2018 International Multi-Conference on Industrial Engineering and Modern Technologies (FarEastCon)

A Hierarchical Method Of Forming Fingerprints Of A Sound Signal

Maksim O. Derbasov, Vasiliy E. Gai, Polina A. Shagalova
Institute of Radio Electronics and Information Technologies
NNSTU
Nizhny Novgorod, Russian Federation
iamuser@inbox.ru

***Abstract*—This paper is devoted to the description of the storage model and the search for sound sequences based on the theory of active perception. The theory of active perception is used to form an indicative description of the sound signal. The class of problems solved by the proposed model has a wide scope and includes, among other things, the search for musical plagiarism. It shows the possibility of creating on the basis of the proposed model a software system for identifying audio signals. The software implementation of the search model is made in the programming language R. To approbate this model computational experiments have been carried out, where the database size is 10,000 musical compositions and the search accuracy reaches 96%. The stability of the proposed model is also analyzed for distortion of the sought signal by noise. A comparison is made with similar existing systems in terms of search accuracy.**

Keywords—theory of active perception; search for sound signals; database

I. Introduction

The rapidly growing volume of collections of digital audio materials creates the need for effective mechanisms for searching content in these collections. The traditional approach to search is based on the use of text metadata describing the contents of sound files. Often, such metadata is not available or does not contain enough information to find. In this case, a strategy based on analyzing the contents of audio files is used. The article proposes a search strategy using the pattern-based search method: when setting an audio query, the task is to extract from the music collection all the signals that are similar to the query [1].

Various solutions are known for this problem. For example, Philips [2] for the description of audio signal fragments offers a compact, 32-bit representation of the differences in subband energy together with the search for an exact match in the hash table. Search in the Shazam [3] system is based on the use of many key signatures representing pairs of peaks in Fourier spectrograms. Fraunhofer [4] offers AudioID technology. This technology is built on the classic pattern classification framework, using the standard neighbor method on MPEG-7 descriptors coded by vector quantization. Academic studies include improvements to the Shazam method using the Constant-Q transform [5], computer vision techniques on spectrograms [6] and the search strategy research for binary sound prints [7].

The proposed work for implementing the search strategy uses the theory of active perception (TAP) [8]. The basic transformation in the theory of active perception is the *U*-transformation. The use of the *U*-transformation in the analysis of sound signals is considered in [9, 10]. Comparing TAP with the known approaches to the formation of an indicative description of the signal, for example, by the Fourier transform, by the mel-frequency transformation, we can note the following: in comparison with the wavelet transformation and the Fourier transformation, TAP makes it possible to calculate, with respect to spectral coefficients, signs of a higher level (due to the use of group algebra); in comparison with the models of in-depth training in TAP, a feature description is calculated without the use of training, but according to predetermined templates; addition and subtraction operations only are used in calculating the *U*-transformation. Conceptually, the system of searching for a sound signal in the database can be represented in the form of the scheme shown in Fig. 1.

This work is devoted to the description of the first three modules responsible for decoding and norming, generating fingerprints and searching the database. As a database, it is supposed to use general-purpose relational DBMS, as their technological development provides performance and scalability.

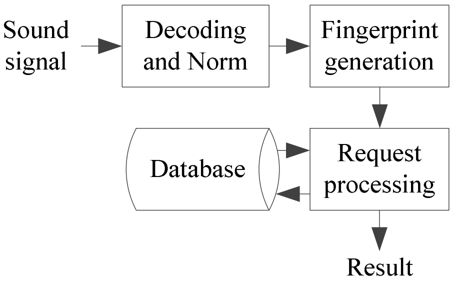

Fig. 1. Example of a figure caption.

II. Pretreatment and Signal Description of the Signal

A continuous sound signal is a continuous real function $s_a(t)$ of the real argument t, defined on a finite time interval T ($t \in T \subset R^1$). The discrete sound signal s_{dsc} is a numerical sequence, each element of which is the value of the continuous sound signal $s_a(t)$ at the time $n\Delta t$, where Δt is the sampling step, n= 0:(N – 1) and N is the number of samples of the discrete audio signal. The digital audio signal s_{dig} is formed as a result of sampling by the sampling level of the discrete audio signal s_{dsc}. In the technical sense, the digital audio signal is the values taken with some known constant sampling rate from the ADC module. Further, for the sake of brevity, the digital audio signal will be denoted by the symbol s.

The stage of preliminary signal processing includes its resampling (in order to bring all signals in the database to the same sampling frequency) and norming according to formula (1) to the range [0, 1]

$$s_{dsc} = s_{dsc} - \min\,(s_{dsc}),\ s_{dsc} = s_{dsc} / \max\,(s_{dsc}) \qquad (1)$$

The necessity of the normalization is explained by the requirements of the U-transformation to the dynamic range of the signal [9].

The analyzed signal is represented as a set of segments **s**, where a segment is understood as a certain non-zero sequence of values from the original signal S: $\mathbf{s} = \{s_k\}$, $k = 1{:}K$, where K is the number of segments, the segment length is M samples, the segments are selected from the signal with a uniform step in N counts.

An indicative description of the signal segment is formed as a result of applying the U-transformation to the i-th segment. The first stage of the U-transformation is the division of the i-th segment by M equal in the number of parts readings and the application of the summation operation to each of the parts (this stage is called the Q-transformation [9]):

$$q_k[i] = \sum_{l \in T_i} s_k[l]$$

where T_i are the indices of the samples of the segment s_k belonging to the i-th part of the segment.

The second stage is the application to the resulting vector of partial sums q_k of Walsh filters of the Harmut system, as a result of which a vector of spectral coefficients μ_k:

$$\mu_k[i] = \sum_{j=0}^{M-1} q_k[j] F_i[j], i = 1{:}M, k = 1{:}K \qquad (2)$$

where M is the number of used Walsh filters of the Harmut system.

III. Information model of the storage and retrieval system

The efficiency of the system being developed depends on the method of storing and searching for the fragment of the sound signal. In actual conditions, the search can take place among millions of records. To solve this problem, a two-level model for storing and searching for fingerprints, presented in the form of multiple segments, has been developed. At the storage phase at the upper level (see Fig. 2), the segments of the analyzed signal, using the recursive descent method, are located in the terminal nodes of the binary tree. This allows us to put in the design of the system the upper level of parallelism, for example, division by various search servers. Then, for each segment, a key is generated, which is used to search for the signal, but not across the entire database, but only among the segments belonging to the terminal node.

A. Placement of Segments in a Binary Tree

Placement of the element (signal segment) in the tree takes place according to the transitions in the recursive descent along the segment:

- The N-level binary tree is initialized, the first level is set as the current analysis level: $i = 1$ ($i \leq N$).
- The U-transformation of the signal segment s_k is calculated using formula (2), the result of calculation (2) is the vector μ_k from M values (in general, M values, in the present paper M is equal to 16) to which the signum function is applied

$$\mu'_k = \mathrm{sgn}\,(\mu_k) \qquad (3)$$

- The Hamming distance between μ'_k and binary vectors is calculated: $T_1 = \{1,1,1,1,1,1,1,1,-1,-1,-1,-1,-1,-1,-1,-1\}$ and $T_2 = \{-1,-1,-1,-1,-1,-1,-1,-1,1,1,1,1,1,1,1,1\}$:

$$d_1 = DIST\,[\mu'_k, T_1],\ d_2 = DIST\,[\mu'_k, T_2]$$

  where $DIST$ [] is the Hamming distance calculation operator. If $d_1 < d_2$, the left half of the analyzed segment is selected for further analysis, and the segment information is placed on the left child vertex (relative to the current vertex), otherwise - the right half of the segment is selected for analysis and the segment information is placed in the right child vertex.

- the transition to the next level of the segment s_k analysis is performed: as the segment s_k its left or right part is now considered (depending on the decision taken at the previous step of the algorithm), the current tree level is set to $i = i + 1$, the transition to step 2 of the algorithm is performed. If $i > N$, the segment is considered to be processed. The algorithm works until all the segments of the source signal are placed in a binary tree.

Thus, as a result of the operation of the algorithm, information about the signal segment is placed in one of the terminal nodes of the binary tree.

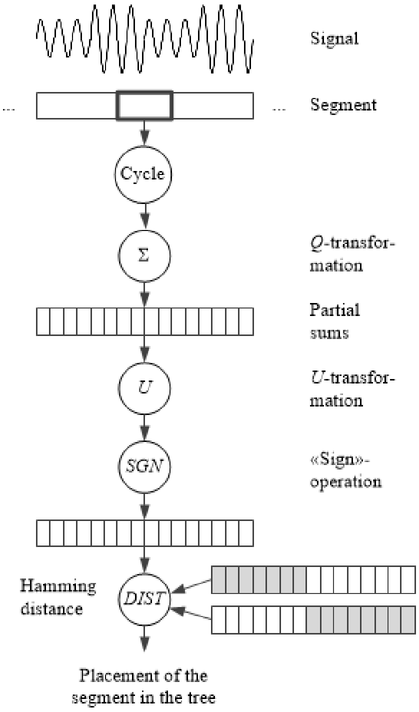

Fig. 2. Generating a segment print

B. Forming a key by a signal segment

At the lower level, the key is generated by the segment s_k, which will then be used when searching for signals in the database (within the specified terminal node). The key K_k for the segment s_k is suggested to be formed by the segment s_k itself, as well as on its neighbors:

$$K_k = KEY\,[s_k, s_{k+1}, s_{k+2}, ..., s_{k+L}],$$

where KEY[] is the key generation operator, L is the number of neighbors participating in the key generation. Thus, at this level, the processing of segments is performed taking into account their temporary connection with each other. This allows you to significantly improve the accuracy of searching for sound signals in the database.

To generate the key, the vectors of the values of the signum function are combined after the first U-transformation of the segment of each of the segments included in the key. Further, the resulting vector is translated into a set of bits, where, for example, the positive sign is converted to zero, and the negative one becomes one.

In other words, it is assumed that the probability of occurrence of short identical sequences of segments between two signals stored in the database is less than the probability of one segment belonging to two different signals.

This produces a sequence that can be interpreted as an unsigned integer. This number will be used as a key when searching.

C. Analysis of possible collisions of search keys

The rationale for using the second level of signal analysis is as follows. Let's spend a little thought experiment. Let the data have sufficient entropy and all random events will be assumed to be equally probable.

Suppose there are 10^9 audio segments that need to be placed in the database. This number may seem large, but it is worth mentioning that the size of such a segment is a fraction of a second.

Let the depth of the partition be six. Then a tree with a depth of six levels will be built on the upper level. Therefore, using this tree, you can divide the data space into 2^6 parts, each of which will have an average of about 16 million segments.

Suppose that a key is generated for the segment based on the method described above, and only the segment itself was used to generate the key, without neighbors ($L = 0$). The range of key values is from 0 to 32767, since the dimension of the binary vector after (3) is 15 elements. Let each key on the second level correspond to the order of 240 segments. The number seems to be small, but since the sequence of segments is searched, it is necessary to select each next 240 elements of the segment, which will increase the number of I / O operations and negatively affect the performance.

If the key is to be calculated for at least two consecutive segments, its range will be from zero to 2^{30}-1. Then the filling factor will be much less than unity.

In spite of the fact that some assumptions are made in the mental experiment, in reality the data will have much less entropy, so the imbalance in the number of records in a couple of orders will not create a lot of collisions in individual keys.

D. Searching for a sound signal in the database

The algorithm for searching for an arbitrary signal includes the same steps as the key generation algorithm.

First, an input signal generates a plurality of segments with a predetermined shift step. Then, for each segment, the terminal node of the binary tree is defined, to which it belongs and a key is formed along the segment. After the key is generated, it is searched for in the audio signal base. As a result, a response is formed: the signal in the database was found or not found.

When searching for an arbitrary signal in the database, one of the main requirements is its duration. The minimum signal duration must be one segment greater than the number of segments participating in the lower level in the formation of the search key. This requirement is a consequence of the search mechanism, which moves from the first sample of the input signal with some step, searches for a possible sequence of segments. This process may seem computationally complex, but with good clipping during the search process must pass quickly enough. The software implementing the proposed method was developed using the approaches described in [11, 12].

IV. Computational experiment

A. Results

The computational experiment was conducted on the basis of 10,000 sound signals representing musical compositions. The average length of the track is about three minutes. The offset step in the formation of a set of segments for each signal is selected equal to 10 samples. The table shows the most representative values obtained during the computational experiment.

From Tab. 1 it follows that the use of a longer sequence makes it more likely to find the signal of interest, the depth of the recursive partitioning of the audio segment positively affects the accuracy of the search.

B. Known results

Tab. 2 provides an estimation of the accuracy of the search in the noise environment of the Shazam system, as well as the developed system based on the proposed method of forming a print of the sound signal.

TABLE I. Results

Number of tree levels	Duration of the input request (seconds)	Signal to noise ratio (dB)	Search accuracy (%)
5	1	0	45.8
		10	62.5
		20	75
	2	0	41.6
		10	50
		20	62.5
	4	0	50
		10	66.6
		20	75
7	1	0	50
		10	66.6
		20	79.2
	2	0	79.2
		10	83.3
		20	91.6
	4	0	87.5
		10	91.6
		20	95.8

TABLE II. Estimation of the stability of the proposed description to noise

Noise level, dB	Developed method, %	Shazam, %
0	87.5	80
10	91.6	95
20	95.8	99

Despite the fact that the proposed method is somewhat inferior to the well-known system, the difference in quality is not dramatic and can potentially be improved by a modification of the method.

Conclusion

The paper proposes an approach to the construction of an information model and an algorithm for finding a signal in sound databases. In practice, this development can be used to search for interesting sound fragments. For example, in the music industry to identify plagiarism among performers and composers. Unlike the known search methods, which basically use an exact search and comparison of the signal characteristics, in this algorithm there is a sequential, first rough, and then, with each step of the algorithm, a more accurate finding of the sound fragment in the database.

References

[1] V. E. Gai and I. V. Polyakov, “Sound Identification Model in Terms of the Active Perception Theory”, International Journal of Imaging and Robotics, Vol. 16; Issue No. 3, pp. 51-63, 2016.

[2] Jaap Haitsma and Ton Kalker, “A highly robust audio fingerprinting system,” in Proc. ISMIR ’02, October 13-17 2002.

[3] Avery Li-Chun Wang, “An industrial-strength audio search algorithm,” in Proc. ISMIR ’03, 2003.

[4] Eric Allamanche, Jurgen Herre, Oliver Hellmuth, Bernhard Froba, Throsten Kastner and Markus Cremer, “Content-based identification of audio material using MPEG-7 low level description,” in Proc. ISMIR ’01, 2001.

[5] Sebastien Fenet, Gael Richard, and Yves Grenier, “A scalable audio fingerprint method with robustness to pitch-shifting,” in Proc. ISMIR ’11, Miami, Florida, USA, pp. 121–126, 2011.

[6] Bilei Zhu, Wei Li, Zhurong Wang, and Xiangyang Xue, “A novel audio fingerprinting method robust to time scale modification and pitch shifting” in Proceedings of the ACM International Conference on Multimedia, Firenze, italy, October 25-29 2010, pp. 987–990.

[7] Kimberly Moravec and Ingemar J. Cox, “A comparison of extended fingerprint hashing and locality sensitive hashing for binary audio fingerprints,” in Proc. ICMR ’11, April 17-20 2011.

[8] V. A. Utrobin Physical interpretation of the elements of image algebra J. Advances in Physical Sciences. – 2004. – № 47. – P. 1017–1032.

[9] V. E. Gai, I. V. Polyakov, M. S. Krasheninnikov, A. A. Koshurina and R. A. Dorofeev “Method of monaural localization of the acoustic source direction from the standpoint of the active perception theory”, Journal of Physics: Conference Series, V. 803, 2017, N. 1, P. 012043

[10] Gai V. E., “A study of stability of sound signal description”, Pattern Recognition and Image Analysis, Vol. 24, No. 4, pp. 463-466, 2014.

[11] V.V. Kondratiev and D. V. Zhevnerchuk “Application of Methods of Self-Organization Theory to Problems of Profiling and Configuring Computational Systems”, Doklady Mathematics, Vol. 90, No. 3, pp. 788-190, Pleiades Publishing, Ltd., 2014.

[12] D.V. Zhevnerchuk, A.S. Zaharov, L.S. Lomakina and A.S. Surkova “Semantic Modeling Of The Program Code Generators For Distributed Automated Systems”, Advances in Computer Science Research (ACSR), V. 72, IV International Research Conference "Information Technologies in Science, Management, Social Sphere and Medicine". (ITSMSSM 2017), P. 266-270