In Proc. 12nd IEEE Int. Multimedia Modelling Conf. (MMM'06), Beijing, China, 2006

Effective Similarity Search in Multimedia Databases using Multiple Representations

Hans-Peter Kriegel, Peer Kröger, Peter Kunath, Alexey Pryakhin
Institute for Computer Science, University of Munich, Germany
{kriegel,kroegerp,kunath,pryakhin}@dbs.ifi.lmu.de

Abstract

Similarity search in large multimedia databases is an important issue in nowadays multimedia environment. Multimedia objects such as music videos usually consist of multiple representations such as audio or video features. Since each representation may be of significantly different quality for a given multimedia object, similarity search methods could greatly benefit from taking these multiple representations into account. An intelligent similarity search technique should consider all available representations of the database objects and should automatically choose the best representation(s), i.e. those representations that model the object in the best possible way. In this paper, we propose a novel approach for similarity search in multimedia databases taking multiple representations of multimedia objects into account. In particular, we present weighting functions to rate the significance of a feature of each representation for a given database object. This allows to weight each representation during query processing. A broad experimental evaluation shows the suitability and the effectiveness of multi-represented similarity search in video databases.

1 Introduction

In this paper, we propose a novel framework for video similarity search that takes the multi-represented nature of the data objects into account. In particular, our framework is able to integrate multiple representations such as audio and video features into the query processing. The most important issue for multi-represented similarity search is the weighting of each representation, i.e. the decision "how significant is a given representation for a given object". We propose methods for this task that can be applied to both types of summarization techniques, i.e. higher-order and first-order summarization, that are commonly used in multimedia similarity search. In addition, we propose a method for combining multiple representations for similarity search by weighting each representation. A broad experimental evaluation of our methods using a database of music videos demonstrates the benefit of our methods for similarity search in multimedia databases.

2 Related Work

Usually, multimedia objects consist of thousands or even millions of feature vectors. In order to handle such data efficiently, summarization techniques are usually applied to the original data, i.e. the original feature vectors are grouped together and each group is represented by a summarization vector or summarization representative. Similarity is then defined on these summarizations or the according summarization representatives.

In general, we can distinguish two classes of summarization techniques, namely higher-order and first order summarization. Higher-order summarization techniques are usually generated by applying optimization algorithms on feature vectors. They describe a video as a mix of statistical distributions or cluster representatives. First-order techniques calculate a small set of representative feature vectors as summarization vectors in order to describe a video. A randomized technique for summarizing videos, called video signature, is proposed in [1].

3 Multi-represented Similarity Search in Multimedia Databases

In the following, we assume $\mathcal{DB}$ to be a database of N multimedia objects. Each object $O_i \in \mathcal{DB}$, $i = 1, \ldots, N$, is represented by a given set of D representations $R_1, \ldots, R_D$, where each representation is a feature space, i.e. $R_i \subseteq \mathbb{R}^{d_i}$, and $d_i \in \mathbb{N}$ denotes the dimensionality of the feature space of representation R_i ($1 \leq i \leq D$). The j-th representation of O_i is denoted by O_i^j, i.e. $O_i = (O_i^1, \ldots, O_i^D)$. We further assume that each representation O_i^j of O_i consists of a series of feature vectors of length n_j, i.e. $O_i^j = (o_{i\,1}^j, \ldots, o_{i\,n_j}^j)$ with $o_{i\,l}^j \in R_i$.

In addition, we assume that the distances within each representation are normalized sufficiently over all representations, e.g. using any of the methods of [4]. In order to

combine multiple representations within the similarity evaluation, we have to determine for each object $O_i \in \mathcal{DB}$ and for each of its representations O_i^j a weight for each of the n_j feature vectors $o_{i\,1}^j, \ldots, o_{i\,n_j}^j$. Having weights for each feature vector of each representation of each object, we can use any common distance measure between sets of points such as the Hausdorff distance in order to compute a weighted distance between two multi-represented multimedia objects. We will first introduce novel methods to determine the weights for a feature vector of a given representation and then describe how these weights can be used to improve similarity search on multimedia objects.

3.1 Weighting Functions For Summarizations

For efficiency reasons, these large sets of feature vectors are usually summarized within each representation. The derived summarizations can be classified as first-order or higher-order summarizations (cf. Section 2). Thus, the feature vectors $o_{i\,1}^j, \ldots, o_{i\,n_i}^j$ of object $O_i \in \mathcal{DB}$ of representation R_j are representative points of the derived summarizations $S_{i\,1}^j, \ldots, S_{i\,n_i}^j$. In the following, an original point p belongs to a summarization S if it is a member of the according cluster (in case of higher-order summarizations) or if the according representative of S is the representative with the lowest distance to p among the representatives of all summarizations.

A Weighting Function Based on Support. The idea behind our first weighting function is that each summarization vector represents a given amount of original feature vectors. This amount is a good indication on the significance of this representative, i.e. how good this summarization represents the original feature vectors. Thus, in our first approach, the weight of the l-th feature vector $o_{i\,l}^j$ of the j-th representation of object $O_i \in \mathcal{DB}$, denoted by $\mathcal{W}_{\text{supp}}(o_{i\,l}^j)$, is computed by the fraction of points that are represented by $o_{i\,l}^j$. Formally, if $|S_{i\,l}^j|$ denotes the fraction of original points that are summarized by $S_{i\,l}^j$, then the weight of the representative $o_{i\,l}^j$ is computed by

$$\mathcal{W}_{\text{supp}}(o_{i\,l}^j) = |S_{i\,l}^j| / n_j.$$

A Weighting Function Based on Specific Quality Measures. The first weighting function only considers the number of objects the given summarization vector represents. However, it does not take the distances to the representative object into account. A better idea might be to consider the distances of the original points within one summarization to their representative.

Usually, the summarization is generated optimizing a specific quality function. For example, for higher-order summarizations, the summarization is derived from a clustering algorithm such as k-means or EM, which optimizes a clustering quality criterion (e.g. TD^2, log-likelihood). In case of first-order summarization techniques, we can e.g. use the method described in [1] and the according quality function. Our second quality measure is based on the quality criterion upon which the summarization is generated. Intuitively, a summarization vector with high representative power should be weighted high.

Let $CQ(o_{i\,l}^j)$ be the quality measure for the l-th summarization vector of the j-th representation of object $O_i \in \mathcal{DB}$, based on which the summarization is generated, e.g. TD^2 in case of higher-order features generated by k-means. Then, the weight of $o_{i\,l}^j$ is computed by:

$$\mathcal{W}_{\text{qual}}(o_{i\,l}^j) = CQ(o_{i\,l}^j).$$

A Weighting Function Based on Local Neighborhood. The second weighting function takes each original object into account when computing the weights for the derived summarizations. When handling e.g. noisy objects, it would be more reliable to rate the weight of a representative point r based only on the original points in the local neighborhood of r.

Our third weighting function follows this idea. Let $\mathcal{N}_\varepsilon(r_i^j) = \{q_i^j | dist(r_i^j, q_i^j) \leq \varepsilon\}$ be the ε-neighborhood of a representative r_i^j of the i-th database object $O_i \in \mathcal{DB}$ in the j-th representation R_j. Let us note that $\mathcal{N}_\varepsilon(r_i^j)$ only contains original feature vectors q_i^j of O_i in representation R_j. We define the weight of $o_{i\,l}^j$ by the number of objects in its local neighborhood, formally

$$\mathcal{W}_{\text{local}}(o_{i\,l}^j) = |\mathcal{N}_\varepsilon(o_{i\,l}^j)| / n_j.$$

A Weighting Function Based on Entropy. The three weighting functions which we have introduced so far are rather local in the following sense: in order to compute the weight of a representative o of a summarization S_o, they only consider the objects that are summarized by S_o, i.e. belong to S_o. However, it may be more appropriate to consider all original features of a given representation R_i in order to rate a summarization vector o^i of this representation. Our fourth weighting strategy follows this idea.

When computing the weight of a summarization vector o^i of a representation R_i, we want to take the distances of all original feature vectors $q_1^i, \ldots, q_m^i$ of representation R_i to o^i into account. In fact, the distances of q_l^i to o^i can be considered as a random variable x following a Gaussian distribution $G(x)$. The information content of such a random variable can be measured by its entropy. For example, if the entropy of the variable x equals 1, the distances $dist(q_l^i, o^i)$ are randomly distributed, whereas if the entropy of the variable x is considerably low, the distances $dist(q_l^i, o^i)$ are most likely densely packed around the mean value of x and thus, o^i is a good representation of the vectors $q_1^i, \ldots, q_m^i$.

Formally, let $x_{o^i} = \{dist(o^i, q_l^i) \mid 1 \leq l \leq m\}$ be a random variable. The Gaussian distribution $G(x_{o^i})$ of this random variable x_{o^i} is represented by the mean

$$\mu_{G(x_{o^i})} = \frac{\sum_{l=1}^{m} dist(o^i, q_l^i)}{m}$$

and the standard deviation

$$\sigma_{G(x_{o^i})} = \sqrt{\frac{1}{m} \cdot \sum_{j=1}^{m} (dist(o, q_j) - \mu_{G(x_{o^i})})^2}.$$

The entropy of x_{o^i} is then defined as

$$\mathcal{H}(x_{o^i}) = \int_{-\infty}^{+\infty} G(x_{o^i}) \cdot \log G(x_{o^i}) \; dx_{o^i}.$$

Let $o_{i\,l}^j$ be the l-th summarization vector of object $O_i \in \mathcal{DB}$ in representation R_j and let $x_{o_{i\,l}^j}$ be the random variable built by the distances of the original features of O_i in representation R_j to $o_{i\,l}^j$ as defined above. The weight of $o_{i\,l}^j$ is defined as the entropy of the random variable $x_{o_{i\,l}^j}$, formally

$$\mathcal{W}_{\text{entropy}}(o_{i\,l}^j) = 1 - \mathcal{H}(x_{o_{i\,l}^j}).$$

3.2 Combining Multiple Representations for Similarity Detection

Having defined a weighting function for each summarization vector for each representation of a database object, we can combine multiple representations for the process of similarity detection. The key step for effective similarity search is the design of a dedicated distance measure that takes the weights of each summarization vector into account.

In general, we can adopt any distance measure that has been designed for multimedia objects to consider the weights of each feature vector. Let $O = (O^1, \ldots, O^D) \in \mathcal{DB}$ be an arbitrary database object and let $Q = (Q^1, \ldots, Q^D)$ be the query object. Furthermore, let $dist^i$ be the distance function for comparing the i-th representation of O and Q, i.e. a^i and Q^i. Then, the distance between query object Q and a database object O can be computed by

$$dist(Q, O) = \sum_{i=1}^{D} \lambda^i \cdot dist^i(O^i, Q^i).$$

The most important part is to determine the weight λ^i of representation R_i. Obviously, λ^i should be derived from the weights of summarization vectors of the i-th representation of the query object Q, i.e. from $\mathcal{W}(q_1^i), \ldots, \mathcal{W}(q_{n_i}^i)$. The use of the weights of the query object Q only rather is more intuitive than using the weights of both Q and O because we want to ensure that we find database objects that are most similar to Q. Thus, the weights of Q are much more important than that of the database object O.

Regarding the distance function which should be used on the summarizations in each representation, we propose to distinguish between higher-order summarizations and first-order summarizations. Of course, we can combine representations of higher-order summarizations with representations of first-order representations.

Higher-order Summarizations. For higher-order summarizations, we use the Hausdorff distance which is an approved and frequently used distance measure in multimedia similaritry search to compute the similarity between a database object $O^i = \{o_1^i, \ldots, o_n^i\}$ and a query object $Q = \{q_1^i, \ldots, q_n^i\}$ w.r.t. a given representation R_i. In fact, the Hausdorff distance relies on the distance of two specific summarizations, one from Q^i, say q_h^i, and one from O^i, say o_h^i. In other words, there are two summarizations $q_h^i \in Q^i$ and $o_h^i \in O^i$, such that $H(Q^i, O^i) = dist(q_h^i, o_h^i)$. Then the weight of the i-th representation λ^i is determined by the the weight of q_h^i, formally

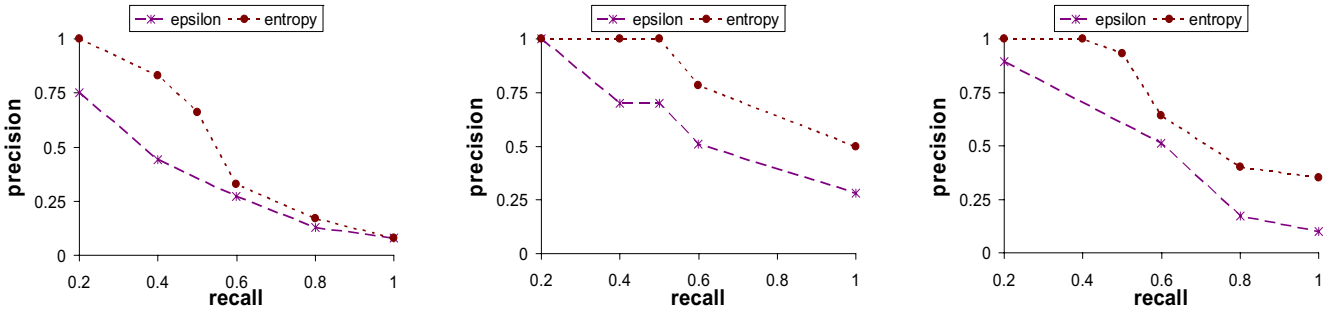
$$\lambda^i = \mathcal{W}(q_h^i).$$

Let us note that the distance function $dist(a, b)$ between two summarization representatives a and b can be arbitrary. If the summarization representatives are feature vectors, any common distance measure such as the Euclidean distance can be used. If the summarization technique generates Gaussian distributions, e.g. using EM clustering, we use the Kullback-Leibler distance [3].

First-order Summarizations. For first-order summarizations, we use the distance function proposed in [1] called ranked ViSig Similarity. This similarity measure relies on a set of distances between summarizations of the query Q and a database object O in each representation. Analogously to higher-order features, we weight each distance with the weight of the participating query summarization.

4 Experimental Evaluation

We evaluated our concepts using a database of 500 music videos collected by a focused web crawler. We extracted the image representations of the videos on a per-frame basis. From each image, we extracted four representations, namely a color histogram and three textural features. For the color histogram, we used the HSV color space. The textural features were generated from 16 gray-scale conversions of the images. We computed contrast, entropy and inverse difference moment using the co-occurrence matrix [2]. For extracting the audio features, we divided the audio signal of a video into short time frames, each having a length of $1/50$ second. Every audio frame is represented by two features in the time- and frequency-domain.



(a) ViSig (b) K-Means (c) EM

Figure 1. Precision vs recall for different weighting strategies when performing similarity search for videos of the same artist.

We computed autocorrelation and threshold-crossing for the time-domain, spectral flux and mel-frequency cepstral coefficients for the frequency-domain [5].

Multi-represented vs. Uni-represented Similarity Search. In a first experiment, we performed video similarity search. As setup step, we picked 50 query videos from our database and manually selected a set of videos which are similar to the query videos. We compared recall and precision achieved on the best single representation to the query result computed by using the ε-neighborhood and entropy weighting functions. Furthermore, we investigated the performance of our weighting strategies on three summarization techniques, namely video signatures (ViSig), K-Means and expectation maximization (EM). For all evaluated summarization techniques, we observed a significant performance improvement when using multiple representations in comparison to the best single representation. Furthermore, our weighted approach leads to better results on all considered summarization techniques.

Using the same test setup as described before, we compared different standard combination techniques for multi-represented objects, such as product, sum, minimum and maximum, to our weighted combination method. In most cases, our weighted approach is more effective than the standard combination algorithms. Especially the ε-neighborhood and entropy weighting methods show good precision and recall values for all considered summarization strategies.

Multi-represented Similarity Search Applications. Given a query video of a specific artist, we want all videos of this artist in our database. Obviously, in this application, a more global notion of similarity is necessary. In order to demonstrate this idea, we selected a set of 20 query videos associated with different artists. For each video in our query set, we extracted all videos of the same artist from our database. The results of our artist search are depicted in Figure 1. In all experiments, the entropy-based weighting function outperforms the ε-neighborhood approach. This can be explained by the fact that the entropy weighting function takes all distances into account in opposite to the local character of the ε-neighborhood function.

5 Conclusions

In this paper, we presented a method for effective similarity search in multimedia databases that takes multiple representations of the database objects into account. In particular, we proposed several weighting functions for summarization vectors of different representations of each database object. Our concepts are independent of the underlying summarization method and compute a weight for each summarization vector of each representation for each object separately. Using these weighting factors, we further show how well-known distance measures for non-multi-represented multimedia objects can be adopted to multi-represented objects. In our experiments we showed the benefits of our approach.

References

[1] S. Cheung and A. Zakhor. Efficient video similarity measurement with video signature. In *IEEE International Conference on Image Processing (ICIP 02)*, volume 1, pages 621–624, 2002.

[2] R. M. Haralick, S. K., and D. I. Textural features for image classification. *IEEE Transactions on Systems, Man, and Cybernetics*, 3(6):610–621, 1973.

[3] G. Iyengar and A. B. Lippman. Distributional clustering for content-based retrieval of images and videos. In *Proc. Int. Conf. Image Processing*, pages 81–84, 2000.

[4] J. Smith, A. Jaimes, C.-Y. Lin, M. Naphade, A. Natsev, and B. Tseng. Interactive search fusion methods for video database retrieval. In *IEEE International Conference on Image Processing (ICIP 03)*, volume 1, pages 741–744, 2003.

[5] G. Tzanetakis and P. Cook. Musical genre classification of audio signals. *IEEE Transactions on Speech and Audio Processing*, 10(5):293–302, 2002.