GeM-Tree: Towards a Generalized Multidimensional Index Structure Supporting Image and Video Retrieval

Kasturi Chatterjee and Shu-Ching Chen
School of Computing and Information Sciences
Florida International University, Miami, FL 33199, USA
{kchat001, chens}@cs.fiu.edu

Abstract

In this paper, we propose a tree-based multidimensional structure, GeM-Tree, which indexes both images and videos within a single general framework utilizing Earth Mover's Distance. It can support different content-based image and video retrieval approaches, and can accommodate applications where the cross-similarity between images and videos need to be considered during content-based retrievals. Furthermore, it is flexible enough to index different video classification units and can maintain the hierarchical relationship between them. In addition, it uses a construct called Hierarchical Markov Model Mediator to introduce high-level semantic relationships among images and different levels of video units. The experimental results indicate that GeM-Tree is a promising generalized index structure for multimedia data with low computational overhead, is flexible enough to support different retrieval approaches and generates query results with high relevance.

1. Introduction

As it is said "a picture speaks a thousand words", multimedia data like images and videos carry more information than traditional text-based data. The popularity of such data in applications like social networks, bit torrent libraries, search-engines, etc., soared rapidly in the recent years. But, carrying that extra information makes multimedia data complex and need special representation, storage, and retrieval mechanisms. Multimedia data is generally represented with a multidimensional feature vector and has semantic information attached to it. Traditional database management is incapable of accommodating the multidimensional representation and high-level semantic information efficiently. Thus, the necessity of a multimedia database management system, comparable in robustness and capability to a traditional text-based database framework, became strong.

Index structures are an integral part of any database management design. Thus, researches were performed to come up with different multidimensional index structures like [1][2][3] to accommodate the multimedia data. The high level semantic relationship was introduced in the multidimensional index structures in [4][5]. Videos are considered to be more complex than images as videos are further divided into several units like frames, shots, etc. and can carry much semantic information within one shot or one video unit. An index structure for videos needs to be capable of storing the different video units and should be able to answer different levels of similarity queries. [1][2][3] were not able to handle such hierarchical structural nature of video data and answer different levels of content-based video retrieval queries. Thus [6] was developed to lay down a multidimensional tree-based index structure which can store and access videos efficiently for different levels of queries like frame-level, shot-level or entire video-level.

Though multidimensional index structures for multimedia data were developed, they were separate structures dedicated for either images or videos alone. To the best of our knowledge, there was no attempt to develop a common framework which can index images as well as videos and handle both content-based image and content-based video retrieval with the same efficiency. Having separate index structures for different types of multimedia data poses two major problems. First, integrating an index structure into a database kernel needs the modification of the Query Optimizer, Query Processor, SQL Compiler/Interpreter etc. in order to tune the performance of the components of a database management system with the corresponding index structure to be embedded. The process itself is complicated, tricky and time consuming [7][8]. Thus, modifying the database kernel components to support multiple different index structures and access methods is not a welcoming idea and might have performance conflict issues (for example, modifying a Query Optimizer for one particular index type for a certain multimedia object

978-0-7695-3454-1/08 $25.00 © 2008 IEEE
DOI 10.1109/ISM.2008.96

might have a degrading effect on the performance of another index structure for a different multimedia data object). Second, for some applications (like multimedia concept search) where the similarity between videos and images needs to be determined to answer queries, having separate index structures is inconvenient and inefficient.

In this paper, we propose a distance-based multidimensional tree-based structure called the Generalized Multimedia Tree (GeM-Tree) which provides a general indexing framework for images as well as videos. It uses Earth Mover's Distance (EMD) [9] as the distance function to calculate the (dis)similarity among the multimedia data objects in a metric space. To capture and utilize the high-level semantic relationships among the multimedia data objects and to introduce the relationships between the different levels of video units, we utilized a probabilistic mathematical construct called the Hierarchical Markov Model Mediator. Further, we introduce a flexible k-NN based similarity search algorithm that can support different techniques of content-based image and video retrievals while considering the high-level semantic relationships. Though EMD was used as a distance function in VP-tree [10], a distance based metric tree, to develop VP-EMD tree [11]., VP-EMD tree doesn't have the capability to index videos and was meant to serve as an index structure supporting only content-based image retrieval for feature sets with variable lengths. Also, VP-Tree is not a balanced structure as it is built in a top-down fashion. GeM-Tree is a balanced structure as it is built from the bottom following an approach similar to M-Tree [2].

The rest of the paper is organized as follows. In Section 2, we provide a brief discussion about EMD. In Section 3, the framework of the GeM-Tree is laid down. Section 4 discusses the k-NN based similarity search algorithm for images and videos. Section 5 presents the experimental results and analysis which is followed by Section 6 that presents the conclusion.

2. Earth Mover's Distance

The Earth Mover's Distance (EMD) [9] is a general and flexible distance function between two distributions and is based on deriving the minimum cost that must be paid to transform one distribution to another. It was derived from the transportation problem viz. the Monge-Kantorovich Problem [12] which determines the minimum cost of transporting goods from a set of *m* sources or suppliers to a set of *n* destinations of demanders.

To use a EMD function, a multimedia object is represented as a *signature* or a *finite distribution x* as $x = \{(x_1, w_1), (x_2, w_2), \ldots\ldots, (x_n, w_m)\} \equiv (X, w) \in D^{K,m}$, where $X = [x_1, x_2, \ldots\ldots, x_m] \in R^{Kxm}$ and *m* is the number of points. Given two distributions $x = (X, w) \in D^{K,m}$ and $y = (Y, u) \in D^{K,n}$, a flow between x and y is a matrix $F = (f_{ij}) \in R^{mxn}$. The main approach is to find a flow between x and y that minimizes the overall cost of $Work(x, y, F) = \sum_{i=1}^{m}\sum_{j=i}^{n} d_{ij} f_{ij}$. Four conditions need to be satisfied by *f* viz. $f_{ij} \geq 0$, $\sum_{j=1}^{n} f_{ij} \leq w_x$, $\sum_{i=1}^{m} f_{ij} \leq w_{y_j}$ and $\sum_{i=1}^{m}\sum_{i=1}^{n} f_{ij} = \min(\sum_{i=1}^{m} w_{x_i}, \sum_{j=1}^{n} w_{y_j})$, where w_x and w_y are the weights of the distributions of *x* and *y*, respectively and $1 \leq i \leq m$ and $1 \leq j \leq n$. An optimal flow *F* is calculated using the transportation problem and EMD is defined as $EMD(x, y) = \left(\sum_{i=1}^{m}\sum_{j=i}^{n} d_{ij} f_{ij}\right) \Big/ \left(\sum_{i=1}^{m}\sum_{j=i}^{n} f_{ij}\right)$. $d_{ij} = d(x_i, y_j)$ is the ground distance between x_i and y_j. EMD is a metric i.e., it follows the laws of symmetry, positivity, and triangular inequality when the total weights of the distributions are equal and the ground distance is a metric [9].

3. GeM-Tree

GeM-Tree is a distance based multidimensional balanced index structure providing a common platform to organize both images and videos based on their (dis)similarity with one another in a metric space. We used EMD as the distance function to build GeM-Tree by utilizing the (dis)similarity between the multimedia data objects in a R^q metric space, where *q* is the total number of features used to represent a multimedia object (an image or any video unit). We chose Euclidean Distance function (L_2) as the ground distance to keep EMD metric. The main benefits of utilizing EMD as a distance function are its capabilities of calculating the (dis)similarity between variable size distributions [9] allowing for partial matches, and its ability to better match perceptual (dis)similarity [9] by providing the flexibility to use different approaches of content based retrieval like region-based [13] methods. The EMD has been used to measure image similarity with respect to color and texture [9][13], but to the best of our knowledge, EMD was not previously utilized to calculate video similarities or to determine the relationship between two different types of multimedia objects like between an image and a video. In this paper, we demonstrated the capability of GeM-Tree to support content based image and video retrieval approaches which treat the entire image or the frames of a video as a single region or class (as discussed in Section 3.1), but it should be pointed out here that GeM-Tree is perfectly capable of supporting other approaches of content based retrievals with different feature/signature representation with equal efficiency.

3.1. Multimedia Data Signatures

A multimodal approach (i.e., both visual and audio features) is adopted. We extracted 19 visual features consisting of color (HSV color space) and texture information from each image and each frame of each video and averaged them over the collection of frames depicting a shot. In addition, we extracted 19 features dedicated to video data from the frames to capture the sequential relationships among the collection of frames that share a common semantic idea or a camera angle and have been tagged together as a shot utilizing [15]. We represent a multimedia object in the form of feature distributions (also can be called *signatures*) which can be divided into three sub-distributions for sake of clear representation as $F_A=\{x_1,x_2,\ldots\ldots,x_i\}$, $F_B=\{y_1,y_2,\ldots\ldots,y_j\}$ and $F_C=\{object\_id, v\_id, s\_id\}$. The feature vector representing the distribution of each multimedia data object is a union of the three sub-distributions $F=\{F_A U F_B U F_C\}$. F_A represents the color features in the HSV space and the texture features. F_B represents the visual and audio features related to videos only, and F_C captures the id information required to distinguish between the different object types and also captures the hierarchical relationship between them (if any). For example, *object_id* is the identification number of the particular multimedia data object, *v_id* stores the *object_id* of the video data object of which a particular frame or a shot is a part and *s_id* stores the *object_id* of the shot of which a frame is a part. By manipulating *v_id* and *s*_id, all the different multimedia data objects and their hierarchical relationships can be captured and represented efficiently. All the feature values are normalized using a [0, 1] norm. A single class of distribution is used for each signature and a fixed weight of 1 is assigned to each signature to keep EMD a metric.

3.1. Node Structures of GeM-Tree

GeM-Tree has two main node types, the leaf nodes storing the actual indexed multimedia data objects and the intermediate nodes which maintain the sub-tree structure within a tree. Further, depending upon the *signature* of the particular multimedia data object that the intermediate or the leaf nodes are storing, they can be subdivided into *image_intermediate* and *image_leaf* nodes, *frame_intermediate* and *frame_leaf* nodes, *shot_intermediate* and *shot_leaf* nodes, and *video_intermediate* and *video_leaf* nodes respectively. Each intermediate node contains the pointer to the sub-tree it points to; a covering radius, which is the distance between the root of the sub-tree under consideration and its farthest child and four place holders for the promoted high-level similarity. These four place holders for the high-level semantic relationship value hold the high-level similarity value for the four possible types of multimedia data object viz. an image, a frame, a shot or a video. These values change with each query issued but the covering radius and the pointer to the sub-tree remains the same after the GeM- Tree is built until the structure of the tree is modified via an insertion or deletion operation. Each leaf node contains the *object_id* of the indexed database object.

3.2. Node Insertion

To insert a node into a GeM-Tree, the tree is recursively traversed until a candidate leaf node is identified. A particular sub-tree leading to the leaf node is chosen by selecting an intermediate node for which there is no ($d(O_r,O_n)\le r(O_r)$) or minimum increase ($d(O_r,O_n)-r(O_r)$ is minimum) in the covering radius. In case of a tie, the sub-tree, whose object type matches with the object to be inserted is chosen, i.e., $O_{candidate}=O_r$ for which $O_r\rightarrow object\_type == O_n\rightarrow object\_type$. Essentially, a new object is inserted at the leaf node, and if it is full, a split is required followed by a rearrangement of the tree with an increase in the number of levels. Thus, it can be seen that GeM-Tree indeed grows in a bottom-up manner and hence maintains the balanced structure.

4. Similarity Search

GeM-Tree uses a metric distance function, Euclidean Distance (L_2) as the ground distance of the EMD to determine the (dis)similarity between multimedia data objects. As explained in Section 3.1, the number of features used to represent each data object is the same, only that some features may be absent in case of some particular type of data object and have 'zero' values. In our case, the images do not have the features relevant to videos only and have 'zero' values for all the 19 values for F_B. The applied signature representation for multimedia data objects makes sure that the similarity between two multimedia data objects is correctly translated and projected into the metric space, thus creating an effective index structure where similar data objects can be retrieved with minimum computation overhead and false dismissals.

For example, the signature of an image and a video shot can be represented as

$$F_{image} = \left\{ \underbrace{(x_1, x_2, \ldots\ldots\ldots, x_i)}_{F_A}, \underbrace{(0,0,0,\ldots\ldots,0)}_{F_B}, \underbrace{(1,0,0),1}_{F_C} \right\} \quad (1)$$

$$F_{shot} = \left\{ \underbrace{(z_1, z_2, \ldots\ldots\ldots, z_i)}_{F_A}, \underbrace{(y_1, y_2, \ldots\ldots, y_j)}_{F_B}, \underbrace{(1,1,0),1}_{F_C} \right\} \quad (2)$$

The similarity between an image and a shot can be related to the similarity between two shots as follows:

$$\text{if, } d(F_{image1}, F_{shot1}) \leq d(F_{shot1}, F_{shot2}) \quad (3)$$

We can conclude,

$$d(F_{A\_image1}, F_{A\_shot1}) \leq d(F_{A\_shot1}, F_{A\_shot2}) \quad (4)$$

and

$$d(F_{B\_image1}, F_{B\_shot1}) \geq d(F_{B\_shot1}, F_{B\_shot2}) \quad (5)$$

From Equations (3), (4), and (5), it is clear that in terms of similarity measure, if the Euclidean Distance measurement indicates that the similarity between an image and a video shot is more than that between two video shots, it is correct to organize the image and the video shot together rather than the two video shots (even if they belong to different categories of multimedia data). As can be seen from Equation (5), the similarity between the video parts of the signature (F_B) between image1 and shot1 is always less than that of shot1 and shot2. This is because the F_B part of an image has all "zero" values, thus the Euclidean Distance between it and the F_B of any video shot having non-zero values will be always more than the Euclidean Distance between that shot and any other shot (which always has a non-zero F_B values). Also, this dissimilarity could not override the similarity between the image part (F_A) of image1 with shot1 as depicted in Equation (4). Thus, it can be concluded that shot1 is more related to image1 in terms of F_A than it is with shot2 in terms of F_B. Also, in our case, we use a signature consisting of only one distribution, the EMD will be directly proportional to the ground distance when the weights are equal since there will be only one possible flow between two signatures. For scenarios when multimedia data objects might need to be represented with a variable length feature distribution, the weight assignment is a crucial step and should be carefully handled so that the ground distance translates to meaningful relationships when the entire EMD between two multimedia data objects is calculated.

4.1. High Level Semantic Relationship

One of the major contributions of GeM-Tree is the utilization of a high-level semantic relationship capturing mechanism in the retrieval and similarity search methods of the index tree, which does not depend solely on the costly and error-prone feature-level semantic similarity translation approach. We use a mathematical construct called Hierarchical Markov Model Mediator (HMMM) [15]. It is represented by an 8-tuple $\lambda = (d, S, F, A, B, \prod, O, L)$ and each element of the tuple was discussed in details in [15]. For GeM-Tree, we use the A matrix which captures the similarity between two data objects based on the frequency with which they are accessed together by the user. For our retrieval application, we use three A matrices viz. affinity between images, affinity between shots, and affinity between videos. Thus we use only three of the four place holders in the intermediate nodes. As discussed in [5], the affinity relationships cannot be introduced into any distance based index structure and need to be promoted from the leaves to the intermediate nodes before each query. The main idea behind the affinity promotion is to ascertain that there is no false dismissal and no unnecessary sub-tree traversal. An affinity promotion technique similar to one discussed in [5] is used with slight modifications so as to handle the three different affinity kinds.

4.1. k-NN Search

The k-NN algorithm for GeM-Tree supports both content-based image and video retrieval considering the high-level semantic relationships between the multimedia data objects. In addition, GeM-tree is capable of answering queries that involve both images and videos together. The pseudo-code for the k-NN search of GeM-Tree is presented in Table 1. The k-NN algorithm of GeM-Tree is flexible and can accommodate different kinds of video unit classifications. For our application, we used a shot as the lowest unit of a video and adjusted the algorithm to reflect it. Also, if the query object and the object in the candidate intermediate node are not the same, we get their high-level semantic relationship indirectly by traveling upwards/downwards in the hierarchy. For example, if the query object is a video and we encounter a shot, we try to find out if there is any high level relationship between them by checking the affinity between the video to which the shot belongs and the query video.

Table 1: k-NN Search Algorithm for GeM-Tree

```
k-NN_GeneralSearch (Q, N, k,) {
  Promote_Affinity(Q); //affinity promotion for
  // image_affinity, shot_affinity and video_affinity
   if (N != leaf) {
     ∀ O_r in N  do: {
         if (| d(O_r, Q) – r(O_r) | ≤  d_k) {
           if (O_r → object_type == Q → object_type) {
             if (aff(O_r, Q)  ≥ aff_k) {
               Update(d_k);
               Update(aff_k);
               k-NN_GeneralSearch (Q, T(O_r), k);
              //T(O_r) points the root of the subtree of O_r
             }}
            elseif (Q is a video) {
               if (O_r is a shot) {
                 if ((aff(O_r → v_id, Q → object_id)  ≥ aff_k)) {
                   Update(d_k);
                   Update(aff_k);
                    k-NN_GeneralSearch (Q, T(O_r), k);
             }}}
             else if (Q is a shot) {
               if (O_r is a video) {
                 if ((aff(O_r → object_id, Q → v_id)  ≥ aff_k)) {
                   Update(d_k);
                   Update(aff_k);
                    k-NN_GeneralSearch (Q, T(O_r), k);
                 }}}
            Update(d_k);
            k-NN_GeneralSearch (Q, T(O_r), k);
      }}}
      //For the leaf node, perform all the checks as the
     //intermediate nodes and if it qualifies, instead of
    //recursion, add the node pointer to the result set and
   //update d_k.
 }
```

Another flexibility of the GeM-Tree is that it allows for video or only image searches as well. For example, if one wishes to search only videos, the distance function can be modified to compare the feature similarity of only the video part, i.e., F_B of the multimedia object signature, and the k-NN search algorithm will automatically pick up the k nearest videos or shots to a submitted query. For dedicated content based image retrieval, the above technique is slightly modified and though F_A of the signature is used in the distance function, but the result might contain both shots as well as images. In this case, a second stage of refinement is performed, where for each shot in the result set, the frames, of which the shot is made up, are checked for similarity with the query image, and the k nearest images/frames are picked up from the multimedia database as the result set.

Table 2: Experimental Results

Query	# of distance computations				Accuracy			
	GeM	I	II	III	GeM	I	II	III
Only Image	98	80	**X**	147	90%	93%	**X**	98%
Only Video	63	**X**	50	147	90%	**X**	91%	95%
Cross Query	80	**X**	**X**	147	80%	**X**	**X**	90%

5. Empirical Study

To analyze the performance of GeM-Tree, we did an extensive study of both the computation cost as well as the accuracy of the query results for only images, only videos, and concept level queries where the user wishes to get both images and videos corresponding to a submitted query. We compared our results with an index tree for images only (presented by I in Table 2), an index tree for videos only (presented by II in Table 2), a sequential search approach (presented by III in Table 2). The first 2 comparisons were made to check the performance of GeM-Tree in terms of computation overhead and the third was made to check the performance of GeM-Tree in terms of accuracy of the query results.

We used about 1000 images from various sources and about 5 videos of a couple of hours of duration. Each video on an average consists of about 150 video shots. We first extracted the 19 color and texture features of the images and from the frames of the videos. Then, we averaged the features of the frames contained in each shot and formed the F_A part of the signature for the video shots. Similarly, we extracted the 19 video features from individual shots and formed the F_B part. We conducted about 10 queries for only images, 10 for only videos, and 10 concept level cross-queries and averaged the results, which is presented in Table 2. The 'X' marks in the table indicate that the particular index structure was incapable of accommodating the particular query type. We can see that the computation cost of GeM-Tree is slightly more than the computation cost for method I and method II when only image retrieval and only video retrieval were considered. This is due to the fact that the total number of multimedia objects indexed by GeM-Tree was more than either I or II as the multimedia database for GeM-Tree consists of both images and video; whereas in case of both I and II, either only images or only videos were present. Also, the accuracy of GeM-Tree was a little less than both I and II due to the mixed data object types present in GeM-Tree, which increased the probability of false dismissals. The computation cost of method III is much higher than that of GeM-Tree because during sequential scan, the entire database is traversed. This also accounts for the higher accuracy in case of method III, as there will be no false dismissal which was achieved at the cost of high computation overhead. Thus, we can conclude that GeM-Tree has a computation cost comparable with dedicated image indexing or video indexing structures, and an acceptable accuracy value with the added functionality of being able to deal with concept-level cross data type queries and providing a common general framework to index both images and videos. GeM-Tree fulfils the two basic criteria of an index structure viz. having a low computation overhead and acceptable accuracy of query results. In addition, it supports different approaches of content-based retrieval from within the same framework with the flexibility of varied feature representations.

6. Conclusion

In this paper, we propose a common platform for indexing multimedia data objects with the help of a distance based multidimensional index structure called the GeM-Tree. GeM-Tree is a flexible structure and can accommodate different techniques of content-based retrieval by utilizing a variable length multimedia feature distribution and using EMD as the underlying distance function. To the best of our knowledge, GeM-Tree is the first attempt to organize two different types of multimedia objects with a single index structure and support queries that involve both. It is a very promising framework in the multimedia index genre and has ample potential to be improved and utilized for different applications.

Acknowledgment

For Shu-Ching Chen, this research was supported in part by NSF HRD-0317692 and HRD-0833093. We would like to thank Dr. Marco Patella, University of Bologna, Italy for all the valuable discussions on M-Tree.

References

[1] Guttman, "R-trees: A Dynamic Index Structure for Spatial Searching," in *Proc. ACM SIGMOD International Conference on Management of Data*, Boston, USA, June 1984, pp. 47-57.

[2] P. Ciaccia, M. Patella and P. Zezula, "M-tree: An Efficient Access Method for Similarity Search in Metric Spaces," in *Proc. 23rd VLDB International Conference*, Athens, Greece, August 1997, pp. 426-435.

[3] J.T. Robinson, "The K-D-B-Tree: A Search Structure for Large Multidimensional Dynamic Indexes," in *Proc. of ACM-SIGMOD International Conference on Management of Data*, Ann Arbor, USA, April 1981, pp. 10-18.

[4] K. Chakrabarti, K. Porkaew, M. Ortega, and S. Mehrotra, "Evaluating Refined Queries in Top-k Retrieval Systems," *IEEE Trans. on Knowledge and Data Engineering (TKDE)*, 16(10), 2004, pp. 1185-1199.

[5] K. Chatterjee and S.-C. Chen, "Affinity Hybrid Tree: An Indexing Technique for Content-Based Image Retrieval in Multimedia Databases," In *Proc. of the IEEE International Symposium on Multimedia*, San Diego, USA, December 2006, pp. 47-54.

[6] K. Chatterjee and S.-C. Chen, "Hierarchical Affinity-Hybrid Tree: A Multidimensional Index Structure to Organize Videos and Support Content-Based Retrievals," In *Proc. of 2008 IEEE International Conference on Information Reuse and Integration*, July 2008, Las Vegas, USA, pp. 435-440.

[7] F. Ramsak, V. Markl, R. Fenk, M. Zirkel, K. Elhardt, R. Bayer, B. Forschungszentrum, and T. München, "Integrating the UB-Tree into a Database System Kernel," in *Proc. 26th International Conference on VLDB*, Cairo, Egypt, Sep. 2000, pp. 263-272.

[8] J.M. Hellerstein, J.F. Naughton, and A. Pfeffer, "Generalized Search Trees for Database Systems," In *Proc. 21st International Conference on VLDB*, Zurich, Switzerland, Sep. 1995, pp. 562-573.

[9] Y. Rubner, C. Tomasi, and L. J. Guibas. "A Metric for Distributions with Applications to Image Databases," In *Proc. of IEEE International Conference on Computer Vision*, January 1998, Bombay, India, pp. 59-66.

[10] J. K. Wu, "Content-based Indexing of Multimedia Databases," *IEEE Trans. on Knowledge and Data Engineering (TKDE)*, 9(6), 1997, pp. 978-989.

[11] L. Wu, T. Bretschneider, "VP-EMD tree: An efficient Indexing Strategy for Data with Varying Dimension and Order," In *Proc. of International Conference on Imaging Science, Systems and Technolog*y, June 2004, Las Vegas, USA, pp. 421–426.

[12] S. T. Rachev, "The Monge-Kantorovich Mass Transference Problem and its Stochastic Applications," In *Theory of Probability and its Applications*", 29(4), 1984, pp. 647-676.

[13] A. Basharat, Y. Zhai, and M. Shah, "Content Based Video Matching Using Spatiotemporal Volumes," In *Journal of Computer Vision and Image Understandin*g, 110(3), 2008, pp. 360-377.

[14] S.-C. Chen, M.-L. Shyu, C. Zhang, L. Luo and M. Chen, "Detection of Soccer Goal Shots Using Joint Multimedia Features and Classification Rules," in *Proc. of the 4th International Workshop on Multimedia Data Mining*, Washington, USA, Aug. 2003, pp. 36-44.

[15] S.-C. Chen, N. Zhao, and M.-L. Shyu, "Modeling Semantic Concepts and User Preferences in Content Based Video Retrieval," *International Journal of Semantic Computing (IJSC)*, 1(3), September 2007, pp. 377-402.