Timbre Similarity Search with Metric Data Structures

Francisco Costa
Msd student
FCT – Universidade Nova de Lisboa
Portugal
fmc17606@fct.unl.pt

Fernanda Barbosa
CITI / DI
FCT – Universidade Nova de Lisboa
Portugal
fb@di.fct.unl.pt

Abstract—Similarity search is essential in music collections, and involves finding all the music documents in a collection, which are similar to a desired music, based on some distance measure. Comparing the desired music to all the music in a large collection is prohibitively slow. If music can be placed in a metric space, search can be sped up by using a metric data structure. In this work, we evaluate the performance of the timbre range query in music collections with 6 metric data structures (LAESA, GNAT, VP-Tree, HDSAT2, LC and RLC) in 2 metric spaces. The similarity measures used are the city-block and the Euclidean distances. The experimental results show that all the metric data structures speeds the search operation, i.e. the number of distance computed in each search process is small when compares to the number of objects in the database. Moreover, the LAESA data structure has the best performance in the two metric spaces used, but the RLC data structure is the only data structure that never degraded its performance and competes with the other metric data structures, in all the experimental cases.

I. INTRODUCTION

With the rapid increase in the use of digital technology, large amounts of music collections will soon be accumulated, like iTunes, emusic and amazon.com. Music browsing are based on the concept of similarity between musical documents, i.e. searching music documents which are very similar or close to a given music document. There are different dimensions to take care in music similarity search [1]. Some of them are melody, harmony, timbre, rhythm and orchestration. But in all of these dimensions, each music document is represented as a vector of numeric properties (features) extracted from the contend-based music.

Currently there are many works related to music similarity search. In these works, the similarity criterion may be based in the melodic dimension [2, 3, 4, 5], in the timbre dimension [6, 7, 8] or in the rhythm dimension [9, 10]. In the most of them the similar searching for a given music document leads to an exhaustive search in the music collection, so the response time will be very long and the search will became ineffective. For this reason, it is necessary to introduce new techniques that can deal with this problem effectively.

The similarity between two music documents is associated with a function, which measures the distance between their respective feature vectors. Some measures used in the music similarity are: Euclidean distance, city-block distance, global edit distance [2, 4], earth mover's distance [11], dynamic time warping and proportional transportation distance [3]. Some of these functions are metric, as Euclidean, city-block and global edit distances. When this function is metric, the set of music documents defines a metric space.

In order to have efficient similar searching in metric spaces, several metric data structures have been proposed [12,13]. These data structures partition the database based on distances between a set of selected objects and the remaining objects. Space partitions seek to minimize the exhaustive search, i.e. at search time, some subsets are discarded and others are exhaustively searched. The distance-based indexing method may be pivot based or cluster based [12]. Some of the data structures using the pivot-based method are the VP-Tree [14] and the MVP-Tree [15]. There are variants of the pivot-based method, used in LAESA [16]. Some of the data structures using cluster-based method are the GNAT [17], the HDSAT [18], the LC [19] and the RLC [20]. The VP-Tree was used in melody search [2] and the RLC metric data structure was already evaluated in different application domains [21, 22, 23].

Our main goal is to evaluate the use and the efficiency of similar searching with metric data structures in music collections. In this work, we address the problem of timbre similarity search in music collections using the metric data structures. This work involves two similarity criterions: the Euclidean distance and the city-block distance. And comprises 6 metric data structures: VP-Tree, LAESA, GNAT, HDSAT2, LC and RLC.

The rest of the paper is structured as follows. In Section II, we recall some basic notions on similarity search in metric spaces. Section III is devoted to the characterization of the metric spaces over music collections. Then Section IV reports the experimental results of the timbre similarity search over music collections. Conclusions and future work are drawn in Section V.

II. SIMILARITY SEARCH IN METRIC SPACES

A metric space is a pair (U,d), where U is a set of objects, called the universe, and $d: U \text{ x } U \Rightarrow \Re^{+}_{o}$ is a function, called distance, that satisfies the three following properties:

- Strict positiveness: $d(x,y) \geq 0$ and $d(x,y) = 0 \Leftrightarrow x = y$;
- Symmetry : $d(x,y) = d(y,x)$;

- Triangle inequality: $d(x,y) \leq d(x,z) + d(z,y)$.

A database over a metric space (U,d) is a finite set $B \subseteq U$.

The output produced by similarity search is a set of objects in the database, which are similar to a given query object q, based on some distance measure d. The similarity search can be range query or k-nearest neighbor.

The result of the range query is the set of objects in the database whose distance to a given object does not exceed a certain amount. Formally, given a database B over a metric space (U,d), a query point $q \in U$, and a query radius $r \in \Re^+$, the answer to the range query (q,r) is the set X, defined in (1).

$$X = \{x \in B \mid d(x,q) \leq r\} \quad (1)$$

The result of the k-nearest neighbor search is the set of closest objects to a given object in the database, where the cardinality of the set is k. Formally, given a database B over a metric space (U,d), a query point $q \in U$, and a positive integer k, the answer to the k-nearest neighbor search $(q)_k$ is the set X with $|X| = k$, defined in (2).

$$X = \{x \in B \mid \forall\, u \in B\text{-}X,\ d(x,q) \leq d(u,q)\} \quad (2)$$

Metric data structures seek to minimize the number of distance computations performed in similarity search. During the computation of similarity searching in a database over a metric space (U,d), triangle inequality and symmetry are used to discard some elements of the database without computing the associated distance to the query object. Given a query element q and a radius r, an element x may be left out without the evaluation of d(q,x), if there is an object o where $|d(q,o) - d(x,o)| > r$. In these cases, it is not necessary to compute d(q,x) since we know that $d(q,x) > r$, based on the triangle inequality.

It is important to remark that similarity search is hard to compute in high dimension metric spaces. The calculation in real metric spaces is still an open problem [13]. But it is well know that the metric space dimension grows with the mean and decreases with the variance.

III. Music metric space

Our experiments involve only one music collection[1], where each music document is in audio form with sampling rate of 11025 Hz. The collection contains 250 music documents with different music genres [24]: alternative, alternative rock, ambient, dance, electronic, goth and dark era, gothic rock, gothic metal, hard rock, instrumental, metal, new age, nu metal, punk, punk rock, rock, soundtrack, symphonic metal and trance.

Our goal is the timbre similarity search, so we need to characterize the timbre dimension in each music document and use a metric that measures the timbre similarity. In our experiments, we used two metric spaces, i.e. we used two measures: Euclidean and city-block distances.

A. *Timbre Representation*

The perception of the timbre is related to the structure of the spectrum of a signal, i.e. by its representation in the frequency and temporal domains. The spectrogram and other alternative time-frequency representation are not suitable content descriptors, because of their high dimensionality [1]. A set of descriptors that have been extensively used in music information retrieval are the Mel-Frequency Cepstral Coefficients (MFCCs) [25].

For each music document, we compute an audio signature. The audio signature process was based on the process used by Alfie Tan Kok Leong [26] (see Figure 1). The steps involved in creating an audio signature are:

- Dividing the audio signal into frames with 25.6 milliseconds with an overlapping of 15.6 milliseconds;
- Computing the MFCC for each frame. In our process only the first 13 coefficients are used, because the addition of more MFCC coefficients does not improve performance, as we can see in [26];
- Computing K-means clustering with 1 cluster specified, i.e. the coefficients mean of the music frames, in order to discover the song structure and have a short description or reduction in the information of the waveform.

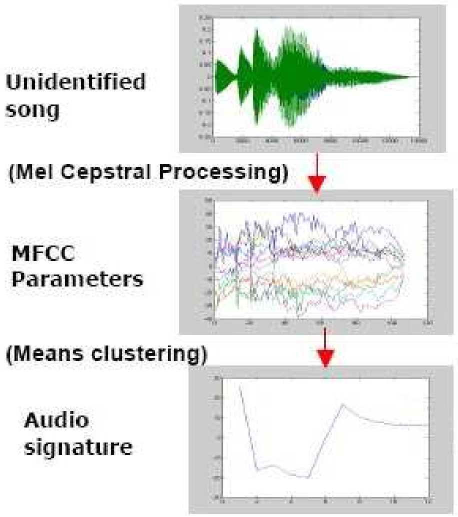

Figure 1. The audio signature process (adapted of [26])

So, each music document has an associate feature vector with size 13.

B. *Timbre Similarity*

In our experiment, the similarity between two music documents is based on the similarity between the associated feature vectors, which is computed with the Euclidean distance and with the city-block distance.

Let $<s_1, \ldots, s_{13}>$ and $<t_1, \ldots, t_{13})>$ be feature vectors associated with two music documents S and T, respectively.

[1] We do not find any online music collection with songs in audio format. So we create a personal music collection.

The Euclidean distance between S and T, denoted by ED(S,T), is defined in (3).

$$ED(S,T) = \sqrt{\textstyle\sum_{i=1..13} (s_i - t_i)^2} \quad (3)$$

The city-block distance between S and T, denoted by CBD(S,T), is defined in (4).

$$CBD(S,T) = \textstyle\sum_{i=1..13} |s_i - t_i| \quad (4)$$

These measures were evaluated in [26], and had satisfactory results according to the music similarity.

In order to study the metric spaces, we computed the histogram of distances between any music documents of the database, using the two measures. In Table I, we presented the mean and the variance of the histogram of distances.

TABLE I. MEAN AND VARIANCE OF DISTANCES

	city-block distance	**Euclidean distance**
Mean	5.791928	2.413186
Variance	5.075894	1.244004
Mean/ Variance	1.14	1.94

An immediate conclusion is that our metric spaces have different dimensions. The dimension is highest for the metric space with Euclidean distance, where the quotient between the mean and the variance is 1.94. The metric space with city-block distance has lowest dimension.

IV. EVALUATION OF METRIC DATA STRUCTURES

The goal of this section is to understand how range queries with metric data structures behave in the music collection, when the similarity criterion is based on timbre dimension (music metric spaces define in Section III).

For the music collection, four files were generated. The smallest is the set of query music documents and the other three are random permutations of the collection. The use of three equal sets lies in the fact that the final shape of some data structures depends on the order in which the objects occur in the input of the construction algorithm. The size of the queries set is 25% of the database size (63 music documents). For each artist/album, we selected 25% of his music documents, in order to have a query set representative of the database, i.e. all the artists, albums and genres are presented in this set.

For each set associated to the database, we submitted the set of query music documents to range query with different query radii in the two metric spaces. The query radii selected for each metric space were equal to 40%, 60% and 80% of the metric space's mean distance. So for the metric space with Euclidean distance, the query radii were 0.964, 1.5 and 1.928. And for the metric space with city-block distance, the query radii were 2.316, 3.5 and 4.632.

In Tables II and III, we presented the average number of music documents retrieved in range queries, and the associated percentage of the database size, for each pair of metric space and query radius.

TABLE II. AVERAGE NUMBER OF MUSIC DOCUMENTS AND PERCENTAGE OF THE DATABASE RETRIEVED WITH THE CITY-BLOCK DISTANCE

city-block distance					
Query Radius 2.316		**Query Radius 3.5**		**Query Radius 4.632**	
Num	Percent	Num	Percent	Num	Percent
7.7	3.07%	38.8	15.51%	92.1	36.84%

TABLE III. AVERAGE NUMBER OF MUSIC DOCUMENTS AND PERCENTAGE OF THE DATABASE RETRIEVED WITH THE EUCLIDEAN DISTANCE

Euclidian distance					
Query Radius 0.964		**Query Radius 1.5**		**Query Radius 1.928**	
Num	Percent	Num	Percent	Num	Percent
13.3	5.33%	59.2	23.66%	102.9	41.18%

In each experimental case (a given metric space and a given query radius), we computed the average number of distance computations done for each music query. So, the results presented are the mean of results obtained to query the three sets associated to the database.

A. Parameterization of Metric Data Structures

In our experiments, we used 9 metric data structures: LAESA, VP-Tree, DSAT, HDSAT1, HDSAT2, LC with fixed cluster size, LC with fixed radius, GNAT and RLC. The metric data structures were parameterized in order to obtain the best results for the music collection. But we've decide to present only the results of 6 metric data structures, so we selected only one data structure for each "representation", which had best results. In the group DSAT, HDSAT1 and HDSAT2, we choose HDSAT2, and in the group LC, we choose LC with fixed cluster size. The parameterization used in each selected data structure is:

- LAESA – Linear Approximating and Eliminating Search Algorithm, with 19 and 11 pivots for the metric spaces with city-block and Euclidean distances, respectively;
- VP-Tree – Vantage Point Tree, which does not have variants or parameters;
- HDSAT2 – Hybrid Dynamic Spatial Approximation Tree2, with arity 5 and 8 for the metric spaces with city-block and Euclidean distances, respectively;
- LC – List of Clusters, with fixed cluster size 8 and 38 for the metric spaces with city-block and Euclidean distances, respectively;
- GNAT – Geometric Near-neighbor Access Tree, with degree 10;

- RLC – Recursive List of Clusters, with array capacity 30 and radius 6.1 for the metric space with city-block distance, and with array capacity 24 and radius 1.86 for the metric space with Euclidean distance.

B. *Experimental Results*

Figures 2, 3 and 4 depict the average number of distances computed to query the music collections with the three query radii in the metric space with the city-block distance.

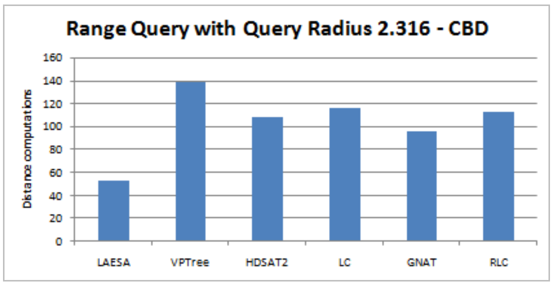

Figure 2. The average number of distance computed to querying the database with city-block distance (query radius 2.316)

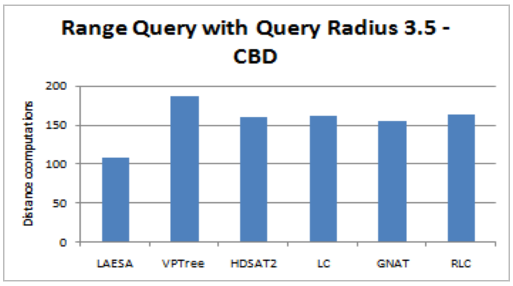

Figure 3. The average number of distance computed to querying the database with city-block distance (query radius 3.5)

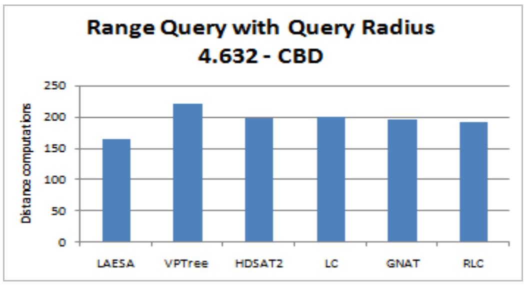

Figure 4. The average number of distance computed to querying the database with city-block distance (query radius 4.632)

Figures 5, 6 and 7 depict the average number of distances computed to query the music collections with the three query radii in the metric space with the Euclidean distance.

In an exhaustive searching, for each query we need to compare the music document query with the all music documents in the database. In our experimental, for all the metric data structures the average number of distance computations by each query is small when comparing with the database size. The corresponding percentage of the database size, for each range query, is shown in Table IV.

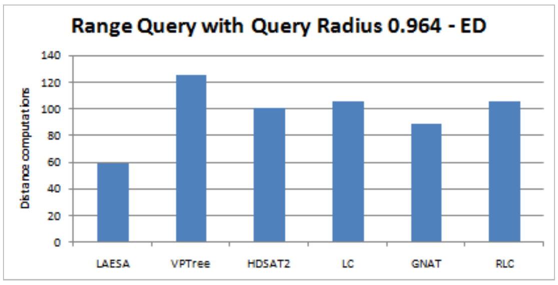

Figure 5. The average number of distance computed to querying the database with Euclidean distance (query radius 0.964)

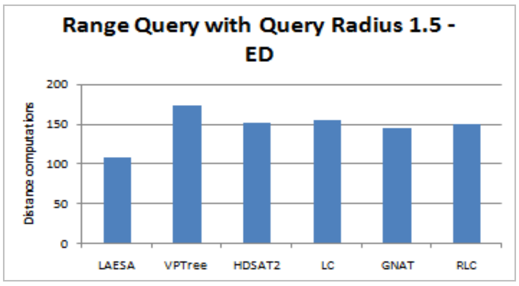

Figure 6. The average number of distance computed to querying the database with Euclidean distance (query radius 1.5)

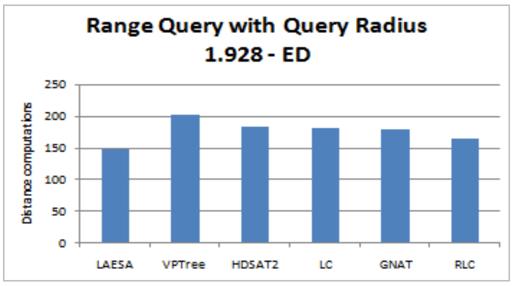

Figure 7. The average number of distance computed to querying the database with Euclidean distance (query radius 1.928)

TABLE IV. THE PERCENTAGE OF DISTANCE COMPUTATIONS ACCORDING TO THE DATABASE SIZE

	City-block distance			**Euclidean distance**		
Query Radius	**2.316**	**3.5**	**4.632**	**0.964**	**1.5**	**1.928**
LAESA	21%	43%	66%	24%	44%	59%
VP-Tree	55%	75%	88%	50%	70%	81%
HDSAT2	43%	64%	79%	40%	61%	73%
LC Fixed Cluster Size	46%	65%	80%	42%	62%	73%
GNAT	38%	62%	78%	36%	58%	71%
RLC	45%	66%	77%	42%	60%	66%

These results confirm that a lot of music documents are discarded without computing the associated distance to the music document query.

We can observe that the best results were obtained in LAESA data structure, for all the query radii in the two metric spaces. And the worse results were obtained in VP-Tree data structure. Excluding LAESA, all data structures were very competitive, but it is important to remark two situations in these data structures:

- The GNAT had the best results with the smallest query radius in the two metric spaces;
- The RLC was the only data structure that never degraded its performance when the query radius increased, i.e. with the two smallest query radii, the RLC competes with the other data structures, but in biggest query radii, RLC is the best data structures.

All the metric data structures had best results in the metric space with the Euclidean distance, which is the highest dimension space.

V. Conclusions and future work

The need to speed the music browsing, lead us to evaluate the performance of range queries with metric data structure in music similar search.

The size of our music collection is not very representative to the real size of music collections. So our next priority is to make this evaluation in large music collections, and we expected to have better results in these collections.

With respect to our metric spaces (timbre representation, and city-block and Euclidean distances), we know that there are many other ways to search for music similarity. So, we have an ongoing work related to search for melodic similarity. We also pretend to evaluate the metric data structures in the rhythm similarity search.

With respect to the efficiency of the range queries with metric data structures, the results leaves us to conclude that the metric data structures speed the range query in the two metric spaces used. This conclusion is based on the observation that a lot of music documents are discarded without computing the associated distance to the music query. The LAESA data structure has the best performance in the two metric spaces, but the RLC data structure is the only data structure that never degraded its performance and competes with the other metric data structures, in all the experimental cases.

References

[1] N. Orio, *Music Retrieval: A Tutorial and Review*, Now Publishers Inc, 2006.

[2] M. Skalak, J. Han, and B. Pardo, “Speeding Melody Search With Vantage Point Trees in *Proc. of International Society for Music Information retrieval (ISMIR),* 2008.

[3] R. Typke, P. Giannopoulos, R. C. Veltkamp, F. Wiering, and R. Van Oostrum, “Using Transportation Distances for Measuring Melodic Similarity”, in *Proc. of International Society for Music Information retrieval (ISMIR),* 2003.

[4] R. B. Dannenberg, W. P. Birmingham, B. Pardo, N. Hu, C. Meek, and G. Tzanetakis, “A Comparative Evaluation of Search Techniques for Query-by-Humming Using the MUSART Testbed”, *Journal of the American Society for Information Science and Technology, pp. 687-701,* 2007.

[5] B. Pardo, and W. Birmingham, “Encoding Timing Information for Musical Query Matching”, in *Proc. of International Society for Music Information retrieval (ISMIR),* 2002.

[6] B. Logan, and A. Salomon, “A music similarity function based on signal analysis”, *Proc. of International Conference on Multimedia and Expo (ICME)*, August ,2001.

[7] J.-J. Aucouturirer, and F. Pachet, “Music Similarity Measures: What’s the Use?”, in *Proc. of International Society for Music Information retrieval (ISMIR),* 2002.

[8] E. Pampalk, “A Matlab toolbox to compute music similarity from audio”, in *Proc. of International Society for Music Information retrieval (ISMIR),* 2004.

[9] J. Foote, M. Cooper, and U. Nam, “Audio Retrieval by Rhythmic Similarity.” in *Proc. of International Society for Music Information retrieval (ISMIR),* 2002.

[10] J. Foote, and U. Nam, “The Beat Spectrum: A New Approach to Rhythm Analysis”, *Proc. of International Conference on Multimedia and Expo (ICME)*, 2001.

[11] Y. Rubner, C. Tomasi, and L. Guibas, “The Earth Mover's Distance as a Metric for Image Retrieval”, *Tech. Report,* Stanford University, 1998

[12] H. Samet, *Foundations of Multidimensional and Metric Data Structures*, Morgan Kaufmann Publishers, San Francisco, USA, 2006.

[13] E. Chávez, G. Navarro, R. Baeza-Yates, and J. Marroquín, Searching in metric spaces, *ACM Computing Surveys*, 33, 3, pp. 273-321, 2001.

[14] P. Yianilos, “Data structures and algorithms for nearest neighbor search in general metric spaces”, in *Proc. of the 4th Annual SIAM Symposium on Discrete Algorithms*. ACM Press, USA, pp. 311-321, 1993.

[15] T. Bozkaya, and M. Ozsoyoglu, “Distance-based indexing for high-dimensional metric spaces”, in *Proc. of the SIGMOD International Conference on Management of Data (SIGMOD’97).* ACM Press, New York, UK, pp. 357-368, 1997.

[16] M. Micó, J. Oncina, and E. Vidal, A new version of the nearest-neighbour approximating and eliminating search algorithm. *Pattern Recognition Letters*, 15, 1, pp. 9-17, 1994.

[17] S. Brin, “Near neighbor search in large metric spaces” in *Proc. of 21st International Conference on Very Large Data Bases (VLDB’95).* Morgan Kaufmann Publishers, Zurich, Switzerland, pp. 574-584, 1995.

[18] D. Arroyuelo, F. Muñoz, G. Navarro, and N. Reyes, “Memory-adaptative dynamic spatial approximation trees”, in *Proc. Of the 10th International Symposium on String Processing and Information Retrieval (SPIRE),* 2003.

[19] E. Chávez, and G. Navarro, A compact space decomposition for effective metric indexing. *Pattern Recognition Letters*, 26, 9, pp. 1363-1376, 2005.

[20] M. Mamede, “Recursive Lists of Clusters: a dynamic data structure for range queries in metric spaces”, in *Proc. of the 20th International Symposium on Computer and Information Sciences (ISCIS 2005).* Springer-Verlag, Berlin, Germany, pp. 843-853, 2005.

[21] M. Mamede, and F. Barbosa, “Range Queries in Natural Language Dictionaries with Recursive Lists of Clusters”, in *Proc. of the 22th International Symposium on Computer and Information Sciences (ISCIS 2007),* IEEE Xplore (doi:10.1109/ISCIS.2007.4456857), 2007.

[22] F. Barbosa, “Similarity-based retrieval in high dimensional data with Recursive Lists of Clusters: a study case with Natural Language Dictionaries”, in *Proc. of the International Conference on Information management and engineering (ICIME 2009)*, IEEE Computer Society (ISBN: 978-1-4244-3774-0), 2009.

[23] F. Barbosa, and A. Rodrigues, “Range Queries over Trajectory Data with Recursive List of Clusters: a case study with Hurricanes data”, in *Proc. of Geographic Information Science Research UK (GISRUK 2009),* UK, 2009.

[24] Lastfm, Music community website, www.lastfm.com

[25] B. Logan, “Mel Frequency Cepstral Coeffcients for Music Modeling”, in *Proc. of International Society for Music Information retrieval (ISMIR),* 2000.

[26] Alfie Tan Kok Leong**,** *A Music Identification System Based on Audio Content Similarity,* Doctoral Thesis, University of Queensland, Queensland, Australia, 2003.