

Title	[課題研究報告書] ハードウェアにおける高速なオーディオフ ィンガープリントを用いた楽曲全体のマッチング手法
Author(s)	山名, 友也
Citation	
Issue Date	2021-03
Type	Thesis or Dissertation
Text version	author
URL	http://hdl.handle.net/10119/17157
Rights	
Description	Supervisor: 井口 寧, 先端科学技術研究科, 修士(情報科 学)

概要

オーディオフィンガープリンティング (AFP) 技術は、楽曲の知覚的な特徴を用いて生成されるフィンガープリント ID (FPID) によって楽曲の識別を行う技術である。この技術はインターネット上で使用されているが、昨今の高速なインターネット上で使用するには高速な楽曲の検索機能が必要あり、楽曲データから FPID を生成する時間は重要である。

離散ウェーブレット変換 (DWT) を用いた高速でロバスト性の高い FPID を生成する HiFP2.0 アルゴリズムがある。これは、基底関数に Haar ウェーブレットを用いることで整数の加減算とビットシフトのみで処理を構成でき、FPGA などのハードウェア上で高速な処理を実現する。HiFP2.0 アルゴリズムは楽曲の冒頭 2.97 秒間のサンプルデータのみを使用して FPID を生成ので、冒頭 2.97 秒間に問題が起こった場合、楽曲検索が正しく行えない可能性がある。

そこで、本研究では、HiFP2.0 の改良として、サンプル抽出の範囲を楽曲データ全体に広げ、楽曲データ全体から一定量のサンプルデータを周期的に抽出し、FPID を生成する HiFP2.1 アルゴリズムを提案する。

HiFP2.1 アルゴリズムでは、サンプルを抽出する領域 (chunk 領域) を楽曲データ全体に細分化し散在させることで確率的に楽曲データ中の問題発生部位を回避する。これによって HiFP2.0 アルゴリズムよりも高い検索精度を実現できる。

このアルゴリズムは、FPGA などのハードウェアに実装する場合、chunk 領域のサンプルデータを抽出する実装形式として 2 つのパターンが考えられる。一つは FPGA 上で送信してきたデータに対して逐次的に行うもの、一つは FPGA を制御するソフトウェアドライバ側で事前に行うものである。

この時、FPGA 上で行うものにおいては、固定的な回路を形成してパイプライン化などで並列処理できることが利点であるが、ホストとなる PC とハードウェアとの通信がボトルネックとなる。一方で、ソフトウェアで実装する場合は、ハードウェアでの実行時のようなインターフェイスを介した通信時間は考慮しなくてよいが、メモリアクセスなどが頻発すると実行時間が大きく増加してしまう欠点がある。

これらの実装形式を比較した場合、chunk 領域数が増加するとソフトウェアではメモリアクセスの回数などが大きく増加して FPGA 上で行った場合より大きなボトルネックとなってしまう。

このことから本研究では FPGA 上で行うものを選択し、提案した。

また、HiFP2.1 を実装する場合、最適な chunk 領域の楽曲データ全体における chunk 領域の個数を決定しなければならない。

前述した FPGA 上での chunk 領域の抽出方法においては、chunk 領域数が増加しても全体のデータ通信処理量に大きな差が生じない。また、HiFP2.1 が chunk 領域の分散による問題部位の確率的な回避を特徴としている。それに加え、離れた特徴量同士の比較から生成される FPID ビットの割合が大きく増加数とエラーが

起きやすいことが考えられる。

よって、本研究では chunk 領域数 1024 の実装として提案を行った。

このような方式で実装した HiFP2.1 は、HiFP2.0 アルゴリズムに比楽曲データ全体を使用するため、検索精度が向上するが、実行時間が悪化してしまうと考えられる。

そこで本研究では、実際に HiFP2.0 および提案手法の HiFP2.1 を実装し、検索精度及び実行時間について実験を行い、その性能比較を行う。

本研究では提案手法である HiFP2.1 アルゴリズムを FPGA に実装し、提案を行った chunk 領域の抽出方式および chunk 領域数の決定方法についての調査と検証を、AFP による検索精度や実行速度の観点から行った。この検証では、HiFP2.1 のソフトウェア実装、ソフトウェアドライバ上で chunk 領域を抽出する HiFP2.1 の FPGA 実装、HiFP2.0 の FPGA 実装を比較対象とした。

結果として、まず、提案を行った chunk 領域の抽出方法および chunk 領域数の決定方法が最適であると結論付けられた。また、検索失敗率は大幅に改善し最大で HiFP2.0 における検索失敗率と比べ、0.000175 倍となった。一方で、実行時間は 30 秒の楽曲データにおいて HiFP2.0 の実行時間に対して最大で 2.069 倍になった。

今後の展望としては、HiFP2.1 アルゴリズムの並列化を行いたい。また、楽曲データ全体に対するサンプル抽出範囲の大きさと検索精度の関係性を調査し、実行速度を抑えることが出来るかどうかを検証し、見込みがあれば実装したい。

Keywords: オーディオフィンガープリント, FPGA, 音響技術, 離散ウェーブレット変換, ブロードキャストモニタリング

Abstraction

There is a HiFP2.0 algorithm for generating fast and robust fingerprints based on the discrete wavelet transform (DWT). The HiFP2.0 algorithm generates FPIDs using only the first 2.97 seconds of a song. The HiFP2.0 algorithm generates FPIDs using only the sample data of the first 2.97 seconds of the song, so if a problem occurs in the first 2.97 seconds, the song retrieval may not be performed correctly.

Therefore, in this study, as an improvement of HiFP2.0, I propose the HiFP2.1 algorithm, which extends the scope of sample extraction to the entire music data and generates FPIDs by periodically extracting a certain amount of sample data from the entire music data.

In the HiFP2.1 algorithm, the area where samples are extracted (chunk area) is subdivided and scattered over the entire music data to probabilistically avoid problematic areas in the music data. As a result, the proposed method can achieve higher search accuracy than the HiFP2.0 algorithm.

As a result, I have implemented the proposed method. It was concluded that the method of extracting chunk regions and the method of determining the number of chunk regions proposed at the same time were optimal. In addition, the search failure rate was greatly improved, and the maximum search failure rate was 0.000175 times higher than that of HiFP2.0. On the other hand, the maximum execution time was 2.069 times longer than that of HiFP2.0 for 30-second music data.

Keywords: Audio Fingerprinting, FPGA, Acoustic Techniques, Discrete Wavelet Transform, Broadcast Monitoring