Automated Multimedia Systems Training Using Association Rule Mining

Na Zhao[1], Shu-Ching Chen[1], Stuart H. Rubin[2]
[1]*Distributed Multimedia Information System Laboratory*
School of Computing and Information Sciences
Florida International University, Miami, FL 33199, USA
[2]*SPAWAR Systems Center (SSC)*
Code 2734, 53560 Hull Street, San Diego, CA 92152-5001, USA
[1]*{nzhao002, chens}@cs.fiu.edu,* [2]*stuart.rubin@navy.mil*

Abstract

User feedback is widely deployed in recent multimedia research to refine retrieval performance. However, most of the existing online learning algorithms handle interactions of a single user, which may pose restrained performance due to the limited size of positive feedback. An alternative solution is to learn general user perceptions via collecting feedback from different users. The training process is initiated only when the number of feedbacks reaches a certain threshold. This could improve the performance but it becomes a manual process to decide the threshold and initiate the training process. To address this challenge, we propose an advanced training method by adopting the association rule mining technique, which can effectively evaluate accumulated feedback and automatically invoke the training process. Training is performed per video rather than for all videos in the database, making the process more efficient and robust. In addition, it can further improve semantic modeling in the video database and continuously improve retrieval performance in the long run. As an example, the proposed method is applied to a mobile-based soccer video retrieval system and the experimental results are analyzed.

1. Introduction

Semantic retrieval of media objects may well extend textual consequents to include all forms of multimedia. However, it is a challenging task for a multimedia system to perform content-based retrieval on multi-dimensional audio/visual data; while it is even harder to refine the retrieval results iteratively and interactively based on user preferences.

Users are usually interested in specific semantic concepts and/or the associated temporal-based event patterns when querying a large scale video archive. In this study, a temporal event pattern is defined as a series of semantic events with some particular temporal relations. For instance, in soccer video retrieval, an example temporal event pattern query can be expressed as “Search for those soccer video segments where a goal results from a free kick.”

Using the algorithm we have proposed in [13], the system should be able to search for the video clips, which contain the desired pattern and rank them based on a certain similarity measurement method. However, not all of the returned video clips will be chosen by the user as positive results. The possible reasons are (i) some video clips may not exactly match the requested events due to the accuracy constraints of the automatic event annotation algorithm, and (ii) though some video clips match the correct event pattern, they do not satisfy user’s particular interests. Furthermore, the ranking may not reflect the user expectations initially. Thus, the system should allow user feedback and learn from them to filter out inaccurate results as well as refine searching & ranking performance.

In this paper, the association-rule mining (ARM) technique [1, 2] is applied to automate the training process. The automated training process has the following advantages. First, the multimedia system is updated to check the threshold effectively in real time and initiate the training automatically using accumulated user feedback. In other words, no manual process is required. Second, the overall training process becomes more efficient and robust since only part of the video database that contains enough historical retrieval data and positive patterns needs to be updated. Finally, the training process can further improve semantic video database modeling and continuously improve system retrieval and ranking performance in the long run.

The rest of this paper is organized as follows. Section 2 reviews research approaches in multimedia retrieval and system learning. In Section 3, the proposed methodology is presented with more detailed technical discussions in video database modeling, ARM, and the automatic system training method in Section 4. Section 5 demonstrates the implementation and analyzes the experimental results. Finally, conclusions and a summary are presented in Section 6.

1-4244-1500-4/07/$25.00 ©2007 IEEE

2. Related Work

One of the most challenging tasks in multimedia information retrieval is to perform the training and learning process such that the retrieval performance of the multimedia search engine can be refined efficiently and continuously. In general, existing multimedia system training and learning mechanisms can be categorized into online learning and offline training.

Relevance feedback (RF) [9] is designed to bridge the semantic gap and provide more accurate results based on the user’s responses. Incorporating RF is an online solution for improving retrieval accuracy, especially for still-image applications. However, existing RF approaches have several constraints and limitations such that it is difficult to employ RF in video retrieval approaches. For example, it does not incorporate any methodology to model all layers of multimedia objects and consequently it does not offer efficient learning for multimodal video retrieval to satisfy general users’ interests. In addition, as mentioned by Muneesawang and Guan [8], RF does not offer a decent solution for video database representation to incorporate sequential information for analytic purposes.

Research efforts have been conducted to extend and refine the RF method for video retrieval and learning purposes. Several multimedia system training approaches try to utilize other possible learning mechanisms such as Support Vector Machine (SVM) and Neural Network techniques. For example, a template frequency model was proposed and a self-learning neural network was employed to implement an automatic RF scheme by Muneesawang and Guan [8]. Yan et al. [10] describes a negative pseudo-relevance feedback (NPRF) approach to extract information from the retrieved items that are not similar to the query items. Unlike the canonical RF approach, NPRF does not require the users to make judgment in the retrieval process as the negative examples can be obtained from the worst matching results. In Bruno et al. [5], a query-based dissimilarity space (QDS) was proposed to cope with the asymmetrical classification problem with query-by-examples (QBE) and RF, where system learning in QDS is completed through a simple linear SVM. However, this linear based method failed to satisfy the complicated requirements for content based video retrieval and learning. Therefore, more sophisticated strategies should be considered.

For the offline training algorithms, the current research mainly focuses on one-time training using certain kind of data sets or classification information. For some cases, user feedback is not the major data source in system training. For instance, Hertz et al. [7] introduced a learning approach using the form of equivalence constraints which determine whether two data points come from the same class. It provides the relational information about the labels of data points, rather than the labels themselves. An automatic video retrieval approach was proposed by Yan et al. [11] for the queries that can be mapped into four predefined user queries, e.g. named persons, named objects, general objects, and scenes. It learned the query-class dependent weights utilized in retrieval offline. This kind of offline training processes is time-consuming, not fully automatic, and limited to the pre-defined query types.

In summary, most of the current online learning algorithms mainly deal with interactions with a single user. Due to the small amount of feedback, they can be performed in real-time, but the performance could only be improved in a limited degree, especially when handling a large-scale multimedia database. On the other hand, some offline training methods try to learn the knowledge from not only collected user feedback, but also some other training data sources. The performance could be better as it considers more training data, but the major drawback is that a manual process needs to be executed to initiate the training process. Moreover, since training is performed through the entire multimedia database, it becomes a tedious task and can only run offline.

3. Overall Process

In this study, an advanced multimedia system learning approach is proposed to make improvements on most of the abovementioned problems. As we have presented in [12][13], our proposed system training solution can fully consider the general user interests as well as help on the long-term continuous improvement on video retrieval performance. In particular, this paper enhances the training process to become more automatic and efficient; while the database model can be trained in a robust and effective manner.

Figure 1 shows the overall process of the proposed method. When a user issues a query pattern, the background server executes the query and ranking process such that the system can return to the user with the ranked video clips that match the query pattern. The user is allowed to choose his/her favorite video clips as positive patterns and issue the feedback. The server engine receives the feedback and accumulates them in the video database. The system then checks if the number of new feedbacks reaches a pre-defined threshold. If yes, a background checking mechanism is invoked automatically to evaluate the feedback using ARM. Otherwise, the next round query is performed. For efficiency purposes, the system will check if there is any video containing enough positive patterns. If no video satisfies the qualification, then the training process will not proceed. Otherwise, the system will initiate the training process on certain video(s) based on the evaluation results.

After the training, the positive patterns used in training the video database model will be removed from the untrained feedback data set, and accordingly, new counts begin for the next query. After the underlying video database has been updated, all the users can have the opportunity to achieve the refined ranked results based on the trained video database models.

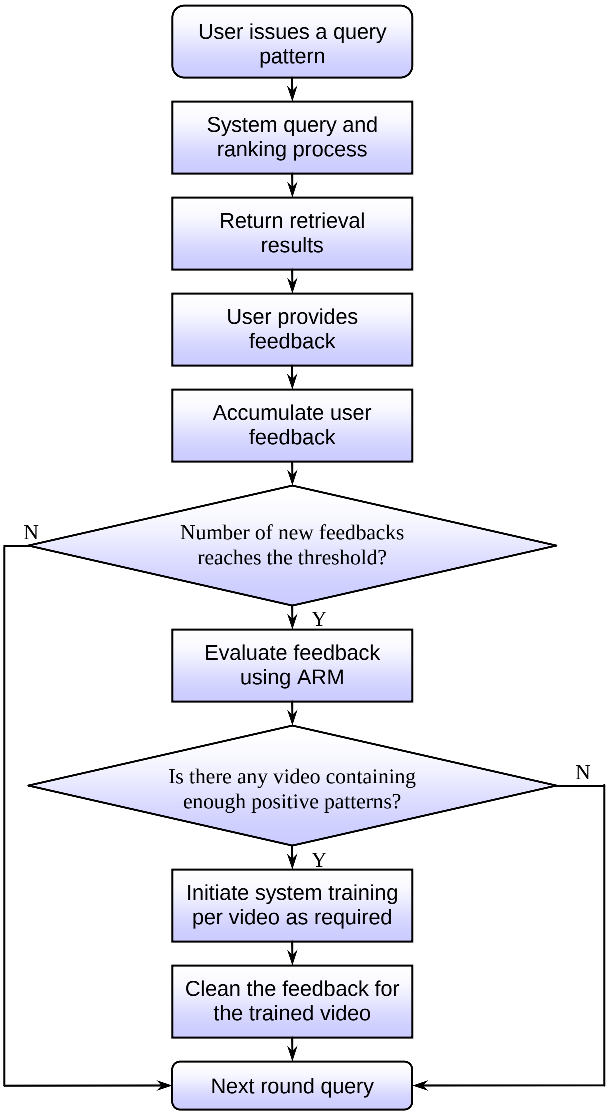

User issues a query pattern
System query and ranking process
Return retrieval results
User provides feedback
Accumulate user feedback
Number of new feedbacks reaches the threshold?
N
Y
Evaluate feedback using ARM
Is there any video containing enough positive patterns?
N
Y
Initiate system training per video as required
Clean the feedback for the trained video
Next round query

Figure 1. Overall process for the automated training

Although this figure only shows the process sequence for one user, the system actually collects feedbacks from multiple users. Only one evaluation measure is calculated for each video and updated with the accumulated feedback from the common users. Of course, the mutual exclusive issue should be considered such that when the feedback from one user invokes system training for a certain video, the thread which processes the request from another user should be aware of this situation and certain actions should be restricted to avoid any conflicts.

Besides, it is worth mentioning that this proposed framework can also be easily adjusted and applied into image retrieval applications. The basic idea is to evaluate if there are enough positive image patterns in a local image database. That is, the training process is performed on independent local image databases rather than the overall image repository.

4. Automated Multimedia System Training

4.1. Video Database Modeling

In order to satisfy the requirements of hierarchical video database modeling, the Hierarchical Markov Model Mediator (HMMM) [12, 13] was proposed to manage a series of multimedia objects, including videos, video shots, and clusters. Here, an MMM (Markov Model Mediator) model is deployed to model a media object consisting of a set of children units through multimodal descriptions.

Table 1. An HMMM is defined as $\Lambda = (d, \{\lambda_i^j\}, \mathbf{O}, \mathbf{L})$.

Symbol	Descriptions
d	Number of levels in HMMM, e.g., d=3.
λ_i^j	$\{S_i^j, F_i^j, A_i^j, B_i^j, \Pi_i^j\}$, the jth MMM in the ith level.
$\mathbf{O}$	$\{O_{i,i+1}\}$, mportance weights for the ith level feature when describing $(i+1)$th feature concepts, where i=1,…,d-1.
$\mathbf{L}$	$\{L_{i,i+1}\}$, link conditions between the higher level states and the lower level states, where i=1, …, d-1.

Table 2. A general MMM $\lambda = \{s, f, \alpha, \beta, \pi\}$.

Symbol	Descriptions
s	Set of states (media objects).
f	Set of features or semantic concepts.
α	State transition probability distribution ($\alpha \to s \times s$). Indicates the affinity relationships among media objects. The higher the entry is, the tighter the relationship exists.
β	Observation symbol probability distribution ($\beta \to s \times f$). Represents the feature values or semantic concepts of media objects.
π	Initial state probability distribution. Indicates the likelihood of a media object being selected as the query.

In this research, the video database is modeled using a 3-level HMMM model, where each level incorporates one or more MMM models which describe the features and relationships among the same series of media objects. For example, a video can be modeled using a 1st-level MMM model, where the segmented video shots are treated as the states in this MMM. In the meanwhile, the video is also modeled as one state in the 2nd-level MMM model which represents a video cluster. The initialization of all these matrices can be found in [12, 13].

4.2. System Training

The event pattern query can be formalized as follows. Given a temporal pattern with C events $Q=\{e_1, e_2, ..., e_C\}$ sorted by the temporal relationships such that $T_{e_1} \leq T_{e_2} \leq ... \leq T_{e_C}$, the system is requested to search for the video clips that containing pattern Q. A set of temporal patterns are retrieved and G of them $\{R_1, R_2, ..., R_G\}$ are marked as "Positive" by the user. Here, R_k represents the kth "Positive" pattern, 1<k<G.

4.2.1. Offline Training without ARM

For the sake of effectiveness, all user access patterns and access frequencies, during a training period, should be utilized to train the underlying HMMM model. The system should collect the feedbacks specified by the users in the online learning process. Once the number of new feedbacks reaches a certain threshold, the system should update both the state transition probability distribution feature matrix and initial state probability distribution matrix. All of the calculations are executed offline in the background server.

The calculations of two affinity relationship matrices A_1 (for the video shots) and A_2 (for the videos) are similar except that they use different training data sets collected from MMMs in different levels. In this research, we mainly introduce the update methods for A_1, while more complete descriptions can be found at [6, 13]. The AF_1 matrix is defined to capture the temporal-based affinity relationships among all the video shots using user access patterns (use_1) and access frequencies ($access_1$). For the kth pattern R_k,

- $access_1(k)$ represents its access frequencies, and
- $use_1(i, k)$ = 1 if s_i (the ith video shot) was accessed in pattern R_k, and $use_1(i, k)$ = 0 otherwise.

Moreover, both s_m and s_n should belong to the "Positive" temporal pattern R_k and follow certain temporal sequence. That is, s_m should occur on or before s_n. Let G be the number of "Positive" patterns on the shot level, the entries $aff_1(m,n)$ in matrix AF_1 can be calculated as shown in Equation (1). A_1 can then be updated via normalizing AF_1 per row and thus MMM represents the relative affinity relationships among all the video shots in the MMM model of this particular video.

$$aff_1(m,n) = A_1(m,n) \times \sum_{k=1}^{G} use_1(m,k) \times use_1(n,k) \times access_1(k), \quad (1)$$
$$iff \;\; s_m \in R_k,\ s_n \in R_k,\ T_{s_m} \leq T_{s_n}$$

$$A_1(m,n) = \frac{aff_1(m,n)}{\sum_{j=1}^{N} aff_1(m,j)}, \quad (2)$$

where 1≤m≤N and 1≤n≤N.

Let $|S_i|$ represent the number of states in the ith MMM, the initial state probability matrices can be updated as follows.

$$\Pi_i = \{\pi_m\} = \frac{\sum_{k=1}^{|S_i|} use_i(m,k)}{\sum_{l \in S_i} \sum_{k=1}^{|S_i|} use_i(l,k)}, \quad (3)$$

where 1≤i≤2 and $1 \leq m \leq |S_i|$.

4.2.2. Automated Training with ARM

The challenge for such a multimedia training process is to determine a suitable threshold value to invoke model re-training for a video v. Due to the fact that the support measure used in ARM [1, 2] can well capture the percentage of data tuples for which the pattern is true, we investigate how to best adopt this concept for the purpose of inspecting whether the underlying HMMM model for a particular video v needs to be re-trained.

As first introduced by Agrawal et al. [1], ARM is designed to discover items that co-occur frequently within a data set. Given a set of transactions in market basket analysis applications, where each transaction contains a set of items, an association rule is defined as an expression $X \Rightarrow Y$, where X and Y are sets of items and $X \cap Y$ = Ø. The rule implies that the transactions of the database containing X tends to also contain Y. In ARM, the *support* constraint concerns the number of transactions that support a rule. The *support* value is defined to be the fraction of transactions that satisfy the union of items in the consequent and antecedent of the rule.

This idea can be mapped and applied to mine the association rules in the positive feedback. Here, each positive event pattern is treated as a transaction, and the historical access pattern database is defined as the set of all transactions. To satisfy our requirements, we modify the definition of target rules and define them as two itemset association rules which follow certain temporal sequences. For example, $s_m \Rightarrow s_n$ can be treated as a target rule, where s_m and s_n are video shots in video v and

$T_{s_m} < T_{s_n}$. Accordingly, the *support* measure can be defined as below:

$$Support\ (m,n) = \frac{Count\ (s_m \Rightarrow s_n)}{NumTrans}, \qquad (4)$$

where $Count(s_m \Rightarrow s_n)$ returns the number of positive event patterns that contain the rule of $s_m \Rightarrow s_n$, and *NumTrans* represents the total number of all temporal event patterns (transactions) in the data set of positive feedbacks which were not used in the previous training process.

In this application, we are more concerned with the number of all rules in a certain video than the number of a specific rule. For a given video *v*, we can sum all the counts for the identified temporal rules to get this number as $\sum_{s_m}\sum_{s_n} Count(s_m \Rightarrow s_n)$, which can also be represented as $\sum_{s_m}\sum_{s_n} Support\,(m,n) \times NumTrans$.

In our video retrieval and training system, a novel means for representing the percentage of s_m and s_n in video *v* that are accessed in the positive pattern R_k with $T_{s_m} < T_{s_n}$ can be utilized to define an "*Evaluation*" measurement for each video for checking purposes.

$$Evaluation\ (v) = \frac{\sum_{s_m}\sum_{s_n} Support\ (m,n) \times TotalNum}{\sum_{s_m}\sum_{s_n} aff_1(m,n)} \qquad (5)$$

Equation (5) captures the percentage of s_m and s_n appearing in the positive temporal patterns versus to the overall affinity relationship between them. If this percentage reaches a certain value, it indicates that such a relationship should be reflected more frequently in the model training process. Here, the threshold is defined as *H* to see if the video is ready for the next round of training.

- When $Evaluation\ (v) < H$:

  The database model for video *v* will not be trained and the feedback is simply accumulated in the server side.

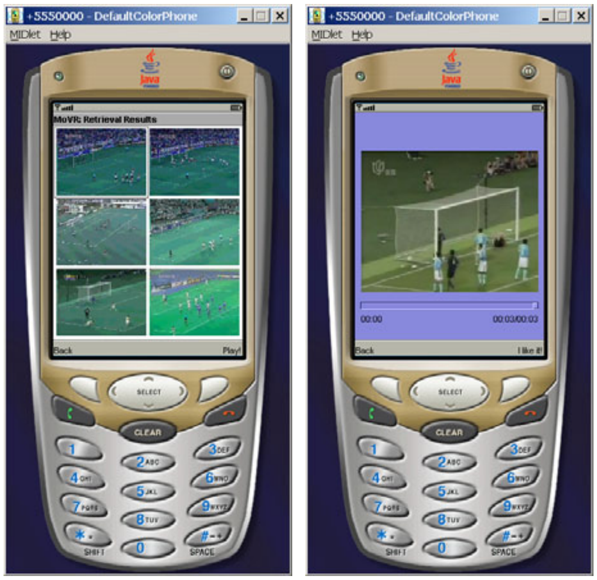
- When $Evaluation\ (v) \geq H$:

  $$aff_1\ (m,n) = A_1(m,n) \times Support\,(m,n) \times NumTrans, \quad iff\ \ s_m \in R_k,\ s_n \in R_k,\ T_{s_m} \leq T_{s_n} \qquad (6)$$

  The *aff* values are then utilized for updating the corresponding affinity relationship matrix and initial state probability matrix for the particular video *v*.

5. Implementation and Experiments

The proposed approach is applied to a distributed multimedia system environment with simulated mobile clients, which was first proposed in [12]. As shown in Figure 1, after a user issues the event pattern query, the system will search, rank the results, and return the key frames to the user. Due to the limited size of wireless devices, each screen is designed to show up to 6 candidate video clips as presented in Figure 2(a). By clicking the user preferred key frame, the corresponding video segment will be displayed as shown in Figure 2(b) and the user can provide positive feedback by using the upper right button to trigger the choice of "I like it!".



(a) (b)

Figure 2. System interfaces for the Mobile based Video Retrieval System

For ARM, we utilize the source codes for the Apriori algorithm from [14], which provides an efficient program (Borgelt et al. [3][4]) to find association rules and frequent itemsets with the Apriori algorithm. Apriori is designed to operate on the data sets containing transactions (i.e., the positive event patterns in the historical feedback). The current system contains totally 45 videos and around 10,000 video shots. The ARM based system evaluation results are recorded in Table 3. Initially, the system performs ARM based evaluation every 50 historical queries. However, it seems that the knowledge captured in the first 50 historical queries is not enough and no video needs to be trained. When it reaches 100 historical queries, more association rules are discovered. For example, there are totally 240 distinct items (positive video shots), 588 transactions (positive event patterns), 44 identified association rules, and accordingly 1 video passing the evaluation threshold. That is, the database model of this video is trained separately. After that, all the positive feedback patterns that are in this trained video are removed from the unused feedback dataset. Then, the system starts ARM-based evaluations per 100 new feedbacks. The same procedures are applied when the numbers of queries reaches 200 and 300, where 3 videos and 2 videos are trained respectively.

The system is designed to conduct training per video, rather than for all the videos in the database. We believe such a design can improve semantic modeling in the second layer and lead to further improvements in the overall retrieval performance. For example, based on 200 historical queries, the system evaluates all videos and determines that 3 videos need to be trained. The historical data that are already utilized for the training process will then be excluded from the next round of evaluation. When the number of historical queries reaches 300, the database models for 2 more videos are required to be trained. Comparing with system training which needs to update the MMM models for 45 videos, the proposed approach reduces the training time by approximately 900 percent, while achieving a similar degree of performance improvement.

Table 3. Experimental results for ARM-based feedback evaluations

Num of Historical Queries	Items	Patterns (Transactions)	Num of All Rules	Videos need to be trained
50	149	286	18	0
100	240	588	44	1
200	268	1069	196	3
300	274	1559	185	2

6. Conclusions

In this paper, an approach that can automate system training by evaluating user feedback in real time is proposed. The proposed approach utilizes the HMMM mechanism coupled with the ARM-based feedback evaluations to support both offline training and online learning and eliminates the need for manually initiating the training process. By utilizing the proposed evaluation measurement, the training process can be automatically triggered to update only those videos that contain enough positive event patterns. The learning and refining mechanism then becomes more effective when dealing with the general user perceptions. Furthermore, as the whole process is run at the backend server, it can remain transparent to general users while the system performance is continuously improved.

7. Acknowledgements

Shu-Ching Chen's research was supported in part by NSF EIA-0220562 and HRD-0317692. Stuart Rubin's research was supported in part by an SSC S&T initiative. Na Zhao's research was supported in part by Florida International University Dissertation Year Fellowship.

8. References

[1] R. Agrawal, T. Imielinski, and A. Swami, "Mining Association Rules between Sets of Items in Large Databases," In *Proc. of the ACM SIGMOD International Conference on Management of Data (SIGMOD)*, USA, 1993, pp. 207–216.

[2] R. Agrawal and R. Srikant, "Fast Algorithms for Mining Association Rules," in *Proc. of 20th International Conference Very Large Data Bases (VLDB)*, 1994, pp.487–499.

[3] C. Borgelt, "Efficient Implementations of Apriori and Eclat," In *Proc. of 1st Workshop of Frequent Item Set Mining Implementations (FIMI)*, USA, 2003.

[4] C. Borgelt and R. Kruse, "Induction of Association Rules: Apriori Implementation," In *Proc. of 15th Conference on Computational Statistics (Compstat)*, Germany, 2002.

[5] E. Bruno, N. Moënne-Loccoz, and S. Marchand-Maillet, "Asymmetric Learning and Dissimilarity Spaces for Content-based Retrieval," in *Proc. of International Conference on Image and Video Retrieval (CIVR)*, USA, 2006, pp. 330–339.

[6] S.-C. Chen, N. Zhao, and M.-L. Shyu, "Modeling Semantic Concepts and User Preferences in Content-Based Video Retrieval," accepted for publication, *International Journal of Semantic Computing*.

[7] T. Hertz, N. Shental, A. Bar-Hillel, and D. Weinshall, "Enhancing Image and Video Retrieval: Learning via Equivalence Constraints," in *Proc. of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2003, pp. 668-674.

[8] P. Muneesawang and L. Guan, "Automatic Relevance Feedback for Video Retrieval," in *Proc. of IEEE International Conference on Acoustics, Speech, and Signal Processing*, Hong Kong, 2003, vol. 3, pp. 1–4.

[9] Y. Rui, T. S. Huang, M. Ortega and S. Mehrotra, "Relevance Feedback: A Power Tool for Interactive Content-based Image Retrieval," *IEEE Trans. On Circuit and Video Technology*, vol. 8, no. 5, 1998, pp. 644–655.

[10] R. Yan, A. G. Hauptmann, and R. Jin, "Negative Pseudo-Relevance Feedback in Content-based Video Retrieval," In *Proc. of ACM Multimedia*, USA, 2003, pp. 343–346.

[11] R. Yan, J. Yang, and A. Hauptmann, "Learning Query-Class Dependent Weights in Automatic Video Retrieval," In Proc. of *ACM Multimedia 2004*, USA, 2004, pp. 548–555.

[12] N. Zhao, M. Chen, S.-C. Chen, and M.-L. Shyu, "User Adaptive Video Retrieval on Mobile Devices," accepted for publication, *Mobile Intelligence: When Computational Intelligence Meets Mobile Paradigm*, John Wiley & Sons Inc, 2007.

[13] N. Zhao, S.-C. Chen, and M.-L. Shyu, "Video Database Modeling and Temporal Pattern Retrieval using Hierarchical Markov Model Mediator," In *Proc. of the First IEEE International Workshop on Multimedia Databases and Data Management (IEEE-MDDM), in conjunction with IEEE International Conference on Data Engineering (ICDE)*, April 8, 2006, Atlanta, Georgia, USA, p. 10.

[14] Source code of Apriori algorithm (written in C) for frequent item set mining/association rule induction. http://fuzzy.cs.uni-magdeburg.de/~borgelt/software.html#assoc