Auditory Content Analysis Based Digital Media Information Compression

Haifeng Li#1, Tian Zhang#2, Lin Ma#3

1School of Computer Science and Technology, Harbin Institute of Technology, China
2lihaifeng@hit.edu.cn; 3zhangtian@hit.edu.cn, 3Malin_li@hit.edu.cn

Abstract- Lots of repetition units exist in digital multimedia especially television programmes and other kinds of broadcastings. In this paper a new information compression algorithm named Auditory Content-Based Information Compression (ACBIC) is proposed to detect repeated programme segments in order to further reduce information redundancy. This algorithm is quite different from former data compression methods in that it realizes compression in semantic level through extracting auditory contents rather than signal statistic information. In our algorithm, Linear Predictive Coding (LPC) features are extracted at acoustic level, an LPC distance measure is defined to divide digital media stream into programme segments, and then auditory content description is created by statistically weighting LPC coefficients for each programme segment. Applying the idea of the dictionary table in LZW lossless data compression algorithm, a content-based information retrieval algorithm is developed to complete information comparison, selection, combination, and updating of the segmented programme units, similarly as data compression and decompression operations. Finally the experiments show an approving information compression result and confirm the completeness of the audio information redundance detection and the reversibility of the compression process.

Keywords- Auditory Content-Based Information Compression; Dictionary Table; Information Redundancy; Auditory Content Description; LPC Distance Measure

I. INTRODUCTION

Data especially multimedia data result in the problem on information storage and transmission at present time. It becomes more and more difficult to store and retrieve information because the information explosion brings so many redundances. As the advanced form of data, information reflects the content of data. As a result content-based compression approaches are superior to conventional data compression ones in that the compression process here is not taken on the data or signal itself but on the content description and statistics. In this paper an information compression method is proposed for the compression of digital multimedia especially television programmes and other kinds of broadcastings named Auditory Content-Based Information Compression (ACBIC) which is improvement and supplement to conventional data compression.

Several researches have been done in the field of auditory compression in recent years. References [1] and [2] used IIR prediction and Huffman coding for the prediction error signal which is used in DVD standard. Lossless Transform Audio Compression (LTAC) took use of transform coding for decorrelation in [3]. In [4] a predictive type computational structure where adaptively varied approximations are subtracted from the original signal was used to produce an error signal that is losslessly coded. Liebchen et al. discussed in [5] the Audio Lossless Coding used in MPEG-4 in detail. Moon et al. estimates virtual source location information (VSLI) as the side information for encoding multichannel signals, which proved a better audio quality than the former methods in [6]. Yu et al. presented Advanced Audio Zip (AAZ), a fine grained scalable to lossless (SLS) audio coder that has been adopted as the reference model for MPEG-4 audio SLS work in [7]. Hotho et al. proposed in [8] a backward-compatible multichannel audio codec combining a high audio quality and a low parameter bit rate. Motlicek et al. applied frequency-domain linear prediction not only for coding speech but also well suited for coding other important acoustic signals such as music and mixed content in [9]. And Fatimah et al. present a tool to evaluate Lossy compression techniques according to the characteristics of the signal, in term of quality, size, and context of use called Efficient Audio Compression Tool (EACT) in [10].

As can be seen, most of these studies are on the compression coding of the audio signal. Comparably there are few researches on content-based information compression. It is worth noticing that television programmes and other kinds of broadcastings always have repetitions in content on which the ACBIC algorithm focuses. ACBIC aims to losslessly and quickly compress digital media files or streams based on redundant programme segments detection and deletion. As one of the most successful lossless compression methods, LZW owns outstanding characters including the dictionary table algorithm and ACBIC borrows the idea of the dictionary table for the application of the compression of digital multimedia information redundance. The main research focuses on the following two aspects: 1. the extraction of the description vectors of auditory content; 2. lossless compression and programme reconstruction based on the produced description vectors. Fig. 1 shows the system briefly.

![Fig. 1 Flow diagram of the system](image-url)
The rest of this paper is organized as follows. In Section II we present the LPC distance based digital media segmentation and content description methods. In Section III the detail of the ACBIC algorithm is proposed. Section IV illustrates the experimental results on the proposed algorithm. Finally the conclusion is given in Section V.

II. LPC DISTANCE BASED SEGMENTATION AND CONTENT DESCRIPTION

The semantic structure must be extracted first in order to achieve content-based information compression, which means that the content-independent programme segments should be produced based on the content segmentation. And then we derive the LPC distance based segmentation algorithm and acquire the content description subsequently.

A. LPC Distance Description

The purpose of programme segmentation is to bring about several programme segments that are made up of frame sequences and are regarded as the minimum unit of auditory information compression. Different segments contain different information, so some statistically changing points must exist. LPC analysis can reflect auditory content information and was used to partition audio files (e.g. [11] and [12]). In fact Mel Frequency Cepstrum Coefficient (MFCC) can also be used for the whole process. However, the time cost of processing waveform files using MFCC is much longer than using LPC (more than 6 times in detail) and the performance is more or less the same as using LPC. So it is more effective and convenient to adopt LPC as the feature used.

In the LPC distance based segmentation if \( N \) continuous frames are detected as changing points, the interval is regarded as the gap between two audio units and is named segmentation point.

Define LPC distance \( d(t, \tau) \) below:

\[
d(t, \tau) = |C_t - C_\tau|
\]  

(1)

Here, 

\( t \) is the processing frame number; 
\( \tau \) is the reference tab; 
\( C_t \) represents the value accumulation of each dimension of the LPC coefficients of the presently processing frame \( t \); 
\( C_\tau \) represents the reference LPC statistics.

The detail of the computation of \( C_t \) and \( C_\tau \) can be found in [13].

And we revise the algorithm in the way judging the states of the LPC distance. Now we redefine the continous frames (within one audio unit) as \( N \) frames or more satisfying \( d(t, \tau) < \theta \) (where \( \theta \) is an experience value obtained by experiments), meanwhile allowing \( M \) concatenated frames or less unsatisfying.

B. LPC Distance Based Segmentation

Now two signal numbers Big and Small are set. Another two variables: Begin and End represent the beginning and ending frame number respectively. And \( \text{SIGN} \) represents a 2-dimensional array which is used to record every beginning and ending frame number of segmented programme segments. The procedure of the judgment is (see the detailed flowchart in Fig. 2):

1. if \( d(t, \tau) < \theta \), the \( t^{th} \) frame is in the same programme segment with the previous frame; set Small++ and Big=0; 
2. if \( d(t, \tau) > \theta \), the situation will be the following:
   1) Small>N and Big>M. It means that there is a gap, so End is set to the frame number of present number minus Big. Record Begin and End in \( \text{SIGN} \) and update these signal values;
   2) Small>N and Big<=M. It means the present frame is still in the programme segment, so no action is needed in this situation except for reading the next frame;
   3) 0<Small<=N and Big>M. That means this whole piece is not a programme segment. Set Begin=Begin+1, and Big=Small=0, and then read the next frame;
   4) 0<Small<=N and Big<=M. Read the next frame;
   5) Small=0. Set Small++, and Big=Small=0.

By considering all possible conditions, we can make reasonable judgment to obtain the right segmentation points.

![Fig. 2 Flow diagram of LPC distance algorithm](image-url)
C. Content Description Production

The content description vector is derived via the statistical calculation of the LPC coefficients. We use the frame number as the weight to assure that the statistics relies on the sequence order. Equation (2) gives the computation of the content description vector.

\[
S(k) = \left\{ S_i(k) \right\}_{i=1,...,I} = \left\{ \frac{2}{N_k(N_k + 1)} \sum_{n=1}^{N_k} nx_k(n)(i) \right\}_{i=1,...,I}
\]  

where \(1 \leq k \leq K\).

Here,

- \(S_i(k)\) is the value of the \(i^{th}\) dimension of the \(I\)-dimensional content description vector for the \(k^{th}\) programme segment;
- \(x_i(n)(i)\) is the value of the \(i^{th}\) dimension of the \(I\)-dimensional LPC coefficients of the \(n^{th}\) frame in the \(k^{th}\) programme segment;
- \(N_k\) is the number of frames in the \(k^{th}\) segment (weight);
- \(K\) represents the number of programme segments;
- \(I\) is the number of dimensions of LPC coefficient.

III. AUDITORY CONTENT-BASED INFORMATION COMPRESSION ALGORITHM

Through the compression algorithm, the produced content descriptions are compared, combined and updated. Content redundancy-free programme segments are achieved as the compression production. During the compression (coding) process we utilize the accessory information for programme reconstruction (decoding) and realize the lossless revert of the original digital media.

A. Information Comparison

During compression, two content description vectors representing the content information usually need to be compared to make sure whether they are “matched”. Then according to the result of comparison, the algorithm can decide whether to keep the processing segment or not. In conventional compression algorithms (e.g. LZW), the objects to be compressed are in text-like format, so the comparison stage is simple and direct. But it is not the case for digital media files. Distance measures are introduced to solve such kind of problems: Euclidean Distance, Cosine Distance, Mahalanobis Distance, et al. (e.g. [14]). Euclidean Distance is adopted for its simpleness and effectiveness. Given two description vectors: \(S(k)\) and \(S(h)\) (here lists \(S_i(k)\) and \(S_i(h)\) below)

\[
S_i(k) = \frac{2}{N_k(N_k + 1)} \sum_{n=1}^{N_k} nx_k(n)(i)
\]

\[
S_i(h) = \frac{2}{N_h(N_h + 1)} \sum_{n=1}^{N_h} nx_h(n)(i)
\]

The formula of comparison using Euclidean Distance is

\[
D(k, h) = \sqrt{\sum_{i=1}^{I} \left( S_i(k) - S_i(h) \right)^2}
\]

\[
= \sqrt{\sum_{i=1}^{I} \left( \frac{2}{N_k(N_k + 1)} \sum_{n=1}^{N_k} nx_k(n)(i) - \frac{2}{N_h(N_h + 1)} \sum_{n=1}^{N_h} nx_h(n)(i) \right)^2}
\]

We set the rule that the two vectors match if \(D(k, h) \leq 0.2\) or vice versa.

B. Programme Combination

The compression process requires the combination of two description vectors to form the content description vector of the two corresponding segments united. To reduce the time and space complexity of the algorithm, we deduce the formula from the values of the two given vectors which is equivalent to the one deduced from the two corresponding segments united instead of extracting LPC coefficients over again after combining the two segments. The formula is derived in the following.

Set the two description vectors: \(S(k)\) and \(S(h)\). Equation (6) shows the content description vector using extracted LPC coefficients after uniting the two programme segments.

\[
S = S(k \cup h) = \left\{ S_i(k \cup h) \right\}_{i=1,...,I}
\]

\[
= \left\{ \frac{2}{(N_k + N_h)(N_k + N_h + 1)} \sum_{n=1}^{N_k+N_h} nx_i(k, h)(n)(i) \right\}_{i=1,...,I}
\]

where \(k \cup h\) represents the segment produced by uniting the two segments.

Now we can prove \(S\) can be deduced from \(S(k)\) and \(S(h)\). Given Equations (3) and (4), we can split the sum of Equation (6) to obtain the following equation:

\[
S_i(k \cup h) = \frac{2}{(N_k + N_h)(N_k + N_h + 1)} \sum_{n=1}^{N_k} nx_i(k)(n)(i)
\]

\[
= \frac{2}{(N_k + N_h)(N_k + N_h + 1)} \sum_{n=1}^{N_k} nx_i(k)(n)(i)
\]

\[
+ \sum_{n=N_k+1}^{N_k+N_h} nx_i(h)(n)(i)
\]

\[
= \frac{2}{(N_k + N_h)(N_k + N_h + 1)} \sum_{n=1}^{N_k} nx_i(k)(n)(i)
\]

\[
+ \sum_{n=N_k+1}^{N_k+N_h} nx_i(h)(n)(i)
\]

Now substitute the involved items, and Equation (8) is derived below:

\[
S = \frac{S(k)N_k(N_k + 1) + 2N_kH + S(h)N_h(N_h + 1)}{(N_k + N_h)(N_k + N_h + 1)}
\]  

\[
\frac{S(k)N_k(N_k + 1) + 2N_kH + S(h)N_h(N_h + 1)}{(N_k + N_h)(N_k + N_h + 1)}
\]
where

\[ H = \left\{ \sum_{n=1}^{N_h} x_{h}(n)(i) \right\}_{i=1,...,I} \]  \tag{9} \]

As we can conclude, the content description vector of two united segments can be acquired via deduction from the two corresponding segments.

C. Information Compression Algorithm Implementation

We borrow the idea of the dictionary table in LZW compression algorithm (e.g. [15] and [16]), and derive our ACBIC algorithm.

Define the key variables: Featuretable, Prefix, NSeg, and Combine.

ACBIC Algorithm:

1) Initialization: Read all content description vectors into Featuretable, and then assign a code value to each vector sequentially; Set Prefix=NULL; Read the processing vector into NSeg;

2) Comparation, Combination and Update: Use Equation (8) to combine NSeg and Prefix to form Combine; Use Equation (5) to judge whether Combine matches any item in Featuretable: if yes, reassign the value of Prefix with Combine, and read the next vector into NSeg; if not, output the value of Prefix as a part of the compressed result; add Combine to Featuretable and reassign the value of Prefix with NSeg; Read the next vector;

3) Output: Output the unique segment as compressed result into a text file; Record the reference relationship between the unsaved segments and the output segments.

Now all necessary segments are saved in the compressed results, meanwhile the repeated ones are recorded only in text form in order to recover when reconstruction. The whole operation is shown in Fig. 3.

D. Programme Reconstruction

Conventional LZW algorithm contains no coding information resulting in complex decoding (e.g. [17] and [18]). But for digital media files the space of the accessory text information is acceptable. We can keep the already existent segments and recover the repeated ones through analyzing the text content.

IV. EXPERIMENTAL RESULTS

Two experiments on the losslessness of ACBIC algorithm and the completeness of the content redundancy are designed and implemented in this section.

In the experiment, we set \( \theta = 0.9 |C_\tau| \) and \( N = 1000 \), \( M = 100 \) in LPC distance based segmentation algorithm. The test digital media files are in MPEG form and selected from the CCTV-1 news media database from which the audio files are extracted quantized at 16 KHz and sampled at 16 bit in monophonic waveform (see [19]). A 12-order LPC is computed once every 20 ms with a 50% overlap. The procedure would be much slower if using more dimensional LPC coefficients and the results don't become better. As a result 12-order is a proper choice.

A. Losslessness Experiment

The losslessness has two meanings here:

- The compressed content is lossless: the number of the segments before compression is the same as the one after reconstruction
- The compression process is reversible: the result of recompressing the reconstructed content is the same as the result of the first compression

Three audio files: U1, U2, and U3 are chosen for the verification of the proposed algorithm. To be simple, in the segmentation stage U1 is taken as an example: first, the LPC distance based segmentation algorithm is applied to the 410s-length U1. The results are listed in Table I and are almost the same as human beings’ segmentation, which proves the validity of the segmentation algorithm. Besides the average time of compression of 1 MB waveform file is less than 1 second which is acceptable.

<table>
<thead>
<tr>
<th>Unit</th>
<th>Begin Frame</th>
<th>End Frame</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unit 1</td>
<td>1</td>
<td>3946</td>
</tr>
<tr>
<td>Unit 2</td>
<td>4112</td>
<td>8040</td>
</tr>
<tr>
<td>Unit 3</td>
<td>8207</td>
<td>12133</td>
</tr>
<tr>
<td>Unit 4</td>
<td>12300</td>
<td>16218</td>
</tr>
<tr>
<td>Unit 5</td>
<td>16385</td>
<td>20312</td>
</tr>
<tr>
<td>Unit 6</td>
<td>20478</td>
<td>24518</td>
</tr>
<tr>
<td>Unit 7</td>
<td>24687</td>
<td>28638</td>
</tr>
<tr>
<td>Unit 8</td>
<td>28804</td>
<td>32720</td>
</tr>
<tr>
<td>Unit 9</td>
<td>32887</td>
<td>36797</td>
</tr>
<tr>
<td>Unit 10</td>
<td>36964</td>
<td>40910</td>
</tr>
</tbody>
</table>
Furthermore the ACBIC algorithm is employed and the results are shown in Table II proving the compression process implemented by the ACBIC algorithm is reversible and lossless as stated before.

**Table II Verification of Losslessness**

<table>
<thead>
<tr>
<th>Name</th>
<th>Original Number/ Compressed Number/ Decompressed Number</th>
<th>Original Size/ Compressed Size/ Decompressed Size (MB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>U1</td>
<td>10/8/10</td>
<td>13/10/10.5/13.1</td>
</tr>
<tr>
<td>U2</td>
<td>5/5/5</td>
<td>1.90/1.90/1.90</td>
</tr>
<tr>
<td>U3</td>
<td>15/10/15</td>
<td>21/15.5/21</td>
</tr>
</tbody>
</table>

B. Completeness Experiment

It is supposed to find all redundant content in the programme file and only keep the unique copy of that content which is called completeness.

First the Content Redundancy Ratio (CRR) is defined in the following:

$$CRR = \frac{\text{size of all repeated units}}{\text{original size}}$$  \hspace{1cm} (10)

And the Compression Ratio (CR):

$$CR = \frac{\text{size of compressed content}}{\text{original size}}$$  \hspace{1cm} (11)

That means

$$CR = 1 - CRR.$$  \hspace{1cm} (12)

**Table III Verification of Completeness of Content Redundancy Detection**

<table>
<thead>
<tr>
<th>Name</th>
<th>Original Size (MB)</th>
<th>CRR (%)</th>
<th>Compressed Size (MB)</th>
<th>CR (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>V1</td>
<td>1.90</td>
<td>0</td>
<td>1.90</td>
<td>100</td>
</tr>
<tr>
<td>V2</td>
<td>43.4</td>
<td>21.1</td>
<td>34.2</td>
<td>78.8</td>
</tr>
<tr>
<td>V3</td>
<td>14.6</td>
<td>50</td>
<td>7.3</td>
<td>50</td>
</tr>
<tr>
<td>V4</td>
<td>71.5</td>
<td>77.4</td>
<td>16.1</td>
<td>22.5</td>
</tr>
</tbody>
</table>

To prove the completeness we choose another 4 representative audio files: V1, V2, V3, and V4. The results in Table III state that the completeness of content redundancy detection is guaranteed.

V. CONCLUSION

We propose an auditory content-based information compression algorithm called ACBIC algorithm in this paper. It can also be used for content-based information retrieval and many other applications. It is able to achieve compression in semantic level through the extraction of auditory information. The LPC distance based programme segmentation is used to segment the media files according to the statistics of the content. Then the content description vectors are extracted and utilized for comparison, combination and update in the compression. As a new kind of compression, the corresponding evaluation standard of performance is built up. Experiments reveal the effectiveness of the proposed algorithm which can guarantee to some extent the losslessness and completeness of the compression process. This algorithm not only can take advantage of semantic structures, but also has promising future in the integration with conventional data compression approaches which will be our near future research.

ACKNOWLEDGMENT

Our thanks to supports from the National Natural Science Foundation of China (61171186), Key Laboratory Opening Funding of MOE-Microsoft Key Laboratory of Natural Language Processing and Speech (HIT.KLOF.2012047 ) and Research Fund for the Doctoral Program of Higher Education No:20050213032. The authors are grateful for the anonymous reviewers who made constructive comments.

REFERENCES


