Multimedia Queries by Example and Relevance Feedback

Leejay Wu Christos Faloutsos Katia Sycara
Terry Payne
Carnegie Mellon University
{lw2j,christos,katia,terryp}@cs.cmu.edu

Abstract

We describe the FALCON system for handling multimedia queries with relevance feedback. FALCON distinguishes itself in its ability to handle even disjunctive queries on metric spaces. Our experiments show that it performs well on both real and synthetic data in terms of precision/recall, speed of convergence, individual query speed, and scalability. Moreover, it can easily take advantage of off-the-shelf spatial- and metric- access methods.

1 Introduction

Interactive multimedia databases such as Informedia [2, 15] present an intriguing challenge. Museum kiosks, public media archives, and similar systems require simple interfaces; museum patrons should not need to learn a query language simply to navigate a catalogue of major exhibits. The obvious user-friendly solution is "query by example", augmented by relevance-feedback with which users iteratively improve the specification of their queries.

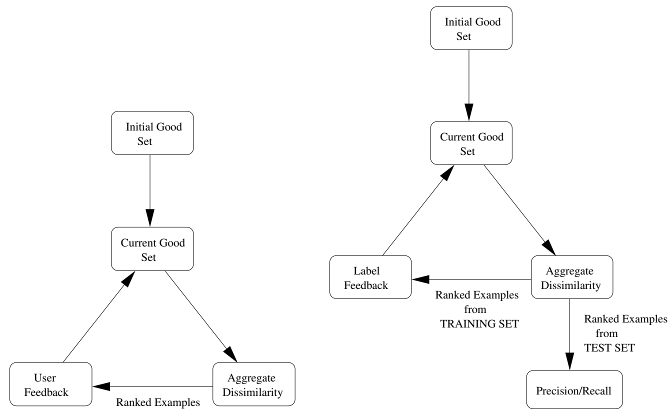
One well-known relevance feedback mechanism is that of Rocchio [10, 1]. It is a basic single-query-point method, in that it produces contiguous hyperspherical isosurfaces in feature spaces centered around *one* query point. To derive its query point, it linearly combines:

- The query point derived from the last iteration.
- A weighted centroid of the documents already judged relevant.
- A weighted centroid of the documents already judged irrelevant. This term receives a negative weight.

Thus, it takes into account relevance feedback, and therefore should gravitate towards positive examples, away from negative ones. Setting the weights for the three components may require some experimentation.

Two more single-query-point systems include MindReader [7], and MARS [11, 9, 12, 13]. MindReader computes the Mahalanobis distance function using the matrix of known positive instances. Since this involves matrix inversion, it can require a significant number of instances when the dimensionality is high. The Mahalanobis space allows MindReader to derive arbitrary hyperelliptical isosurfaces around multiple query points; these isosurfaces can then be refined when the user adds or removes examples from the desired set. MARS takes a more traditional approach in which similarities are computed between known positive instances and candidates, and linearly then combined using user-specified weights.

Bulletin of the IEEE Computer Society Technical Committee on Data Engineering



(a) Normal operation (b) Evaluation

Figure 1: The FALCON relevance-feedback loop, (a) designed for interactive use, and (b) modified for automated performance testing.

PicHunter [4, 5] provides a Bayesian approach to finding a particular image. At each iteration, a user selects zero or more of the presented images. This binary feedback is the only input it uses to develop a probabilistic model as to which image is the target. PicHunter distinguishes itself in another aspect; it explicitly tries to minimize the number of subsequent iterations by taking into account expected information gain when choosing which examples to present to the user, rather than simply always presenting those thought to be closest to the target.

2 Proposed System: FALCON

Herein we describe the FALCON system. Like the aforementioned systems, it incorporates user feedback in order to iteratively improve its internal representation of the query concept. The feedback loop is illustrated in Figure 1(a); Figure 1(b) shows a variation used for non-interactive use during testing, where feedback comes from pregenerated labels rather than a user. This latter variation will be discussed further with the rest of the evaluation methodology.

FALCON maintains a current "good set" $\mathcal{G}$ consisting of objects that have been labeled by the user as good examples. Where it differs is in how it uses $\mathcal{G}$ to rank candidate objects.

For this, FALCON uses a distance-combining function we call "aggregate dissimilarity". Given d, a pairwise distance function the internals of which do not concern us; m distinct good objects g_i; α a tuning parameter we discuss below; and x, any given candidate, the aggregate dissimilarity of x with respect to $\mathcal{G}$ is defined as follows.

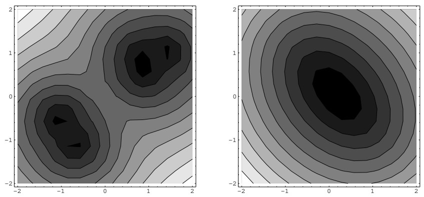
$$(D_{\mathcal{G}}(x))^{\alpha} = \begin{cases} 0 & \text{if } (\alpha < 0) \bigwedge \exists i \; d(x, g_i) = 0 \\ \frac{1}{m} \times \sum_{i=1}^{m} d(x, g_i)^{\alpha} & \text{otherwise} \end{cases} \tag{1}$$

Weights can be included as follows, with weight w_i corresponding to good object g_i:

$$(D_{\mathcal{G}}(x))^{\alpha} = \frac{1}{\sum_{i=1}^{m} w_i} \cdot \sum_{i=1}^{m} w_i (d(x, g_i))^{\alpha} \tag{2}$$

2.1 Properties of $D_{\mathcal{G}}$

Since $D_{\mathcal{G}}$ is used to measure the probable relevance of objects to return to the user, the exact behavior of $D_{\mathcal{G}}$ is clearly important. The tunable parameter α has significant effects on the isosurfaces generated by $D_{\mathcal{G}}$, as listed



(a) $\alpha = -5$ (b) $\alpha = 2$

Figure 2: Sample contours with two values of α. Both are based on the same $\mathcal{G}$ – three points chosen from one circular region, two from another.

below.

- $D_{\mathcal{G}}$ handles disjunctive queries well when $\alpha < 0$. If $\alpha = -\infty$, then $D_{\mathcal{G}}$ is analogous to a fuzzy OR, in that it results in the minimum distance. With $\alpha \in (-\infty, 0)$, all distances are taken into account but the bias is towards the lesser ones. Thus, it can generate contours such as shown in Figure 2(a), derived from a disjunctive $\mathcal{G}$ consisting of five points from two disjoint circular regions.
  Disjunctive queries are important, because many queries that may appear simple to people are disjunctive. One such simple query would be for stocks whose trends resemble V's or inverted V's, which might be useful for a short-term market-timer. Any system that represented queries via a single convex region will fail to capture the nature of the query, no matter how many positive or negative examples were specified.
- Positive values of α generate fuzzy ANDs. The extreme value of $\alpha = \infty$ results in $D_{\mathcal{G}}$ using only the maximum distance, while values of $\alpha \in (0, \infty)$ account for all distances but bias towards the higher ones. As a consequence, a negative α is a more natural choice for the greater flexibility via accepting disjunctive queries.
- $D_{\mathcal{G}}$ is undefined at $\alpha = 0$.
- For $\alpha = 2$, the isosurfaces of $D_{\mathcal{G}}$ become hyperspheres centered on the centroid of $\mathcal{G}$ [16]. This is reflected in Figure 2(b), which is identical to (a) except for α. Despite $\mathcal{G}$ corresponding to a disjunctive query, $\alpha = 2$ yields concave isosurfaces.

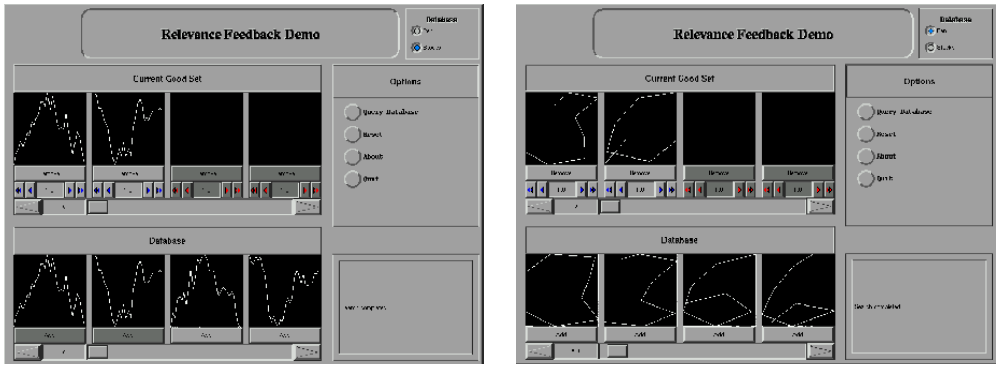
In addition, since $D_{\mathcal{G}}$ never directly uses features, FALCON is not intrinsically restricted to vector spaces; theoretically, it could be implemented using completely unrestricted spaces and sequential scanning. A more practical approach would be to use fast indexing structures; later in this paper we present algorithms for range and k-nearest-neighbor queries in $D_{\mathcal{G}}$ space, and these can take advantage of indices that support single-query-point versions of the same. Metric spaces – where we lack features, but instead have a symmetric distance function that obeys the triangle inequality – then become feasible with such indices as the M-tree [3] and Slim-tree [14].

2.2 User Feedback

Now that we have defined one major component $D_{\mathcal{G}}$, and therefore know how to rank examples, an obvious question is how FALCON incorporates user feedback.

Consider a given $\mathcal{G}$ and a pre-determined α. These define $D_{\mathcal{G}}$, and therefore provide a hybrid distance function for evaluating examples. For any arbitrary $k \in \mathcal{Z}^{+}$, we can retrieve and present the best k candidates to the user based upon $D_{\mathcal{G}}$.

The user may then change $\mathcal{G}$ by selecting one or more candidates to add to $\mathcal{G}$. Likewise, current members of $\mathcal{G}$ can be removed if the user changes his mind. If the weighted $D_{\mathcal{G}}$ variation is used, then the user should be able to alter weights for individual members of $\mathcal{G}$.



(a) on the STOCK dataset (b) on the PEN dataset

Figure 3: GUI of FALCON. The top row contains the "good" objects; the bottom row gives the top matches. The implicit query in (a) is for stock vectors that climbed sharply, then fell, or vice versa; in (b), we seek both 3's and 6's. Notice that FALCON handles these disjunctive queries quite well, returning examples of the desired sets instead of objects that each have slight resemblances to all good objects.

2.3 Algorithms and Scalability

Finding candidates for feedback in an interactive system using a non-trivial database clearly requires fast, scalable algorithms. Here, we describe algorithms for performing both range and k-nearest-neighbor queries in $D_{\mathcal{G}}$ space. The latter fits in naturally with the feedback model, and utilizes the former.

Both require range and k-nearest-neighbor queries in the original space, as defined by the pairwise distance function d. If d is a distance metric, metric structures such as M-trees [3] and Slim-trees [14] apply; for vector sets, there are legions of spatial indices with the required properties [6].

Given support for the single-point versions, we can derive versions that operate in $D_{\mathcal{G}}$ space using multiple query points as found in $\mathcal{G}$. These algorithms scale well and are provably correct. For reference, we label these two algorithms as the "MRQ" (Multiple Range Query) and "k-NN+MRQ" (k-Nearest Neighbor + Multiple Range Query) algorithms; details and proofs of correctness are in [17] and [16], respectively.

In a nutshell, the MRQ algorithm computes range-queries in $D_{\mathcal{G}}$ space by finding the union of standard range queries with the same desired radius for each of the "good" objects g_i, followed by filtering via $D_{\mathcal{G}}$. Similarly, the k-NN+MRQ algorithm uses multiple k-NN queries centered on each g_i. The results are then used to overestimate the $D_{\mathcal{G}}$ distance to the k^{th} neighbor; given this range, it applies the MRQ algorithm and selects the top k results as ranked by $D_{\mathcal{G}}$.

2.4 Prototype implementation

Figures 3(a) and (b) shows what a user might see. These screenshots show a prototype operating on two data sets, STOCK and PEN, both described later. The top row of panels shows $\mathcal{G}$, while the bottom row shows an excerpt from the database in whatever order is current – sorted according to the previous query; or the original order if no queries have been executed or the system has been reset.

In both cases, the queries are disjunctive; the STOCK example shows a request for stocks with curves resembling V's or inverted V's, while the PEN example requests both 3's and 6's.

3 Experiments

In this section, we present empirical support for the FALCON system. There are three obvious questions that pertain to relevance-feedback systems in general.

1. In terms of quality, how good are the candidates returned by FALCON?
2. In terms of iterations, how quick does the feedback loop converge?
3. In terms of speed, how scalable is FALCON?

In addition, since FALCON has one free parameter α, we ask whether or not there is an optimal value of α; and what values of α seem to perform acceptably well.

3.1 Evaluation Methodology

The testing cycle differs from the original flow in two areas, as shown in Figure 1(b).

The first is that feedback is that all instances have already been labeled by a script, rather than a user. Two major limitations are imposed. One is that only a subset of the data is available for feedback, which should accelerate convergence — preferably without severely limiting performance. To understand why, consider that a k-nearest-neighbor query may return a cluster of points close to each other, if such exist. A tight cluster is less likely to be useful for determining the underlying query concept than an equal number of well-separated points; and a sample should, in theory, increase the typical separation between candidates.

A second limitation imposed upon automated feedback is that no more than twenty previously unseen instances are selected per iteration. If any of these are positive instances, they get added to $\mathcal{G}$. Otherwise, the feedback loop terminates.

The second way in which the feedback loop differs is that the method computes precision/recall in order to quantitatively evaluate performance. For any level of recall $r \in [0, 1]$, a database of size n, and a total of n_p positive examples, there is a minimum number n_e such that the top-ranked t objects among the n total include at least $p = \lceil rn_p \rceil$ positive examples. Then, we can say that precision/recall is $\frac{p}{t}$ at r recall. Tracking precision/recall at multiple levels of recall over a series of iterations measures speed of convergence.

We also measured wall-clock time required for multi-query-point range and k-nearest-neighbor queries, which are critical to FALCON's performance. For reference, these tests were performed on a 400 MHz Intel Pentium II computer with 128 MB of RAM running Linux.

3.2 Data and Parameters

Five data sets were used, each paired with a different query. Four – two low-dimensional synthetic, two high-dimensional real – were used to measure precision/recall, convergence speed, and sensitivity to α. Real data received realistic queries; the two synthetic sets tested non-convex and disjunctive queries, respectively. Regarding α sensitivity, we tested with $\alpha \in \{-\infty, -100, -10, -5, -2, 2, 5\}$.

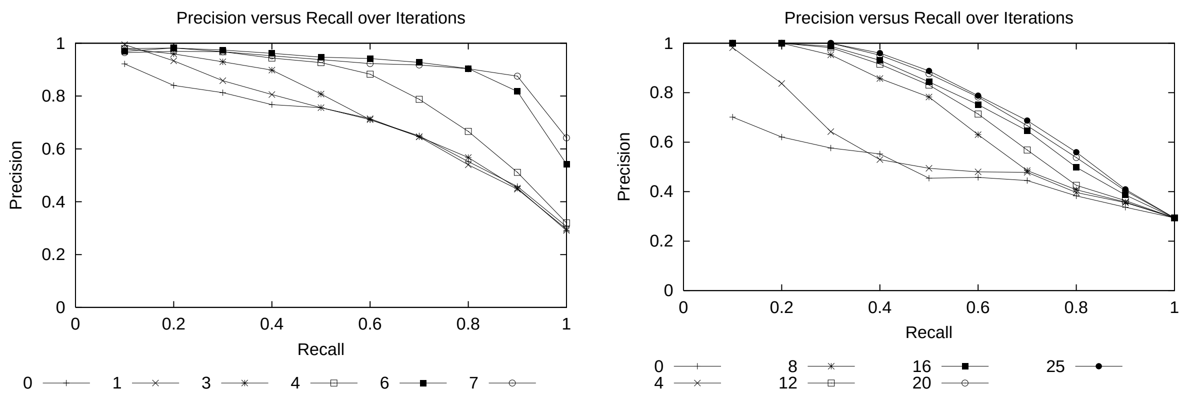
The fifth and largest data set was used for scalability testing. For this purpose, most tests used $\alpha = -5$; a few range-query tests were run with $\alpha = -\infty$, and are labeled accordingly.

2D_50K/RING: The 2D_50K data set has 50,000 points in 2-D Cartesian space, uniformly distributed within the orthogonally-aligned square (-2,-2)-(2,2). The RING query selects points that are between 0.5 and 1.5 units from the origin as measured by Euclidean distance; 19,734 qualify.

2D_20K/TWO_CIRCLES: The 2D_20K data set was generated identically to 2D_50K except that it has only 20,000 points. The TWO_CIRCLES query accepted points at most 0.5 units away in Euclidean distance from either (-1,-1) or (1,1); 1,899 qualified.

PEN/PEN_4: The Handwriting Recognition data set was found at the UCI repository [8]. Each of the 10,992 objects is a 16-dimensional vector corresponding to the scaled Cartesian coordinates of eight pen positions sampled in time as a test subject wrote a digit. The PEN_4 query selects vectors corresponding to handwritten 4's; 1,144 qualified.

STOCKS/FLAT_STOCKS: Fifty-one stocks were arbitrarily chosen and their closing prices obtained for up to the previous five years. These sequences were split into 1,856 non-overlapping vectors of length 32 and then subjected to a discrete wavelet transformation. The FLAT_STOCKS query accepted stocks with the slope within [-0.02, 0.02]; 540 qualified. The DWT results from five artificial, perfectly flat price vectors were used for the first $\mathcal{G}$.



TWO_CIRCLES FLAT_STOCKS

Figure 4: Precision/recall over iterations for two of the queries at $\alpha = -5$.

COVER: The forest cover database came from the UCI repository [8], and includes 581,012 54-dimensional vectors, with both discrete and continuous attributes. No split was used since the purpose was not to assess query accuracy, but instead to measure performance of a single multi-point query. Uniform random sampling without replacement was used to find $\mathcal{G}$'s of varying size.

3.3 Experimental Results

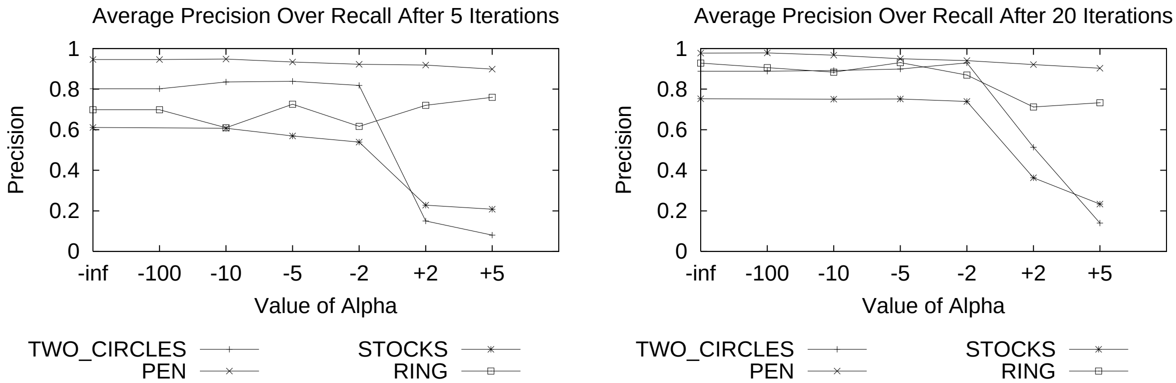
A summary of the results follows; for a more in-depth treatment, see [17, 16].

Values of α Accuracy testing showed that $\alpha = -2$ and $\alpha = -5$ were both reasonable values with $\alpha = -10$ not substantially inferior. With $-\infty$, the strict MIN function is too strict; and with positive values, disjunctive queries such as TWO_CIRCLES become impossible. No value was optimal for all sets; intuitively, this makes sense as there is little reason to expect a global optimum over the space of different queries.

Quality and Convergence Figure 4 shows precision/recall for TWO_CIRCLES and FLAT_STOCKS with $\alpha = -5$. The former completes in seven iterations, with excellent precision-recall; the latter nears its final values after 8-12 iterations, but marginal gains continue for a total of 25. These results are typical; a fairly low number of iterations generally resulted in good precision at low levels of recall early, with subsequent iterations converging for the higher levels of recall.

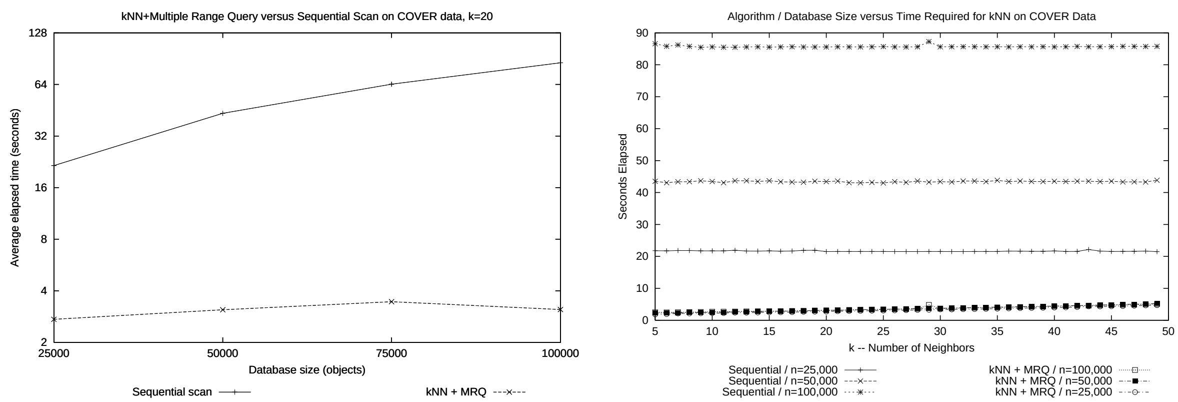
Figure 5 shows average precision/recall values for at multiple values of α with the four data sets used for accuracy testing. Averages were computed as the mean of precision at each level of recall from 10% to 100% at 10% intervals; in cases where 20 iterations were not required due to faster convergence, final values are presented. Notably, negative values of α yield better results, but otherwise the performance is rather insensitive to the exact value of α. In addition, average precision on a full database can be respectably high even after only 5 iterations of feedback on a randomly chosen subset.

Scalability Figure 6 shows that the k-NN+MRQ algorithm, when backed by an M-tree, performs very well compared to sequential scanning as the sample size chosen from COVER is varied from 25,000 to 100,000 and k goes from 5 to 50. In particular, k-nearest-neighbor performance is not nearly as affected by database cardinality as is sequential scanning. Clearly, the MRQ algorithm, as a significant part of the k-NN+MRQ algorithm, must itself scale accordingly.



After 5 iterations After 20 iterations

Figure 5: Average precision/recall after 5 and 20 iterations.



Cardinality scaling Dimensionality scaling

Figure 6: k-NN+MRQ performance testing with variable database size and variable k.

4 Conclusions

We have described FALCON, the first relevance feedback system that can handle disjunctive queries in multimedia databases. In addition, FALCON has the following desirable properties:

- It can be applied to metric datasets, too, in addition to vector ones.
- It reaches good precision and recall, after few iterations. For instance, with all queries, $\alpha = -5$ yielded at least 80% precision at 50% recall with 10 iterations.
- When backed by an indexing data structure, it scales well. For instance, k-NN+MRQ with $k = 50$ on a 100,000-object set was approximately four times as fast as a sequential scan with $k = 5$ on a 25,000-object set.

Acknowledgements

This research was partially funded by the National Science Foundation under Grants No. DMS-9873442, and IIS-9910606; also, by DARPA/ITO through Order F463, issued by ESC/ENS under contract N66001-97-C-851;

and the Office of Naval Research Grant N-00014-96-1-1222. Additional funding was provided by donations from NEC and Intel. Views and conclusions contained in this document are those of the authors and should not be interpreted as representing official policies, either expressed or implied, of the Defense Advanced Research Projects Agency or of the United States Government. All trademarks are property of their respective owners.

References

[1] Chris Buckley, Gerard Salton, and James Allan. The effect of adding relevance information in a relevance feedback environment. In *Proc. of SIGIR*, pages 292–300, 1994.

[2] M. Christel, T. Kanade, M. Mauldin, R. Reddy, M. Sirbu, S. Stevens, and H. Wactlar. Informedia digital video library. *Communication of the ACM*, 38(4):57–58, April 1995.

[3] Paolo Ciaccia, Marco Patella, and Pavel Zezula. M-tree: An efficient access method for similarity search in metric spaces. In Matthias Jarke, Michael J. Carey, Klaus R. Dittrich, Frederick H. Lochovsky, Pericles Loucopoulos, and Manfred A. Jeusfeld, editors, *Proc. of 23rd International Conference on Very Large Data Bases*, pages 426–435. Morgan Kaufmann, sep 1997.

[4] Ingemar J. Cox, Matt L. Miller, Stephen M. Omohundro, and Peter M. Yianilos. PicHunter: Bayesian relevance feedback for image retrieval. In *Proceedings of International Conference on Pattern Recognition*, Vianna, Austria, 1996.

[5] Ingemar J. Cox, Matthew L. Miller, Thomas P. Minka, and Peter N. Yianilos. An optimized interaction strategy for Bayesian relevance feedback. In *Proc. of IEEE Conf. on Comp. Vis. and Pattern Recognition*, pages 553–558, 1998.

[6] Volker Gaede and Oliver Günther. Multidimensional access methods. *ACM Computing Surveys*, 30(2):170–231, 1998.

[7] Yoshiharu Ishikawa, Ravishankar Subramanya, and Christos Faloutsos. Mindreader: Querying databases through multiple examples. In Ashish Gupta, Oded Shmueli, and Jennifer Widom, editors, *Proc. of 24rd International Conference on Very Large Data Bases*, pages 218–227. Morgan Kaufmann, aug 1998.

[8] P.M. Murphy and D.W. Aha. UCI Repository of Machine Learning Databases. Department of Information and Computer Science, University of California, Irvine, CA. [http://www.ics.uci.edu/~mlearn/MLRepository.html], 1994.

[9] Kriengkrai Prokaew, Sharad Mehrotra, Michael Ortega, and Kaushik Chakrabarti. Similarity search using multiple examples in mars. In *1999 International Conference on Visual Information Systems*, June 1999.

[10] Joseph John Rocchio. Relevance feedback in information retrieval. In Gerard Salton, editor, *The SMART Retrieval System – Experiments in Automatic Document Processing*, pages 313–323. Prentice Hall, Englewood Cliffs, N.J., 1971.

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

[12] Yong Rui, Thomas S. Huang, and Sharad Mehrotra. Content-based image retrieval with relevance feedback in MARS. In *Proceedings of IEEE International Conference on Image Processing '97*, Santa Barbara, CA, October 1997.

[13] Yong Rui, Thomas S. Huang, and Sharad Mehrotra. Human perception subjectivity and relevance feedback in multimedia information retrieval. In *Proceedings of IS&T and SPIE Storage and Retrieval of Image and Video Databases VI*, San Jose, CA, January 1998.

[14] Caetano Traina Jr., Agma J. M. Traina, Bernhard Seeger, and Christos Faloutsos. Slim-trees: High performance metric trees minimizing overlap between nodes. In Carlo Zaniolo, Peter C. Lockemann, Marc H. Scholl, and Torsten Grust, editors, *Proc. of 7th International Conference on Extending Database Technology*, volume 1777 of *Lecture Notes in Computer Science*, pages 51–65. Springer, mar 2000.

[15] Howard D. Wactlar, Takeo Kanade, and Michael A. Smith. Intelligent access to digital video: Informedia project. *IEEE Computer*, 29(5):45–52, May 1996.

[16] Leejay Wu, Christos Faloutsos, Katia Sycara, and Terry Payne. Falcon: Relevance feedback in metric spaces. Submitted to ACM TODS.

[17] Leejay Wu, Christos Faloutsos, Katia Sycara, and Terry R. Payne. Falcon: Feedback adaptive loop for content-based retrieval. In *Proceedings of the 26th International Conference on Very Large Databases*, pages 297–306, Cairo, Egypt, September 2000. VLDB.