEE Trans. on IP*, vol 9, no.1, 20-37, 2000.
- [6] X. S. Zhou, T. S. Huang, "Small Sample Learning During Multimedia Retrieval Using Biasmap," In *Proc. IEEE CVPR*, 2001.
- [7] X. S. Zhou, T. S. Huang, "Comparing Discriminating Transformations and SVM for Learning during Multimedia Retrieval," In *Proc. ACM Int. Conf. on MM*, 2001.
- [8] P. Jing, "Multi-class Relevance Feedback Content-based Image Retrieval," *Computer Vision and Image Understanding*, pp. 42-67 2003.
- [9] J.H. Friedman, "Flexible Metric Nearest Neighbor Classification," *Technique Report*, Dept. of Statistics, Stanford U. 1994.
- [10] L. Zhang, F. Lin, and B. Zhang, "Support Vector Machine Learning for Image Retrieval," In *Proc. IEEE ICIP*, 2001.
- [11] P. Hong, Q. Tian, and T. S. Huang. "Incorporate Support Vector Machines to Content-based Image Retrieval with Relevant Feedback," In *Proc. IEEE ICIP*, 2000.
- [12] Y. Chen, X. S. Zhou, and T. S. Huang, "One-class SVM for Learning in Image Retrieval," In *Proc. IEEE ICIP*, 2001.
- [13] G. Guo, A. K. Jain, W. Ma, and H. Zhang, "Learning Similarity Measure for Natural Image Retrieval with Relevance Feedback," *IEEE Trans. on NN*, vol. 12, no. 4, pp.811-820, July 2002.
- [14] Vapnik, V. "The Nature of Statistical Learning Theory," Springer-Verlag, New York (1995).
- [15] J. Burges, "A Tutorial on Support Vector Machines for Pattern Recognition," *Data Mining and Knowledge Discovery* 2, pp. 121-167, 1998.
- [16] K.R. Muller, S. Mika, G. Ratsch, K. Tsuda, and B. Scholkopf, "An Introduction to Kernel-based Learning Algorithms," *IEEE Trans. on NN*, vol 12, no. 2, Mar. 2001.
- [17] X.S. Zhou, T.S. Huang, "Relevance Feedback for Image Retrieval: a Comprehensive Review," *ACM Multimedia Systems Journal*, vol. 8, no. 6, pp. 536-544, Apr. 2003.
- [18] M.J. Swain and D.H. Ballard. "Color Indexing," *IJCV*, vol. 7, no. 1 pp.11-32, 1991.
- [19] B. S. Manjunath and W. Y. Ma. "Texture Features for Browsing and Retrieval of Image Data," *IEEE Trans. on PAMI*, vol.18 no. 8 pp. 837-42, Aug. 1996.
- [20] A. K. Jain and A. Vailaya. "Image Retrieval Using Color and Shape," *Pattern Recognition*, vol. 29, no.8 pp.1233-1244, Aug. 1996.

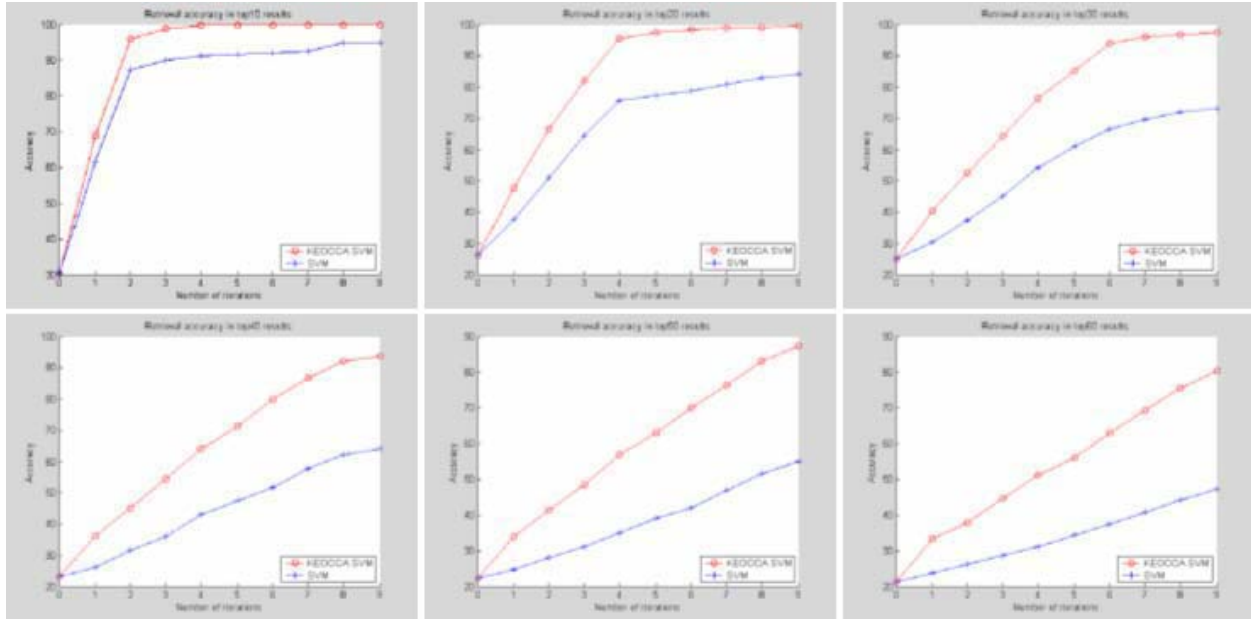


Figure 3. Evaluation Experimental Results on Large-Scale Corel Photo Gallery with 17,800 images. The top-left, top-middle, top-right, bottom-left, bottom-middle, and bottom-right figures show the mean accuracy curve with 9 RF iterations in the top 10, 20, 30, 40, 50, and 60 retrieved images, respectively.

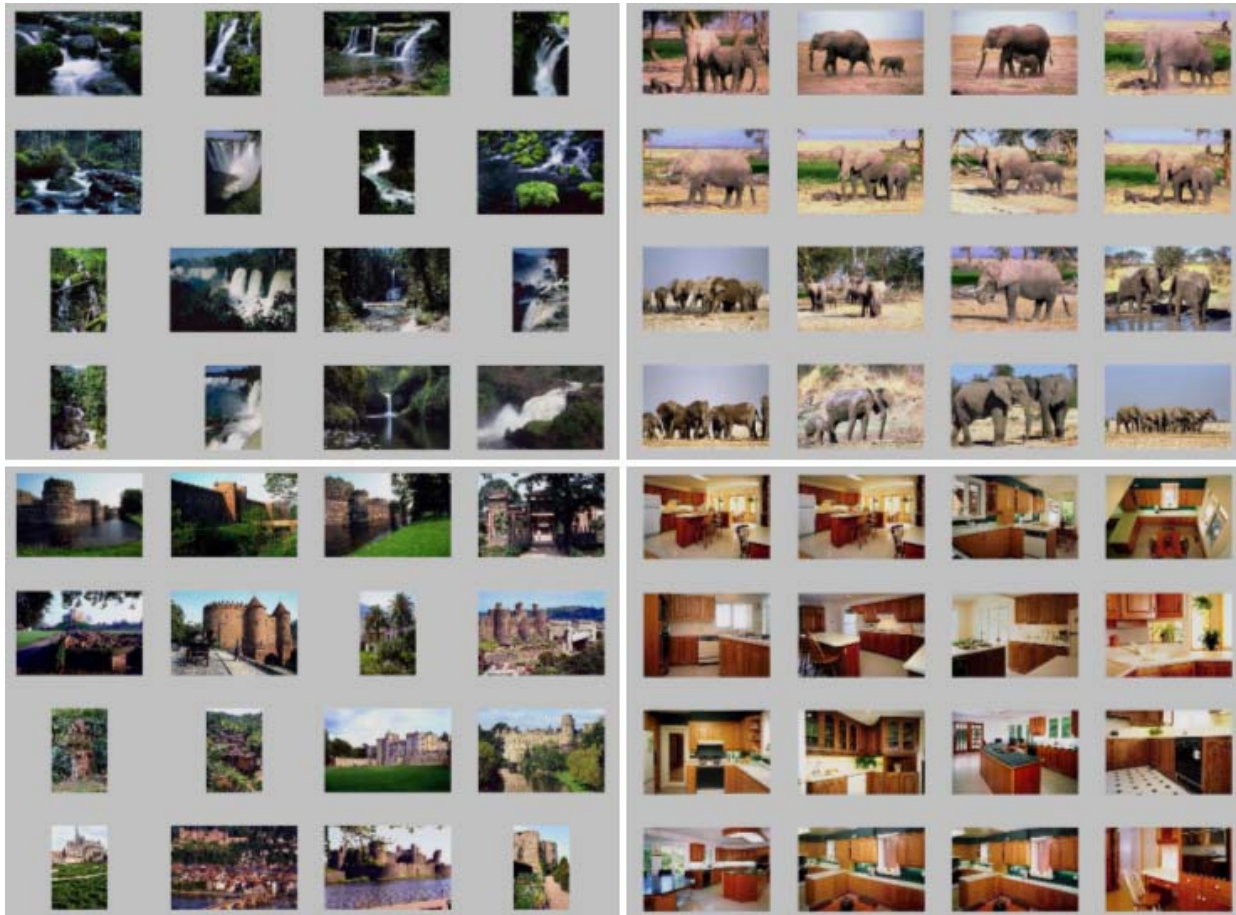


Figure 4. Real-World Experimental results in the 4th RF iteration. The top-left image of each subfigure is the query.