The University of Osaka
Institutional Knowledge Archive

Title	パターンマッチングを用いた音楽音響信号探索法および画像圧縮法に関する研究
Author(s)	永野, 秀尚
Citation	大阪大学, 2005, 博士論文
Version Type	VoR
URL	https://hdl.handle.net/11094/168
rights	
Note	

The University of Osaka Institutional Knowledge Archive : OUKA

https://ir.library.osaka-u.ac.jp/

The University of Osaka

パターンマッチングを用いた音楽音響信号探索法
および画像圧縮法に関する研究

２００４年１２月

永野 秀尚

内容梗概

本論文は，筆者がこれまでに大阪大学及び日本電信電話株式会社 NTT コミュニケーション科学基礎研究所にて行ってきたパターンマッチングを用いた音楽音響信号探索法と画像圧縮法に関する研究，及びこれらのメディア処理に関連する研究をまとめたものである．

近年の情報通信機器及びネットワークの普及・高性能化や放送のディジタル化及び多チャンネル化などにより，我々の身の回りには多種多様な音や映像などのメディア情報が溢れている．特に音楽コンテンツや映画などの映像コンテンツの普及はすさまじく，大量のコンテンツが作成され流通している．そのため，単にメディア情報を蓄積，再生するだけでなく，大量のメディア情報の中から必要なメディア情報を高速かつ正確に探し出すメディア探索の技術や，メディア情報を高効率に圧縮する圧縮技術などの，メディア情報を有効に利用するためのメディア処理技術が必要とされている．筆者はメディア情報（特に音楽音響信号）の探索及び画像圧縮に関する研究を行い，特にパターンマッチングを用いた手法に関する研究を進めてきた．パターンマッチングはメディア処理の基本となる技術の一つであり，メディア情報の探索や画像圧縮において幅広く用いられている．そして，本論文では，特に，変動に頑健かつ高速な音楽音響信号の探索手法と，画像圧縮における画像間の距離計算を効率的に並列処理する手法を提案する．

本論文は全 6 章から構成され，その内容は以下の通りである．まず，第 1 章において序論を述べる．そして，第 2 章と第 3 章では，音楽音響信号の探索手法を提案する．音楽音響信号はメディア情報の中でも重要なものの一つであり，その探索技術は強く求められている．例えば，ユーザの利便性のための音楽探索ツールや，著作権管理のための特定音楽の検出や類似音楽の探索技術としても求められている．第 2 章では，多数の小領域スペクトログラムの探索に基づく背景音楽の高速探索法を提案する．ここで，背景音楽の探索とは，音声が音楽に重畳した音響信号（蓄積信号）中から，参照信号として与えられた音楽が音声の背景に含まれる区間（目的区間）を見つけ出すことである．背景音楽の探索では，探索の目的となる目的区間においても，他の音声の重畳により，その音響信号は参照信号と著しく異なり，参照信号を蓄積信号と直接照合する手法での探索は難しい．そのため，高速・高精度な背景音楽の探索は困難であった．そこで，妨害音である音声は時間周波数領域においては疎であることに着目し，参照信号のスペクトログラムを時間周波数領域において多数の小領域に分割し，これらの小領域スペクトログラムと類似度の高い小領域スペクトログラムをもつ時点を蓄積信号のスペクトログラムから探索し，その探索結果を統合することで，目的区間を探索する手法を提案する．本手法では，スペクトログラムを小領域に分割し，かつ類似度の高

い小領域スペクトログラムの組み合わせにだけ着目することで，高精度な背景音楽探索を可能にしている．さらには，類似する小領域スペクトログラムの組み合わせだけに着目することで，そのような小領域スペクトログラムの探索において，非類似部分での照合を探索精度を損なわずに省略することが可能になる．これにより，背景音楽探索全体を高速に行うことができる．背景音楽探索は，例えばラジオ放送中やテレビ放送中で背景音楽として使われた音楽の検出に応用することができる．また，背景音楽として使われている音楽音響信号を探索キーとするビデオ検索システムへの応用なども考えられる．

第3章では，多重奏音響信号を参照信号として多重奏音響信号から類似する音楽を探索する類似音楽探索法を提案する．上述の背景音楽の高速探索法が，重畳している音声以外は同一の音楽音響信号を探索する手法であるのに対し，本手法では，類似する音楽を探索することを目的としている．類似する音楽とは，同一曲を再演奏したもの，他の楽器を用いて演奏したもの，テンポを変えて演奏したもの，もしくは移調して演奏したものなどを指す．本類似音楽探索においては多重奏音響の類似性に基づいた探索を高速に行う必要がある．そこで，まず，多重奏において同時に発生する複数の音の有無に着目した2値多重音響特徴ベクトルを提案する．本特徴は多重奏音響信号の類似性に基づいた高速探索に適している．そしてさらに，本特徴を用いた探索における高速化手法を導入する．本手法は類似度行列のスパース化により探索空間を制限し，より高速な探索を行う．

第4章では，パターンマッチングを用いた画像圧縮において必要な画像間の距離計算を再構成可能ハードウェアを用いて行う手法を提案する．第2章，第3章の探索技術は計算機を用いたソフトウェア処理によるものであるが，第4章は画像の照合をハードウェアを用いて効率的に行う手法に関するものである．組合せ回路や順序回路を再構成できるハードウェアを用いて，圧縮したい画像に応じてハードウェアを再構成することで，必要な距離計算を限られたハードウェア資源で効率よく並列に行うことが可能になる．

第5章では，メディア処理高信頼化のための誤り訂正符号復号法について述べる．誤り訂正符号はメディア処理技術を実際の応用に適用する際に不可欠な技術であり，その復号法はパターンマッチングとも関連が深い．本論文では最尤復号法や準最尤復号法の平均計算時間の削減について述べる．さらに再構成可能ハードウェアを用いた実現についても述べる．

最後に，第6章では，本研究全体のまとめを行った上で，今後の研究課題について述べる．

関連発表論文

1. 学会論文誌発表論文

1. H. Nagano, T. Fujiwara, and T. Kasami, "Average Complexity Evaluation of an MLD Algorithm Using the Trellis Structure for a Linear Block Code," *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol.E78-A, no.9, pp.1209–1214, Sept. 1995.
2. H. Nagano, A. Matsuura, and A. Nagoya, "An Efficient Implementation Method of a Metric Computation Accelerator for Fractal Image Compression Using Reconfigurable Hardware," *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol.E84-A, no.1, pp.372–377, Jan. 2001.
3. 永野秀尚, 柏野邦夫, 村瀬洋, "2 値多重音響特徴ベクトルを用いた類似音楽探索とその高速化," 電子情報通信学会論文誌 D-II, vol.J86-D-II, no.11, pp.1657–1667, Nov. 2003.
4. 永野秀尚, 柏野邦夫, 村瀬洋, "多数の小領域スペクトログラムの探索に基づく背景音楽の高速探索法," 電子情報通信学会論文誌 D-II, vol.J87-D-II, no.5, pp.1179–1188, May 2004.

2. 研究会等発表論文（査読付）

1. H. Yamamoto, H. Nagano, T. Fujiwara, and T. Kasami, "Recursive MLD Algorithm Using the Detail Trellis Structure for a Linear Block Code and Its Average Complexity Analysis," *Proceedings of 1996 International Symposium on Information Theory and Its Applications (ISITA '96)*, vol.II, pp.704–708, Sept. 1996.
2. H. Nagano, T. Suyama, and A. Nagoya, "Soft Decision Maximum Likelihood Decoders for Binary Linear Block Codes Implemented on FPGAs," *Proceedings of 1998 ACM SIG/DA Sixth International Symposium on Field Programmable Gate Arrays (FPGA '98)*, p.261, Feb. 1998.
3. H. Nagano, T. Suyama, and A. Nagoya, "Acceleration of Linear Block Code Evaluations Using New Reconfigurable Computing Approach," *Proceedings of Asia and South Pacific Design Automation Conference 1999 (ASP-DAC '99)*, pp.161–164, Jan. 1999.

4. A. Matsuura, H. Nagano, and A. Nagoya, "A Method for Implementing Fractal Image Compression on Reconfigurable Architecture," *Proceedings of ACM International Symposium on Field-Programmable Gate Arrays (FPGA '99)*, p.251, Feb. 1999.

5. H. Nagano, A. Matsuura, and A. Nagoya, "An Efficient Implementation Method of Fractal Image Compression on Dynamically Reconfigurable Architecture," *Proceedings of 6th Reconfigurable Architectures Workshop (RAW '99) in IPPS/SPDP'99 Workshops —Parallel and Distributed Processing*, LNCS 1586, pp.670–678, Springer-Verlag, April 1999.

6. H. Nagano, K. Kashino, and H. Murase, "Fast Music Retrieval Using Polyphonic Binary Feature Vectors," *Proceedings of 2002 IEEE International Conference on Multimedia and Expo (ICME 2002)*, vol.I, pp.101–104, Aug. 2002.

7. H. Nagano, K. Kashino, and H. Murase, "A Fast Search Algorithm for Background Music Signals Based on the Search for Numerous Small Signal Components," *Proceedings of 2003 International Conference on Acoustics, Speech, and Signal Processing (ICASSP '03)*, vol.V, pp.796–799, April 2003 (canceled due to SARS), also in *Proceedings of 2003 IEEE International Conference on Multimedia and Expo (ICME 2003)*, vol.I, pp.165–168, July 2003.

3. その他の研究会等発表論文

1. H. Nagano, T. Fujiwara, and T. Kasami, "Complexity Evaluation of an MLD Algorithm for a Linear Block Code Using Its Trellis Structure," *Proceedings of 17th Symposium on Information Theory and Its Applications (SITA '94)*, pp.209–212, Dec. 1994.

2. T. Koumoto, H. Nagano, T. Takata, T. Kasami, T. Fujiwara, and S. Lin, "A New Iterative Soft-Decision Decoding Algorithm," 電子情報通信学会技術研究報告, IT-95-28, pp.19–24, July 1995.

3. H. Nagano, T. Fujiwara, and T. Kasami, "A Decoding Algorithm for a Linear Block Code Using the Detailed Structure of its Trellis Diagram with Partial Search," *Proceedings of 18th Symposium on Information Theory and Its Applications (SITA '95)*, pp.449–452, Oct. 1995.

4. T. Koumoto, H. Nagano, T. Takata, T. Kasami, T. Fujiwara, and S. Lin, "An Iterative Soft-Decision Decoding Algorithm," *Proceedings of 18th Symposium on Information Theory and Its Applications (SITA '95)*, pp.557–560, Oct. 1995.

5. 松浦昭洋, 永野秀尚, 名古屋彰, "フラクタル画像圧縮の再構成可能アーキテクチャによる実現法," 電子情報通信学会技術研究報告, CPSY98-83, pp.15–22, Sept. 1998.

6. 永野秀尚, 須山敬之, 名古屋彰, "再構成可能なハードウェアを用いた線形ブロック符号の性能評価の高速化," 電子情報通信学会技術研究報告, ICD98-290, VLD98-144, pp.25–32, March 1999.

7. 永野秀尚, 柏野邦夫, 村瀬洋, "類似度に基づく曖昧文字列照合法と音楽検索への適用," 情報処理学会研究報告, 2001-MUS-41, pp.67–72, Aug. 2001.

8. 永野秀尚, 柏野邦夫, 村瀬洋, "メディア探索のための曖昧文字列照合の高速計算法," 電子情報通信学会 情報・システムソサイエティ大会講演論文集, p.197, Sept. 2001.

9. 永野秀尚, 柏野邦夫, 村瀬洋, "多数の小区間信号の探索に基づく背景音楽の探索," 電子情報通信学会技術研究報告, PRMU2002-86, WIT2002-29, pp.37–41, Sept. 2002.

10. 永野秀尚, 柏野邦夫, 藤原融, "2 値多重音響特徴ベクトルを用いた類似音楽探索法の頑健性評価," 情報処理学会研究報告, 2004-MUS-58, pp.39–44, Dec. 2004.

以上

関連公開特許出願

1. 松浦 昭洋, 永野 秀尚, 名古屋 彰, “再構成可能アーキテクチャを用いた画像符号化方法及びそのための装置, 及びそのための方法を記述したプログラムを記録した記録媒体,” 公開番号 特開 2000-102005, 公開日 2000.4.7, 出願番号 特願平 10-266480, 出願日 1998.9.21, 出願人 日本電信電話株式会社.

2. 永野 秀尚, 松浦 昭洋, 名古屋 彰, “再構成可能なハード ウェアを用いた画像符号化方法, 画像符号化装置および画像符号化のためのプログラム記録媒体,” 公開番号 特開 2000-295613, 公開日 2000.10.20, 出願番号 特願平 11-102010, 出願日 1999.4.9, 出願人 日本電信電話株式会社.

3. 柏野 邦夫, 永野 秀尚, 村瀬 洋, ブライアン トス, “類似音楽検索装置ならびにその方法, および類似音楽検索プログラムならびにその記録媒体,” 公開番号 特開 2003-44082, 公開日 2003.2.14, 出願番号 特願 2001-236948, 出願日 2001.8.3, 出願人 日本電信電話株式会社.

4. 永野 秀尚, 柏野 邦夫, 村瀬 洋, “類似部分系列検出方法および装置, 類似部分系列検出プログラムならびにその記録媒体,” 公開番号 特開 2003-50823, 公開日 2003.2.21, 出願番号 特願 2001-236949, 出願日 2001.8.3, 出願人 日本電信電話株式会社.

5. 永野 秀尚, 柏野 邦夫, 村瀬 洋, “特定音響信号検出方法, 信号検出装置, 信号検出プログラム及び記録媒体,” 公開番号 特開 2004-102023, 公開日 2004.4.2, 出願番号 特願 2002-265427, 出願日 2002.9.11, 出願人 日本電信電話株式会社.

以上

目 次

第 1 章　序章 **1**
1.1　研究の背景 1
1.2　研究の目的 4
1.3　本論文の構成 6

第 2 章　多数の小領域スペクトログラムを用いた背景音楽の高速探索法 **7**
2.1　まえがき 7
2.2　対象とする背景音楽探索と従来の探索法 9
2.3　背景音楽探索法 12
2.3.1　スペクトログラムの分割一致探索による背景音楽探索 12
2.3.2　時系列アクティブ探索法を用いたスペクトログラム分割一致探索による背景音楽探索 14
2.4　提案手法の評価 17
2.4.1　探索精度の評価実験 17
2.4.2　探索速度の評価実験 21
2.5　むすび 23

第 3 章　2 値多重音響特徴ベクトルを用いた類似音楽探索法 **25**
3.1　まえがき 25
3.2　対象とする類似音楽探索と従来の探索法 27
3.3　類似音楽探索法 28
3.3.1　曖昧系列照合による類似音楽探索 28
3.3.2　2 値多重音響特徴ベクトルを用いた曖昧系列照合による類似音楽探索とその高速化 29
3.4　提案手法の評価 40
3.4.1　探索精度の評価実験 41
3.4.2　探索速度の評価実験 45
3.5　むすび 46

第 4 章　フラクタル画像圧縮のための画像間距離の高速計算回路の実現法　49
4.1 まえがき 49
4.2 対象とするフラクタル画像圧縮 50
4.3 画像間距離計算回路の再構成可能ハードウェアを用いた実現法 53
4.3.1 再構成可能ハードウェア 53
4.3.2 計算回路のアーキテクチャとその実現法 54
4.4 提案手法の評価 57
4.4.1 回路規模の評価 57
4.4.2 画像のマッチング回数の評価 60
4.5 むすび 61

第 5 章　メディア処理高信頼化のための誤り訂正符号復号法　63
5.1 まえがき 63
5.2 対象とする誤り訂正符号復号法 65
5.3 準最尤復号アルゴリズムとヴィタビ復号法の並列処理 73
5.3.1 トレリスダイアグラムの詳細構造を用いた部分探索による準最尤復号アルゴリズム 73
5.3.2 ヴィタビ復号法の再構成可能ハードウェアを用いた実現法 74
5.4 各手法の評価 76
5.4.1 分割トレリス復号法の平均計算量の評価 76
5.4.2 部分探索復号法の平均計算量と復号誤り率の評価 82
5.4.3 再構成可能ハードウェアを用いたヴィタビ復号器の速度評価 84
5.5 むすび 85

第 6 章　結論　87
6.1 本論文のまとめ 87
6.2 今後の研究課題 88

謝辞　91

参考文献　93

第1章　序章

1.1　研究の背景

近年の情報通信機器及びネットワークの普及・高性能化や放送のディジタル化及び多チャンネル化などにより，我々の身の回りには多種多様な音や映像などのメディア情報が溢れている．特に音楽コンテンツや映画などの映像コンテンツの普及はすさまじく，大量のコンテンツが作成され流通している．そのため，単にメディア情報を蓄積，再生するだけでなく，メディア情報を有効に利用するためのメディア探索技術，メディア圧縮技術，メディア伝送技術などが必要とされている．メディア探索技術は，必要なメディア情報を高速かつ正確に探し出すことが要求され，メディア圧縮技術は，高速かつ高効率に圧縮することが要求される．また，メディア伝送技術は，高速かつ正確に情報を伝送することが要求される．筆者はメディア情報（特に音楽音響信号）の探索及び画像圧縮に関する研究を行い，特にパターンマッチングを用いた手法に関する研究を進めてきた．パターンマッチングはメディア処理の基本となる技術の一つであり，メディア情報の探索や画像圧縮において幅広く用いられている．そして，本論文では，特に，変動に頑健かつ高速な音楽音響信号の探索手法，画像圧縮における画像間の距離計算を効率的に並列処理する手法，さらに，メディア伝送技術において重要な役割をもつ誤り訂正符号の復号法を提案する．以降では，音楽音響信号の探索，画像圧縮，そして，誤り訂正符号復号法について，その研究の背景を述べる．

音楽はメディア情報の中でも重要なものの一つであり，その探索技術は強く求められている．例えば，ユーザの利便性のための音楽探索ツールや，著作権管理のための特定音楽の検出や類似音楽の探索技術としても求められている．また，音楽については，その音響信号自体に限らず曲そのものにも著作物としての価値があり，曲として同じものを探索することも重要である．このような，メディア探索の重要な応用例である音楽探索についてはこれまで多くの研究がなされてきたが，それらは，その目的により二つに大別できる．一つは音楽CDなどと信号レベルでほぼ同一の音楽の探索をねらう一致音楽探索であり，もう一つは信号レベルでは必ずしも一致していないが，演奏者が異なる同一の曲，主旋律が同じ曲，アレンジの異なる曲などのように，何らかの意味で類似している音楽の探索をねらう類似音楽探索である．

本論文では，これらの探索において，探したい音楽を探索キーと呼ぶ．探索においては，音響信号などの音楽を表すデータの集合から探索キー（またはその類似の音楽）が含まれている区間を探すことになるが，この音楽のデータの集合を蓄積データベースと呼ぶ．以下では蓄積データベースを単に DB と書き，DB 内の一つの音楽のデータを蓄積データと呼ぶ．さらに，探索キーが音響信号として与えられるとき，これを特に参照信号と呼ぶ．そして，蓄積データが音響信号であるとき，これを特に蓄積信号と呼ぶ．

一致音楽探索と類似音楽探索のうち，一致音楽探索は，参照信号の音響特徴を蓄積信号の音響特徴と照合し，一致する部分を検出する同一信号探索により実現できる．この同一信号探索については，それを高速に行う手法や，特徴歪みに頑健にする手法などが提案されている [1, 2, 3, 4, 5]．しかし，これらの手法における探索の目的となる区間では，音響信号そのものが探索キーとほぼ同一であることを前提としている．このため，例えば，ナレーションが重畳した音楽を探索する場合，たとえ元の音楽の音響信号は参照信号と同一であっても，他の音の重畳により音響信号が激しく変化するため，探索精度が低下するという問題があった．

一方，類似音楽探索についても様々な手法がその目的に応じて提案されている [6, 7, 8, 9, 10, 11]．一般に類似音楽探索においては，蓄積信号と参照信号で音響特徴がかなり異なる．また，探索の対象となる蓄積データが音響信号以外に楽譜などで表現されるため，音響特徴以外の特徴を用いた探索が必要となる．特に Ghias らは単旋律の音高系列を “S（同じ）”，“U（上がる）”，“D（下がる）” の相対音高の文字列で表す旋律輪郭（melodic contour）を用いた単旋律の照合手法を提案し，ハミングを探索キーとして，類似する単旋律を探索する類似音楽探索においてその有効性を示した [7]．旋律輪郭は旋律の表現としては楽譜などに比べ単純であるが，単旋律の識別，照合には有効であり，その他の研究においても単旋律の探索における特徴として用いられている．旋律輪郭を用いた探索法を含め，これまでの類似音楽探索は単旋律同士の照合による探索であった．また，これら多くの手法における DB の蓄積データは，MIDI や楽譜情報，もしくは探索キーと同じくハミングによる主旋律の入力から得られる，既に主旋律などが抽出された単旋律の音高系列であった．音響信号からなる DB を対象とした類似音楽探索については，橋口らが，ハミングによる単旋律音高系列を探索キーとし，単旋律対多重奏の照合を前提とした手法を提案している [11]．しかし，この手法においても探索キーはハミングという単旋律を前提としている．

一方，類似音楽探索の応用を考えた場合，探索キー及び探索対象の DB も音楽音響信号としながら類似する音楽を探索することも強く求められている．しかし，通常，多重奏である音楽音響信号を探索キーとし，同じく通常の音楽音響信号からなる DB から類似する音楽を探索する場合に単旋律の探索キーや DB を前提とした照合法を用いるには，多重奏からの主

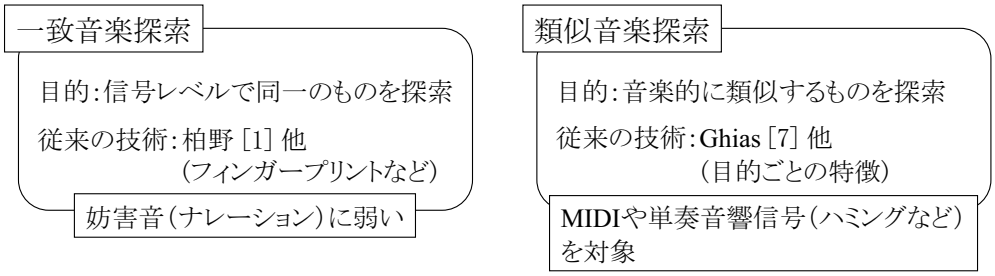


図 1.1: 音楽探索技術の状況

旋律抽出が必要となること，もしくは多重奏と多重奏の間で主旋律を照合する必要があることなど困難な問題が多い [12, 13]．このため，音楽音響信号（多重奏音響信号）同士の比較に基づく類似音楽探索の手法は提案されていなかった．

以上のように音楽探索については多くの研究がなされてきたが，対象を音楽音響信号とした場合，一致探索の枠組においても変動に対する頑健性は十分でない．また，類似音楽探索の枠組においては，探索キーと DB をともに音楽音響信号とした手法の検討自体が十分になされているとはいえない．音楽探索において，一致音楽探索と類似音楽探索の両方の枠組で，音響信号を対象とすることは，その応用面からも重要である．そして，その応用においては，変動に対する探索技術の頑健性が必要になる．実際，一致音楽探索の代表的な応用例である，放送における音楽使用検出においては，音楽へのナレーションの重畳が非常に多くある．また，類似音楽探索の応用では，例えばライブ演奏などのように曲としては同一であるが音響信号としては異なる音楽の検出などが求められる．このように，音楽音響信号を対象とした音楽探索を考えた場合，妨害音の重畳のような信号レベルの変動から，演奏の違いという変動まであり，これらの変動に対する音楽探索技術の頑健性が求められるが，それはまだ十分に実現されているとはいえない．図 1.1 にこれまでの音楽探索技術の状況をまとめる．

このような状況下，本論文においては，変動に頑健な音楽音響信号の探索法を提案する．これらの手法で目的とする音楽音響信号の探索は，現実の音楽音響信号を対象とし，これらの信号間の類似性を測り，信号が目的の信号であるかを判定するということを計算機に行わせることが必要である．そのため，提案手法は信号というパターンを照合し，その類似性を測るパターンマッチングを基本とする．

パターンマッチングはパターンの類似性を測るという基本的な処理であり，上述のメディア探索においても基本的な処理となる以外，画像や映像の圧縮においても基本的な処理と

して用いられる．そのため，パターンマッチングを高速に行うことは意義がある．そこで，本論文では，画像圧縮のなかでもフラクタル画像圧縮を例にとり，ハードウェアを用いてこのようなパターンマッチングを効率よく処理する手法も検討する．ここでは，特に再構成可能ハードウェアを用いることで，汎用的な計算手段としてパターンマッチングを高速かつ効率的に処理する方法を検討する．

また，先に述べたメディア処理技術を実際にサービスとして提供する際には，それらのサービスにおいて，メディアデータを正確に伝送する必要がある．例えば，メディア探索技術においても，ユーザが流れている音楽を携帯電話でキャプチャ・伝送し，これを探索キーとして，これと同じ音楽を DB から探索することでその音楽の曲名をユーザに教えるような楽曲探索サービスを考えた場合，探索キーの伝送を正確に行うことは非常に重要である．また，圧縮したメディアデータを伝送する際も，元のメディアデータから冗長性が失われるだけに，圧縮されたデータをより正確に伝送する必要がある．信頼性の高いデータの伝送のための技術としては誤り訂正符号がある [14, 15, 16]．誤り訂正符号はデータの誤りの検出・訂正を目的として構成される符号である．対象とする誤りは通信路で発生するランダム誤りやバースト誤りの他に，磁気テープや磁気ディスクなどの外部記憶装置で発生する誤りや演算装置で発生する誤りなどがあり，それぞれに対する誤り訂正符号が考えられ，通信，計算機，また，オーディオ・ビデオなどの分野で幅広く応用されている．そして特に通信では，近年の通信量の著しい増大に伴い，帯域幅をできるだけ小さくしながら高速かつ信頼性の高い通信を行うことが求められている．そのため，より高性能な誤り訂正符号の研究開発が進められている．本論文では，メディア処理技術の高信頼化のために，誤り訂正符号復号法の平均処理時間の削減やハードウェアでの実現を行う．

1.2　研究の目的

前節で述べたように，これまで音楽探索においては様々な手法が提案されてきた．しかし，音楽音響信号を対象とした場合，一致音楽探索においてもナレーションの重畳などの変動への頑健性は十分でなく，さらに音楽的な変動を扱う類似音楽探索の枠組においては音楽音響信号を対象とした探索手法の検討自体が十分ではなかった．

そこで，本論文では，まず，変動に頑健な音楽音響信号の高速探索法の開発を目的とし，パターンマッチングを用いた音楽音響信号の探索法を二つ提案する．最初の探索法は，多数の小領域スペクトログラムを用いた背景音楽の高速探索法である．本背景音楽探索法は，信号レベルで同一の音楽を探索する一致音楽探索の一つであるが，目的の音楽にナレーションなどの他の音声がかぶさった場合にも高速かつ高精度に目的の音楽を探索する手法である．

本背景音楽探索法は，音楽の著作権管理を目的とした音楽使用検出などに応用でき，例えばラジオ放送中や TV 放送中で背景音楽として使われた音楽の検出に応用することができる．この場合，各放送を録音したものを蓄積信号として蓄積しているとする．そして，ある音楽の CD の一部を参照信号とし，蓄積信号で他の音声の背景に参照信号と同じ音楽が流れた区間を検出することで，その音楽が背景音楽として使われた区間を検出することができる．また，他の応用としては，背景音楽として使われている音楽音響信号を探索キーとするビデオ検索システムなども考えられる．

第二の探索法は，2 値多重音響特徴ベクトルを用いた類似音楽探索法である．本手法は，多重奏音響信号を参照信号とし，同じく多重奏音響信号の中から，参照信号に類似する音楽を探索する．背景音楽の高速探索法が，重畳している音声以外は同一の音楽音響信号を探索する手法であるのに対し，本手法では，類似する音楽を探索することを目的としている．類似する音楽とは，同一曲を再演奏したもの，他の楽器を用いて演奏したもの，テンポを変えて演奏したもの，もしくは移調して演奏したものなどを指す．このような類似音楽探索の枠組は，類似の音楽を探索する音楽探索ツールとしてだけでなく，背景音楽探索と同様に，音楽の著作権管理のための音楽検出ツールとしても重要である．

上述の音楽音響信号の探索法はパターンマッチングを基本としている．また，先に述べたようにパターンマッチングはメディア処理の基本技術の一つであり，パターンマッチングを効率よく行うことは重要であり，パターンマッチングの効率的な実現法も本論文の目的である．そこで，本論文では，フラクタル画像圧縮を例にとり，フラクタル画像圧縮において行われるパターンマッチングである画像間の距離計算を再構成可能ハードウェアを用いて行う手法を提案する．特にここでは，再構成可能ハードウェアを用い，画像に特化して回路を構成することで，画像間の距離計算を少ないハードウェア資源で効率的に並列処理する手法を提案する．

また，前節で述べたように，誤り訂正符号はメディア処理技術を実際の応用に適用する際に不可欠な技術である．そこで，本論文では，最後に，メディア処理高信頼化のための誤り訂正符号復号法について述べる．誤り訂正符号はデータの誤りの検出・訂正を目的として構成される符号である．その本質は，雑音などの変動が加わった情報から誤りを訂正し，元の情報を復元することであり，パターンの認識・探索や信号処理との関連が深い．特に本論文では，パターンマッチングとの関連が深く，そして通信分野においても重要な線形ブロック符号とその軟判定最尤復号を対象とし，その復号法について述べ，また，より高速な準最尤復号法と，最尤復号の再構成可能ハードウェアによる実現法を提案する．

1.3　本論文の構成

本論文は全6章から構成され，その内容は以下の通りである．第2章では，多数の小領域スペクトログラムの探索に基づく背景音楽の高速探索法を提案する．背景音楽の探索では，探索の目的となる目的区間においても，他の音声の重畳により，その音響信号は参照信号と著しく異なり，参照信号を蓄積信号と直接照合する手法での探索は難しい．そこで，妨害音である音声は時間周波数領域においては疎であることに着目し，参照信号のスペクトログラムを時間周波数領域において多数の小領域に分割し，これらの小領域スペクトログラムと類似度の高い小領域スペクトログラムをもつ時点を蓄積信号のスペクトログラムから探索し，その探索結果を統合することで，目的区間を探索する手法を提案する．本手法では，スペクトログラムを小領域に分割し，かつ類似度の高い小領域スペクトログラムの組み合わせにだけ着目することで，高精度な背景音楽探索を可能にしている．さらに，類似する小領域スペクトログラムの組み合わせだけに着目することで，そのような小領域スペクトログラムの探索において，非類似部分での照合を探索精度を損なわずに省略できる．

第3章では，多重奏音響信号を参照信号として多重奏音響信号から類似する音楽を探索する類似音楽探索法を提案する．類似する音楽とは，同一曲を再演奏したもの，他の楽器を用いて演奏したもの，テンポを変えて演奏したもの，もしくは移調して演奏したものなどを指す．本類似音楽探索においては多重奏音響の類似性に基づいた探索を高速に行う必要がある．そこで，多重奏において同時に発生する複数の音の有無に着目した2値多重音響特徴ベクトルを提案する．本特徴は多重奏音響信号の類似性に基づいた高速探索に適している．さらに，本特徴を用いた探索における高速化手法を導入する．本手法は類似度行列のスパース化により探索空間を制限し，より高速な探索を行う．

第4章では，パターンマッチングを用いた画像圧縮において必要な画像間の距離計算を再構成可能ハードウェアを用いて行う手法を提案する．組合せ回路や順序回路を再構成できるハードウェアを用いて，圧縮したい画像に応じてハードウェアを再構成することで，必要な距離計算を限られたハードウェア資源で効率よく並列に行うことが可能になる．

第5章では，メディア処理高信頼化のための誤り訂正符号復号法について述べる．誤り訂正符号はメディア処理技術を実際の応用に適用する際に不可欠な技術であり，その復号法はパターンマッチングとも関連が深い．本論文では最尤復号法や準最尤復号法の平均計算時間の削減について述べる．さらに再構成可能ハードウェアを用いた実現についても述べる．

最後に，第6章では，本研究全体のまとめを行った上で，今後の研究課題について述べる．

なお，第2章は文献 [17, 18, 19]，第3章は文献 [20, 21, 22, 23, 24]，第4章は文献 [25, 26, 27, 28]，第5章は文献 [29, 30, 31, 32, 33, 34, 35, 36, 37] で公表した内容に基づき論述する．

第2章　多数の小領域スペクトログラムを用いた背景音楽の高速探索法

2.1　まえがき

第1章で述べたように，音や映像のメディア情報の増加，多様化により様々なメディア情報の探索技術が必要とされている．筆者の属する研究グループでは，時系列探索，すなわち，長時間の音や映像の信号（蓄積信号）と，探したい音や映像の信号（参照信号）が探索キーとして与えられたとき，蓄積信号中の参照信号に類似する区間を探し出す探索技術について研究を進めてきた [1, 2, 4, 17, 20, 38]．そして，特に，本章では，音楽に音声が重畳した蓄積信号中から参照信号として与えられる音楽が音声の背景に含まれる区間（目的区間）を見つけ出す背景音楽探索を高速に行う手法について述べる．背景音楽探索においては，ナレーションなど他の音声が音楽へ重畳されうるものとする．そのため，目的区間では，参照信号として与えられる音楽と同一の音が背景に鳴ってはいるが，前景である音声が重なっているために，音響信号そのものは参照信号と大きく異なりうる．背景音楽探索は，例えばラジオ放送中やテレビ放送中で背景音楽として使われた音楽の検出に応用することができる．この場合，各放送を録音したものを蓄積信号として蓄積しているとする．そして，ある音楽の CD の一部を参照信号とし，蓄積信号中の目的区間を検出することで，その音楽が背景音楽として使われた区間を検出することができる．また，他の応用としては，背景音楽として使われている音楽音響信号を探索キーとするビデオ検索システムなども考えられる．

第1章で述べたように，音楽探索は，その目的により二つに大別できる．一つは音響信号レベルで同一の音楽の探索をねらう一致音楽探索で，もう一つは類似音楽探索である．そして，この一致音楽探索は，目的区間の音響信号自体が参照信号とほぼ同一である場合の探索を行う全景音楽探索と，目的区間において音声の背景に流れる音楽が参照信号と信号レベルで同一である場合の探索を行う背景音楽探索に分けられる．このうち，全景音楽探索は，参照信号と信号レベルで同一の音響信号区間を蓄積信号中から検出する同一信号探索 [1, 2, 3, 4, 5, 39] により実現できる．この同一信号探索は，参照信号を蓄積信号上でずらしながら蓄積信号上のすべての時点で照合するずらし照合法（全探索法と呼ぶ）により実現

技術的課題：音声が目的の音楽にかぶさる

・時系列アクティブ探索法（TAS）：
非類似部分との照合をスキップし，
一致部分のみを探索 → 探索精度低下

・自己最適化スペクトル相関法（SSC）：
類似性を詳細に計算し全探索 → 計算量大

図 2.1: 背景音楽探索における従来法の課題

できる．このとき，音響信号の照合は，スペクトログラムを照合し相関を計算することなどで行われる．スペクトログラムは音響信号中の時間・周波数ごとの音の強さ（パワー）を表したものである [40]．一方，この同一信号探索の高速な手法として時系列アクティブ探索法（time-series active search，本論文では以下 TAS と表記する）が提案されている [1]．TAS ではスペクトログラムの照合においてヒストグラム特徴を用いることと，類似しないスペクトログラムとの照合を省略し，探索区間をスキップすることで，スペクトログラムの相関を用いた全探索法（相関法）よりも高速な探索が行える．そして，TAS を特徴ひずみに対し頑健にする手法 [2, 4] や参照信号中の部分区間の探索に用いる応用手法なども提案されている [5]．しかし，これらの同一信号探索の手法は，探索の目的となる区間では，音響信号そのものが参照信号とほぼ同一であることを前提としており，背景音楽探索のように，他の音声など，参照信号以外の音が重畳している場合には探索精度が低下する．

一方，背景音楽探索に関連する手法としては，最近，自己最適化スペクトル相関法（self-optimized spectral correlation method，本論文では SSC と表記する）が提案されている [41]．SSC は背景音楽同定のための手法であるが，以下のように，スペクトログラムの相関を計算しながら全探索を行うことで背景音楽探索を行うことができる．SSC においては参照信号のスペクトログラムを時間周波数領域で小領域のスペクトログラムに分割し，この小領域スペクトログラムと蓄積信号のスペクトログラム中の小領域スペクトログラムとの類似度を，スケールパラメータの最適化により最大化された相関とする．SSC では，参照信号のスペクトログラムを分割して得られる各小領域スペクトログラムについて，全探索法により，この小領域スペクトログラムと，蓄積信号中の各時点における小領域スペクトログラムとの相関を計算する．そして，その計算結果を統合し探索結果を得る．このようにして，SSC では背景音楽探索が可能になる．しかし，SSC は，その計算量の大きさから，長時間の蓄積信号からの探索には時間がかかるという問題点がある．

以上の背景音楽探索における従来法の課題をまとめると図 2.1 のようになる．従来の高速な同一信号探索である TAS は，音声が目的の音楽に重畳すると探索精度が低下するため適用できず，SSC は探索に時間がかかるという問題がある．

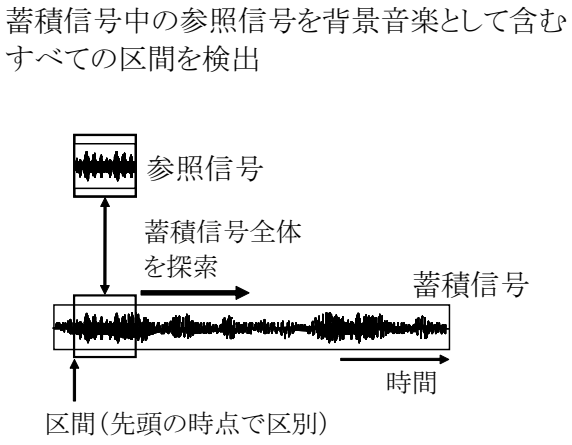


図 2.2: 対象とする背景音楽探索

そこで，本論文では，背景音楽探索の高速かつ高精度な手法を提案する．提案の内容は 2 点に要約される．第 1 点はスペクトログラムの分割一致探索法による背景音楽探索である．分割一致探索法においては，信号のスペクトログラムを小領域スペクトログラムに分割し，参照信号の小領域スペクトログラムと一致する小領域スペクトログラムを多く含む区間を蓄積信号から探索することにより背景音楽探索を実現する．提案の第 2 点は，このような一致する小領域スペクトログラムの組合せを TAS により高速に見つけ出す点である．上述の分割一致探索法においては，一致する小領域スペクトログラムの組合せだけに着目することにより，一致する小領域スペクトログラムの探索を同一信号探索としてよく，そのため，この探索に TAS の適用が可能になる．この TAS の適用により高速な背景音楽探索が可能になる．

以下，2.2 節で対象とする背景音楽探索について述べ，2.3 節で提案手法について述べる．そして，2.4 節で実験結果について述べ，2.5 節を本章のまとめとする．

2.2　対象とする背景音楽探索と従来の探索法

本論文で目的とする背景音楽探索は，図 2.2 に示すように，参照信号で与えられる音楽音響が，蓄積信号中で他の音声の背景に含まれる区間をすべて見つけ出すことである．なお，参照信号の音楽音響が含まれる蓄積信号中の区間を目的区間と呼ぶ．また，信号中の各区間はその先頭の時点で区別する．

本論文における背景音楽探索では，目的区間において他の音声の背景にある音楽の音響信号自体は音量を除いて参照信号と同一の音響信号であるものとする．また，この目的区間の音楽の音量は，それに重畳する他の音声の音量より小さいこともありえるとする．実際のテ

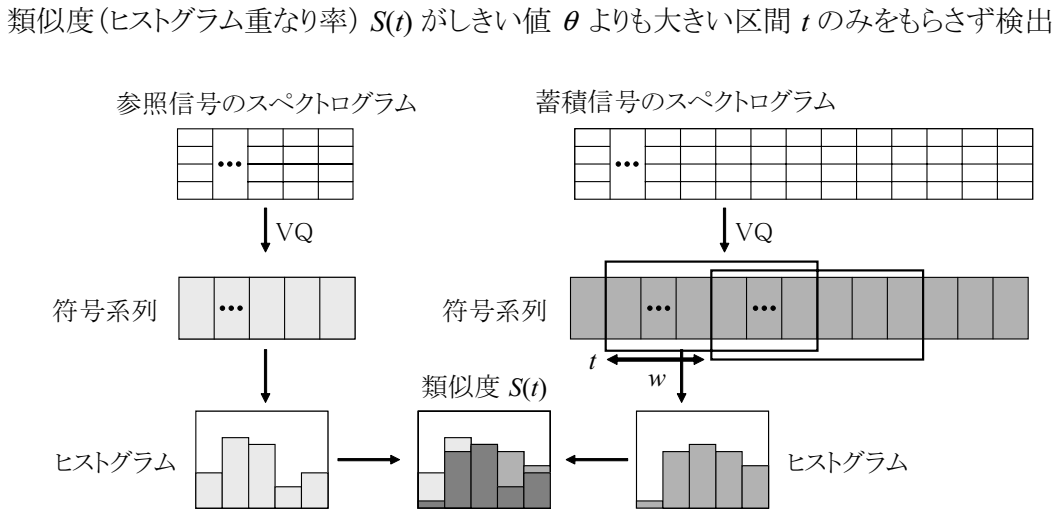


図 2.3: TAS の概要

レビ放送などでは，背景音楽は他の音声と同等の音量もしくは，他の音声よりも小さい音量となる．そのため，音楽音響信号をソース，他の音声をノイズとしたとき，SN 比としては +5dB から −15dB 程度を想定する．

参照信号については，実際の放送からの特定楽曲の検出を想定し，15 秒程度の他の音声などが重畳していない元の音楽の音響信号が与えられるものとする．また，蓄積信号については実際のテレビ放送番組などを想定し，数時間程度の音響信号を想定する．

ここで述べた背景音楽探索では，目的の音楽に他の音声が重畳されうるものの，目的の音楽自体は参照信号と同じである．そのため，この背景音楽探索は一致音楽探索の一つと考えられる．2.1 節で述べたように，一致音楽探索のうち，全景音楽探索については，同一信号探索の高速な手法として TAS が提案されている．また，背景音楽探索については SSC を基にすることが可能である．以下，本論文に関連の深い TAS と SSC について説明する．

TAS の概要を図 2.3 に示す．TAS は，蓄積信号のスペクトログラムから，参照信号のスペクトログラムとのヒストグラム重なり率による類似度がしきい値 θ より大きいスペクトログラムを持つ参照信号と同じ長さの区間を探索する．TAS では，まず，各スペクトログラムについて，スペクトログラム上の各時点のスペクトル特徴を正規化[1]した上で，ベクトル量子化（VQ）[2]により符号化し，各スペクトログラムについて，VQ 符号の系列（符号系列）を作成しておく．このとき，あるサイズのスペクトログラムについて，その中でのこの VQ 符号の出現回数を数え上げたヒストグラムをこのスペクトログラムのヒストグラム特徴という．時間方向と周波数方向で同じサイズのスペクトログラム X と Y について，それらの

[1]本論文での正規化は，各時点のスペクトル特徴の各要素を，それらの平均値で正規化することで行った．
[2]本論文では，VQ の符号帳（code book）の学習には LBG アルゴリズムを用いた．

ヒストグラム特徴を h^X と h^Y としたとき，X と Y のヒストグラム重なり率 $S_h(h^X, h^Y)$ を

$$S_h(h^X, h^Y) = \frac{1}{D}\sum_{i=1}^{M}\min(h_i^X, h_i^Y) \tag{2.1}$$

とする．ここで，h_i^X と h_i^Y は，それぞれ h^X と h^Y の i 番目のビンに含まれる度数であり，M はヒストグラムのビンの数，D は一つのヒストグラムの総度数である．TAS においては，このヒストグラム重なり率をスペクトログラムの類似度に用いる．そして，参照信号のスペクトログラムと蓄積信号中の区間 t のスペクトログラムのヒストグラム重なり率を $S(t)$ とするとき，t で照合の後，$S(t)$ を用いて次の照合位置までのスキップ幅 w を

$$w = \begin{cases} \lfloor D(\theta - S(t)) \rfloor + 1, & S(t) < \theta \text{ のとき} \\ 1, & \text{それ以外のとき} \end{cases} \tag{2.2}$$

として計算する．ここで，$\lfloor x \rfloor$ は x を超えない最大の整数を表す．次に，照合位置を w だけずらして照合を行い，新たにスキップ幅を計算する．TAS では，これを繰り返していくことで，探索を行う．そして，照合した区間でのヒストグラム重なり率が θ より大きい場合，その区間を検出する．TAS では，このようにしてスキップにより照合回数を削減しながらもヒストグラム重なり率がしきい値 θ より大きい区間をもらさず探索できる．

以上が TAS の概要である．TAS ではこの手続きにより参照信号と信号レベルでほぼ同一の目的区間を高速に探索することが可能である．しかし，背景音楽探索のように目的の音楽に他の音声が重畳した場合には，探索精度が低下する．そこで，次に，このような背景音楽探索に応用可能な SSC について説明する．

SSC は背景音楽同定のための手法であるが，以下のように，スペクトログラムの相関を計算しながら全探索を行うことで背景音楽探索を行うことができる．SSC においては参照信号のスペクトログラムを時間周波数領域で小領域のスペクトログラムに分割し，この小領域スペクトログラムと蓄積信号のスペクトログラム中の小領域スペクトログラムとの類似度を，スケールパラメータの最適化により最大化された相関とする．本論文ではこの最大化された相関で得られる小領域スペクトログラムの類似度（小領域類似度）を特に小領域相関類似度と呼ぶ．SSC では，参照信号のスペクトログラムを分割して得られる各小領域スペクトログラムについて，全探索法により，この小領域スペクトログラムと，蓄積信号中の各時点における小領域スペクトログラムとの小領域相関類似度を計算する．そして，これらの小領域相関類似度をスケール及び蓄積信号中の小領域スペクトログラムの時点に関する投票空間への投票により統合し，蓄積信号中の目的区間を探索する．この投票は，蓄積信号中の同じ時点で，同じスケールで類似する小領域スペクトログラムの小領域相関類似度を，そのスケールと時点における類似度として積算することで行う．この投票は，参照信号の小

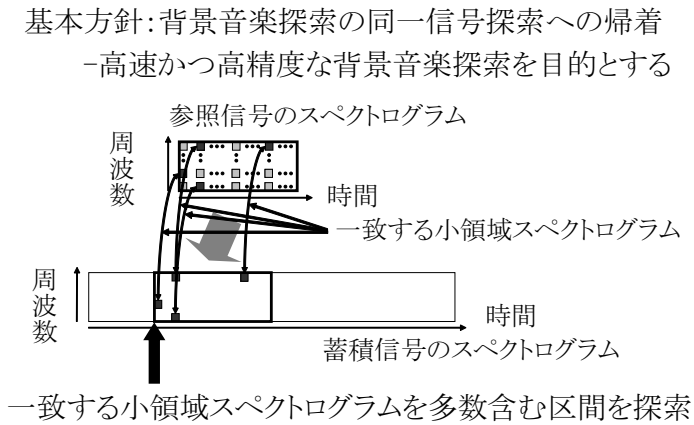


図 2.4: スペクトログラムの分割一致探索による背景音楽探索

領域スペクトログラムの参照信号内での時点を考慮したものである．例えば，参照信号の先頭から t_r の時点の小領域スペクトログラムと蓄積信号中の時点 t_s の小領域スペクトログラムとの小領域相関類似度が s で，スケールパラメータが a のとき，s は，時点–スケールの投票空間内で，時点が $t_s - t_r$，スケールが a の類似度として投票され積算される．次に，この投票の結果，あるスケールと時点において積算された類似度が大きい場合，この時点に目的区間があったとする．

以上が SSC による背景音楽探索の概要である．SSC においては小領域ごとの照合で選ばれたスケールパラメータを投票法により選び，重畳する音声の影響が少ない時間周波数領域で最適化されたスケールパラメータを用いて，参照信号と照合区間全体の類似度を測っているといえる．このように，SSC は背景音楽探索に応用可能であるが，その計算量の大きさから，長時間の蓄積信号からの探索には時間がかかるという問題点がある．

2.3 背景音楽探索法

2.3.1 スペクトログラムの分割一致探索による背景音楽探索

図 2.4 に分割一致探索法の概要を示す．本探索法では参照信号のスペクトログラムを多数の小領域スペクトログラムへ分割し，この小領域スペクトログラムと一致する小領域スペクトログラムをもつ時点を蓄積信号より探索し，そして，参照信号の小領域スペクトログラムと一致する小領域スペクトログラムを多数もつ時点を探索結果とする．すなわち，分割一致探索法では，信号の探索を，参照信号のスペクトログラムを小領域スペクトログラムに分割し，参照信号と同一ではないが，部分的には同一となる小領域スペクトログラムを多く含む

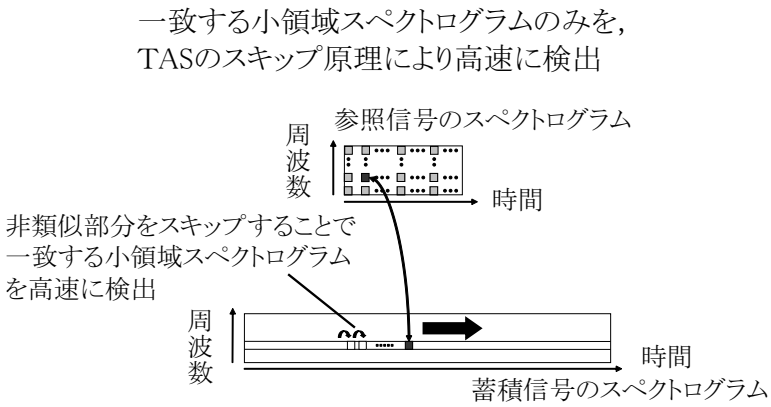


図 2.5: 時系列アクティブ探索法（TAS）を用いた分割一致探索法

区間を探索することと考える．そして，この分割一致探索法により高精度な背景音楽探索が可能になると考えられる．実際，テレビ放送などで音楽が背景音楽に用いられる場合の前景音響は多くの場合音声などであり，時間方向においても息つぎや間などにより，空白が存在する．また，音声による音響は周波数方向においても偏在し，音楽音響の周波数領域をすべて覆うことはまれなので，分割一致探索法による背景音楽探索へのアプローチは有効と考えられる．

さらに，分割一致探索法においては一致する小領域スペクトログラムの組合せだけに着目するため，一致する小領域スペクトログラムの探索に高速な同一信号探索の手法の適用が可能になる．本論文ではこの同一信号探索に，高速な手法である TAS を用いる．これにより相関法に比べ高速な一致箇所の検出が可能と考えられる（図 2.5）．

本提案手法における参照信号のスペクトログラムの小領域スペクトログラムへの分割と投票法による小領域類似度の統合の基本的な発想は SSC に共通するものがある．しかし，SSC では，参照信号の各小領域スペクトログラムについて，全探索法により，蓄積信号中の全時点での相関を計算し，その結果を統合しているのに対し，提案手法では，参照信号の各小領域スペクトログラムについて，類似度の高い小領域スペクトログラムのみを TAS により高速に検出したのち，それら検出された小領域スペクトログラムの類似度のみを統合することで探索結果を得るという違いがある．

従って，本提案手法の特長は以下の 2 点である．第一の特長は，参照信号を小領域スペクトログラムに分割し，一致する小領域スペクトログラムにのみ着目して背景音楽探索を行うことで，背景音楽探索を高精度に行うこととである．第二の特長は，この一致する小領域スペクトログラムのみに着目することにより，高速な同一信号探索の適用を可能にし，背景音楽探索全体を高速に行うことにある．

2.3.2 時系列アクティブ探索法を用いたスペクトログラム分割一致探索による背景音楽探索

2.3.1の提案内容を，処理の手順に沿ってまとめると以下のようになる．提案手法では，参照信号のスペクトログラムを多数の小領域スペクトログラムに分割し，各小領域スペクトログラムについてそれに類似する小領域スペクトログラムをもつ時点を蓄積信号からTASにより高速に探索する．そして，この類似する小領域スペクトログラムが検出された時点に基づき，その時点の小領域スペクトログラムの類似度を投票法により統合し，背景音楽探索を行う．なお，提案手法では，TASの適用に伴い，小領域スペクトログラムの類似度には小領域スペクトログラムのヒストグラム特徴の重なり率（小領域ヒストグラム類似度）を用いる．また，探索に先立ち，参照信号と蓄積信号のスペクトログラムは既に抽出してあるものとする．なお，ヒストグラム重なり率による小領域類似度を特に小領域ヒストグラム類似度と呼ぶ．

以下，提案手法の説明に先立ち，説明に用いる用語・記法について説明する．

1. スペクトログラムから切り出した一定のサイズの小領域のスペクトログラムを小領域スペクトログラムと呼ぶ．小領域スペクトログラムは周波数軸方向，時間軸方向に，各々一定の個数のスペクトログラム上の点をもとのスペクトログラムから切り出すことで得る．なお，小領域スペクトログラムの大きさは一定とする．

2. 参照信号のスペクトログラム中の，先頭の時点が t，周波数帯が w_m の小領域スペクトログラムを F_{t,w_m} と書く．

3. 蓄積信号のスペクトログラム中の，先頭の時点が t，周波数帯が w_m の，小領域スペクトログラムを G_{t,w_m} と書く．

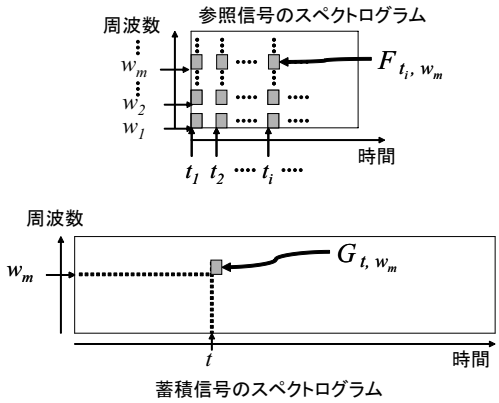
なお，提案手法では，各小領域スペクトログラムのパワー値は，各小領域スペクトログラムごとに正規化する[3]．

次に，図2.6に提案手法の概要を示し，以下に処理手順を述べる．

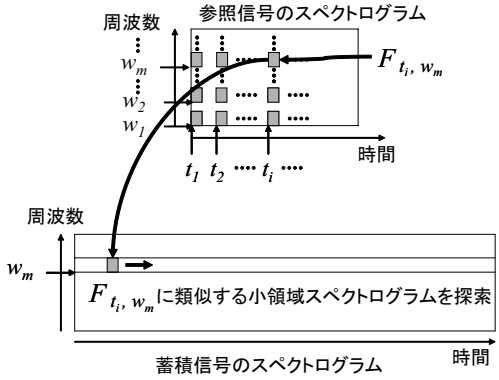
提案手法

（ステップ1） 図2.6(a)にあるように，参照信号のスペクトログラムから小領域スペクトログラムを等間隔に切り出す．ここで，切り出した参照信号の小領域スペク

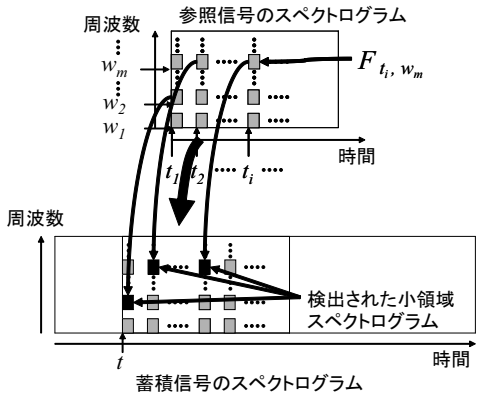
[3]小領域内の各時点においてその時点の各周波数でのパワー値を小領域内のその時点のパワー値の平均値で正規化する．



(a) 参照信号のスペクトログラムから一定のサイズの小領域スペクトログラムを等間隔に切り出す.



(b) 参照信号から切り出した各小領域スペクトログラム F_{t_i,w_m} について，蓄積信号のスペクトログラムから類似する小領域スペクトログラムをすべて，時系列アクティブ探索法（TAS）により探索する.



(c) 検出された小領域スペクトログラムの小領域類似度を統合し，各時点 t の区間類似度を計算する.

図 2.6: 提案手法の概要

トログラムを F_{t_i,w_m} とする．また，小領域スペクトログラムを切り出した時点 t_i と周波数帯 w_m の集合を，各々，$T_R(T_R = \{t_1, t_2, \ldots\})$，$W(W = \{w_1, w_2, \ldots\})$ とする．

（ステップ 2） 図 2.6(b) にあるように，参照信号から切り出した各 F_{t_i,w_m} について，蓄積信号から F_{t_i,w_m} に類似する時点を，周波数帯 w_m において探索する．この探索は TAS を用いて行われる．なお，ここで，F_{t_i,w_m} に類似する時点とは，F_{t_i,w_m} と G_{t,w_m} の小領域ヒストグラム類似度 $s^p(F_{t_i,w_m}, G_{t,w_m})$ が小領域探索しきい値 s^p_{th} を超えるような時点 t である．

（ステップ 3） 図 2.6(c) にあるように,（ステップ 2）の各小領域スペクトログラム F_{t_i,w_m} についての探索結果をもとに，蓄積信号中の各時点 t について，小領域類似度を統合し，t における参照信号との類似度（区間類似度）$S(t)$ を，

$$S(t) = \frac{1}{|T_R|} \sum_{t_i \in T_R} \left(\max_{w_m \in W} (s^p(F_{t_i,w_m}, G_{t+t_i,w_m})) \right) \tag{2.3}$$

として求める．ここで，$|T_R|$ は T_R の要素数を表す．なお，F_{t_i,w_m} についての探索の結果，蓄積信号の時点 t において，G_{t+t_i,w_m} が F_{t_i,w_m} に類似する小領域スペクトログラムとして検出されなかった場合，すなわち，

$$s^p(F_{t_i,w_m}, G_{t+t_i,w_m}) \leq s^p_{th} \tag{2.4}$$

の場合，式 (2.3) においては，

$$s^p(F_{t_i,w_m}, G_{t+t_i,w_m}) = 0 \tag{2.5}$$

とする．実際の探索においては，G_{t+t_i,w_m} が F_{t_i,w_m} に類似する小領域スペクトログラムとして検出された場合のみ，式 (2.3) において $s^p(F_{t_i,w_m}, G_{t+t_i,w_m})$ を積算する．

以上のようにして得られた区間類似度をもとに，区間類似度 $S(t)$ が探索しきい値 S_{th} よりも大きいような時点 t を先頭とする区間に参照信号が検出されたとする．

□

なお，式 (2.3) においては，

$$\max_{w \in W} (s^p(F_{t_i,w_m}, G_{t+t_i,w_m})) \tag{2.6}$$

として，$s^p(F_{t_i,w_m}, G_{t+t_i,w_m})$ について，その値が最大となるような周波数帯 w_m を全周波数帯の集合 W から選んでいる．これは，重畳する音声が最も無音に近く，目的の音楽への重

畳が少ないと考えられる周波数帯を選択するためである．そのために，参照信号中の同じ時点の複数の異なる周波数帯の小領域スペクトログラムについて，これらに類似する小領域スペクトログラムとして，蓄積信号中の同じ時点の異なる複数の周波数帯の小領域スペクトログラムが検出された場合に，最も小領域ヒストグラム類似度が高い周波数帯を選んでいる．

提案手法におけるスペクトログラムの小領域スペクトログラムへの分割については，小領域スペクトログラムの時間軸方向の長さがスペクトルの抽出間隔に比べて短すぎると，小領域スペクトログラム内のスペクトルの抽出数が少なくなり，小領域ヒストグラム類似度の信頼性が下がる．その結果，背景音楽探索における探索精度の低下を招く．しかし，小領域スペクトログラムの時間方向の長さが長すぎると，小領域スペクトログラムの照合において，前景音響の影響を受ける確率が高くなり，やはり探索精度が低下する．これらは予備実験などにより適切に設定する必要がある．2.4 節で行った実験では，スペクトルの抽出間隔を 0.5 ms とし，小領域スペクトログラムの時間方向の長さを 100 ms とした．また，周波数軸方向への分割については前景音響の影響を避けるためにはより細かく分割することが望ましいが，その場合，小領域スペクトログラムの数が増えることにより，探索時間が増加するという問題がある．周波数軸方向への分割は探索精度との兼ね合いとなるが，探索精度を保ちながら，できるだけ少ない分割数にすることが望ましい。実際には，できるだけ少ない分割数から始め，探索精度が足りない場合には，より大きな分割数にすることになる．また，参照信号からの小領域スペクトログラムの切り出しの際の時間軸方向の切り出し間隔も，短い方が探索精度が向上する．しかし，この場合も，参照信号から切り出される小領域スペクトログラムの数が増えると探索時間が増加する．周波数軸方向への分割と同様，探索速度の向上のためには，できるだけ大きな切り出し間隔から始め，探索精度が足りない場合には，より細かな切り出し間隔にすることになる．

2.4 提案手法の評価

2.4.1 探索精度の評価実験

提案手法を用いた背景音楽探索の探索精度の比較実験について述べる．提案手法の探索精度の評価のため，SSC，TAS，提案手法の 3 つの手法で背景音楽探索実験を行い，各々の探索精度を比較評価した．SSC においては参照信号を小領域スペクトログラムに分割し，小領域スペクトログラムを蓄積信号と全探索により照合し，蓄積信号のすべての時点での小領域相関類似度を時点–スケールの投票空間に投票した．そして，投票空間において積算された類似度（SSC における区間類似度とする）が探索しきい値より大きい時点を先頭とする

区間を探索結果とすることとした．TASにおいては，まず，スペクトログラムを分割しない従来のTASによって，参照信号に類似する区間（参照信号とのヒストグラム重なり率による類似度が0.6より大きい区間）を蓄積信号から探索する．そして，さらに，このTASによる検出結果に対してしきい値処理を行い，ヒストグラム重なり率による類似度（TASにおける区間類似度とする）が探索しきい値よりも大きい区間を探索結果とすることとした．また，(c)における小領域探索しきい値は0.6とした．

実験では，約30分の音楽の音響信号に音声の音響信号を計算機上で混合したものを蓄積信号とした．各信号はサンプリング周波数44.1 kHz，16 bit量子化のものを用いた．ここで，音楽音響信号にはポップスとロックを用いた．また，音声音響信号には，DJ（ディスクジョッキー）のトーク及びアーティスト同士の対談を録音したものを用いた．ただし，この音声音響信号については，1秒以上の無音区間がないように編集した．そして，音声音響信号の音楽音響信号への混合においては，テレビ放送などで音楽がBGMとして使われた場合を想定し，両信号の電力比（混合比と呼ぶ）を5〜−20 dBの間で5dBごととした．なお，ここで，混合比は

$$10\log_{10}\left(\frac{\text{音楽音響信号の平均電力}}{\text{音声音響信号の平均電力}}\right)\quad(\mathrm{dB}) \tag{2.7}$$

である．混合比が負数ということは音楽音響信号の方が音量が小さいことを意味する．参照信号はもとの音楽音響信号からランダムに切り出した長さ15秒の信号100個とした．

そして，各参照信号について，もとの音楽音響信号から参照信号を切り出した時点と同じ蓄積信号中の時点をこの参照信号に対する目的区間とし，蓄積信号から探索した．探索により検出された区間を検出区間と呼ぶ．なお，本実験においては，ある区間tが検出され，tとtからのずれが前後15秒未満の範囲内の検出区間のうち，tの区間類似度が最大であった場合，この範囲においてはtのみが検出区間として検出されたとした．すなわち，区間類似度が前後15秒未満の範囲で極大かつ探索しきい値より大きい区間のみを検出区間とした．そして，検出区間のうち，目的区間からのずれが前後15秒未満の検出区間があればそのうちの一つを正しい検出区間とし，他のすべての検出区間は正しくない検出区間とした．また，目的区間からのずれが前後15秒未満の位置に検出区間が検出された場合，この目的区間は検出されたとし，探索結果に含まれたとした．

探索精度については，検出された検出区間のうちの正しい検出区間の割合（正答率）と，目的区間のうち探索結果に含まれた割合（検出率）の100回の試行における平均値が等しくなるように探索しきい値を各混合比の場合で調整し，等しくなったときのこれらの割合の値を探索精度とした．なお，各混合比においては，100回の試行を通して探索しきい値は一定とした．本章の探索においては，探索しきい値S_{th}を大きくすれば正答率は上がり，検出率

は下がる．そのため，探索精度は，探索しきい値を調整し，両割合の平均値が同じになったときの両割合の値で測ることとした．また，このとき，正しい検出区間の数を高々一つとすることで，(検出率 − 正答率) が探索しきい値に関する広義の単調減少関数となる．その結果，(検出率の平均値 − 正答率の平均値) も探索しきい値に関する広義の単調減少関数となり，両割合の平均値が等しくなるときの探索しきい値を容易に求めることができる．

次に，SSC，TAS，提案手法の 3 つの手法について，実験に用いた他のパラメータを述べる．すべての手法で，探索に用いる周波数帯域は 500 Hz から 2 kHz とし，スペクトルの周波数軸方向の抽出点数は 28 とした．この帯域は，携帯電話などの音声通信を目的とした帯域の狭い音響信号での探索を想定したものである．また，SSC，提案手法とも，参照信号から小領域スペクトログラムを切り出す数を増やすと探索精度が上がることが期待できる．しかし，両手法とも，参照信号から小領域スペクトログラムを切り出す数に比例した探索時間がかかる．そのため，SSC と提案手法の参照信号の小領域スペクトログラムへの分割については，両手法で同じ 個数の小領域スペクトログラムを切り出すこととし，予備実験による探索精度と探索速度の予備評価から，周波数軸方向は四つに分割し，小領域スペクトログラムの参照信号からの切り出し間隔は 600 ms とした．

なお，SSC と提案手法の比較においては音響特徴の抽出法，抽出間隔，分析窓長，並びに，小領域スペクトログラムの時間方向の長さ，小領域スペクトログラムへの周波数軸方向への分割の仕方についても，同じ条件で行うことを検討すべきである．そこで，SSC では文献 [41] に従い，FFT で音響特徴を抽出し，小領域スペクトログラムへの周波数軸方向の分割は等間隔に行うこととし，提案手法でも同様とすることを検討した．しかし，提案手法の場合の探索精度を評価した予備実験において，提案手法で，SSC と同様に，音響特徴の抽出に FFT を用い，小領域スペクトログラムへの分割は周波数軸上で等間隔に分割する場合と，音響特徴の抽出には中心周波数を対数軸上で等間隔に配置した帯域通過フィルタバンクを用い [1]，小領域スペクトログラムへの分割は対数軸上で等間隔に分割するように工夫した場合の探索精度を比較したところ，後者の方がより良い探索精度を得られた．そのため，提案手法では後者の方法をとった．SSC では文献 [41] に従い，FFT により得られる振幅を用い最適化した小領域相関類似度を計算しているため，FFT で音響特徴抽出を行い，小領域スペクトログラムへの分割も線形軸上で等間隔に分割した．そしてこれらの条件の下，音響特徴の抽出間隔及び分析窓長，小領域スペクトログラムの長さは，SSC と提案手法の場合で，各々，探索精度が最大になるよう実験的に定めた．これは，SSC と提案手法で，各々，探索精度が最大となるパラメータが異なるためである．そして，TAS においては提案手法と同じ 音響特徴を用いることとした．

以下，実験に用いたパラメータを詳細に述べる．SSC では文献 [41] に従い，FFT により

表 2.1: 実験パラメータ

音響特徴抽出	SSC（従来法）	TAS（従来法）	提案手法
抽出法	FFT	帯域通過フィルタバンク	
分析窓長	50 ms	2 ms	
抽出間隔	5 ms	0.5 ms	
特徴抽出点数（周波数軸上）	28		
小領域への分割	**SSC（従来法）**	**TAS（従来法）**	**提案手法**
分割の有無	あり	なし	あり
小領域の長さ	50 ms	–	100 ms
周波数軸方向	4 等分（線形軸上）		4 等分（対数軸上）
切り出し間隔	600 ms		600 ms

音響波形信号からスペクトルを抽出した．なお，FFT の分析窓長は 50 ms とし，5 ms ごとにスペクトルを抽出した．そして，周波数軸上で 500 Hz から 2 kHz の間に等間隔にある 28 個の周波数において得られた FFT の振幅を実験に用いるスペクトルとした．そして，参照信号の小領域スペクトログラムへの分割については，周波数軸方向には，上記の 28 個の出力を 7 点ずつ，四つに分割し，時間軸方向には，長さ 50 ms とすることとした．一つの小領域スペクトログラムには周波数軸方向で 7 点のスペクトルが時間軸方向に 10 組あることになる．そして，このような小領域スペクトログラムを参照信号から 600 ms おきに切り出した．TAS の場合は，音響特徴の抽出には上述のように帯域通過フィルタバンクを用いた．帯域通過フィルタは計 28 個のフィルタを 500 Hz から 2 kHz の帯域に対数軸上で等間隔に配置して用いた．そして，各フィルタで 2 ms の時間区間の分析を 0.5 ms ごとに行い，各時間，各周波数におけるパワーを抽出し，各時間の 28 個のフィルタの出力をその時点のスペクトルとした．なお，TAS では参照信号の小領域スペクトログラムへの分割などは行わず，蓄積信号中から参照信号に類似する区間を単に従来の TAS を用いて探索した．また，提案手法では TAS と同じ音響特徴を用いた．そして，参照信号の小領域スペクトログラムへの分割については，周波数軸方向には，上記の 28 の帯域通過フィルタの出力を周波数の順に七つずつに分けた 4 区間への分割とし，小領域スペクトログラムの時間軸方向の長さは 100 ms とした．すなわち，各小領域スペクトログラムは七つの帯域通過フィルタから得られる 100 ms 分，200 個所の分析によるスペクトログラムとした．そして，このような小領域スペクトログラムを参照信号から 600 ms おきに切り出した．表 2.1 にこれらのパラメータをまとめる．

表 2.2: 探索精度（%）

混合比 (dB)	SSC（従来法）	TAS（従来法）	提案手法
5	99.8	93.0	100.0
0	98.0	88.1	99.0
−5	92.9	81.0	96.8
−10	72.0	72.9	94.8
−15	44.9	64.0	90.3
−20	24.8	–	73.2

各手法の各混合比における探索精度を表 2.2 に示す．混合比が 5 dB の場合，提案手法の探索精度は 100%であり，検出もれも過剰検出もないことが分かる．また，各混合比において提案手法の探索精度が最も良いことがわかる．しかし，SSC，提案手法のいずれの場合においても混合比が小さくなるにつれて，探索精度が低下している．このような背景音楽が非常に小さい場合に探索精度を上げるためには，周波数軸方向及び時間軸方向において，より細かい小領域スペクトログラムへの分割が必要になると思われる．なお，TAS の場合は各混合比で探索精度が低く，背景音楽探索への適用は難しいと考えられる．また，TAS で混合比が −20 dB の場合では，しきい値を 0.6 とした TAS の段階での検出もれが多く，この TAS の検出結果に対するしきい値処理において探索しきい値を調整しても，検出率と正答率が等しくなることはなかった．また，SSC においては，音響特徴の抽出間隔を短くすることで精度の向上を図ることも考えられるが，予備的に行った実験では探索精度の向上は見られず，探索時間は増加した．

2.4.2 探索速度の評価実験

次に提案手法における探索速度の評価実験について述べる．本実験では，提案手法の探索速度を，背景音楽探索が可能な SSC，及び，提案手法で小領域スペクトログラムの探索に全探索法を用いた場合と比較して評価した．SSC と提案手法について，探索時の実験条件は 2.4.1 と同じである．小領域スペクトログラムの探索を単純に全探索を用いて行う場合は，各小領域スペクトログラムを蓄積信号上のすべての時点において照合し，小領域ヒストグラム類似度を求めることになる．以下，提案手法で小領域スペクトログラムの探索に全探索法を用いた場合をヒストグラム全探索法と呼ぶ．ヒストグラム全探索法のその他の実験条件は提案手法と同じである．

表 2.3 に，2.4.1 の背景音楽探索において，SSC，ヒストグラム全探索法，提案手法の 1 回

表 2.3: 30 分の蓄積信号の探索における平均照合回数

混合比 (dB)	SSC（従来法）	ヒストグラム全探索法（従来法）	提案手法
5	3.3×10^7	3.3×10^8	3.4×10^6
0	同上	同上	3.4×10^6
−5	同上	同上	3.3×10^6
−10	同上	同上	3.3×10^6
−15	同上	同上	3.3×10^6
−20	同上	同上	3.3×10^6

の背景音楽探索で行われた小領域スペクトログラムの照合回数の平均値を示す．2.4.1 で述べたように，いずれの混合比においても，提案手法の小領域探索しきい値は 0.6 である．なお，SSC と，ヒストグラム全探索法については，小領域スペクトログラムを蓄積信号のスペクトログラム上のすべての時点で照合するため，混合比による照合回数の変動はない．SSC とヒストグラム全探索法の照合回数をみると，ヒストグラム全探索法の場合の音響特徴の抽出間隔が SSC の場合の 1/10 であるため，ヒストグラム全探索法の場合の照合回数が SSC の場合の照合回数の 10 倍になっていることが分かる．しかし，ヒストグラム全探索法と提案手法では，音響特徴の抽出間隔は同じであるが，提案手法では TAS により，不要な時点での照合を省略しながら小領域スペクトログラムに類似する時点を探索するため，提案手法では照合回数を削減することができる．表 2.3 からは，いずれの混合比においても，提案手法の照合回数はヒストグラム全探索法の照合回数の約 1%にまで大幅に削減されていることが分かる．そして，その結果，提案手法の照合回数は SSC の照合回数の約 10%となっている．SSC における小領域スペクトログラムの一度の照合にかかる計算時間と，ヒストグラム全探索法，提案手法における小領域スペクトログラムの一度の照合にかかる計算時間は異なるが，照合回数の観点からは，提案手法により，より高速な背景音楽探索が可能になることが考えられる．なお，提案手法においては混合比が小さくなるにつれて，照合回数が減少する．これは，前景の音声信号が大きくなるにつれ，音楽音響信号である参照信号の小領域スペクトログラムを蓄積信号と照合したときの小領域類似度が小さくなり，TAS におけるスキップ幅が大きくなるためと考えられる．

次に，1 回の背景音楽探索にかかった CPU 時間の平均値を表 2.4 に示す．なお，表 2.4 の探索時間には，前処理（音響波形信号からのスペクトルの抽出など）にかかる時間は含まれていない．また，CPU 時間の計測には PC（CPU:Pentium 4 2 GHz， L2 キャッシュ 256 kByte，OS:Red Hat Linux 7.2）を用いた．表 2.3 にある照合回数の削減にともない，表 2.4 においても提案手法の探索時間がヒストグラム全探索法の場合の約 2%まで大幅に削減され

表 2.4: 30 分の蓄積信号の探索における平均探索時間 (CPU time)

混合比 (dB)	SSC（従来法）	ヒストグラム全探索法（従来法）	提案手法
5	30.3 s	48.5 s	0.86 s
0	30.6 s	48.5 s	0.86 s
−5	30.7 s	48.3 s	0.86 s
−10	30.7 s	48.4 s	0.85 s
−15	30.7 s	48.3 s	0.85 s
−20	30.7 s	48.2 s	0.85 s

ていることが分かる．そして，提案手法の探索時間は SSC の約 3%であることが分かる．以上の実験から，提案手法が背景音楽探索を高速に行えることが分かる．

なお，表 2.3 において，提案手法の照合回数はヒストグラム全探索法の照合回数の約 1%であるが，表 2.4 において，提案手法の探索時間はヒストグラム全探索法の探索時間の約 2%となっている．これは，提案手法の方が，1 回の照合において，蓄積信号上の小領域スペクトログラムのヒストグラムの作成にかかる計算時間が長いためである．ヒストグラム全探索法の場合は，蓄積信号上で小領域スペクトログラムの照合位置を一つずらしたとき，蓄積信号上の小領域スペクトログラムのヒストグラムは，一つ前の時点のヒストグラムから 1 要素を取り除き，そして，新たに 1 要素を加えることで作成できる．提案手法では 1 回の小領域スペクトログラムの照合を行う際に，前に照合を行った時点のヒストグラムから，TAS によるスキップ幅だけ要素を入れ換えたヒストグラムを作成する必要がある．

また，表 2.3 において，ヒストグラム全探索法の照合回数は SSC の照合回数の 10 倍であるが，表 2.4 において，ヒストグラム全探索法の探索時間は SSC の探索時間の約 1.6 倍である．これは，ヒストグラム全探索法の各照合における小領域ヒストグラム類似度の計算時間が SSC の各照合における小領域相関類似度の計算時間よりも小さいためと考えられる．また提案手法の照合回数は SSC の照合回数の約 10%であるが，同じ理由により，提案手法の探索時間は SSC の探索時間の約 3%にまで削減されている．

2.5 むすび

本章では，音響信号の部分一致に着目した，高速な背景音楽探索の手法を提案した．本手法は参照信号のスペクトログラムを小領域スペクトログラムに分割し，各小領域スペクトログラムに類似する小領域スペクトログラムをもつ時点を蓄積信号のスペクトログラムから

TAS により高速に探索し，検出された時点の小領域類似度を統合し，その結果から参照信号を含む区間を検出する．この類似する小領域スペクトログラムの探索に TAS を用い，小領域スペクトログラム同士の照合回数を削減することで，高速な背景音楽探索が可能になる．実験では小領域スペクトログラムの探索に全探索を用いた場合に比べ，探索時間を約 2%にまで短縮することができた．そして，その結果，15 秒の参照信号を用いた約 30 分の蓄積信号からの探索を 1 秒未満で行うことができた．

なお，本章では，参照信号を他の音声などが重畳していない音楽の音響信号とし，蓄積信号を他の音声などが重畳した音響信号としたが，提案手法は参照信号を音楽に他の音声などが重畳した音響信号とし，蓄積信号を他の音声などが重畳していない音楽の音響信号とした場合にも適用可能である．この場合，例えば，放送の音響信号から，随時，指定された長さの音響信号を参照信号として切り出し，音楽音響信号からなる DB から参照信号の背景に流れてる音楽を探索することで，DB に登録されている音楽が放送中に流れた場合に自動的に検出するような応用が考えられる．

第3章　2値多重音響特徴ベクトルを用いた類似音楽探索法

3.1　まえがき

本章では音楽演奏の音響信号の探索において，多重奏音響信号を探索キー（参照信号）とし，同じく多重奏音響信号（蓄積信号）の中から，参照信号に類似する音楽を探索する類似音楽探索法とその高速化手法を提案する．ここで類似する音楽とは，参照信号と同一の音響信号だけではなく，例えば，再演奏されたもの，他の楽器を用いて演奏されたもの，テンポの異なる演奏，もしくは移調されたものなど，同一の音響信号ではないが同じ曲も含むこととする．また，音楽は電子楽器などだけではなく，自然楽器などでも演奏される．このような多重奏音響信号による類似音楽探索は，例えば，通常多重奏音響信号である音楽CDの断片などを参照信号とし，それに類似する音楽を音楽音響信号のDBから探索するために必要な技術である．

第1章で述べたように，演奏者が異なる同一の曲，主旋律が同じ曲，アレンジの異なる曲などのように，何らかの意味で類似している音楽の探索をねらう類似音楽探索についても，これまで様々な研究が行われてきた [6, 7, 8, 9, 10, 11]．このような類似音楽探索においては，参照信号と蓄積信号で音響特徴が大きく異なるためや，探索キーや蓄積データが音響信号以外に楽譜などで表現されるため，音響特徴以外の特徴を用いた探索が必要となる．特にGhiasらは単旋律の音高系列を “S（同じ）”，“U（上がる）”，“D（下がる）” の相対音高の文字列で表す旋律輪郭（melodic contour）を用いた単旋律の照合手法を提案し，ハミングを探索キーとし，類似する単旋律を探索する類似音楽探索においてその有効性を示した [7]．旋律輪郭は旋律の表現としては楽譜などに比べ単純であるが，単旋律の識別，照合には有効であり，その他の研究においても単旋律の探索における特徴として用いられている．そして，これまでの類似音楽探索の多くは，ハミングなどで入力された単旋律の音高系列を探索キーとし，同じく単旋律の音高系列からなるDBから探索キーに類似するものを，単旋律同士の照合により探索するものであった．また，そのほとんどの手法におけるDBの蓄積データは，MIDIや楽譜情報，もしくは探索キーと同じくハミングによる主旋律の入力から得ら

れる，既に主旋律などが抽出された単旋律の音高系列であった．

音響信号からなるDBを対象とした類似音楽探索については，橋口らが，ハミングによる単旋律音高系列を探索キーとし，単旋律対多重奏の照合を前提とした手法を提案している [11]．しかし，本研究で対象とする多重奏音響信号を探索キーとした多重奏音響信号からの類似音楽探索の場合，単旋律の探索キーやDBを前提とした照合法を用いるには，多重奏からの主旋律抽出が必要となることや，もしくは多重奏と多重奏の間での主旋律の照合など困難な問題が多い [12, 13]．

以上の多重奏音響同士の比較に基づく類似音楽探索の課題をまとめると以下の二つになる．第一の課題は，音楽音響信号が変動するために一致音楽探索のような音響特徴が使えないことである．第二の課題は，変動のある音楽音響信号を対象とし，かつ，多重奏音響信号の照合を行わなくてはならないことである．

本論文ではこのような多重奏音楽の音響信号の照合のための2値多重音響特徴ベクトルを提案し，この特徴を用いた類似音楽探索法を提案する．これまで，このような多重奏音響信号同士の比較に基づく類似音楽探索の手法は提案されていなかった．本提案は，Ghiasらの旋律輪郭と同様に，音楽演奏の時系列を単純な符号の一次元系列で表現し，この符号系列に対し，系列照合を行うことでロバストかつ高速な類似音楽の探索をねらうものである．2値多重音響特徴ベクトルは，各成分を2値とする特徴ベクトルであり，この各成分は各々対応する音高の音の有無に大まかに対応する．この特徴はGhiasらの旋律輪郭が旋律の正確な記述を目的としたものではないのと同様に，同時に発音している音の正確な記述を目的とするものではないことに注意したい．3.3節で述べるように，多重奏音楽の音響信号から，同時に発生する複数の音に基づき順次2値多重音響特徴ベクトルを抽出し，特徴ベクトルの時系列を得る．そして，蓄積信号と参照信号から得られた2値多重音響特徴ベクトルの系列について，2値多重音響特徴ベクトルの類似度に基づいた曖昧系列列照合を行うことにより，類似音楽探索を行う．この2値多重音響特徴ベクトルにより，多重奏音響信号同士の照合が可能になり，本章で目的とする多重奏音響信号間の類似音楽探索が可能になる．更に，本章ではこの2値多重音響特徴ベクトルを用いた曖昧系列照合による類似音楽探索の高速化手法を提案する．

以下，3.2節で本章で対象とする類似音楽探索について述べる．次に3.3節では2値多重音響特徴ベクトルを用いた類似音楽探索法とその高速化手法について述べ，3.4節で提案手法の探索精度及び探索速度に関する評価実験について述べる．そして，3.5節を本章のまとめとする．

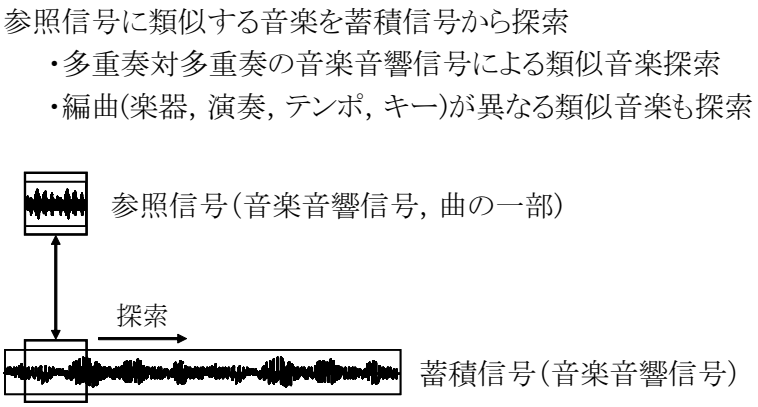


図 3.1: 対象とする類似音楽探索

3.2 対象とする類似音楽探索と従来の探索法

図 3.1 に本章で対象とする類似音楽探索の概要を示す．ここで参照信号は探索キーとなる多重奏音楽音響信号であり，音楽音響信号のある区間を切り出したものとする．また，蓄積信号も多重奏の音楽音響信号とする．なお，音楽音響信号は電子楽器に限らず自然楽器などでも演奏されるものとする．そして，対象とする類似音楽探索は，蓄積信号から参照信号と同じ楽譜に基づいて演奏されている区間を探索することとする．本類似音楽探索においては，参照信号と同一の音楽演奏だけではなく，再演奏されたもの，他の楽器を用いて演奏されたもの，テンポの異なる演奏，もしくは移調されたものなど，同一の音響信号ではないが同じ楽譜に基づく演奏も探索することを目的とする．以降，本章で類似する音楽とは同じ楽譜に基づいて演奏された音楽とする．そして，これらの演奏の違いを編曲変動と呼ぶ．

本章で目的とするような，多重奏の音楽音響信号同士を比較照合し，類似する音楽を探索するための方法としては，まず，音楽の楽譜情報を音楽音響信号から抽出し，それを比較することで同じ楽譜に基づく音楽音響信号を探索することが考えられる．音楽音響信号からの楽譜情報の計算機による抽出は自動採譜と呼ばれ，これまでにも多くの研究が行われている [42, 43]．しかし，電子楽器音を対象とした場合に比べ，自然楽器音の自動採譜は難しく，単旋律の音響信号を対象としても簡単ではない．また，複数の楽器がある多重奏音楽音響信号を対象とした場合，問題はより難しくなる．更に，複数の音の情報をどのように分離するかという音源分離の問題 [44] や，得られた複数の音からどのようにして旋律を抽出，分離するかという問題もある．

また，楽譜情報が得られたとしても，どのようにして楽曲の類似性を測り，同じ音楽を探索するかという問題がある．採譜が誤り無く正確にできれば，楽譜情報自体を照合すればよいが，誤りなどがある場合，それを考慮した照合を行う必要がある．また，楽譜情報の類

似性判定に関する研究も行われているが [13, 45, 46]，単旋律の照合を基本とする方法では，先に述べた旋律の分離や主旋律抽出が正確に行われている必要がある [12].

このように，多重奏の音響信号同士を照合し，類似音楽探索を行う際に，自動採譜した情報に基づき探索を行うには，困難な問題が多い．そして，これまで，多重奏の音響信号同士を照合し，類似音楽探索を行う方法は提案されていなかった.

一方で，Ghias らの旋律輪郭は，音楽の詳細な記述を目的とした楽譜に比べると，旋律の表現は非常に単純である．しかし，目的を音楽の正確な記述でなく類似する単旋律の探索とした場合，旋律輪郭は，単旋律の識別，照合のために非常に有効であるといえる．そして，旋律輪郭はその他の研究においても単旋律の探索における特徴として用いられている．そこで，本論文では，対象とする類似音楽探索において，旋律輪郭と同様に，探索を目的とした符号系列を抽出し，これを照合することで目的の類似音楽探索を行うアプローチをとる．この提案は，Ghias らの旋律輪郭と同様に，音楽演奏の時系列を単純な符号の一次元系列で表現し，この符号系列に対し，系列照合を行うことでロバストかつ高速な類似音楽の探索をねらうものである．そして，この符号化のための特徴として，2 値多重音響特徴ベクトルを提案する．2 値多重音響特徴ベクトルは，各成分を 2 値とする特徴ベクトルであり，この各成分は各々対応する音高の音の有無に大まかに対応する．この特徴は Ghias らの旋律輪郭が旋律の正確な記述を目的としたものではないのと同様に，同時に発音している音の正確な記述を目的とするものではないことに注意したい．次節でこれらの手法について述べる.

3.3　類似音楽探索法

3.3.1　曖昧系列照合による類似音楽探索

図 3.2 に曖昧系列照合による探索処理の概要を示す．ここで，蓄積信号と参照信号は多重奏音楽の音響信号であり，探索においては，蓄積信号中の参照信号に類似するすべての区間を検出することを目的とする．まず，探索の準備として，各音響信号について周波数分析を行い特徴ベクトルを時系列順に抽出する．そして，抽出された特徴ベクトルを順次符号化し符号系列を得る．ここで蓄積信号と参照信号から得られる符号系列を，各々，蓄積符号系列，参照符号系列と呼ぶ．そして，探索においては，蓄積符号系列中の参照符号系列に類似する長さ w のすべての部分系列を探索し，検出する．ここで w は探索窓の幅として与えられるものとする．そして，参照符号系列と長さ w の部分系列の類似度は符号間の類似度に基づいた DP 照合により求める．これは，演奏速度の違いなどによる時間軸方向の信号伸縮に対応するためである．なお，この DP 照合により得られる類似度を系列間類似度と呼び，

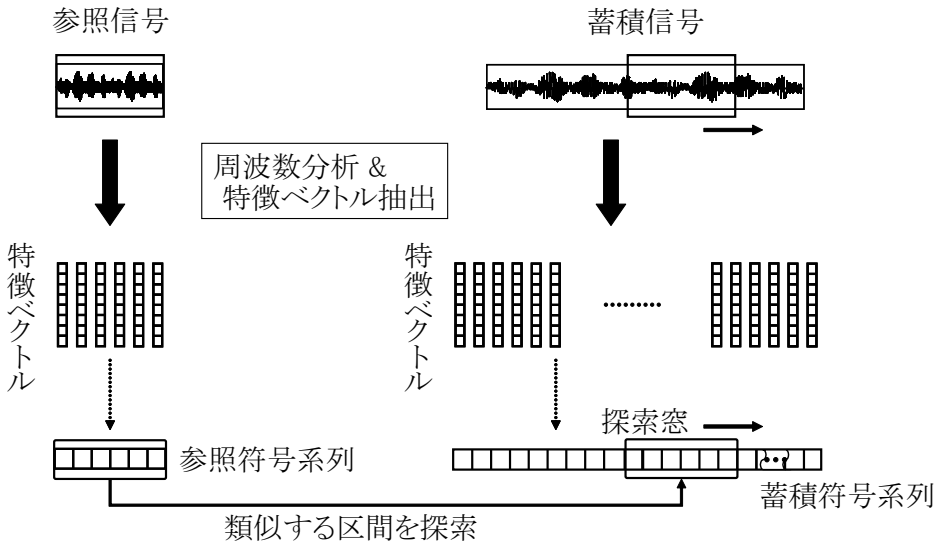


図 3.2: 曖昧系列照合による探索処理の概要

照合を行う符号系列の長さ，すなわち参照符号系列の長さと w により正規化されているものとする．以上，本探索で行う処理についてその概要を述べたが，本探索は，形式的には，参照符号系列との系列間類似度が探索しきい値 s_t より大きい蓄積符号系列中の長さ w の部分系列をすべて検出することとする．

3.3.2 2 値多重音響特徴ベクトルを用いた曖昧系列照合による類似音楽探索とその高速化

(1) 2 値多重音響特徴ベクトルの抽出

曖昧系列照合による類似音楽探索においては，用いる特徴ベクトルは多重奏対多重奏の音響信号の照合という課題と，編曲変動による音響信号の変動を吸収するという課題に対処しなければならない．そこで，本論文ではこれらの課題に対処するため 2 値多重音響特徴ベクトルを提案する．2 値多重音響特徴ベクトルは各成分が 0 または 1 の 2 値の特徴ベクトルであり，各成分（各ビット）は下位から順に A，A#，B，. . . ，G#の各音に対応しており，その値は対応する音高の有 (1) と無 (0) を表す．2 値多重音響特徴ベクトルは同時に鳴っている音高を表す特徴ベクトルである．そして，このような特徴ベクトルを一定の組合せの音高が鳴っている区間単位で抽出する．また，2 値多重音響特徴ベクトル間の類似度は共通の音高に基づいた類似度とする．提案手法では，この 2 値多重音響特徴ベクトルとその間の類似度に基づき曖昧系列照合を行うことで対象とする類似音楽探索を行う．図 3.3 に対象とす

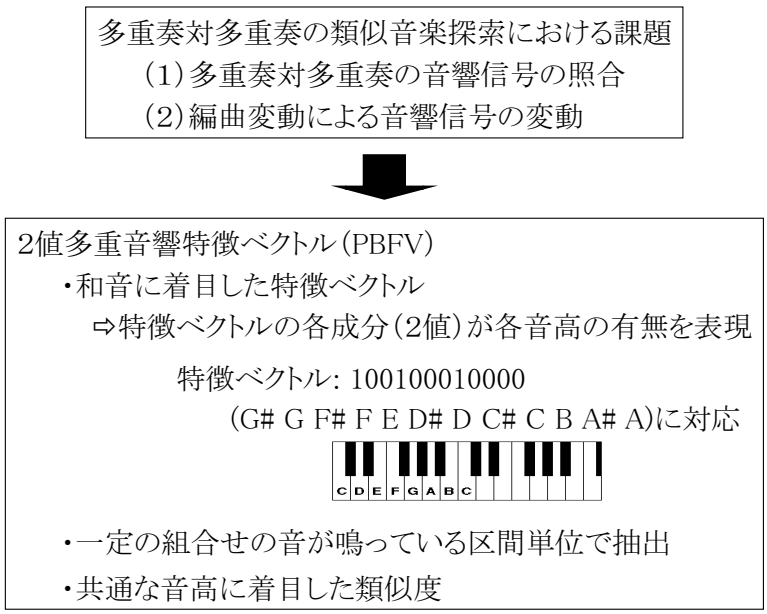


図 3.3: 類似音楽探索における課題と提案する特徴ベクトル（2値多重音響特徴ベクトル）

る類似音楽探索における課題とそれに対する2値多重音響特徴ベクトルの特長をまとめる．

次に，2値多重音響特徴ベクトルの抽出過程について述べるが，そのためにSNAP（Simultaneous-Note-set Alteration Point；同時単音集合変化時刻）の概念を提案し導入する．

多重奏音楽は，複数の単音の組合せから構成されているが，どれか一つの単音が立ち上がる時刻，あるいは同時に複数の単音が立ち上がる場合にはその時刻をSNAPと呼ぶ．すなわち，SNAPは多重奏音楽において，単音の立上りにより，同時に鳴っている単音の組合せの変化が起きた時刻を示す．ただし，ある単音の組合せの演奏のあとに続けて同じ単音の組合せが再度演奏された場合，あとの音の立上りにより二つの単音の組が分離している場合は，単音の組合せが変化せずとも，後者の演奏の立上りの時刻をSNAPとみなす．例えば，ピアノで{C，E，G}の和音を弾き，そのあと続けて再度同じ和音を弾いた場合には後者の和音の立上りはSNAPとみなす．そして，隣り合うSNAPとSNAPの間の一定の組合せの音が鳴っている区間をsnapshotと呼ぶ．そして，このsnapshotごとに一つの2値多重音響特徴ベクトルを抽出し，一つの2値多重音響特徴ベクトルを一つの符号に符号化する．このsnapshot単位の符号化には，分析窓に重なりをもたせながら抽出した多数の短時間スペクトル特徴の系列を順に一つひとつ符号化する場合に比べ，一般に，符号系列長を短くできるというメリットがある．また，snapshot単位の符号化では，演奏速度の違いなどによる音の長さの伸縮を吸収することも期待できる．

SNAPに似た概念としては拍があり，拍位置の抽出手法もいくつか提案されている[47, 48]．このような拍位置を抽出し，拍単位で符号化することも考えられる．しかし，今回の音楽

の演奏の分割の目的は，音楽的な拍位置を厳密に抽出することではなく，探索のために音楽を符号系列として表現する際に 1 符号が対応する区間を得ることである．ここで，各符号は複数の音の有無を大まかに表現するものである．そのため，音の組合せの変化に着目した snapshot 単位で符号化することとした．

以下に，SNAP の概念を用いた 2 値多重音響特徴ベクトルの抽出過程について，順に述べる．

1. **音響特徴抽出**

   まず，帯域通過フィルタを用い音響波形信号から各周波数におけるパワースペクトルを得る．帯域通過フィルタは 75Hz から 9600Hz まで 1 オクターブごとに 48 ずつ 7 オクターブ分，計 336 個のフィルタを周波数の対数軸上で等間隔に配置して用いた．そして，各フィルタで，44ms の時間区間の分析を，11ms ごとに行い，各時間，各周波数におけるパワーを抽出した．

2. **SNAP 抽出**

   上記で得た時間–周波数–パワー空間のスペクトログラムを用い，SNAP を抽出する．本論文では，文献 [48] の拍位置抽出の考え方に基づき，周波数成分の立上りとなるスペクトログラム上の点が集中する時刻を SNAP として求めることとした．また，複数の単音が同時に発生する場合にも，厳密には発生時刻にばらつきがある．そこで，このようなばらつきは基本拍間隔 [48] に基づき吸収し，複数の単音が同時刻に発生したとする．そして，この時刻を一つの SNAP とすることとした．ただし，本論文では，基本拍間隔も周波数成分の立上りとなる点に着目して計算する．以下，本論文の SNAP 抽出手続きにおいて用いる用語を述べた後，その抽出手続きについて述べる．

   本論文で周波数成分とは，スペクトログラムを時間–周波数–パワー空間における曲面としてみなしたとき，その曲面上のパワーが一定のしきい値を超える周波数軸方向のピークについて，ピーク近傍における曲面の尾根方向の接線ベクトルを接続したものと考える．そして，周波数成分の立上りとなっている点としては以下の抽出開始端点と端点候補点を抽出し用いることとした．抽出開始端点は，周波数成分上の点で，スペクトログラム上で周波数軸方向の極大値であり，その周波数成分上でパワーに関する抽出しきい値を時間軸方向に見て初めて上回る点である．端点候補点は周波数成分の尾根上の点で，尾根上でくぼみとなった後，尾根を登り始める点である．以降，抽出開始端点と端点候補点をまとめて SNAP 形成候補点と呼ぶ．

次に，本論文で用いたSNAPの抽出手続きについて述べる．

SNAPの抽出手続き

(i) 最大パワーがしきい値以上で，時間方向の長さもしきい値以上の各周波数成分からSNAP形成候補点を抽出する．

(ii) 得られた各SNAP形成候補点iについて，iの時刻がSNAPになる確率（SNAP確率）P_iを次式のようにして計算する[48]．

$$P_i = 1 - \prod_{j \in \Gamma_i} \left(1 - p_j \exp\left(-\frac{(t_i - t_j)^2}{\Delta T_i^2}\right)\right) \tag{3.1}$$

ここで，p_iはSNAP形成候補点iの端点確率であり，iが周波数成分の立ち上がりであることの尤度を表す．端点確率は，各SNAP形成候補点について，その点周辺のスペクトログラムの形状などから求めることも考えられるが，本論文ではすべてのSNAP形成候補点で同じ定数を与えた．また，t_iはiの時刻であり，Γ_iはiからの時刻の差がしきい値（本論文では5秒とした）以下のSNAP形成候補点の集合である．また，ΔT_iはiにおける基本拍間隔の$\frac{1}{8}$の値である．そして，iからの時刻の差がしきい値（上記しきい値と同様に5秒とした）以下のSNAP形成候補点の集合をΨ_iとし，Ψ_iにおける二つのSNAP形成候補点の時刻の差のうち最頻のものを基本拍間隔とする．本論文では，Ψ_iにおけるSNAP形成候補点の時刻の差を0.2秒ごとに分類し，最頻となる時刻の差を基本拍間隔とした．

(iii) 各SNAP形成候補点について，そのSNAP形成候補点及びそのSNAP形成候補点とほぼ同時刻（本論文では時刻の差が0.5秒以内とした）のSNAP形成候補点のSNAP確率の総和（SNAP近傍総確率）を計算する．

(iv) SNAP近傍総確率がその近傍（時刻が近いSNAP形成候補点，本論文では時刻の差が0.5秒以内とした）で最大の点で，その近傍内のSNAP形成候補点のSNAP確率の最大値がしきい値以上の点の時刻をSNAP候補時刻とし，その集合をTとする．

(v) Tにおいて，Tの各SNAP候補時刻での基本拍間隔を求める．本論文では，TのSNAP候補時刻t_iの基本拍間隔は，TのSNAP候補時刻のうちt_iとの時刻の差がしきい値（本論文では5秒とした）以下のSNAP候補時刻からなるTの部分集合をΦ_{t_i}とし，Φ_{t_i}におけるSNAP候補時刻の差のうち最頻のものを基本拍間隔とする．本論文では，Φ_{t_i}におけるSNAP候補時刻の差を0.2秒ごとに分類し，最頻となる時刻の差を基本拍間隔とした．

(vi) T から SNAP 近傍総確率が最大の時刻 t を選ぶ.

(vii) T の SNAP 候補時刻のうち，t との時間差が t の基本拍間隔の $\frac{1}{8}$ 以内の SNAP 候補時刻は SNAP でないとし，T から取り除く.

(viii) T のうち，まだ選ばれていない時刻を，SNAP 近傍総確率の大きい順に t として選び，(vii) を繰り返す．これを T の選ばれていない時刻がなくなるまで繰り返す.

□

上記 (i)〜(viii) の操作で得られる T の時刻を SNAP とする．このようにして得られた各 SNAP を用いて，時間-周波数-パワー空間のスペクトルを “snapshot”–周波数–パワーのスペクトルにする．ここでは，各 snapshot で，各周波数における snapshot 内でのパワーの最大値をその周波数でのパワーとした．なお，上記の SNAP 抽出手続きにおいて，本論文で用いたしきい値などのパラメータは実験的に定めた.

3. **倍音除去**

次に，上記で得られた各 snapshot において，倍音除去を行い，各音（半音単位の音高で表現）の有無を調べる．この倍音除去は，同時に発音している音の基本周波数に大まかに対応した表現を得るために行う.

まず，ある周波数について，対数軸上で最も周波数の近い半音単位の音高をその周波数の対応する基本音と呼ぶこととする．そして，各 snapshot で周波数軸方向のパワーのローカルピークについて，その対応する基本音を求め，その音が存在したとする．このようにして，この snapshot でローカルピークであった音を $a_1, a_2, \ldots, a_N$ とし，そして，それらが対応する基本音を $A_1, A_2, \ldots, A_J$ とする．なお，ここでは，異なるピーク a_i，a_j が同じ基本音 A_k に対応し得る．そして，$h_{i,1}, h_{i,2}, \ldots, h_{i,\delta}$ を a_i の 1 倍音から δ 倍音とし，このうち，この snapshot 内で存在する a_i の倍音の個数を c_i とする．そして，$c_i \geq \varepsilon$ のとき，a_i の対応する基本音 A_j が存在したとする．ここで ε は倍音除去のしきい値である．すなわち，a_i の δ 倍音のうち ε 個以上が存在すれば，この snapshot 内で a_i の対応する基本音が存在したとする．本論文では，

$$c_i = \sum_{k=1}^{\delta} e(h_{i,k}) \tag{3.2}$$

とし，$e(h_{i,k})$ は，$h_{i,k}$ の対応する基本音と同じ基本音に対応する音が存在した[1]場合に 1，そうでないとき 0 とした.

[1]ここでは，snapshot 内で各周波数のパワーを正規化し，ある周波数について，そのパワーがしきい値より大きい場合，その周波数の音が存在するとした.

4. **2値多重音響特徴ベクトルへの変換**

上記で得られた基本音の有無に着目し，各 snapshot について 12 ビットの 2 値多重音響特徴ベクトルを構成する．

特徴ベクトルの各ビットは下位から順に A，A#，B，. . . ，G#の各音に対応しており，この snapshot 内で同じ音名となる基本音が存在したとき，対応するビットを 1 に，そうでないとき 0 にする．例えば，A(220Hz)，E(330Hz)，A(440Hz)，C(523Hz)，A(880Hz) が snapshot 内に存在した場合，この snapshot の特徴ベクトルは 2 進表現で “000010001001” とする．2 値多重音響特徴ベクトルは，多重音響信号において同時に発生する音の有無に大まかに対応していることになる．

上記の 1 の音響特徴抽出から 4 の 2 値多重音響特徴ベクトルへの変換までの手続きにより 2 値多重音響特徴ベクトルを抽出する．

(2)　2値多重音響特徴ベクトルの類似度

2 値多重音響特徴ベクトル $\mathbf{x} = (x_1, x_2, \ldots, x_{12})$ と $\mathbf{y} = (y_1, y_2, \ldots, y_{12})$ の間の類似度は

$$\frac{\sum_{k=1}^{12} u_k}{\sum_{j=1}^{12} (x_j + y_j)} \tag{3.3}$$

と定義する．ここで u_k は，$x_k = y_k = 1$ のとき $u_k = 2$ で，そうでないとき $u_k = 0$ と定義される．なお，$\mathbf{x} = \mathbf{y} = \mathbf{0}$ の場合，$\mathbf{x}$ と $\mathbf{y}$ の類似度は 1 とする．これは無音の snapshot 同士は同じ和音として取り扱うことを意味している．なお，類似度としてはハミング距離を 0 から 1 の範囲に正規化したものなども考えられるが、予備実験の結果，比較的良好な探索結果が得られた式（3.3）を用いた．

(3)　2値多重音響特徴ベクトルの符号化

2 値多重音響特徴ベクトルの符号化は 4 で述べた 2 値多重音響特徴ベクトルを単に 12 ビットの 2 進数とみなすことで行う．この符号化により，2 値多重音響特徴ベクトルを簡潔に符号化できると同時に，探索時に参照符号系列の符号をシフトしてから探索することにより，移調された音楽の探索も可能となる．なお，符号間の類似度（符号間類似度）も対応する 2 値多重音響特徴ベクトル間の類似度（式 (3.3)）により定義するものとする．

(4) 探索の高速化

提案の高速化手法の説明に先立ち，本探索で用いた DP 照合について説明する．$\mathbf{T} = [c_T(1), c_T(2), \ldots, c_T(n)]$ を蓄積符号系列，$\mathbf{P} = [c_P(1), c_P(2), \ldots, c_P(m)]$ を参照符号系列とする．そして，探索窓が蓄積符号系列中のある位置に置かれたとき，その中の部分符号系列（以下，単に部分系列と呼ぶ）を $\mathbf{W} = [c_W(1), c_W(2), \ldots, c_W(w)]$ と表す．$\mathbf{P}$ と $\mathbf{W}$ の DP 照合において，$c_P(x)$ と $c_W(y)$ の累積類似度 $S(x, y)$ を，

$$S(x,y) = \max \begin{bmatrix} S(x, y-1) + s(c_P(x), c_W(y)), \\ S(x-1, y-1) + 2s(c_P(x), c_W(y)), \\ S(x-1, y) + s(c_P(x), c_W(y)) \end{bmatrix} \tag{3.4}$$

ただし，

$$S(x,1) = S(x-1,1) + s(c_P(x), c_W(1)) \tag{3.5}$$

$$S(1,y) = S(1,y-1) + s(c_P(1), c_W(y)) \tag{3.6}$$

$$S(1,1) = 2s(c_P(1), c_W(1)) \tag{3.7}$$

とする．ここで，$s(c_P(x), c_W(y))$ は $c_P(x)$ と $c_W(y)$ の符号間類似度であり 0 から 1 の値をとるものとする．なお，極端な符号の対応付けを防ぐため，適当な整合窓を用いることとし，整合窓外の格子点 (x, y) を通るパスは無視し，式 (3.4)〜(3.7) では，そのようなパスを選択しないこととした．そして，$\mathbf{P}$ と $\mathbf{W}$ の系列間類似度は $S(m, w)$ を

$$\frac{S(m,w)}{m+w} \tag{3.8}$$

として正規化したものとする．

次に，探索の高速化手法について述べる．この探索は単に DP 照合を用いたずらし照合法（以下，単にずらし照合法と呼ぶ），すなわち，蓄積符号系列上で探索窓を順に 1 符号ずつずらしながら探索窓内の部分系列と参照符号系列を DP 照合で照合し，蓄積符号系列のすべての部分系列と照合を行う手法でも行うことができる．しかし，より高速な探索のため，ここで説明する探索法を提案する．この手法の高速化の要点は，参照符号系列と蓄積信号中の部分系列の照合を類似し得る部分系列との照合に限定することと，この照合にかかる計算量自体を軽減することにある．なお，符号間の類似度が同じであれば提案探索法とずらし照合法では同一の探索結果が得られる．しかし，符号間の類似度が “スパース” であるとき，提案探索法ではより高速に探索結果が得られる．なお，符号間の類似度がスパースとは符号間類似度が 0 より大きい符号と符号の対が少ないことを意味している．図 3.4 に提案探索法の高速化の基本アイデアを示す．

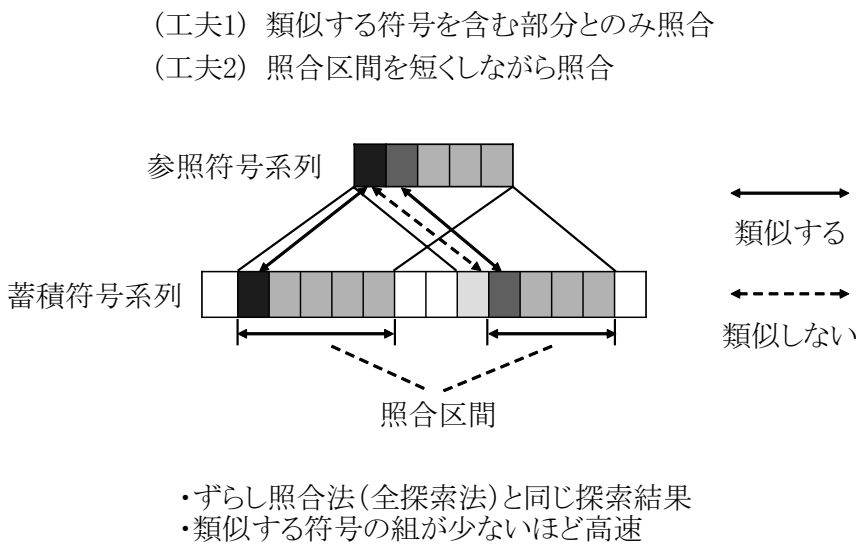
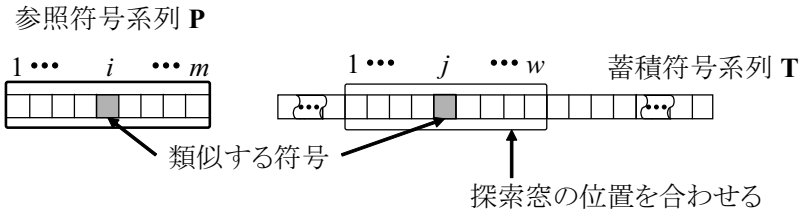


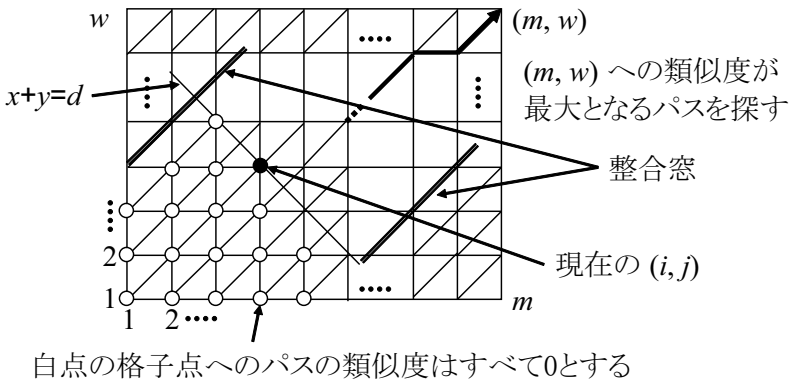
図 3.4: 提案探索法の高速化の基本アイデア

ここで，符号間の類似度を表す参照テーブルを導入する．本論文ではこれを類似度行列と呼ぶ．類似度行列の (u, v) 成分は符号 u と v の符号間類似度を表す．この類似度行列はずらし照合法においても符号間類似度の参照に用いられる．しかし，この類似度行列をスパースにすること，すなわち，ほとんどの成分の値が 0 となるような類似度定義を用いることで，提案探索法ではより高速な探索が行える．

図 3.5 に提案探索法の概要を示す．本手法では，まず各符号について蓄積符号系列中での出現位置を調べておく．そして，探索窓の位置は，図 3.5(a) に示すように，$\mathbf{P}$ の一符号と類似する符号を含む位置に合わせる．図 3.5(a) では，$\mathbf{P}$ の i 番目の符号が探索窓内の j 番目の符号に類似している．ここで符号が類似するとは符号間類似度が 0 より大きいということである．そして，蓄積符号系列中の探索窓内の部分系列と参照符号系列を図 3.5(b) のようにして照合する．図 3.5(b) のグラフは DP 照合での符号の対応付けを行うグラフを表しており，水平方向の軸（x 軸）は $\mathbf{P}$ の各符号に，垂直方向の軸（y 軸）は探索窓内の符号系列の各符号に対応する．なお，図中の二重線は整合窓の境界を表す．そして，通常の DP 照合では整合窓内の各格子点について累積類似度を計算するが，提案手法では，後述するように，白点については累積類似度は計算せずとも 0 となる．このようにして，累積類似度を計算する点の数を減らすことにより，DP マッチングの計算量を削減しながら，参照符号系列と類似する符号を持つ部分系列とだけ照合を行うことで探索を高速に行う．



(a) 類似する符号がある未照合の部分系列に探索窓を合わせる.



(b) 白点の格子点への累積類似度を 0 として，他の整合窓内の格子点について累積類似度を計算し，(m, w) への累積類似度最大のパスを探す.

図 3.5: 高速な探索法（提案探索法）の概要

以下に，提案手法を詳細に説明する．

提案手法

（ステップ 1） 各符号について $\mathbf{T}$ での出現位置を調べる．そして，符号の種類ごとに，その符号の出現位置をリストとして保持する．このリストは（ステップ 4）の探索窓の位置合せに用いる．

（ステップ 2） $\mathbf{T}$ の q 番目を先頭とする探索窓で未照合か照合済かを表すフラグを $f_q(1 \leq q \leq n-w+1)$ とし，すべての q について f_q を未照合に初期化．

（ステップ 3） 照合においては，ある d について，$x+y<d$ となる格子点 (x,y) では累積類似度は 0 となるように，直線 $x+y=d$ 上の格子点 (x,y) から以下の（ステップ 4）の手順に従い位置合せを行い，照合する．具体的には，d の値を 2 から z まで 1 ずつ増やしながら（ステップ 4）を繰り返す．z は $(1-s_t)(m+w)+2$ より小さい最大整数である．

（ステップ 4） 1 から $\min(m, d-1)$ までのすべての整数 i について，1 から順に以下を行う．j を $d-i$ とし，格子点 (i,j) が整合窓内の点ならば，$c_P(i)$ と類似する符号を類似度行列から抜き出し，その符号の出現位置を（ステップ 1）のリストから順に選ぶ．選ばれた出現位置を k としたとき，$1 \leq k-j+1 \leq n-w+1$ かつ f_{k-j+1} が未照合であるなら，探索窓の先頭位置を $\mathbf{T}$ の $k-j+1$ 番目に合わせ（図 3.5(a)），参照符号系列と探索窓内の部分系列を図 3.5(b) のようにして照合し，f_{k-j+1} を照合済とする．なお，$k-j+1<1$ または $k-j+1>n-w+1$ の場合や，f_{k-j+1} がもとより照合済の場合は照合しない．そして，照合の結果，系列間類似度が s_t より大きい場合，現在の探索窓の位置に類似区間が見つかったとする．上記の照合を，$c_P(i)$ に類似するすべての符号について,（ステップ 1）のリストから得られる $\mathbf{T}$ 上でのすべての出現位置を用いて行う.（ステップ 4）では，以上をすべての i について順に行う．

□

（ステップ 4）の照合は，図 3.5(b) のように行われる．ここで，図の白点の格子点の累積類似度は 0 とし，整合窓内の白点以外の点についてだけ累積類似度を計算する．なお，直線 $x+y=d$ 上の格子点 (i,j) 及び格子点 (i,j) より右下の格子点については，整合窓内であれば累積類似度の計算が必要である．

提案手法では以下の (a)，(b) 及び更に (c) により，累積類似度 $S(x,y)$ の計算回数を減らし，高速化を図っている．

(a) （ステップ 3）で d の値の上限を z とし，系列間類似度の上限値が s_t 以下になった場合，探索を打ち切る．

(b) （ステップ 4）では，探索窓が未照合の位置に合わされた場合に照合が行われる．ゆえに，この照合においては，図 3.5(b) で白点で表される格子点 (x,y)，すなわち，x と y がすでに i と j の値として用いられたような x と y については $c_P(1), c_P(2), \ldots, c_P(x)$ と $c_W(1), c_W(2), \ldots, c_W(y)$ の間に類似する符号はなく，$S(x,y)$ は 0 となり，$S(x,y)$ の計算を行わない．

(c) 図 3.5(b) の照合では白点以外の格子点 (x,y) について，式 (3.4), (3.5), (3.6), (3.7) に従い，x と y を 1 から, 各々，m と w まで増やしていき $S(x,y)$ を順次求め，各格子点への生き残りパスを求めていく．ここで，格子点 (x,y) を通る格子点 (m,w) へのパスの累積類似度の上限値 $S_u(x,y)$ は

$$S_u(x,y) = S(x,y) + (m-x) + (w-y) \tag{3.9}$$

となる．そして，

$$\frac{S_u(x,y)}{m+w} \leq s_t \tag{3.10}$$

の場合，(x,y) を通るパスでは s_t を越えない．そこで,（ステップ 4）の図 3.5(b) の照合においては，$S(x,y)$ を求めていく過程で，$\frac{S_u(x,y)}{m+w} \leq s_t$ となった格子点 (x,y) を通るパスの探索は打ち切る．そして，すべてのパスの探索が打ち切られた場合，その探索窓の位置での照合を打ち切る．

(5) 計算量の比較

ここでは提案探索法とずらし照合法の計算量について述べる．両探索法においては $S(x,y)$ の計算が主である．そして，この計算は式 (3.4), (3.5), (3.6), (3.7) により行われ，計算量は定数オーダである．そこで $S(x,y)$ の計算回数（$S(x,y)$ を計算する格子点の数）で両手法の計算量を見積もる．以降，簡単のため，式 (3.4), (3.5), (3.6), (3.7) による 1 回の $S(x,y)$ の計算を符号照合と呼ぶ．

探索窓の幅を w，蓄積符号系列の長さを n とし，このとき，一度の DP 照合において整合窓内にある格子点の数を c とすると，ずらし照合法における符号照合回数は合計で

$$c(n-w+1) \tag{3.11}$$

となる．提案探索法については，最悪の場合はずらし照合法と同じ符号照合回数が必要である．しかし，符号と符号が類似しない確率を平均 p_d とすると，平均的には符号照合回数は，(4) の (a) と (c) による削減を無視しても，

$$\left(c - \sum_{l=1}^{c} {p_d}^l\right)(n - w + 1) \tag{3.12}$$

となる．式 (3.11) と式 (3.12) より，類似度行列がスパースであればあるほど，提案探索法の符号照合回数はずらし照合法に比べて少なくなることがわかる．また．ずらし照合法に対する提案探索法の符号照合回数の割合は

$$\frac{\left(c - \sum_{l=1}^{c} {p_d}^l\right)(n - w + 1)}{c(n - w + 1)} = 1 - \frac{1}{c}\sum_{l=1}^{c} {p_d}^l \tag{3.13}$$

であり，c が小さいほど，その割合は小さくなり，符号照合回数の削減の効果が大きいことがわかる．

3.4　提案手法の評価

2値多重音響特徴ベクトルを用いた探索の探索精度の評価及び，2値多重音響特徴ベクトルを用い類似度行列をスパース化した場合の探索精度及び提案探索法による探索速度の評価のための実験を行った．

実験は実験用の音楽DBを用い，この中から実際に類似音楽を探索することで行った．音楽DBには，ポップス，クラシック，インストルメンタルなどの様々なジャンルの多重奏の楽曲216曲の音響信号を用いた．なお，ここでは，これらの216曲の録音のそれぞれを楽曲と呼ぶ．このDBの音響信号の長さは合計で約480分であった．なお，音響信号は48kHzサンプリング，16bit量子化，ステレオ録音のものをモノラル化して用いた．実験用の参照信号については，参照信号の長さにより，探索精度，探索速度及び速度向上率が変化するため，各実験で示す長さごとに15個ずつ用意し探索実験を行った．各参照信号は複数の三重奏アンサンブル音楽の音響信号から切り出した楽曲の一部である．なお，これらのアンサンブル音楽については，それぞれ，参照信号を切り出した楽曲（原曲）を含め，計8曲の類似楽曲がDBに含まれている．これらは，原曲と，原曲に移調，楽器変換，テンポ変換のアレンジを行った各編曲を実際に演奏家に2回ずつ演奏してもらい録音したものである．実験ではこれらの八つの類似楽曲（目的楽曲）中の参照信号と同じ演奏区間（目的区間）の探索及び類似楽曲の探索を行うこととした．すなわち，各参照信号に対し，8個の目的区間及び目的楽曲があることになる．なお，これらのアンサンブル音楽の演奏にはピアノ，フルート，

オーボエ，バスーン，バイオリン，チェロを用いた．そして，楽器変換の際には一つまたは二つのパートを異なる楽器で演奏した．

探索精度については区間単位，楽曲単位で評価することとした．区間単位の探索精度については，各探索において，検出された区間のうちの正答の割合 (P) と，目的区間のうち検出されたものの割合 (R) が等しくなる探索しきい値でのこれらの割合の値とした．なお，両割合が等しくなることが複数回ある場合は，両割合が等しくなったときの値のうちの最大値とした[2]．ある探索結果を正答と判断する基準については, その探索結果の区間がある目的区間と重複率 [11] が 0.5 以上のとき, この探索結果を正答とするとした．なお，区間 a と区間 b の重複率は

$$\text{重複率} = \frac{(a \cap b)\ \text{の区間長}}{(a \cup b)\ \text{の区間長}} \tag{3.14}$$

である．また, 目的区間が検出されたかの判断も探索により検出されたある類似区間との重複率が 0.5 以上の時, 検出されたとする．また，検出された区間のうち，重複率が 0.5 以上で重なり合う区間は一つの検出結果とみなした．

楽曲単位の探索精度については，楽曲中に類似区間が一つでも検出された場合，その楽曲は検出されたとする．そして，楽曲単位の場合の探索精度は, 各探索において，検出楽曲中の目的楽曲 (P') の割合と, 目的楽曲のうちの検出された割合 (R') が等しくなる探索しきい値でのこれらの割合の値とした[3]．なお，両割合が等しくなることが複数回ある場合は，両割合が等しくなったときの値のうちの最大値とした．

3.4.1 探索精度の評価実験

探索精度の比較評価のためスペクトル特徴を用いた場合，2 値多重音響特徴ベクトルを用いた場合，2 値多重音響特徴ベクトルを用いスパースな類似度行列を用いた場合の三通りの場合について探索精度を調べた．なお，2 値多重音響特徴ベクトルを用いスパースな類似度行列を用いた場合については 2 値多重音響特徴ベクトル（スパース化）と表記する．

スペクトル特徴の場合は，31 ページの音響特徴抽出で述べた帯域通過フィルタを用い，探索時の極端な速度低下を避けるため，分析区間を 400ms とし 100ms ごとに抽出した．そして，符号化は LBG アルゴリズムを用いて学習した符号帳（code book）に基づくベクトル量子化（VQ）とし，符号間類似度は VQ コードが同じ場合に 1，それ以外は 0 とした．なお，

[2]実際には P と R は探索しきい値に対し不連続であるため，探索しきい値を 0.02 単位で変化させその間は直線で結んだ P と R のグラフの交点を，両割合が等しくなるときとしている．

[3]P' と R' も探索しきい値に対し不連続であるため，探索しきい値を 0.02 単位で変化させその間は直線で結んだ P' と R' のグラフの交点を，両割合が等しくなるときとしている．

表 3.1: 探索精度の比較（区間単位）

参照信号の長さ	スペクトル特徴	2値多重音響特徴ベクトル	2値多重音響特徴ベクトル (スパース化)
符号長 10	28 %	44 %	40 %
符号長 30	47 %	70 %	64 %
符号長 50	54 %	72 %	71 %
符号長 70	55 %	82 %	79 %
四分音符 24 音符分	53 %	73 %	71 %

ベクトル量子化の際の符号帳の大きさは，2値多重音響特徴ベクトルとの比較のため 4096（$=2^{12}$）とした．

そして，2値多重音響特徴ベクトルの場合は式 (3.3) に従った符号間類似度により定義される類似度行列を用い，2値多重音響特徴ベクトル（スパース化）の場合においては，2値多重音響特徴ベクトルの場合の類似度行列の 0.85 より小さい成分の値を 0 として得られるスパースな類似度行列を用いた．なお，2値多重音響特徴ベクトルを用いた場合の符号長は DB 全体で 94744 であった．

実験では 3 通りいずれの場合も，各特徴及び類似度に基づき 3.3.1 の探索手続きにより探索した．ただし，参照符号系列の符号長 m に対し，探索窓の符号長 w については，$w = m$ とし，DP 照合における整合窓については，$|x - y| < \max(2, \lfloor 0.06m \rfloor)$ となる格子点 (x, y) は整合窓内の点であるとして定義した．w は m に対し可変であるが, 本探索実験においては参照信号と同じ演奏区間を探索することを目的としているため $w = m$ とした．

なお，2値多重音響特徴ベクトルと 2値多重音響特徴ベクトル（スパース化）の場合は参照符号系列の 2値多重音響特徴ベクトルを 1 ビットずつ 12 通りにシフトしたものについても探索を行った．

表 3.1，表 3.2 に各特徴における区間単位の探索精度と楽曲単位の探索精度を示す．参照信号の長さは 2値多重音響特徴ベクトルで抽出した際の符号長，または，実際の演奏における音符による長さを表している．上述のように，各長さについて 15 個の参照信号を用いて評価しており，表の探索精度はその平均値である．なお，実際の参照信号の平均の長さは符号長 10，30，50，70 の場合で，各々，約 3，8，14，19 秒であった．このとき，標準偏差は，各々，約 0.2，0.5，0.9，0.9 であり，いずれの場合も，平均値からの偏差は平均値の 20% 以下であり，平均値からの変動はほとんどなかった．四分音符 24 音符分の場合は，参照信号の長さは平均約 21 秒であり，符号長の平均は約 74 であったが，符号長の最小値は 21，最大

表 3.2: 探索精度の比較（楽曲単位）

参照信号の長さ	スペクトル特徴	2 値多重音響特徴ベクトル	2 値多重音響特徴ベクトル (スパース化)
符号長 10	45 %	58 %	55 %
符号長 30	56 %	82 %	77 %
符号長 50	60 %	84 %	81 %
符号長 70	63 %	89 %	86 %
四分音符 24 音符分	65 %	81 %	82 %

値は 114，標準偏差は 34.7 とばらつきがあった．

表 3.1，表 3.2 のいずれにおいても，実用上重要と思われる符号長が 30 から 70，または，四分音符 24 音符分の場合において 2 値多重音響特徴ベクトルの探索精度はスペクトル特徴を大きく上回っている．また，2 値多重音響特徴ベクトルで類似度行列をスパース化しても探索精度はあまり低下しないことがわかる．なお，四分音符 24 音符分の場合，表 3.1 においては 2 値多重音響特徴ベクトルの場合の探索精度は 2 値多重音響特徴ベクトル（スパース化）の場合の探索精度より上であるが，表 3.2 においては 2 値多重音響特徴ベクトル（スパース化）の場合の探索精度の方が高くなっている．2 値多重音響特徴ベクトル（スパース化）の場合の類似度行列のスパース化は，異なる曲の識別にはより有効である場合があるが，例えば同じ曲内の繰り返しのように，似ているが異なる区間を識別する場合には，その識別ができなくなる場合がある．今回の四分音符 24 音符分の場合は，繰返しにより類似する区間が同一曲内に複数回含まれるようなクエリーがあり，そして，目的区間以外は不正解とみなされることから，2 値多重音響特徴ベクトル（スパース化）の場合に区間単位の探索精度が楽曲単位の探索精度に比べ低くなってしまったものと思われる．

次に，表 3.2 の実験において，異なる種類の変動がある類似楽曲を探し出す際に，2 値多重音響特徴ベクトルを用いた場合，2 値多重音響特徴ベクトルを用い類似度行列をスパース化した場合について，変動の種類に応じて検出性能がどのように変化するかを調べた．

原曲，テンポ変換，移調，楽器変換の各変動ごとに，検出率を調べた結果を表 3.3，3.4，3.5，3.6 に示す．ここで，検出率とは表 3.2 での各探索において，各変動をもつ目的楽曲が探索結果に含まれた割合[4]であり，表 3.3，3.4，3.5，3.6 の値は 15 個の参照信号での各変動における検出率の平均値である．

表 3.3，3.4 を見ると，スペクトル特徴, 2 値多重音響特徴ベクトル, 2 値多重音響特徴ベク

[4] 43 ページの脚注 4 で述べた P' と R' のグラフの交点に最も近い探索しきい値での検出率.

表 3.3: 検出率（原曲）

参照信号の長さ	スペクトル特徴	2値多重音響特徴ベクトル	2値多重音響特徴ベクトル (スパース化)
符号長 10	97 %	90 %	93 %
符号長 30	97 %	100 %	100 %
符号長 50	100 %	100 %	100 %
符号長 70	100 %	100 %	100 %
四分音符 24 音符分	100 %	97 %	100 %

表 3.4: 検出率（テンポ変換）

参照信号の長さ	スペクトル特徴	2値多重音響特徴ベクトル	2値多重音響特徴ベクトル (スパース化)
符号長 10	50 %	63 %	70 %
符号長 30	90 %	83 %	80 %
符号長 50	100 %	80 %	77 %
符号長 70	100 %	90 %	90 %
四分音符 24 音符分	100 %	87 %	90 %

トル（スパース化）のいずれの場合にも，高い検出率が得られていることがわかる．これは，いずれの場合にも，照合においてはDP照合を用いており，信号の時間伸縮が吸収されているためと思われる．ただし，スペクトル特徴の場合はスペクトル特徴を細かい間隔で抽出する必要があり，探索時間の観点からは2値多重音響特徴ベクトル，2値多重音響特徴ベクトル（スパース化）のようにSNAP単位での特徴抽出が望ましい．

また，表3.5，3.6からは，2値多重音響特徴ベクトルと2値多重音響特徴ベクトル（スパース化）の検出率がスペクトル特徴の検出率を大きく上回っていることがわかる．

これらの実験結果では，2値多重音響特徴ベクトルを用いた手法は移調や楽器変換の変動に対して，スペクトル特徴に比べ総じて高い検出率を示すことがわかる．しかし，2値多重音響特徴ベクトルを用いた手法においても原曲や，テンポ変換、移調などの変動に比べると，楽器変換による検出率の低下があり，特にスパース化した場合に検出率の低下が見られる．

表 3.5: 検出率（移調）

参照信号の長さ	スペクトル特徴	2 値多重音響特徴ベクトル	2 値多重音響特徴ベクトル (スパース化)
符号長 10	27 %	50 %	40 %
符号長 30	27 %	87 %	83 %
符号長 50	30 %	93 %	97 %
符号長 70	40 %	97 %	100 %
四分音符 24 音符分	50 %	90 %	93 %

表 3.6: 検出率（楽器変換）

参照信号の長さ	スペクトル特徴	2 値多重音響特徴ベクトル	2 値多重音響特徴ベクトル (スパース化)
符号長 10	7 %	27 %	17 %
符号長 30	10 %	53 %	47 %
符号長 50	10 %	70 %	53 %
符号長 70	13 %	77 %	53 %
四分音符 24 音符分	10 %	50 %	47 %

3.4.2 探索速度の評価実験

次に提案探索法の探索速度についての評価実験結果を述べる．ずらし照合法においては 3.4.1 の 2 値多重音響特徴ベクトルの場合について探索を行い，提案探索法においては 2 値多重音響特徴ベクトル（スパース化）の場合について探索を行った．なお，探索窓の幅や整合窓については 3.4.1 と同じ条件であり，両探索手法とも 12 通りにシフトした 参照符号系列についても探索を行っている．探索精度はずらし照合法においては 3.4.1 の 2 値多重音響特徴ベクトルの場合，提案探索法においては 2 値多重音響特徴ベクトル（スパース化）の場合と同じになる．以下は，これらの条件下での探索速度の評価となる．

表 3.7，表 3.8 に表 3.1，表 3.2 の各探索精度における両手法の 1 回の探索における符号照合回数の平均値を，表 3.9，3.10 に 1 回の探索にかかった CPU 時間の平均を示す．表 3.7，表 3.8 の照合回数比はずらし照合法の符号照合回数に対する提案探索法の符号照合回数の割合を示している．そして，表 3.9，表 3.10 の速度比はずらし照合法による探索速度を 1 としたのときの提案探索法の探索速度を示している．なお，CPU 時間の計測には PC（CPU:Pentium 4 2GHz，L2 キャッシュ256 kByte，OS:Red Hat Linux 7.2）を用いた．区間単位，楽曲単位

表 3.7: 符号照合回数の比較（区間単位）

参照信号の長さ	ずらし照合法	提案探索法	照合回数比
符号長 10	31,180,800	728,282	1/43 (2.3 %)
符号長 30	93,434,880	7,895,351	1/12 (8.5 %)
符号長 50	149,468,160	22,677,253	1/6.6 (15.2 %)
符号長 70	457,962,240	114,775,998	1/4.0 (25.1 %)
四分音符 24 音符分	577,487,245	208,073,344	1/2.8 (36.0 %)

表 3.8: 符号照合回数の比較（楽曲単位）

参照信号の長さ	ずらし照合法	提案探索法	照合回数比
符号長 10	31,180,800	777,273	1/40 (2.5 %)
符号長 30	93,434,880	8,765,645	1/11 (9.4 %)
符号長 50	149,468,160	24,099,514	1/6.2 (16.1 %)
符号長 70	457,962,240	119,386,645	1/3.8 (26.1 %)
四分音符 24 音符分	577,487,245	212,282,596	1/2.7 (36.8 %)

の探索精度のいずれの場合にも提案探索法の符号照合回数はずらし照合法の数%から 40%程度であり，CPU 時間にして 18 倍から 1.6 倍の高速化を達成している．符号照合回数の削減率よりも CPU 時間における高速化の割合が小さいのは，式 (3.9) と式 (3.10) の計算のオーバヘッドによるものと考えられる．なお，表 3.9，表 3.10 の提案探索法の CPU 時間には（ステップ 1）の計算にかかる CPU 時間は含まれていない．これは，探索前に，各蓄積符号系列ごとに各符号の蓄積符号系列中での出現位置を求めておくことも可能だからである．なお,（ステップ 1）の計算にかかる CPU 時間は DB 全体に対し，0.01 秒とわずかであり，探索ごとに行っても探索時間はほとんど変わらない．

3.5　むすび

本章では多重奏音響信号による類似音楽探索のための 2 値多重音響特徴ベクトルを提案し，それによる類似音楽探索法を提案した．そして，本探索における高速化手法について述べた．216 曲の実験用音楽 DB を用いた探索実験では，スペクトル特徴に比べて 2 値多重音響特徴ベクトルの方が高精度であることが示された．例えば，楽曲単位の探索において，クエリーの符号長が 70（平均 19 秒）のとき，スペクトル特徴を用いた場合に 62.5%であった

表 3.9: 探索時間の比較（区間単位）

参照信号の長さ	ずらし照合法	提案探索法	速度比
符号長 10	3.07 s	0.17 s	18.1
符号長 30	10.38 s	1.62 s	6.4
符号長 50	17.81 s	4.57 s	3.9
符号長 70	49.60 s	22.01 s	2.3
四分音符 24 音符分	66.90 s	40.14 s	1.7

表 3.10: 探索時間の比較（楽曲単位）

参照信号の長さ	ずらし照合法	提案探索法	速度比
符号長 10	3.07 s	0.17 s	18.1
符号長 30	10.38 s	1.72 s	6.0
符号長 50	17.81 s	4.75 s	3.7
符号長 70	49.60 s	22.58 s	2.2
四分音符 24 音符分	66.90 s	40.79 s	1.6

精度が，提案の 2 値多重音響特徴ベクトルを用いることにより 89.3%にまで向上した．また，本高速化手法においては，複数の類似楽曲を探す場合に，全体としては，精度をほとんど損なわずに，DP 照合を用いたずらし照合法と比べて約 1.6 倍から 18 倍の高速化を達成した．

なお，本章で対象とした類似音楽探索では演奏の元となった楽譜が同一な音楽を探索対象とした．そのため，これらの類似楽曲においては和音の構成音は変更されていない．今後，和音の変更もある場合についても提案手法を適用していくこと考える必要がある．また，2 値多重音響特徴ベクトルを用いた手法は移調や楽器変換の変動に対して，スペクトル特徴に比べ総じて高い頑健性をもつが，2 値多重音響特徴ベクトルを用いた手法においても原曲や，テンポ変換、移調などの変動に比べると，楽器変換による検出率の低下がある．また，類似度行列をスパース化した場合に検出率の低下が見られる．今後は，このような変動に対する性能の改善のため，楽器変換などにより頑健な 2 値多重音響特徴ベクトルの抽出法を検討する必要があると考えられる．また，探索を更に頑健にしながら，高速な探索を行うための特徴ベクトルの符号化法と類似度行列の構成法も検討すべきと考えられる．

第4章　フラクタル画像圧縮のための画像間距離の高速計算回路の実現法

4.1　まえがき

第2章と第3章では，メディア処理技術の中でも，特に音楽音響信号を対象としたメディア探索技術について述べた．一方，メディア情報としては，音楽音響信号の他に，画像情報や映像情報などもある．画像情報，映像情報についても，そのコンテンツの増加は著しく，大量の画像情報，映像情報が流通しており，これらを効率よく扱うメディア処理技術が求められている．そして，画像情報や映像情報はそのデータ量が多いため，必要な情報を必要なときに利用するためには，データを圧縮し，効率良く格納・伝送するためのメディア圧縮技術が不可欠である．このメディア圧縮技術には，高い圧縮率，圧縮歪みが小さいこと，高速にデータの圧縮・復元ができることが望まれる．

本章では，フラクタル画像圧縮を対象として，その効率化について述べる．フラクタル画像圧縮処理では, 画像のマッチングが処理の大部分を占めるため, このマッチングが処理をハードウェアにより効率的に行う手法を提案する．

フラクタル画像圧縮はその圧縮率や復元した画像の忠実性から将来を期待されている画像の圧縮手法である．特に，低ビットレート時の画像の忠実性による期待が大きい．しかし，その圧縮にかかる計算時間のため，その用途は圧縮時間が問題とならない分野に限られてきた．フラクタル画像圧縮の幅広い利用のためには，その圧縮処理の高速化が不可欠であり，そのため，ハードウェアとしてVLSIで実現する符号化器などが提案されてきた [28, 49, 50, 51, 52, 53].

基本的なフラクタル画像圧縮においては，圧縮したい画像を複数の部分画像に分割し，これらの部分画像間の類似性を，ブロック内の濃度値の2乗誤差（2乗距離）で測り，類似する部分画像を探すという処理が行われる．そして，これらの部分画像同士の照合（すなわち距離計算）を非常に多くの回数行う必要があり，このことが圧縮に長い計算時間を要する原因となっている．そのため，既に提案されているいくつかの符号化器では，複数の部分画像の組について，同時（並列）に距離計算をおこなうアーキテクチャをとり圧縮の高速化をはかっている．これらはフラクタル画像圧縮専用のハードウェアとして設計されるものが多

い．一方，本論文で提案する手法は，汎用の再構成可能ハードウェアを用いて，画像間の距離計算自体をパイプライン処理により高速化するアプローチをとる．なお，本論文では，特に画像間の距離計算のための積和演算の高効率処理を目的とするが，この演算は画像の照合において，汎用的に行われる計算であり, 他の画像処理にも適用可能である．

ハードウェアとしてはFPGA(Field Programmable Gate Array)に代表されるプログラマブルロジックデバイス（再構成可能ハードウェア）を用いる．再構成可能ハードウェアを用いることで，専用に設計，製造されたハードウェアを用いることなしに，汎用の再構成可能ハードウェアに所望の回路を実現し，ハードウェア（専用回路）を用いた並列処理による高速計算が可能になる [54]．そのため，メディア処理などにおいても，再構成可能ハードウェアを用い，必要な演算を高速に行う手法などが研究されている [55]．また，再構成可能ハードウェアの中には，回路の動作中に，回路を部分的にも再構成できるデバイスなども開発され [56, 57]，これらの再構成可能性を用いたアプリケーション処理手法の研究なども行われている [58]．そこで，本章では，フラクタル画像圧縮という画像圧縮手法に着目し，この圧縮手法の中で行われる画像間の距離計算を再構成可能ハードウェアを用い，限られた計算資源の中で効率的に並列処理する手法を提案する．なお，本論文では，フラクタル画像圧縮におけるパターンマッチングの高速化を目的とするため，フラクタル画像圧縮自体の詳細には立ち入らない．フラクタル画像圧縮については，その概要及び対象とした圧縮法について説明するにとどめる．

以下，4.2節で本論文で対象とするフラクタル画像圧縮の概要について説明する．次に，4.3節で画像間距離計算回路の再構成可能ハードウェアを用いた実現法について述べる．そして，4.4節において提案手法の評価を行う．そして，最後に4.5節を本章のむすびとする．

4.2 対象とするフラクタル画像圧縮

本節では本論文で対象とするフラクタル画像圧縮の概要について述べる．フラクタルによる画像圧縮の枠組はM. Barnsleyにより最初に提案された反復関数系によるものであり [59]，そして，白黒濃淡画像の符号化への応用はJacquinにより行われた [60, 61]．フラクタル画像圧縮の詳細は文献 [61, 62, 63, 64] を参照されたい．なお，本論文で対象とするフラクタル画像圧縮はquad-tree decompositionを用いた白黒濃淡画像を対象としたものである [61, 62]．

対象とするフラクタル画像圧縮においては，まず，圧縮したい画像（原画像）をレンジブロックを呼ばれる重なりのない部分画像に分割する．そして，各レンジブロックについて，それを最もよく近似する縦横2倍の大きさの部分画像（ドメインブロックと呼ぶ）を原画像から探索する．そして，各レンジブロックについて，レンジブロックの画素値の代わりに，

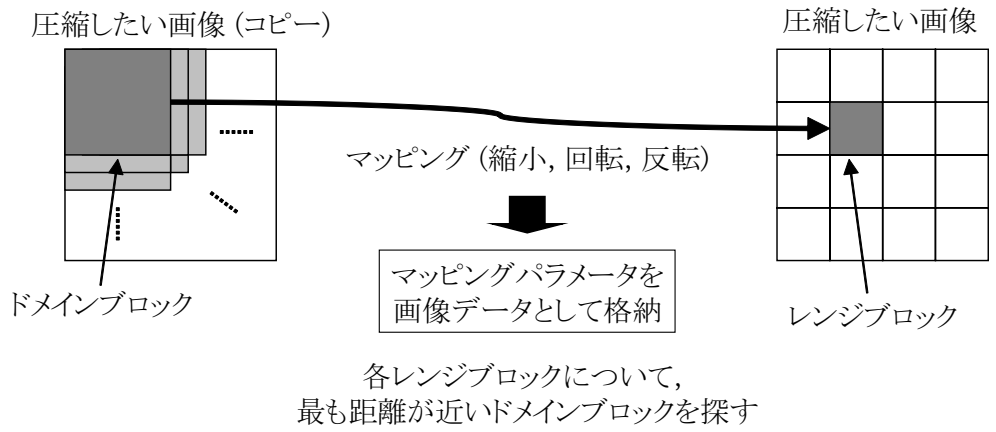


図 4.1: 対象とするフラクタル画像圧縮の概要

そのレンジブロックを近似するドメインブロックからレンジブロックへのマッピングパラメータを保存する．ただし，近似の度合についてはあらかじめ許容度が定められているものとし，設定した許容度未満で近似する部分画像がなかったレンジブロックについては，レンジブロックをさらに縦横半分の大きさの部分画像に分割し，この 4 つの部分画像を新たなレンジブロックとし，これらを近似するドメインブロックを探索する．図 4.1 に対象とするフラクタル画像圧縮の概要を示す．

近似の度合は画像間距離を用いる．$R_1, R_2, \ldots, R_p$ を大きさ $B \times B$ ピクセルのレンジブロックとする．また，$D_1, D_2, \ldots, D_q$ を大きさ $2B \times 2B$ ピクセルのすべての部分画像とする．$D_1, D_2, \ldots, D_q$ がドメインブロックとなる．ドメインブロックは大きさ $2B \times 2B$ ピクセルのすべての部分画像であり，重なりを許すものとする．そして，すべてのドメインブロックからなるドメインブロックの集合をドメインプールと呼ぶ．

次に，各レンジブロックについて，そのレンジブロックを最もよく近似する，すなわち，画像間距離が最も小さいドメインブロックを探す．そして，許容誤差以内で近似するドメインブロックが見つかったレンジブロックについては，covered とし，見つからなかったレンジブロックは uncovered とする．なお，ドメインブロックとレンジブロックを照合し，距離計算を行う際には，ドメインブロックは 2×2 の画素を平均して 1 画素にして大きさを縮小し，レンジブロックと同じ大きさの画像にしてから距離計算を行う．なお，レンジブロック R_k と縮小したドメインブロック D_j の距離 $R(D_j, R_k)$ は，D_j の画素値を $a_1, a_2, \ldots, a_n$，R_k の画素値を $b_1, b_2, \ldots, b_n$ としたとき，

$$R(D_j, R_k) = \sum_{i=1}^{n} (s\, a_i + o - b_i)^2 \tag{4.1}$$

として計算する．ここで，s と o は濃度に関するスケーリングとオフセット成分である．

$R(D_j, R_k)$ を最小にする s と o は，

$$s = \frac{n\sum_{i=1}^{n} a_i b_i - \sum_{i=1}^{n} a_i \sum_{i=1}^{n} b_i}{n\sum_{i=1}^{n} a_i^2 - \left(\sum_{i=1}^{n} a_i\right)^2} \tag{4.2}$$

$$o = \frac{\sum_{i=1}^{n} b_i - s\sum_{i=1}^{n} a_i}{n} \tag{4.3}$$

であり，このとき，$R(D_j, R_k)$ は，式 (4.3) に基づき o を展開し，

$$R(D_j, R_k) = \frac{1}{n}\left(n\sum_{i=1}^{n} b_i^2 - \left(\sum_{i=1}^{n} b_i\right)^2\right) - \frac{s^2}{n}\left(n\sum_{i=1}^{n} a_i^2 - \left(\sum_{i=1}^{n} a_i\right)^2\right) \tag{4.4}$$

として計算できる．式 (4.4) で与えられる画像間距離に基づき，各レンジブロックについて，それを最もよく近似するドメインブロックを探索し，適切なドメインブロックが見つからなかったレンジブロックについてはレンジブロックを再度分割し，ドメインブロックも縦横半分の大きさの部分画像として，同様の処理を繰り返す．なお，近似誤差を小さくするためドメインブロックには元の部分画像に加え，90 度，180 度，270 度回転させたものと，これらを縦方向の軸に関して反転させたものも加える．

以上のようにして，各レンジブロックについて，そのレンジブロックを最も良く近似するドメインブロックと回転，反転の各種類を求める．同時にそのときの s と o の値を求める．それらの数値に最後に無歪みの圧縮を行うのが符号化の概略である．なお，8 ビット階調の濃淡画像について，s と o は各々5 ビット，7 ビットの精度があれば良いことが経験的に知られている．この圧縮手続きを図 4.2 に示す．画像の復元は次のように行う．まず，任意の画像データを初期画像として与え，各レンジブロックについて，その符号に基づき，それを近似するドメインブロックから対応する変換（マッピング）を行い，それをもとのレンジブロックと置き換える．これをすべてのレンジブロックに対して行い，新たな画像データを得る．新たに得られた画像データに対して，このマッピング処理を繰り返すことで復元が行われる．通常，この画像全体の置き換え処理を 10〜20 回繰り返すと，十分な忠実度の画像に復元できるとされている．

上で述べたように，フラクタル画像圧縮においては画像の距離計算を多数回行う必要がある．通常，レンジブロックには 8×8 ピクセルや 4×4 ピクセルの大きさのものが使われている．256×256 ピクセルの画像を圧縮対象として 8×8 ピクセルのレンジブロックを考えた場合，レンジブロックの数は 1,024 になり，16×16 ピクセルのドメインブロックの数は 58,081 となる．この場合，単純には約 6 千万回の画像の距離計算が必要なことになる．この

```
For (k = 1 ; k ≤ num_range ; k++) {
    min_R = tolerance;
    For (j = 1 ; j ≤ num_domain ; j++) {
        compute s;
        if (0 ≤ s < 1.0)
            if (R(D_j, R_k) < min_R) {
                min_R = R(D_j, R_k);
                best_domain[k] = j;
            }
    }
    if (min_R == tolerance)
        set R_k uncovered and divide it into 4 smaller blocks;
}
```

図 4.2: フラクタル画像圧縮処理の概要

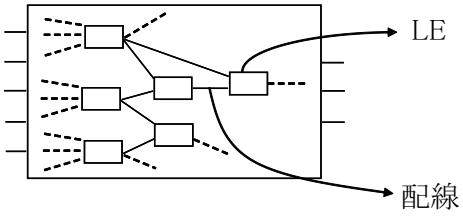
問題に対し，照合するレンジブロックとドメインブロックの組合せを減らす手法も古くから検討されている [63]．しかし，この場合，照合する組合せを減らすと，復元した画像の画質が低下するという問題があり，圧縮速度と画質の両面から見た場合，まだ十分な手法があるとはいえない．そこで，本論文では，単にこれらの画像間の距離計算自体をパイプライン処理により高速化するというアプローチをとる．

4.3 画像間距離計算回路の再構成可能ハードウェアを用いた実現法

4.3.1 再構成可能ハードウェア

ここで，本論文で想定する再構成可能ハードウェアについて述べる．図 4.3 に想定する再構成可能ハードウェアの概念図を示す．本論文で想定する再構成可能ハードウェアは通常の FPGA のように，細粒度の論理関数を構成できるロジックエレメント（Logic Element，本論文では以下 LE と表記する）と，目的の回路に応じて LE を接続できる LE 間の再構成可能な配線からなる．なお，各 LE は 3〜4 入力 1 出力の任意の論理関数を構成できるものと

・再構成可能なロジックエレメント (LE)
　　−ファンクションユニット:3〜4入力1出力の任意の論理関数を実現
　　(3〜4ビット入力1ビット出力のメモリ)
　　−論理関数の出力保持用に1ビットのレジスタが付属
・再構成可能な配線:LE間の配線を再構成可能



LEと配線を動作中に再構成可能

図 4.3: 想定する再構成可能ハードウェアの概念図

し，パイプライン処理の際に論理関数の出力を保持するための1ビットのレジスタが付属しているものとする．また，これらのLEと配線は回路の動作中に部分的にも再構成可能なものとする[1]．LEのアーキテクチャとしてはLUT(Look-Up-Table)と呼ばれるメモリを用いるものやマルチプレクサを用いるものなどがあるが，本論文ではLEは単に3または4入力1出力の任意の論理関数を実現できる回路素子として扱う．

4.3.2 計算回路のアーキテクチャとその実現法

図4.2のアルゴリズムと式(4.2)–(4.4)を見ると，ループの最も内側の計算のうち，画像の距離計算において $\sum_{i=1}^{n} a_i$，$\sum_{i=1}^{n} a_i^2$，$(\sum_{i=1}^{n} a_i)^2$，$\sum_{i=1}^{n} b_i$，$\sum_{i=1}^{n} b_i^2$，$(\sum_{i=1}^{n} b_i)^2$，$(n\sum_{i=1}^{n} a_i^2 - (\sum_{i=1}^{n} a_i)^2)$，そして $(n\sum_{i=1}^{n} b_i^2 - (\sum_{i=1}^{n} b_i)^2)$ といった計算はレンジブロックまたはドメインブロックごとに前もって計算しておくことができることがわかる．実際に画像間の距離計算において，画像の組合せごとに必要な計算は s と $R(D_j, R_k)$ の計算にあり，より具体的に，計算量が大きいのは $\sum_{i=1}^{n} a_i b_i$ の計算である．そこで提案手法ではこの積和計算を行う回路（Processing Element，本論文では以下PEと表記する）を実現し，パイプライン処理により，順次，この計算を高速に行うアプローチをとる．なお，この画像間の積和演算は画像の照合において，汎用的に行われる計算である．

図4.4にPEの基本的なアーキテクチャを示す．本アーキテクチャにおいては，レンジブロックの画素値 $b_1, b_2, \ldots, b_n$ はあらかじめPEに与えられている．そして，ドメインブロッ

[1] このような再構成可能なハードウェアのプロトタイプとしてはXilinx社のXC6200シリーズFPGA[56]やAtmel社のAT40K FPGA[57]などが発表されている．

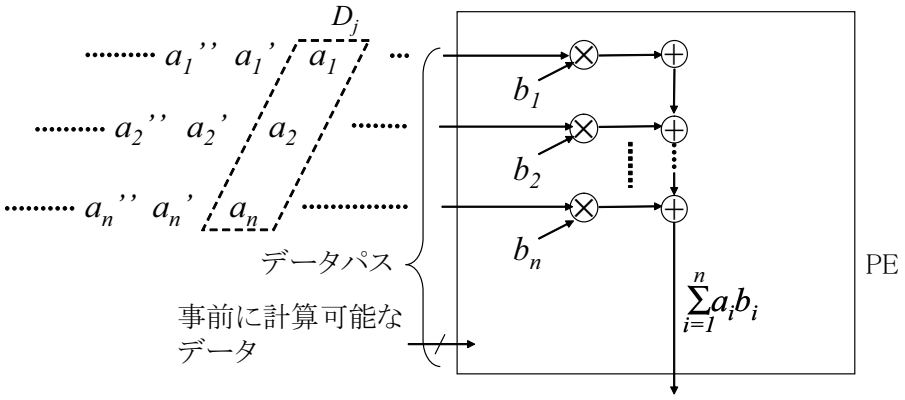


図 4.4: PE のアーキテクチャ

クの画素値 $a_1, a_2, \ldots, a_n$ がパイプライン式に PE に提供され，パイプラインの 1 ステップごとに一組の $a_1, a_2, \ldots, a_n$ について $\sum_{i=1}^{n} a_i b_i$ の積和計算が完了する．なお，本アーキテクチャは n 個の乗算器と $n-1$ 個の加算器で実現できる．また，このときの a_i と b_i の乗算器も b_i の各ビットを a_i に乗算したものをシフトしながら加算していくことで実現される．このとき，1 つの乗算器は b_i のビット幅から 1 を引いた数の個数の加算器で実現される．

次に，ハードウェアの再構成可能性を利用して少ない LE 数で積和演算を行う PE を実現する方法について述べる．上述のように，図 4.4 の PE の乗算器もシフトと加算器により構成される．再構成可能ハードウェアにおいてはシフトは LE 間の配線により実現可能である．そこで，加算器の数を減らすことを考える．まず，整数値 b_i の 2 値表現を $b_{i,l} b_{i,l-1} \cdots b_{i,1}$ とする．ここで，$b_{i,j}$ は 0 か 1 のいずれかである．そして，l は a_i と b_i のビット幅を表す．このとき，$a_i b_i$ の乗算は

$$a_i b_i = \{(a_i b_{i,l}) \ll (l-1)\} + \{(a_i b_{i,l-1}) \ll (l-2)\} + \cdots + \{a_i b_{i,1}\} \tag{4.5}$$

の形で表現できる．ここで，$a \ll j$ は a の j ビット左シフトを表す．この式を用いれば，乗算は $l-1$ 個の加算器で行われる．したがって，PE での $\sum_{i=1}^{n} a_i b_i$ の計算に必要な加算器の数は，

$$(l-1)n + (n-1) = ln - 1 \tag{4.6}$$

となる．

一方，PE を特定のレンジブロックに応じて再構成する場合，PE にとって b_i は定数である．このことを利用すると，$b_{i,j}$ が 0 の場合，$(a_i b_{i,j})$ の部分は式 (4.5) から取り除くことができる．以下では，$(a_i b_{i,j})$ の部分をできるだけ取り除き，より少ない加算器数で積和演算を

実現する．まず，$\sum_{i=1}^{n} a_i b_i$ の計算式を以下のように書き換える．

$$\begin{aligned}\sum_{i=1}^{n} a_i b_i &= \{(a_1 b_{1,1} + a_2 b_{2,1} + \cdots + a_n b_{n,1})\} \\ &\quad + \{(a_1 b_{1,2} + a_2 b_{2,2} + \cdots + a_n b_{n,2}) \ll 1\} \\ &\quad \cdots \\ &\quad + \{(a_1 b_{1,j} + a_2 b_{2,j} + \cdots + a_n b_{n,j}) \ll (j-1)\} \\ &\quad \cdots \\ &\quad + \{(a_1 b_{1,l} + a_2 b_{2,l} + \cdots + a_n b_{n,l}) \ll (l-1)\}\end{aligned} \tag{4.7}$$

そして，式 (4.7) の j 番目の行の

$$(a_1 b_{1,j} + a_2 b_{2,j} + \cdots + a_n b_{n,j}) \tag{4.8}$$

の計算は，

$$\sum_{i=1}^{n} a_i - (a_1 \overline{b_{1,j}} + a_2 \overline{b_{2,j}} + \cdots + a_n \overline{b_{n,j}}) \tag{4.9}$$

としても計算可能である．ここで，$\sum_{i=1}^{n} a_i$ はドメインブロックごとに事前に計算して与えることができる．そして，PE を $b_{1,j}, b_{2,j}, \ldots, b_{n,j}$ のうち 1 であるビットの数が $\frac{n}{2}$ より大きいとき，PE を式 (4.8) の代わりに式 (4.9) のように構成する．これをすべての j について行う．このとき，PE での $\sum_{i=1}^{n} a_i b_i$ の計算に必要な加算器の数は，最悪時でも，

$$\frac{nl}{2} - 1 \tag{4.10}$$

となる．ゆえに，式 (4.6) に比べ少なくとも加算器の数を半分に減らすことができる．なお，式 (4.10) は最悪時の加算器数であり，これはすべての $j (1 \leq j \leq l)$ について，$b_{1,j}, b_{2,j}, \ldots, b_{n,j}$ のうちの 1 であるビットの数が $\frac{n}{2}$ か $\frac{n}{2}+1$ のいずれかである場合に起こる．実際には，$b_i (1 \leq i \leq n)$ は部分画像の画素値であり，j が l に近いときには，$b_{1,j}, b_{2,j}, \ldots, b_{n,j}$ は同じ値になる傾向がある．それゆえ，式 (4.10) の最悪の場合はほとんど起こらず，実際に必要な加算器の数は式 (4.10) よりも少ないことが期待できる．

画像の距離計算のための積和演算を，特定の画像（レンジブロック）に応じて効率よく再構成可能ハードウェアに構成する手法を述べた．このようにして回路規模を削減することと，再構成可能ハードウェアに実現できる回路の規模が大きくなれば，この距離計算回路 PE を 1 チップの再構成可能ハードウェア上に複数実現し，並列に動作させることも考えられる．このとき，再構成可能ハードウェアが動的かつ部分的に再構成可能であれば，各 PE を非同期に異なるレンジブロックに応じて再構成することが可能になる．その場合，図 4.5

```
For (j = 1 ; j ≤ num_range ; j++) {
    flag = 0;
    For (k = 1 ; k ≤ num_domain ; k++) {
        compute s;
        if (0 ≤ s < 1.0)
            if (R(D_j, R_k) < tolerance) {
                best_domain[j] = k;
                flag = 1;
                break;
            }
    }
    if (flag == 0)
        set R_j uncovered and divide it into 4 smaller blocks;
}
```

図 4.5: 高速化アルゴリズム

に示すようなフラクタル画像圧縮の高速化が可能になる．図 4.2 のアルゴリズムでは，各レンジブロックについて，与えられる全てのドメインブロックのうち最も距離の小さいドメインブロックを探索している．一方図 4.5 においては，距離が *tolerance* より小さいドメインブロックを見つけた時点で探索を終了し，そのドメインブロックが，現在のレンジブロックを近似するブロックとする．適切な *tolerance* を選択することで，画像のマッチング回数を削減でき，画像圧縮を高速化できる．

4.4 提案手法の評価

4.4.1 回路規模の評価

本節では，4.3.2 で提案した PE の実現法による PE に必要な LE 数の削減効果を評価する．表 4.1 に，8 ビット階調（1 ピクセルの画素値を 8 ビットの整数で表現）の白黒濃淡画像画像で，レンジブロックの大きさが 4×4 ピクセルの場合と 8×8 ピクセルの場合に $\sum_{i=1}^{n} a_i b_i$ の計算に必要な加算器数を示す．従来法はハードウェアの再構成可能性を用いずに単純に実現

図 4.6: サンプル画像（Girl，256 × 256 ピクセル, 8 ビット階調の白黒濃淡画像）

表 4.1: PE に必要な加算器の数

レンジブロックのサイズ（ピクセル）	従来法（単純に実現した場合）	提案法		
		最悪時	平均	
			Girl	Lena
8 × 8	511 (100.0%)	255 (49.9%)	150 (30.1%)	153 (30.7%)
4 × 4	127 (100.0%)	63 (49.6%)	32 (27.6%)	32 (27.6%)

した場合で，式 (4.6) の値である．提案法の最悪時の値は式 (4.10) の値である．なお，提案法については SIDBA の実験用画像の Girl（256 × 256 ピクセル, 8 ビット階調の白黒濃淡画像）と Lena（512 × 512 ピクセル, 8 ビット階調の白黒濃淡画像）を対象画像とした場合の，$\sum_{i=1}^{n} a_i b_i$ の計算に必要な加算器数の平均も示されている． 図 4.6 と図 4.7 に実験に用いたサンプル画像を示す．表 4.1 からは，提案法により，必要な加算器数が最悪でも従来法の約半分に減り，実際の画像では平均的にはさらに削減されることがわかる．

次に，表 4.2 に PE に必要な LE の数を示す．表の見方は，4.1 と同様である．本実験では，LE は 4 入力 1 出力の LUT ベースのものを想定した．また積和演算に用いる各加算器は桁上げ保存加算器とすることとし，積和計算の最後に，桁上げ保存加算器により桁上げ保存加算器の加算結果と桁上げを桁上げ先見加算器で加算することにした．そのため，表 4.1 の

図 4.7: サンプル画像（Lena，512 × 512 ピクセル, 8 ビット階調の白黒濃淡画像）

表 4.2: PE に必要な LE の数

レンジブロックのサイズ（ピクセル）	従来法（単純に実現した場合）	提案法		
		最悪時	平均	
			Girl	Lena
8 × 8	8976 (100.0%)	5006 (55.8%)	3332 (37.1%)	3372 (37.6%)
4 × 4	2192 (100.0%)	1260 (57.5%)	774 (35.3%)	774 (35.3%)

加算器以外に一つの桁上げ先見加算器が PE に実現される．そして，これらの加算器に必要な LE 数を見積もることで，PE に必要な LE 数を見積もった．桁上げ保存加算器については，1 ビットの全加算器に必要な LE 数を 2 とし，必要な全加算器数を数えることで，必要な LE 数を見積もった．桁上げ先見加算器については，Altera 社の FPGA マッピングツール MAX+Plus II を用いて加算器を設計し必要な LE 数を見積もった．なお，桁上げ先見加算器に必要な LE 数があり，表 4.1 と 4.2 は比例関係にはならない．

表 4.2 からは，提案手法により，最悪でも LE 数を従来の約 60%に削減できることがわかる．また，実際の画像では，さらに少ない LE 数で PE が実現できることが示されている．なお，本構成の LE へのデータの入力に必要な入出力ピンはドメインブロックの画素値と事前に計算した $\sum_{i=1}^{n} a_i$ の入力用のピンとなり，これらの数は 8 × 8 のレンジブロックの場合で合計 526，4 × 4 のレンジブロックの場合で合計 140 となる．また，データ出力用に必要

な入出力ピンの数は 8×8 と 4×4 の場合で，各々，12 と 10 である．

現時点[2]においても，市販の FPGA の LE 数は 10000 を越え，入出力ピンの数は約 500 となっており，提案法により構成した PE は十分な実現可能性がある．また，提案手法による LE 数の削減は 1 チップの再構成可能ハードウェアに LE 数を複数実装するためにも意義がある．この場合，データ入力用の入出力については，パイプライン状に PE を接続して入力データを共有することで，必要な入出力ピンの増加を抑えることができる．

4.4.2 画像のマッチング回数の評価

本節では 4.3.2 で述べた高速化アルゴリズムの効果を評価する．図 4.2 のアルゴリズム（全探索法）と図 4.5 のアルゴリズム（高速化アルゴリズム）とも，*tolerance* を大きくすると，画像のマッチング回数が減り，SN 比が低くなる．*tolerance* を小さくすると，画像のマッチング回数が増え，SN 比が高くなる．そこで，先の Girl と Lena をサンプル画像として，各アルゴリズムについて *tolerance* を変化させて，画像のマッチング回数と復元した画像の SN 比を調べた．なお，画像の忠実度を測る SN 比は，

$$\text{SN 比} = 10\log_{10}(dr(S)^2/d_e(S,S'))(\text{dB}) \tag{4.11}$$

とした．ここで，$dr(S)$ は原画像 S のダイナミックレンジ，$d_e(S,S')$ は S と復元画像 S' の距離であり，

$$d_e(S,S') = \sum_{i=1}^{t}(\mu_i - \mu'_i)^2 \tag{4.12}$$

とした．μ_i と $\mu'_i(1 \le i \le t)$ は，各々，S と S' の画素値である．通常，28 から 32dB の SN 比であれば，十分は忠実度があるといえる．

なお，画像の復元は，得られた圧縮結果から行うマッピングを 20 回繰り返すことで行った．圧縮においては，まず，8×8 ピクセルのレンジブロックを用い，uncoverd であったものを 4×4 ピクセルのレンジブロックに分割した．また，ドメインブロックとしては，元の画像を分割して得られるドメインブロックのみを用いており，回転，反転したものは用いていない．

図 4.8，4.9 に実験結果を示す．全探索法では，Girl で SN 比が 26.5dB のときに，Lena で SN 比が 31.5dB のときに，8×8 ピクセルのレンジブロックが 4×4 ピクセルのレンジブロックに分割されることはなくなった．この場合はそれ以上，*tolerance* を大きくしても，画像のマッチング回数が減ることはない．Girl について SN 比が 28 から 30dB のとき，高速化ア

[2] 1998 年のデータブックによる [65].

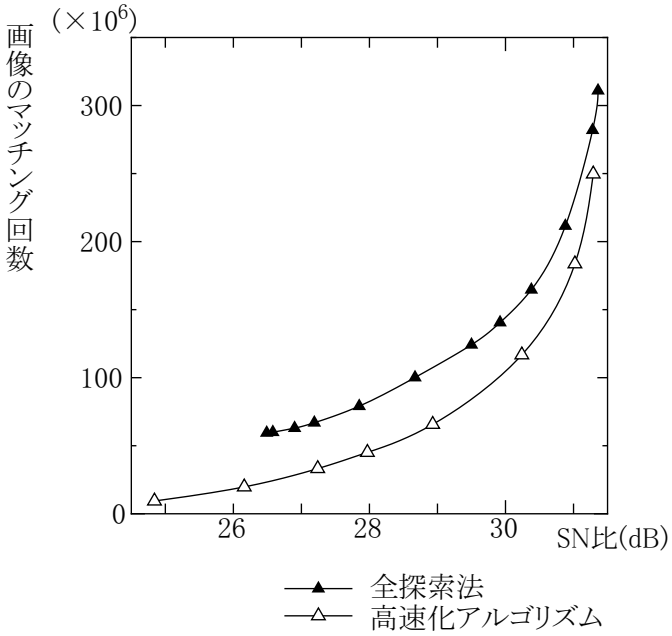


図 4.8: SN 比と画像のマッチング回数の関係（Girl）

ルゴリズムの画像のマッチング回数は全探索法の約 55%から 70%であった．また，Lena については，SN 比が 28 から 31.5dB のときに，高速化アルゴリズムの画像のマッチング回数は全探索法の約 10%から 30%であった．これらのサンプル画像については，実用的な SN 比において，10%から 55%まで画像のマッチング回数が削減されていた．

この高速化は，フラクタル画像圧縮をハードウェア上に実現した場合にも，有効であり，実装することが望ましいと考えられる．

4.5 むすび

本章では，フラクタル画像圧縮のための画像間距離の高速計算回路の実現法を提案した．提案した回路では，画像の距離計算（積和演算）をパイプライン処理により高速に行う．さらに，提案した実現法により，この並列度を保ったまま，必要な回路規模を約 60%にまで削減することができる．

また，提案手法で構成する回路を複数実現し，これらを特定の画像ブロックに応じて非同期に再構成した場合の圧縮の高速化について述べた．ソフトウェアによる手法と同様に，許容範囲内の近似画像ブロックをみつけるるごとに画像のマッチング処理を打ち切ることで，画像圧縮処理を高速化することが期待できる．

なお，提案手法では圧縮したい画像に応じて回路を再構成する．実際の圧縮においてはこ

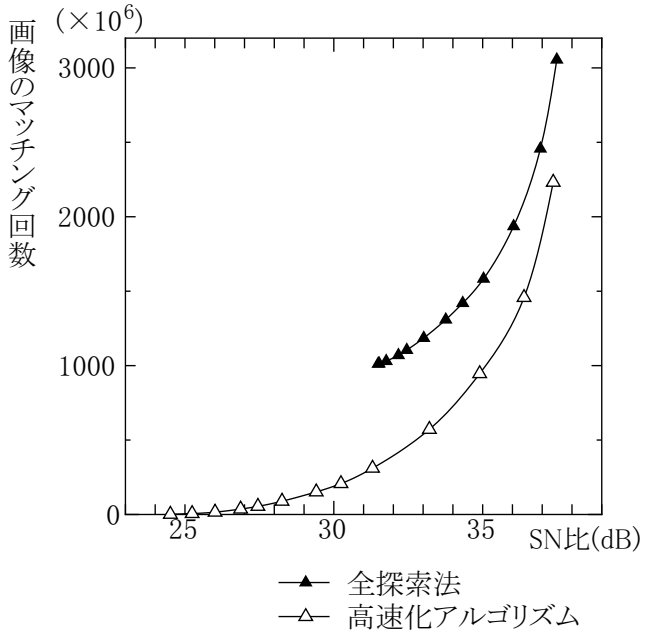


図 4.9: SN 比と画像のマッチング回数の関係（Lena）

の再構成にかかる時間も考慮する必要がある．ただし，提案した手法は，将来，より大規模な再構成可能ハードウェアが実現された場合においても，より多くの計算回路を並列動作させるために有効である．

第5章　メディア処理高信頼化のための誤り訂正符号復号法

5.1　まえがき

本章では，メディア処理高信頼化のための誤り訂正符号復号法について述べる．誤り訂正符号はデータの誤りの検出・訂正を目的として構成される符号である [14, 15, 16]．対象とする誤りは通信路で発生するランダム誤りやバースト誤りの他に，磁気テープや磁気ディスクなどの外部記憶装置で発生する誤りや演算装置で発生する誤りなどがあり，それぞれに対する誤り訂正符号が考えられ，通信，計算機，また，オーディオ・ビデオなどの分野で幅広く応用されている [66]．特に，通信においては，近年の通信量の著しい増大に伴い，帯域幅をできるだけ小さくしながら高速かつ信頼性の高い通信を行うことが求められており，より高性能な誤り訂正符号の研究開発が進められている．

第 2 章や第 3 章で述べたメディア探索技術を実際にサービスとして提供する際にも，誤り訂正符号は重要な技術と考えられる．例えば，携帯電話を通じた楽曲探索サービスを提供する場合などには，探索キーの伝送などに，誤り訂正符号の適用は不可欠であると考えられる．また，誤り訂正符号は，雑音などの変動が加わった情報から誤りを訂正し，元の情報を復元する技術であり，パターンの認識・探索や信号処理との関連が深い．例えば，誤り訂正符号の最尤復号アルゴリズムとして提案されたヴィタビアルゴリズム [67, 68] と同様の手法は，音声認識などの分野でも広く使われている [69]．第 3 章で述べた DP マッチングも基本はほぼ同様の手法と考えられる [40, 70, 71]．更に，情報をいかに符号化するかを取り扱う学問体系である情報理論の枠組において，誤り訂正符号と情報圧縮は密接な関係にあり [72]，メディア処理技術にとって誤り訂正符号は重要な役割を担っているといえる．そこで，本章では，メディア処理高信頼化を目的とし，誤り訂正復号法について述べる．特に本論文では，高い誤り訂正能力をもつ線形ブロック符号とその軟判定最尤復号を対象とし，その復号法について述べ，また，より高速な準最尤復号法と，最尤復号の再構成可能ハードウェアによる実現法を提案する．

線形ブロック符号の軟判定最尤復号には，符号のトレリスダイアグラムを用いたヴィタビ

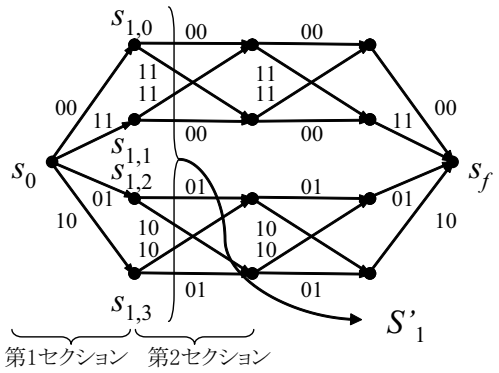


図 5.1: 符号長 8，1 次のリード・マラー符号の 4 セクショントレリスダイアグラム

アルゴリズムが標準的なアルゴリズムとして用いられている．一般に，線形ブロック符号のトレリスダイアグラムはセクションと呼ばれる単位で構成される [73, 74, 75, 76, 77]．2 元 (N, K) 線形ブロック符号 C は N セクショントレリスダイアグラム T を持つ [76]．ここで，T は C を受理集合とする有限オートマトンの状態遷移図と考えられる．T は初期状態 s_0 と最終状態（受理状態）s_f をもち，内部状態間の状態遷移を表す枝は 0 または 1 の 1 ビットのラベルをもつ．また，N 以下の非負整数 h について，S_h を T の h ビット目の位置における状態の集合とする．N を L の倍数としたとき，L セクショントレリスダイアグラムは，各枝が長さ N/L の符号ビットを表すトレリスダイアグラムとして構成される．図 5.1 に符号長 8，1 次のリード・マラー符号の 4 セクショントレリスダイアグラムを示す．このトレリスダイアグラムにおいては隣り合う 2 つの状態は 2 ビットの枝ラベルをもつ 1 本の枝でつながれている．一般には，隣り合う 2 つの状態の間に異なる枝ラベルをもつ枝が複数あることもある．なお，L セクショントレリスダイアグラムは状態数が最小のものが一意に定まることが知られており，以降，単にトレリスダイアグラムといえば最簡であるものとする．

リード・マラー符号（RM 符号）のように，線形ブロック符号のトレリスダイアグラムは，多くの場合，状態間の接続も疎な単純な並列構造をもつ．この並列構造を利用することで，ヴィタビアルゴリズムを用いた復号（ヴィタビ復号法）の実現が簡単化される．さらに，より小さな部分トレリスダイアグラムを並列処理することで，ヴィタビ復号法を高速に行うことができる．この高速化処理は，実際の高速データ通信システムにおいて非常に重要である．

2 元線形ブロック符号の L セクショントレリスダイアグラムの並列構造は，初め嵩らにより解析された [76]．文献 [76] では，与えられた 2 元線形ブロック符号 C の L セクショントレリスダイアグラムについて，トレリスダイアグラムの各セクションは並列成分と呼ばれる相互を接続する枝を持たない同一構造の部分グラフに分割できることが示された．さらに，

各セクションにおける並列成分の数が，C に関連した線形ブロック符号により決定されることも示され，隣合うセクションの並列成分間の接続構造なども解析されている．

L セクショントレリスダイアグラムの並列成分の枝ラベルについても嵩らにより文献 [77] において解析されている．この文献では，トレリスダイアグラムの各並列成分について，その並列成分の枝の枝ラベルの集合を枝ラベル集合とし，あるセクションの二つの並列成分について，それらの枝ラベル集合に共通な要素が一つでもある場合，二つの並列成分は同一の枝ラベルを持つこと，更に，これらの同一の枝ラベルを持つ並列成分の数やそれらの構造までもが C に関連した線形ブロック符号により決定されることが示されている．また，文献 [77] では，これらのトレリスダイアグラムの詳細構造を用いた最尤復号アルゴリズム（本論文では分割トレリス復号法と呼ぶ）とその計算量の上界が示されている．この最尤復号アルゴリズムの計算量は，その上界（最悪時の計算量）でもヴィタビ復号法の計算量より小さい．また，分割トレリス復号法の平均の計算量は，最悪時の計算量よりも非常に小さいことが予想される．

そこで，本章では，分割トレリス復号法の平均の計算量が，最悪時の計算量よりも非常に小さいことを計算機シミュレーションにより示す．そして，そのシミュレーションを分析した結果に基づき，より計算量の小さい準最尤復号法である部分探索復号法を提案する．また，分割トレリス復号法及び部分探索復号法はソフトウェア向けのアルゴリズムである．そこで，ヴィタビ復号法を第 4 章で用いた再構成可能ハードウェアに実現し，トレリスダイアグラムの構造に応じた並列処理を行うことで，ヴィタビ復号法をソフトウェア処理より高速に行う手法について述べる．

以下，5.2 節でヴィタビ復号法と分割トレリス復号法について述べる．次に，5.3 節で部分探索復号法と，ヴィタビ復号法の再構成可能ハードウェアによる実現法を提案する．5.4 節で各手法についての性能評価を行う．最後に，5.5 節を本章のむすびとする．

5.2 対象とする誤り訂正符号復号法

本論文で対象とする誤り訂正復号法は，ヴィタビ復号法，分割トレリス復号法，そして，部分探索復号法である．部分探索復号法については 5.3 節で提案するため，本節では，ヴィタビ復号法と分割トレリス復号法について述べる．まず，各種手法の説明において必要となるトレリスダイアグラムについて用語を定義しその性質について述べる．

符号長 N の 2 元線形ブロック符号 C の L セクショントレリスダイアグラムは C の全ての符号語を受理する有限オートマトンの状態遷移図と考えられる．ここで L は N の約数であり，状態遷移図の各枝は符号語の部分系列を表す N/L ビットの 2 元系列を枝ラベルとして

もつ．なお，L セクショントレリスダイアグラムは状態数が最小のものが一意に定まることが知られており，以降，単にトレリスダイアグラムといえば最簡であるものとする．トレリスダイアグラムにおいては初期状態を s_0，最終状態 (受理状態) を s_f と表すことにする．また, 図 5.1 に示したように, セクショントレリスダイアグラムの部分グラフは初期状態に近い部分から順に第 1 セクション, 第 2 セクションと呼ぶ．そして, 整数 $i(1 \leq i \leq L)$ に対して, S'_i を第 i セクションの終りの状態の集合とし，$S'_0 = \{s_0\}$，$S'_i = \{s_{i,0}, s_{i,1}, \ldots, s_{i,|S'_i|-1}\}$，$S'_L = \{s_f\}$ とする．また，1 つの枝を経由して到達可能な隣り合う 2 つの状態間に複数の枝があるときそれらを並列枝と呼び，1 つの状態対の間の全ての並列枝の集合を並列枝集合, 並列枝集合の枝ラベルの集合を並列ラベル集合と呼ぶ．一般に線形ブロック符号の L セクショントレリスダイアグラムでは 1 つの状態対は複数の枝で接続されていること，すなわち, 並列枝を持つことが多い. なお，ここでは L を N の約数とし N を等間隔に L 個のセクションに分割したが，L セクショントレリスダイアグラムは N ビットを単に L 個のセクションに分けることで構成可能である．ただ，本論文では，L セクショントレリスダイアグラムといえば L は N の約数であり，各セクションは N/L ビットに対応するものとする．一方，単に初期状態から h ビット目の状態の集合を考えた場合は，これを S_h と表すことにする．

以上のトレリスダイアグラムを用いて，線形ブロック符号の軟判定最尤復号を行うヴィタビ復号法について説明する．

線形ブロック符号の軟判定最尤復号はトレリスダイアグラム上の最短経路問題に帰着される．第 i セクションの枝 b の枝ラベルを $l_b(l_b = (l_{b,1}, l_{b,2}, \ldots, l_{b,N/L}))$ とし，受信系列を $\boldsymbol{r} = (r_1, \cdots, r_N)$ とする．このとき，枝 b の枝メトリック $BM(b)$ は受信系列 r に対する l_b の尤度として定義される．実際には，通信路に加法的白色ガウス雑音通信路を想定した場合には，l_b と r の第 i セクション部分との自乗ユークリッド距離などが用いられる．自乗ユークリッド距離をメトリックとし，メトリックが小さいほど，尤度が大きいとする．そして, 初期状態 s_0 からある状態 $s_{i,j}$ への 1 つのパスのパスメトリックはそのパス上の枝の枝メトリックの総和として定義される. 初期状態 s_0 から状態 $s_{i,j}$ への全てのパスのうち最小 (最短) のパスメトリックをもつパスを状態 $s_{i,j}$ への生き残りパスといい, このパスメトリックを $s_{i,j}$ の状態メトリック $SM(s_{i,j})$ という．軟判定最尤復号法は受信系列との間のメトリックが最小の符号語を復号語とする復号法であり，一般には, トレリスダイアグラム上で最終状態 s_f への生き残りパスを探し, そのパス上のラベル系列を復号語とする．以下にこの生き残りパスの探索を行う標準的なアルゴリズムであるヴィタビ復号法のアルゴリズムを示す.

ヴィタビ復号法

（ステップ 1）全ての並列枝集合の各々について，全ての並列枝の枝メトリックを求

め, 最小の枝メトリックをもつ枝を探す.

（ステップ 2） S'_i の各状態についてその状態メトリックを, S'_{i-1} の全ての状態の状態メトリックを求めた後に, 以下の式に従い求める.

$$SM(s_{i,j}) = \min_{s_{i-1,k} \in CS(s_{i,j})} (SM(s_{i-1,k}) + MB(s_{i-1,k}, s_{i,j})) \tag{5.1}$$

ここで，$s_{i-1,k}$ を通る $s_{i,j}$ へのパスが生き残りパスとなる場合，状態番号 k を状態 $s_{i,j}$ で保持する.

以上のようにして第 1 セクションから第 L セクションまで，各状態への生き残りパスをセクション順に求める.

（ステップ 3） 最終状態 s_f への生き残りパス上のラベル系列を復号語とする. □

（ステップ 2）における $CS(s_{i,j})$ は S'_{i-1} の状態のうち $s_{i,j}$ への枝をもつ状態の集合である．また，$MB(s_{i-1,k}, s_{i,j})$ は（ステップ 1）で求められた $s_{i-1,k}$ と $s_{i,j}$ の間の並列枝の最小の枝メトリックである.

1 つのセクションにおける任意の 2 つの並列枝集合については, その並列ラベル集合は等しいか互いに素であるかのいずれかであることが知られている [76]．したがって, 同じ並列ラベル集合をもつ並列枝集合については,（ステップ 1）における計算はただ 1 度だけ行えばよい.

次に，ヴィタビ復号法と同様に線形ブロック符号の軟判定最尤復号を行う分割トレリス復号法について説明するが，分割トレリス復号法はトレリスダイアグラムの詳細構造を使うため，まずは，このトレリスダイアグラムの詳細構造を説明する．なお，通常，分割トレリス復号法の計算量はヴィタビ復号法より非常に小さい.

L セクショントレリスダイアグラムの各セクションは図 5.2 のような並列成分に分けることができる．また，一つの並列成分内の枝ラベルの集合を枝ラベル集合と呼ぶ．並列成分には次のような重要な性質がある [76, 77].

1. 並列成分と並列成分の間には枝がない.
2. 並列成分は，枝ラベルを除き，一つのセクション内で同一の構造をもつ完全 2 部グラフである.
3. 一つのセクション内の二つの並列成分について，そのラベル集合は，一致するか互いに素かのいずれかである.

次に，状態集合 S_h と状態集合 $S_{h'}$ 間のある並列成分を考える．ここで，この並列成分の S_h に含まれる状態の集合を $\bar{S}_h$，$S_{h'}$ に含まれる状態の集合を $\bar{S}_{h'}$ とする.

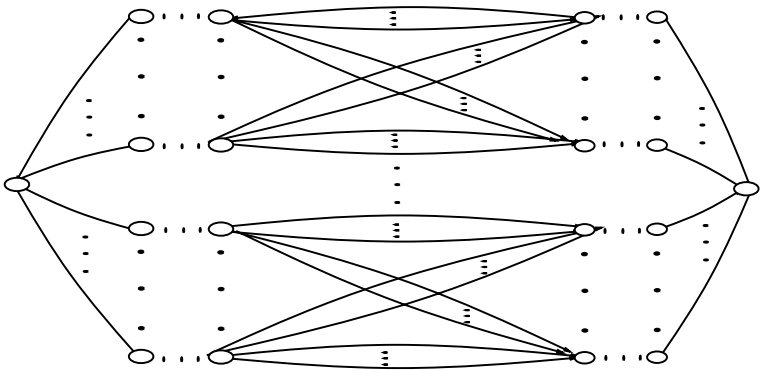


図 5.2: トレリスダイアグラムの並列成分

文献 [77] の定理 3 にあるように，任意の並列成分は次のような性質をもつ部分グラフに分解できる．

1. 状態集合 $\bar{S}_h$ は同一の大きさ ν の集合に分割できる．
2. 状態集合 $\bar{S}_{h'}$ も大きさ ν の集合に分割できる．
3. 上記で分割した状態集合 $B \subseteq \bar{S}_h$ と $B' \subseteq \bar{S}_{h'}$ の状態集合の組 (B, B') について，B と B' の状態及びそれらの間の枝からなる次の (i)，(ii) の性質を持つ部分グラフ（部分トレリス）が構成できる．
   (a) $1 \leq r \leq \nu$ の r について，B と B' の間に，B の全ての状態 s について，$1 \leq r < r' \leq \nu$, $\varphi_r(s) \neq \varphi_{r'}(s)$ となる一対一対応 φ_r が存在する．
   (b) B の各状態 s から，$\varphi_r(s)$ により対応する B' の状態へ，r のみによって決まる並列ラベル集合 P_r をもつ並列枝が存在する．

例 [77]：符号長 16，2 次の RM 符号 RM(16,11) の 4 セクショントレリスダイアグラムを考える．この第 2 セクションは両端が各 4 状態の部分トレリスからなる．この部分トレリスにおいては，$\nu = 4$ であり，状態集合 $\bar{S}_4$ は一つの部分集合 B からなり，状態集合 $\bar{S}_8$ も一つの部分集合 B' から構成される．そして，B の状態から B' の状態への並列ラベル集合は P_1，P_2，P_3，P_4 の 4 種類となる．一対一対応 $\varphi_r (1 \leq r \leq 4)$ を図 5.3(a) に示す．そして，一つの並列成分 T' の詳細構造を図 5.3(b) に示す． □

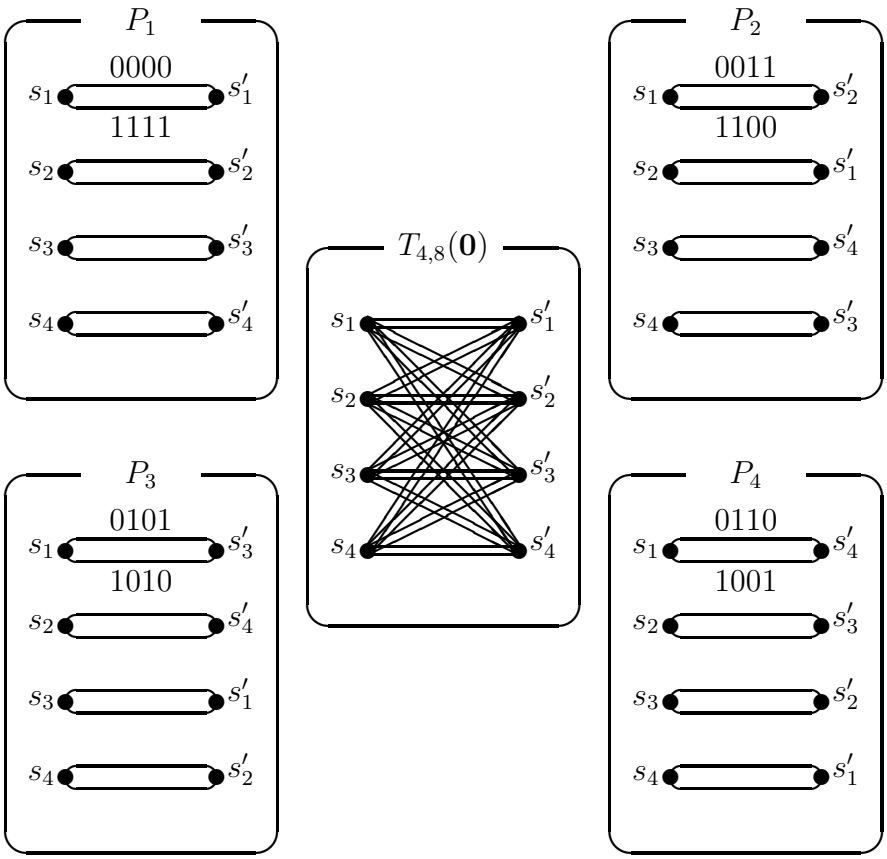
$m(B, B', r)$ は並列ラベル集合 P_r で最小のメトリックをもつ枝ラベルを表すものとする．また，

$$B = \{s_1, s_2, \cdots, s_\nu\} \tag{5.2}$$

$$B' = \{s'_1, s'_2, \cdots, s'_\nu\} \tag{5.3}$$

s	$\varphi_1(s)$	$\varphi_2(s)$	$\varphi_3(s)$	$\varphi_4(s)$
s_1	s'_1	s'_2	s'_3	s'_4
s_2	s'_2	s'_1	s'_4	s'_3
s_3	s'_3	s'_4	s'_1	s'_2
s_4	s'_4	s'_3	s'_2	s'_1

(a) 一対一対応 $\varphi_r(s)(1 \leq r \leq 4)$



(b) T' の詳細構造

図 5.3: RM(16,11) の並列成分 T' の詳細構造

とする．次に，$1 \leq i \leq \nu$ の i について，$sm(i)$ は状態 s_i の生き残りパスのメトリックを表すものとする．ここで，状態 $s'_j (1 \leq j \leq \nu)$ について，B の状態から s'_j へのパスのみを考えた場合の，s'_j の最小メトリック $sm'(B, j)$ は，

$$sm'(B, j) = \min_{\substack{1 \leq i \leq \nu \\ s'_j = \varphi_r(s_i)}} sm(i) + m(B, B', r) \tag{5.4}$$

で与えられる．従って，s'_j の生き残りパスのメトリック $sm'(j)$ は，

$$sm'(j) = \min_{B \subseteq \bar{S}_h} sm'(B, j) \tag{5.5}$$

で得られる．

以下では，文献 [77] に基づく

$$\{sm'(B, 1), sm'(B, 2), \cdots, sm'(B, \nu)\} \tag{5.6}$$

を求めるアルゴリズムを示す．分割トレリス復号法はヴィタビ復号法の（ステップ 1）の処理後に，以下のアルゴリズムを第1セクションから順に各セクションの部分トレリスに適用することで各状態への生き残りパスを求める．

分割トレリス復号法のアルゴリズム（部分トレリスに対する生き残りパスの探索手続き）

（ステップ 1） $\{sm(1), sm(2), \ldots, sm(\nu)\}$ をソートし，

$$sm(\pi(1)) \leq sm(\pi(2)) \leq \cdots \leq sm(\pi(\nu)) \tag{5.7}$$

となるような $\{1, 2, \cdots, \nu\}$ に関する置換 π を見つける．なお，ここで，このソートは各 B について一度だけ行う．また，前のセクションで得られる部分的にソートされたメトリックを用いることでこのソートにかかる計算量は軽減することができる．例えば，符号長64，3次の RM 符号 RM(64,42) の4セクショントレリスダイアグラム及び，その (64,40) 部分符号の4セクショントレリスダイアグラムの場合，第2セクションの B' と第3セクションの B は同一であり，第3セクションにおいては，本ステップのソートは実際には行わなくてよい．

（ステップ 2） $\{m(B, B', 1), m(B, B', 2), \ldots, m(B, B', \nu)\}$ をソートし，

$$m(B, B', \pi'(1)) \leq m(B, B', \pi'(2)) \leq \cdots \leq m(B, B', \pi'(\nu)) \tag{5.8}$$

となるような $\{1, 2, \cdots, \nu\}$ に関する置換 π' を見つける．なお，ラベル集合まで同一な並列成分については，このソートは対応する並列ラベル集合を持つ B と B' の組について一度だけ行う．

（ステップ 3） $J(J \subseteq \{1,2,\ldots,\nu\})$ について，

$$U(J) = \{(i,r) : 1 \le i \le \nu, 1 \le r \le \nu \text{ かつ } \varphi'_r(i) \in J\}$$

となる $U(J)$ を定める．ここで，$\varphi'_r(i) \triangleq j$ は $s'_j = \varphi_{\pi'(r)}(s_{\pi(i)})$ となるような整数 j である．ここで，$\prec$ で表す $U(J)$ についての半順序関係を導入する．$(i,r) \neq (i',r')$ である $U(J)$ の要素 (i,r) と (i',r') について，$(i,r) \prec (i',r')$ の必要十分条件を $i \le i'$ かつ $r \le r'$ あることとする．ここで $(i,r) \prec (i',r')$ の場合，

$$sm(\pi(i)) + m(B,B',\pi'(r)) \le sm(\pi(i')) + m(B,B',\pi(r')) \tag{5.9}$$

となる．そして，$J = \{1,2,\ldots,\nu\}$ とし，以下の (i) と (ii) を J が空になるまで繰り返す．J は $sm'(B,j)$ が見つからなかったような整数 j の集合である．

(i) 以下のような U_0 を得る．

$$U_0 = \{(i,r) \in U(J) : (i',r') \prec (i,r) \text{ となるような } (i',r') \text{ が } U(J) \text{ に存在しない}\} \tag{5.10}$$

(ii) 任意の $(i,r) \in U_0$ について，

$$sm(\pi(i_0)) + m(B,B',\pi'(r_0)) \le sm(\pi(i)) + m(B,B',\pi'(r)) \tag{5.11}$$

となる $(i_0,r_0) \in U_0$ を見つける．
このとき，$sm'(B,\varphi'_{r_0}(i_0))$ が $sm(\pi(i_0)) + m(B,B',\pi'(r_0))$ として得られる．
そして，$\varphi'_{r_0}(i_0)$ を J から取り除く．

□

$J = \{1,2,\ldots,\nu\}$ のとき，$U_0 = \{(1,1)\}$ である．J から $\varphi'_{r_0}(i_0)$ を取り除いた後，新しい U_0 は前の U_0 から，次のようにして得られる．

U_0 が $(i,r) \in U(\{1,2,\ldots,\nu\} - J)$ を含む間，以下の (a)–(c) を行う：

(a) (i,r) を U_0 から取り除く．

(b) $i < \nu$ のとき，$r = 1$ の場合は $(i+1,r)$ を U_0 に加え，$r = 1$ でない場合は $(i+1,r-1)$ を U_0 から取り除く．

(c) $r < \nu$ のとき，$i = 1$ の場合は $(i,r+1)$ を U_0 に加え，$i = 1$ でない場合は $(i-1,r+1)$ を U_0 から取り除く．

$a(B,B')$ と $c(B,B')$ を，各々，上記の（ステップ 3）で行われるメトリックの加算回数と比較回数とし，ma_i と mc_i は i セクションの全ての B と B' についての $a(B,B')$ の平均と $c(B,B')$ の平均とする．このとき，$a(B,B')$ の上界を求めるために以下の補題を導出する：

補題 1 $i_1+r_1>\nu+1$ である任意の整数の組 (i_1,r_1) について，(i_1,r_1) が U_0 の要素になることはない．

（証明）$i_1+r_1>\nu+1$ である (i_1,r_1) が U の要素であるとする．ここで，$j_1 \triangleq \varphi'_{r_1}(i_1)$ と定義し，

$$V_{j_1} \triangleq \{(i,r) : 1 \le i \le \nu, 1 \le r \le \nu, j_1 = \varphi'_r(i)\}$$

と定義する．$|V_{j_1}|$ は ν である．U の定義より，

$$V_{j_1} \cap \{(i,r) : 1 \le i \le i_1, 1 \le r \le r_1\} = \{(i_1,r_1)\} \tag{5.12}$$

である．次に，$i \ne i'$ のとき $\varphi'_r(i) \ne \varphi'_r(i')$，$r \ne r'$ について $\varphi'_r(i) \ne \varphi'_{r'}(i)$ から

$$|V_{j_1} \cap \{(i,r) : i_1 < i \le \nu \ \mathit{and}\ 1 \le r \le r_1\}| \le \min\{r_1-1, \nu-i_1\} \tag{5.13}$$

$$|V_{j_1} \cap \{(i,r) : r_1 < r\}| \le \nu - r_1 \tag{5.14}$$

となる．そして，式 (5.12)–(5.14) から

$$|V_{j_1}| \le 1 + \min\{r_1-1, \nu-i_1\} + \nu - r_1 = 2\nu + 1 - (i_1+r_1) < \nu$$

となり，矛盾する． □

補題 2 任意の $(i,r) \in U_0$ について，$|U_0| \le i+r-1$.

（証明）U_0 の定義より，$(i',r') \ne (i,r)$ の (i',r') について，(i',r') が U_0 の要素となるのは，$i' < i$ または $r' < r$ の場合だけである． □

補題 1 から, $a(B,B')$ の上界が得られる：

$$a(B,B') \le |\{(i,r) : i+r \le \nu+1\}| = \nu(\nu+1)/2 \tag{5.15}$$

（ステップ 3）の (ii) で，j が J から取り除かれるとき，$i+r \le 1+i_0$ である (i,r)（i_0 は $\varphi'_1(i_0)=j$ となる整数）は U_0 の要素となる．このことと，補題 2 より

$$c(B,B') \le \nu(\nu-1)/2 \tag{5.16}$$

となる．最悪時において，式 (5.15) と式 (5.16) の等号が成立する．しかし，$a(B,B')$ の平均と $c(B,B')$ の平均はこれらの上界よりも非常に小さいことが予想される．これらの値の平均

時の評価は 5.4 節で行うが，$a(B, B')$ の平均と $c(B, B')$ の平均はこれらの上界よりも非常に小さい．

（ステップ 1）及び（ステップ 2）における ν 個のメトリックのソートにおけるメトリックの比較回数の上限は，ヒープソートを用いた場合，$2\nu \log \nu$ である [78]．式 (5.6) を全探索で見つける場合のメトリックの加算及び比較の正確な回数は，各々，ν^2 と $(\nu - 1)\nu$ である．ヴィタビ復号法においては，各状態についてその生き残りパスを求めるためとき，その状態への全てのパスについてメトリックを求め，これらのメトリックを比較することで生き残りパスを探索する．すなわち，ヴィタビ復号法において B' の各状態について生き残りパスを求めるときのメトリックの加算と比較の回数は，各々，ν^2 と $(\nu - 1)\nu$ となる．ゆえに，一受信ベクトルを復号するときの分割トレリス復号法でのメトリックの加算回数と比較回数は，ν が全セクションについて非常に小さい場合を除き，どのような線形ブロック符号についても，ヴィタビ復号法の場合よりも小さいといえる．もしいくつかのセクションについて ν が非常に小さい場合には，分割トレリス復号法とヴィタビ復号法を組合せ，このようなセクションについてはヴィタビ復号法を使うことが考えられる．

5.3 準最尤復号アルゴリズムとヴィタビ復号法の並列処理

5.3.1 トレリスダイアグラムの詳細構造を用いた部分探索による準最尤復号アルゴリズム

本節では，トレリスダイアグラムの詳細構造を用いた部分探索による準最尤復号アルゴリズムを提案する．詳細な評価は 5.4 で述べるが，分割トレリス復号法の平均計算量はその上界に比べ非常に小さい．このことは，分割トレリス復号法の（ステップ 3）の処理において，比較的大きな i や r の $sm(\pi(i))$ や $m(B, B', \pi'(r))$ はほとんど参照されないことを意味している．このことに基づき，分割トレリス復号法の部分トレリスに対する生き残りパスの探索手続きを以下の手続きとする準最尤復号アルゴリズム，部分探索復号法を提案する．

部分探索復号法のアルゴリズム（部分トレリスに対する生き残りパスの探索手続き）

（ステップ 1'） 整数 $\mu_1 (1 \leq \mu_1 \leq \nu)$ を選ぶ．$\{sm(1), sm(2), \cdots, sm(\nu)\}$ を部分的にソートし，

$$sm(\pi(1)) \leq sm(\pi(2)) \leq \cdots \leq sm(\pi(\mu_1)) \leq \min_{\mu_1 < i \leq \nu} sm(\pi(i)) \tag{5.17}$$

となる $\{1, 2, \cdots, \nu\}$ 上の置換 π を見つける．なお，このソートは各 B について一度だけ行う．

（ステップ 2'） 整数 $\mu_2 (1 \leq \mu_2 \leq \nu)$ を選ぶ．$\{m(B, B', 1), m(B, B', 2), \ldots, m(B, B', \nu)\}$ を部分的にソートし，

$$m(B, B', \pi'(1)) \leq m(B, B', \pi'(2)) \leq \cdots \leq m(B, B', \pi'(\mu_2)) \leq \min_{\mu_2 < i \leq \nu} m(B, B', \pi'(i)) \tag{5.18}$$

となる $\{1, 2, \cdots, \nu\}$ 上の置換 π' を見つける．なお，このソートは同じ枝ラベルを持つ部分トレリスについて一度だけ行う．

（ステップ 3'） 分割トレリス復号法の（ステップ 3）と同じ． □

本アルゴリズムは $\mu_1 = \mu_2 = \nu$ のとき，分割トレリス復号法と同じであり，最尤復号アルゴリズムとなる．

本アルゴリズムはメトリックのソートにかかる計算量を軽減することで，復号全体における計算量を削減するものである．

5.3.2　ヴィタビ復号法の再構成可能ハードウェアを用いた実現法

分割トレリス復号法と部分探索復号法はヴィタビ復号法に比べ計算量の少ないアルゴリズムであるが，その処理はソフトウェア向けの処理といえる．一方，ヴィタビ復号法は分割トレリス復号法，部分探索復号法と比べると比較的簡単にハードウェアによる並列処理が実現できる．そこで，本節では再構成可能ハードウェアを用いたヴィタビ復号法の実現法について述べる．再構成可能ハードウェアを用いることで，符号のトレリスダイアグラムの構造に応じた並列処理が可能になる．これにより，ヴィタビ復号法をソフトウェアで実装した場合に比べ，より高速な復号処理が可能になる．

(1)　構造と動作

図 5.4 に再構成可能ハードウェア上に実装する軟判定最尤復号器の構造と動作を示す．復号器は (2) 節で詳しく述べるハードウェアコンポーネントにより構成される．$\mathrm{BRANCH_SET}_{i,j}$ は対応する第 i セクションの並列ラベル集合 $p_{i,j}$ の枝メトリックを，受信系列を入力として順に計算し，最小の枝メトリックを出力するハードウェアコンポーネントである．この計算は 5.2 節で述べたヴィタビ復号法の（ステップ 1）の並列枝集合に関する計算に対応する．$\mathrm{STATE}_{i,m}$ は状態 $s_{i,m}$ に対応するハードウェアコンポーネントである．$\mathrm{STATE}_{i,m}$ は順に入力される状態メトリックと枝メトリックを加算し，逐次, パスメトリックを求め，最終的に最小となるパスメトリックを求める．これは（ステップ 2）における状態 $s_{i,m}$ の状態メトリック

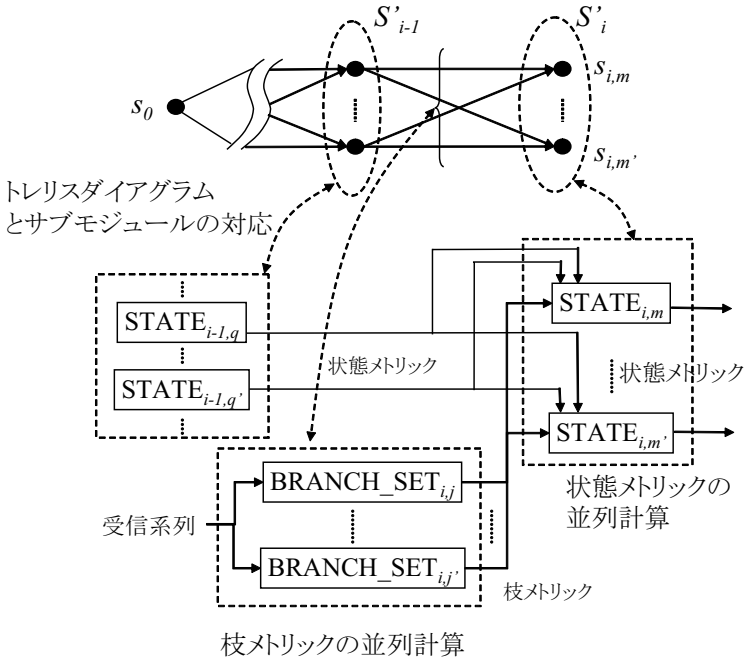


図 5.4: 復号器の構造と動作

を求める計算にあたる．第 i セクションの並列ラベル集合 $p_{i,j}$ をもつ並列枝が状態 $s_{i,m}$ に接続されている場合，それに対応して $\mathrm{BRANCH_SET}_{i,j}$ の出力が $\mathrm{STATE}_{i,m}$ の入力に接続される．そして，状態 $s_{i,m}$ と状態 $s_{i-1,m'}$ の間に枝がある場合に $\mathrm{STATE}_{i-1,m'}$ の出力が $\mathrm{STATE}_{i,m}$ に入力される．

ヴィタビ復号法の（ステップ 1）の枝メトリックを計算し最小の枝メトリックを求める計算は，全てのセクションの全ての並列ラベル集合について同時に行うことが可能である．それゆえ，全ての BRANCH_SET は並列に動作し，上に述べた処理を同時に行う．また,（ステップ 2）における状態集合 S'_i の各状態の状態メトリックの計算も，S'_{i-1} の全ての状態の状態メトリックが計算された後，S'_i の全ての状態について同時に行うことができる．この並列計算を行うため，S'_i の各状態に対応する全ての STATE は並列に動作し，各状態への生き残りパスの探索を同時に行う．この 2 点の並列処理により，復号時間を短縮することができる．符号のトレリスダイアグラムの構造は符号に特有のものであり，このような並列処理は与えられた符号に特化した回路を構成することにより可能になる．

(2) 復号器のサブモジュール

ここでは，軟判定最尤復号器のハードウェアコンポーネント (サブモジュールと呼ぶ) について詳細に説明する．

1. BRANCH_SET$_{i,j}$ は第 i セクションの並列ラベル集合 $p_{i,j}$ の枝メトリックを入力される受信系列から順に計算し，最終的に集合内で最小の枝メトリックとそれをあたえる枝ラベルを求めるためのサブモジュールである．枝メトリックと枝ラベルの番号を格納するレジスタを持ち，最初，それらには枝メトリックの最大値とラベル番号の初期値がセットされる．ラベル番号により与えられる枝ラベルと受信系列から枝メトリックを計算し，その値とレジスタに格納されている枝メトリックの値を比較し，小さい方の枝メトリックとラベル番号をレジスタに格納する．これを順次繰り返し, 最小となる枝メトリックを求め，その値とそれを与える枝ラベルの番号を記憶する．そして，出力が接続されているサブモジュール，例えば STATE$_{i,m}$ などからの要求で，記憶している $p_{i,j}$ の最小の枝メトリックを出力する．これらの機能のため，BRANCH_SET$_{i,j}$ は加算器，比較器，枝メトリックとラベル番号を記憶するレジスタ，そして，並列ラベルを生成する機能回路から構成される．

2. STATE$_{i,m}$ は状態 $s_{i,m}$ に対応するサブモジュールである．STATE$_{i,m}$ は順次入力される状態メトリックと枝メトリックを加算し，パスメトリックを求める．STATE$_{i,m}$ は状態メトリックと状態番号を格納するレジスタを持つ．最初，このレジスタには状態メトリックの最大値と状態番号の初期値がセットされる．順次入力される状態メトリックと枝メトリックの加算結果をレジスタに格納されているメトリックと比較し，小さい方のメトリックとそれを与えるパスが通過する直前の状態の番号をレジスタに格納する．これを順次繰り返し，最小となるパスメトリックを求め状態 $s_{i,m}$ の状態メトリックを求める．また，生き残りパスを与える状態と $s_{i,m}$ 間の並列枝集合のもつ並列ラベル集合を表す番号もあわせて保持する．そして，出力が接続されているサブモジュール，例えば，STATE$_{i+1,m'}$ から要求があると，STATE$_{i,m}$ は保持している状態メトリック $SM(s_{i,m})$ を出力する．これらの機能のため，STATE$_{i,m}$ は状態番号，並列ラベル集合の番号，及び，状態メトリックを保持するレジスタ，加算器，比較器から構成される．

5.4　各手法の評価

5.4.1　分割トレリス復号法の平均計算量の評価

ここでは，(64,45) 拡大 BCH 符号 BCH(64,45)，符号長 64，3 次のリード・マラー符号 RM(64,42)，RM(64,42) の (64,40) 部分符号 RM'(64,40) を用い，分割トレリス復号法の平均計算量を計算機シミュレーションにより評価する．なお，BCH(64,45) は，(64,45) 拡大原始 BCH 符号からビット位置の置換をし，より単純な構造のトレリスダイアグラムが得られる

表 5.1: BCH(64,45) の 4 セクショントレリスダイアグラムの複雑度

i	1	2	3	4
$K_{16(i-1)}$	0	10	13	10
K_{16i}	10	13	10	0
$K_{16(i-1),16i}$	5			
$q_{16(i-1),16i}$	0	4	4	0
$Q_{16(i-1),16i}$	0	6	9	10
$Q'_{16(i-1),16i}$	10	9	6	0
$\lambda_{16(i-1),16i}$	0	3	3	0
ν	1	64	64	1

(1) 第 i セクションの終りの状態数（$16i$ ビット目の状態数）は $2^{K_{16i}}$.

(2) $16i$ ビット目の各状態 s について，s への枝がある $16(i-1)$ ビット目の状態数は $2^{Q_{16(i-1),16i}}$．また，このときの一状態対間の並列枝の数は $2^{K_{16(i-1),16i}}$.

(3) 第 i セクションの並列成分の数は $2^{q_{16(i-1),16i}}$.

(4) 第 i セクションの一並列成分における初期状態側の状態（左側の状態）の数は $2^{Q_{16(i-1),16i}}$.

(5) 第 i セクションの一並列成分における終了状態側の状態（右側の状態）の数は $2^{Q'_{16(i-1),16i}}$.

(6) 第 i セクションの $2^{q_{16(i-1),16i}}$ 個の並列成分は，$2^{q_{16(i-1),16i}-\lambda_{16(i-1),16i}}$ の大きさ $2^{\lambda_{16(i-1),16i}}$ のブロックに分けられる．このとき，同じブロックの並列成分は，その枝ラベルまで同一となる．また，枝ラベルまで同一の二つの並列成分は同じブロックに分けられる．

(7) 第 i セクションの各並列成分は文献 [77] の Theorem 3 で述べられている構造を持つ.

表 5.2: BCH(64,45) の 8 セクショントレリスダイアグラムの複雑度

i	1	2	3	4	5	6	7	8
$K_{8(i-1)}$	0	7	10	13	13	13	10	7
K_{8i}	7	10	13	13	13	10	7	0
$K_{8(i-1),8i}$	1							
$q_{8(i-1),8i}$	0	4	7	7	7	7	4	0
$Q_{8(i-1),8i}$	0	3	3	6	6	6	6	7
$Q'_{8(i-1),8i}$	7	6	6	6	6	3	3	0
$\lambda_{8(i-1),8i}$	0	3	6	6	6	6	3	0
ν	1	8	8	64	64	8	8	1

表 5.3: RM(64,42) の 4 セクショントレリスダイアグラムの複雑度

i	1	2	3	4
$K_{16(i-1)}$	0	10	10	10
K_{16i}	10	10	10	0
$K_{16(i-1),16i}$	5			
$q_{16(i-1),16i}$	0	4	4	0
$Q_{16(i-1),16i}$	0	6	6	10
$Q'_{16(i-1),16i}$	10	10	6	0
$\lambda_{16(i-1),16i}$	0	3	3	0
ν	1	64	64	1

表 5.4: RM(64,42) の 8 セクショントレリスダイアグラムの複雑度

i	1	2	3	4	5	6	7	8
$K_{8(i-1)}$	0	7	10	13	10	13	10	7
K_{8i}	7	10	13	10	13	10	7	0
$K_{8(i-1),8i}$	1							
$q_{8(i-1),8i}$	0	4	7	7	7	7	4	0
$Q_{8(i-1),8i}$	0	3	3	6	3	6	6	7
$Q'_{8(i-1),8i}$	7	6	6	3	6	3	3	0
$\lambda_{8(i-1),8i}$	0	3	6	6	6	6	3	0
ν	1	8	8	8	8	8	8	1

ようにしたものである [75]．RM'(64,40) は文献 [79] の連接符号の内部符号である．各符号について，$L = 4$ 及び $L = 8$ の L セクショントレリスダイアグラムの複雑度を表 5.1，5.2，5.3，5.4，5.5，5.6 に示す．

シミュレーションにおいては，上記の各符号で 4 セクショントレリスダイアグラムと 8 セクショントレリスダイアグラムを用いた場合で復号を行い，5.2 節で定義した ma_i と mc_i について，各々の平均値を評価した．なお，このシミュレーションは加法的白色ガウス雑音通信路において BPSK 変調とともに上記符号を用いたことを想定して行った．また，シミュレーションは通信路の雑音の状況が $E_b/N_o = 1$ (dB) と $E_b/N_o = 4$ (dB) の二通りの場合について行った．このとき，BCH(64,45) のブロック誤り率は $E_b/N_o = 4$ (dB) で 3.9×10^{-1}，$E_b/N_o = 4$ (dB) で 1.4×10^{-3} であった．表 5.7，5.8 に，ma_i の平均値 a_i と mc_i の平均値 c_i をその上界である $\hat{a}_i$ と $\hat{c}_i$ とともに示す．上界は式 (5.15) と式 (5.16) により得られる．表からは，a_i は c_i は $\hat{a}_i$ と $\hat{c}_i$ よりも非常に小さく，特に $L = 4$ の場合に小さいことが分かる．ν の値は 4 セクショントレリスダイアグラムの場合は全ての符号について同じであった．a_i と

表 5.5: RM'(64,40) の 4 セクショントレリスダイアグラムの複雑度

i	1	2	3	4
$K_{16(i-1)}$	0	8	8	8
K_{16i}	8	8	8	0
$K_{16(i-1),16i}$	5			
$q_{16(i-1),16i}$	0	2	2	0
$Q_{16(i-1),16i}$	0	6	6	8
$Q'_{16(i-1),16i}$	8	6	6	0
$\lambda_{16(i-1),16i}$	0			
ν	1	64	64	1

表 5.6: RM'(64,40) の 8 セクショントレリスダイアグラムの複雑度

i	1	2	3	4	5	6	7	8
$K_{8(i-1)}$	0	6	8	11	8	11	8	6
K_{8i}	6	8	11	8	11	8	6	0
$K_{8(i-1),8i}$	1							
$q_{8(i-1),8i}$	0	3	5	5	5	5	3	0
$Q_{8(i-1),8i}$	0	3	3	6	3	6	5	6
$Q'_{8(i-1),8i}$	6	5	6	3	6	3	3	0
$\lambda_{8(i-1),8i}$	0	2	5	5	5	5	2	0
ν	1	8	8	8	8	8	8	1

c_i の値は概ね ν に依存していることが分かる．逆に，枝ラベルまたは SN 比による影響はほとんどないことが分かる．

次に，一度の復号全体にかかる計算量を評価する．この評価は，(M1) 枝メトリックの計算にかかるメトリックの加算回数の総和 AD_{BM}，(M2) 並列ラベル集合の中から枝メトリックが最小の枝ラベルを探すためのメトリックの比較回数の総和 CP_{PB}，(M3) 前状態への生き残りパスのパスメトリックと枝メトリックの加算回数の総和 AD，(M4) 各状態への生き残りパスを見つけるためのメトリックの比較回数の総和 CP により行う．なお，AD と CP について，分割トレリス復号法の場合を，AD^{NEW} と CP^{NEW} とし，ヴィタビ復号法の場合を，AD^{EX} と CP^{EX} とする．CP^{NEW} については，π と π' をみつけるためのソートにおけるメトリックの比較回数も含んでいる．これらのソートにはヒープソートアルゴリズムを用いた．

L セクショントレリスダイアグラムの第 i セクションにおいて，異なる枝ラベルを持つ枝の数を 2^{b_i} とする．このとき，枝メトリックの計算を i セクションにおいて，2^{b_i} の加算器を

表 5.7: メトリックの加算回数の平均値 a_i と比較回数の平均値 c_i，及びそれらの上界値 $\hat{a}_i$ と $\hat{c}_i$.（4セクショントレリスダイアグラムの場合．$L = 4$）

(a) BCH(64,45)

	i	2	3
	ν	64	64
E_b/N_o	a_i	77	76
$= 1$(dB)	c_i	262	247
E_b/N_o	a_i	77	76
$= 4$(dB)	c_i	260	244
Upper	$\hat{a}_i$	2080	2080
Bound	$\hat{c}_i$	2016	2016

(b) RM(64,42)

	i	2	3
	ν	64	64
E_b/N_o	a_i	77	76
$= 1$(dB)	c_i	261	245
E_b/N_o	a_i	77	76
$= 4$(dB)	c_i	245	224
Upper	$\hat{a}_i$	2080	2080
Bound	$\hat{c}_i$	2016	2016

(c) RM'(64,40)

	i	2	3
	ν	64	64
E_b/N_o	a_i	77	76
$= 1$(dB)	c_i	254	241
E_b/N_o	a_i	76	76
$= 4$(dB)	c_i	232	218
Upper	$\hat{a}_i$	2080	2080
Bound	$\hat{c}_i$	2016	2016

用いて行ったとすると，加算回数はこのセクションにおいて合計 $2^{b_i}(64/L - 1)$ となる．また，第 i セクションにおいて各並列ラベル集合の要素数が 2^{p_i} のとき，ここから枝メトリックが最小の並列ラベルを選ぶのに必要な枝メトリックの比較回数は $(2^{p_i} - 1)$ となる．8セクショントレリスダイアグラムの場合の AD_{BM} と CP_{PB} の値はこのようにして計算した．

4セクショントレリスダイアグラムについては，ここで取り上げた符号の場合，各並列ラベル集合は符号長16の RM 符号の剰余類となる．これを利用すると，復号にかかる処理時間を増加させることなく，メトリックの加算回数と比較回数を削減することができる．この剰余類の2セクショントレリスダイアグラムでは $|S_8| = 8$ であり，そしてこの各状態について，初期状態から2本の並列枝があり，最終状態へ2本の並列枝がある．2^5 個の並列枝の枝ラベルから最小の枝メトリックを枝ラベルを見つける際，この2セクショントレリスダイアグラムを用いるとメトリックの加算回数は，BCH(64,45) と RM(64,42) の場合で約 3/125 に，RM'(64,40) の場合で約 1/32 に削減できる．また，ここで挙げた全ての符号について，メトリックの比較回数は 23/31 に削減できる．4セクショントレリスダイアグラムについては，AD_{BM} と CP_{PB} はこのようにして評価した．

4セクショントレリスダイアグラムと8セクショントレリスダイアグラムの場合の各符号についての AD_{BM}，CP_{PB}，AD^{EX}，CP^{EX}，AD^{NEW}，CP^{NEW} の値を表 5.9 に示す．表から，AD^{NEW} と CP^{NEW} は AD^{EX} と CP^{EX} より，非常に小さいことが分かる．L が8より大きい場合には，ν は各セクションにおいて非常に小さくなる．このような場合は，分割トレリス復号法はメトリックのソートにかかる計算が無視できなくなり，復号にかかるメトリックの加算や比較などの演算の回数を減らすことができなくなる．しかし，ヴィタビ復号法におけるメトリックの比較回数と加算回数はセクション数が4のとき最小である [79]．ここで挙げた符号については，セクション数を4とした分割トレリス復号法の計算量が最も小さい

表 5.8: メトリックの加算回数の平均値 a_i と比較回数の平均値 c_i，及びそれらの上界値 $\hat{a}_i$ と $\hat{c}_i$.（8 セクショントレリスダイアグラムの場合．$L = 8$）

(a) BCH(64,45)

	i	2	3	4	5	6	7
	ν	8	8	64	64	8	8
E_b/N_o	a_i	11.7	11.5	76.6	76.2	11.0	10.5
$= 1$(dB)	c_i	13.5	13.0	268.6	263.2	11.6	10.9
E_b/N_o	a_i	11.6	11.5	76.5	76.1	11.0	10.8
$= 4$(dB)	c_i	13.2	12.9	264.5	260.5	11.7	11.3
Upper	$\hat{a}_i$	36	36	2080	2080	36	36
Bound	$\hat{c}_i$	28	28	2016	2016	28	28

(b) RM(64,42)

	i	2	3	4	5	6	7
	ν	8	8	8	8	8	8
E_b/N_o	a_i	11.7	11.5	11.2	11.2	10.9	10.9
$= 1$(dB)	c_i	13.5	13.0	12.1	12.1	11.4	11.4
E_b/N_o	a_i	11.6	11.5	11.2	11.1	10.9	10.9
$= 4$(dB)	c_i	13.2	12.9	12.1	12.1	11.4	11.5
Upper	$\hat{a}_i$	36	36	36	36	36	36
Bound	$\hat{c}_i$	28	28	28	28	28	28

(c) RM'(64,40)

	i	2	3	4	5	6	7
	ν	8	8	8	8	8	8
E_b/N_o	a_i	11.7	11.5	11.2	11.2	10.9	11.0
$= 1$(dB)	c_i	13.7	13.0	12.2	12.2	11.5	11.6
E_b/N_o	a_i	11.7	11.5	11.2	11.2	11.0	11.2
$= 4$(dB)	c_i	13.5	12.9	12.2	12.3	11.6	12.0
Upper	$\hat{a}_i$	36	36	36	36	36	36
Bound	$\hat{c}_i$	28	28	28	28	28	28

表 5.9: L セクショントレリスダイアグラムを用いた場合のヴィタビ復号法と分割トレリス復号法の平均計算量

符号	BCH(64,45)		RM(64,42)		RM'(64,40)	
L	4	8	4	8	4	8
AD_{BM}	47104	14336	47104	14336	15360	7168
CP_{PB}	94208	1024	94208	1024	23552	512
AD^{EX}	1049600	1196160	132096	278656	33024	69696
AD^{NEW} (1dB)	20669	45206	3480	48821	868	12270
AD^{NEW} (4dB)	20658	45557	3476	48718	865	12286
CP^{EX}	1040383	1169407	130047	259071	32511	64767
CP^{NEW} (1dB)	173937	287268	36563	134099	9093	34322
CP^{NEW} (4dB)	173262	287424	35953	133844	8913	34324

といえる．BCH(64,45) で 4 セクショントレリスダイアグラムを用いた場合では，分割トレリス復号法のメトリックの加算回数はヴィタビ復号法の約 1/50，メトリックの比較回数は約 17/100 となっている．これは，復号の際の CPU 時間を減らすために非常に重要である．

5.4.2　部分探索復号法の平均計算量と復号誤り率の評価

5.4.1 の実験により，分割トレリス復号法の平均計算量はその上界に比べ非常に小さいことが示された．このことにより，部分探索復号法が復号誤り率をほとんど下げることなしに，より高速な復号を行えることが期待できる．ここでは，5.4.1 で挙げた符号のうち BCH(64,45) を対象として，4 セクショントレリスダイアグラムを使った場合の，部分探索復号法の復号誤り率と平均計算量を計算機シミュレーションにより評価する．なお，通信路と変調については 5.4.1 と同様に，加法的白色ガウス雑音通信路と，BPSK 変調を想定した．なお，BCH(64,45) の 4 セクショントレリスダイアグラムの複雑度は先の表 5.1 の通りである．シミュレーションは $E_b/N_o = 1, 1.5, 2, 3, 3.5, 4$ (dB) の各場合について行った．このときの軟判定最尤復号によるブロック誤り率は，順に 3.9×10^{-1}，2.5×10^{-1}，1.3×10^{-1}，6.1×10^{-2}，2.2×10^{-2}，6.4×10^{-3}，1.6×10^{-3} であった．

表 5.10 にシミュレーションの結果を示す．この表において，AD_{BM}，CP_{PB}，AD，CP は 5.4.1 と同様であり，枝メトリックの計算にかかるメトリックの加算回数の総和 AD_{BM}，並列ラベル集合の中から枝メトリックが最小の枝ラベルを探すためのメトリックの比較回数の総和 CP_{PB}，前状態への生き残りパスのパスメトリックと枝メトリックの加算回数の総和 AD，各状態への生き残りパスを見つけるためのメトリックの比較回数の総和 CP である．

表 5.10: 部分探索復号法の平均計算量とブロック誤り率の評価（BCH(64,45)．4 セクショントレリスダイアグラムを用いた場合）

AD_{BM}	47104						
CP_{PB}	94208						
μ_1, μ_2	64	32	16	8	4	2	1
AD^{SUB} (1dB)	20669	20228	19583	19059	18629	18364	18199
AD^{SUB} (1.5dB)	20672	20235	19588	19062	18632	18366	18200
AD^{SUB} (2dB)	20674	20234	19586	19062	18634	18368	18201
AD^{SUB} (2.5dB)	20674	20231	19583	19062	18636	18369	18202
AD^{SUB} (3dB)	20672	20228	19579	19062	18638	18371	18203
AD^{SUB} (3.5dB)	20667	20222	19574	19061	18642	18374	18205
AD^{SUB} (4dB)	20658	20216	19569	19060	18646	18378	18207
CP^{SUB} (1dB)	173937	141609	113273	93289	81092	74077	68033
CP^{SUB} (1.5dB)	173911	141699	113359	93313	81151	74168	68106
CP^{SUB} (2dB)	173873	141652	113265	93236	81156	74212	68150
CP^{SUB} (2.5dB)	173800	141574	113121	93110	81146	74249	68193
CP^{SUB} (3dB)	173688	141461	112931	92928	81122	74308	68239
CP^{SUB} (3.5dB)	173510	141293	112671	92674	81090	74375	68297
CP^{SUB} (4dB)	173262	141068	112357	92353	81037	74444	68357
P_{er} (1dB)	3.9×10^{-1}	3.9×10^{-1}	3.9×10^{-1}	3.9×10^{-1}	3.9×10^{-1}	4.0×10^{-1}	4.1×10^{-1}
P_{er} (1.5dB)	2.5×10^{-1}	2.5×10^{-1}	2.5×10^{-1}	2.5×10^{-1}	2.5×10^{-1}	2.6×10^{-1}	2.6×10^{-1}
P_{er} (2dB)	1.3×10^{-1}	1.3×10^{-1}	1.3×10^{-1}	1.4×10^{-1}	1.4×10^{-1}	1.4×10^{-1}	1.4×10^{-1}
P_{er} (2.5dB)	6.1×10^{-2}	6.1×10^{-2}	6.1×10^{-2}	6.2×10^{-2}	6.4×10^{-2}	6.6×10^{-2}	6.9×10^{-2}
P_{er} (3dB)	2.2×10^{-2}	2.2×10^{-2}	2.2×10^{-2}	2.3×10^{-2}	2.4×10^{-2}	2.5×10^{-2}	2.7×10^{-2}
P_{er} (3.5dB)	6.4×10^{-3}	6.4×10^{-3}	6.4×10^{-3}	6.6×10^{-3}	6.8×10^{-3}	7.5×10^{-3}	8.0×10^{-3}
P_{er} (4dB)	1.6×10^{-3}	1.6×10^{-3}	1.6×10^{-3}	1.6×10^{-3}	1.8×10^{-3}	2.0×10^{-3}	2.6×10^{-3}

なお，AD と CP については，部分探索復号法の場合を示すため，AD^{SUB} と CP^{SUB} とした．CP はソートにおけるメトリックの比較回数を含んでいる．また，P_{er} はブロック誤り率である．この 4 セクショントレリスダイアグラムにおいては ν は第 1 セクションと第 4 セクションで 1，第 2 セクションと第 3 セクションで 64 である．そのため，第 1 セクションと第 4 セクションで μ_1 と μ_2 は常に 1 とした．そして，第 2 セクション及び第 3 セクションの μ_1 と μ_2 を表にあるように変化させ，シミュレーションを行った．μ_1 と μ_2 が 64 のときは，部分探索復号法は分割トレリス復号法と同じである．

表からは，$\mu_1 = \mu_2 = 8$ のとき，ブロック誤り率は最尤復号とほぼ同じであり，計算量は分割トレリス復号法の約 75%にまで減少していることが分かる．この計算量はヴィタビ復号法の約 11%である．

5.4.3　再構成可能ハードウェアを用いたヴィタビ復号器の速度評価

ここでは，5.3.2で述べたヴィタビ復号器を再構成可能ハードウェアに実現した場合の復号速度を，ソフトウェアで実現した場合と比較評価する．5.3.2で述べたように，個々の符号の復号器はその符号に特化したものであり，のハードウェア記述は符号に特化したものであり, 符号のトレリスダイアグラムの構造に基づいた並列処理を行う．そのため，目的とする符号の復号器のハードウェア記述を生成行列から自動的に作成するプログラムを作成し，復号器を設計した．

ハードウェア記述言語にはSFL(Structured Function description Language)[80]を用いた．SFLは，ハードウェアを手続き記述のみで記述し，ハードウェアコンポーネント間の接続記述を用いないオブジェクト指向の言語である．SFL記述におけるハードウェアコンポーネントはオブジェクトとして表現され，コンポーネント間の接続は単純な動作記述として表される．そのため，(2)で述べたサブモジュールもオブジェクトとして表現し，符号のトレリス構造に依存したサブモジュール間の複雑な接続構造もオブジェクトの単純な動作として表現することが可能となる．そして，SFL記述の論理合成には高位論理合成システムPARTHENON[80]を用いた．また，復号器を実装する再構成可能ハードウェアとしてはAltera社のFPGAであるFLEX 10Kを想定し，FPGAのマッピングにはAltera社のFPGAマッピングツール，MAX+plus IIを用いた．

なお，本実験で用いる符号は，想定した再構成可能ハードウェアの規模などから，符号長32，1次のRM符号RM(32,6)と符号長32，3次のRM符号RM(32,26)とした．また，トレリスダイアグラムのセクション数は4とした．

なお，ソフトウェアデコーダはC言語で記述し，ヴィタビ復号法の各ステップを全て逐次的に実行する．実験ではSun Ultra2 (Ultra Sparc 200MHz)で実行し復号遅延を評価した．

以上の条件のもと1つの受信系列を復号するのにかかる時間（復号時間）を, 再構成可能ハードウェア上に実装される復号器とソフトウェアデコーダの場合とで評価し比較した．実験結果を表5.11に示す．再構成可能ハードウェアに実装される復号器の動作クロック周波数，クロックサイクル数，各々，復号器の動作クロック周波数，1受信系列の復号に必要なクロックサイクル数である．表5.11より, RM(32,6)の場合, 再構成可能ハードウェア上の復号器の復号時間はソフトウェアデコーダの約1/23となっていることが分かる．また, RM(32,26)の場合は約1/7となっている．以上のことから，ソフトウェアデコーダに代わって再構成可能ハードウェア上の復号器を実装することで，ヴィタビ復号法の並列計算可能な部分を利用し，より高速な復号処理を行えることが分かる．

表 5.11: 再構成可能ハードウェアによるヴィタビ復号法とソフトウェアによるヴィタビ復号法の復号速度評価

符号		RM(32,6)	RM(32,26)
再構成可能ハードウェア	動作クロック周波数 (MHz)	11.5	6.08
	クロックサイクル数	38	185
	復号時間 (μs)	3.31	30.4
ソフトウェア	復号時間 (μs)	76.2	211.5

5.5 むすび

本章では，メディア処理高信頼化のための誤り訂正符号復号法について，特に線形ブロック符号の軟判定最尤復号を対象として述べた．はじめに，分割トレリス復号法を計算機シミュレーションにより評価し，その平均計算量が上界よりも非常に小さいことを示した．次に，この分析結果に基づき，より平均計算量が小さい準最尤復号復号アルゴリズム 部分探索復号法を提案した．BCH(64,45) を用いた評価実験では部分探索復号法の復号誤り率は最尤復号と同等で，その平均計算量は分割トレリス復号法の約 75%で，ヴィタビ復号法の約 11%であった．また，ヴィタビ復号法を再構成可能ハードウェアで実現する手法についても述べた．再構成可能ハードウェアに各符号のトレリスダイアグラムの構造に応じた復号器を構成することで，各トレリスダイアグラムの構造に応じた並列処理を行い，高速な復号処理を行うことが可能になる．

分割トレリス復号法，部分探索復号法の復号処理の高速化は，主に，各並列ラベル集合について最小の枝メトリックを持つ枝を見つけた後の生き残りパスの探索処理部分を対象としている．この高速化の対象とした部分は，ヴィタビ復号法において復号処理の大部分を占めており，その高速化の意義は大きい．しかし，この部分が高速化された結果，並列ラベル集合について最小の枝メトリックを持つ枝を見つけ出す処理も復号処理において大きな割合を占めるようになって来ている（表 5.9，5.10 参照）．5.4.1 と 5.4.2 の実験においては，この部分の処理についてもトレリスダイアグラムを用いることを述べた．今後，このような処理方法を用いることにより，並列ラベル集合からの最小の枝メトリックの探索をより高速に行うことなども検討する必要があると考えられる [34, 81].

第6章　結論

6.1　本論文のまとめ

本論文では，パターンマッチングを用いた音楽音響信号探索法及び画像圧縮法について論じ，特に，変動に頑健かつ高速な音楽音響信号の探索手法と，画像圧縮における画像間の距離計算を効率的に並列処理する手法を提案した．

まず第1章において本研究の目的となる音楽音響信号の探索技術の重要性について述べ，さらにその技術的な状況と課題について述べた．そして，メディア探索技術や画像圧縮などのメディア処理の基本技術の一つにパターンマッチングがあることを述べ，画像圧縮において行われるパターンマッチングを高速化することも本論文の目的とすることを述べた．

第2章では分割一致探索法の枠組を提案し，多数の小領域スペクトログラムを用いた背景音楽の高速探索法を提案した．分割一致探索法では，参照信号のスペクトログラムを時間周波数領域において多数の小領域に分割し，これらの小領域スペクトログラムと類似度の高い小領域スペクトログラムをもつ時点を蓄積信号のスペクトログラムから探索し，その探索結果を統合することで，目的区間を探索する．分割一致探索法では参照信号を小領域スペクトログラムに分割し，一致する小領域スペクトログラムだけに着目して背景音楽探索を行うことで，背景音楽探索を高精度に行うことと，この一致する小領域スペクトログラムだけに着目することにより，一致する小領域スペクトログラムの探索への高速な同一信号探索の適用を可能にし，背景音楽探索全体を高速に行う．実験では，この同一信号探索にTASを適用することで，約30分の音響信号から15秒の音楽を1秒未満で探索可能であった．また，同実験における探索精度は，SN比が -15dBと，重畳している音声よりも音楽の方が非常に小さい場合でも約90%であり，提案手法が高精度に背景音楽を探索できることを確認した．

第3章では，多重奏音響信号を参照信号として多重奏音響信号から類似する音楽を探索する類似音楽探索法を提案した．特に本手法では，同一の音楽音響信号以外に類似する音楽，例えば，再演奏されたもの，他の楽器を用いて演奏されたもの，テンポの異なる演奏，もしくは移調されたものなど，同一の音響信号ではないが同じ曲である類似音楽も探索することを目的としている．そこで，多重奏音響の類似性に基づいた探索のための2値多重音響特徴ベクトルを提案し，2値多重音響特徴ベクトルを用いた曖昧系列照合法による類似音楽探索

法とその高速化手法を提案した．216 曲の実験用音楽データベースを用いた探索実験では，例えば，楽曲単位の探索において，クエリーの長さが平均 19 秒のとき，スペクトル特徴を用いた場合に 62.5%であった精度が，提案の 2 値多重音響特徴ベクトルを用いることにより 89.3%にまで向上した．また，本高速化手法においては，精度をほとんど損なわずに，DP 照合を用いたずらし照合法に比べ約 1.6 倍から 18 倍の高速化を達成した．

第 4 章では，パターンマッチングを用いた画像圧縮において必要な画像間の距離計算を再構成可能ハードウェアを用いて行う手法を提案した．本手法で提案する回路では，画像の距離計算に必要な画素値の積和演算をパイプライン処理により高速に行うことを想定している．このとき，画像に応じて回路を再構成することで，計算の並列度を保ったまま，主要回路のである加算器の数を，最悪でも，単純に回路を構成した場合の半分に削減できる．本手法により，必要な距離計算を限られたハードウェ資源で効率よく並列に行うことが可能と考えられる．また，提案手法で構成する回路を複数実現し，これらを特定の画像ブロックに応じて非同期に再構成した場合の圧縮の高速化についても述べた．ソフトウェアによる手法と同様に，許容範囲内の近似画像ブロックをみつけるるごとに画像のマッチング処理を打ち切ることで，画像圧縮処理を高速化することが期待できる．

第 5 章では，メディア処理高信頼化のための誤り訂正符号復号法について，特に線形ブロック符号の軟判定最尤復号を対象として述べた．はじめに，分割トレリス復号法の平均計算量を計算機シミュレーションにより評価した．シミュレーションの結果からは，分割トレリス復号法の平均計算量は上界よりも非常に小さく，また，例えば (64,45)BCH 符号で 4 セクショントレリスダイアグラムを用いた場合，分割トレリス復号法の平均計算量は，ヴィタビ復号法の約 15%と非常に小さいことを示した．次に，この分析結果に基づき，より平均計算量が小さい準最尤復号復号アルゴリズム 部分探索復号法を提案した．(64,45)BCH 符号を用いた評価実験では部分探索復号法の復号誤り率は最尤復号と同等で，その平均計算量は分割トレリス復号法の約 75%で，ヴィタビ復号法の約 11%であった．また，ヴィタビ復号法を再構成可能ハードウェアで実現する手法についても述べた．再構成可能ハードウェアに各符号のトレリスダイアグラムの構造に応じた復号器を構成することで，各トレリスダイアグラムの構造に応じた並列処理を行い，高速な復号処理を行うことが可能になる．

6.2　今後の研究課題

本論文においては，パターンマッチングを基本とし，音楽音響信号の探索法，画像圧縮で行われる部分画像のマッチングの高速化，メディア処理高信頼化のための誤り訂正復号法について検討した．そして，特に音楽音響信号の探索においては，さらに以下の研究課題が挙

げられる．

背景音楽探索のさらなる高速化

背景音楽探索については第 2 章の提案手法により，高速かつ高精度な背景音楽探索が可能になった．しかし，応用面からは，探索精度は保ったまま，探索速度はより速くなることが望まれる．そのためには，探索の精度を保つため，探索の枠組は分割一致探索法のままとし，一致小領域スペクトログラムの組合せをより高速に探索することが望ましい．

類似音楽探索のさらなる高精度化

第 3 章の類似音楽探索法により多重奏対多重奏の音楽音響信号の照合に基づく類似音楽探索が可能になった．しかし，提案した手法においても楽器を変換した場合の探索精度は十分でない [23]．楽器変動に頑健な 2 値多重音響特徴ベクトルの抽出法を検討する必要がある．また，和音の構成音自体を変化させた場合への対応も検討する必要がある．

汎用的な音楽の認識・探索手法

第 2 章の背景音楽探索法と，第 3 章の類似音楽探索法では，第 2 章の分割一致探索法により重畳妨害音という変動に対応し，第 3 章の 2 値多重音響特徴ベクトルと曖昧系列照合法により，類似音楽における変動に対応している．今後は，多様な変動を統一的に処理する枠組を検討する必要がある．

また，第 4 章のフラクタル画像圧縮のための画像間距離の高速計算回路の実現法については以下の課題が上げられる．

再構成可能なハードウェアによる実装とその評価

第 4 章で提案した計算回路を実装し，動作速度などを評価する必要がある．なお，提案手法において回路を複数実現し並列動作させるためには，再構成可能ハードウェアが部分的に再構成できること，再構成可能ハードウェアに実現できる回路規模ができるだけ大きいことが望ましい．そのためには，文献 [82] で提案されているようなハードウェアを使うことが考えられる．

最後に，第 5 章のメディア処理高信頼化のための誤り訂正符号復号法については以下の課題が上げられる．

復号処理の更なる高速化

分割トレリス復号法，部分探索復号法の復号処理の高速化は，主に，各並列ラベル集

合について最小の枝メトリックをもつ枝を見つけた後の生き残りパスの探索処理部分を対象としている．この部分が高速化された結果，並列ラベル集合について最小の枝メトリックをもつ枝を見つけ出す処理が復号処理において大きな割合を占めるようになって来ている．並列ラベル集合からの最小の枝メトリックの探索をより高速に行うことなども検討する必要があると考えられる．

謝辞

本研究の遂行および，本論文をまとめるにあたり，懇切なる御指導，御助言と格別なるご配慮を賜わりました大阪大学大学院情報科学研究科マルチメディア工学専攻 藤原融教授に謹んで深謝の意を表します．藤原先生には，筆者が大阪大学基礎工学部情報工学科に在籍した時から長年に渡って御指導を頂き，誠に感謝しております．

本論文をまとめるにあたり，貴重な時間を割いて頂き，懇切なる御指導と有益な御助言を賜わりました大阪大学大学院情報科学研究科マルチメディア工学専攻 西尾章治郎教授，並びに岸野文郎教授に心より感謝致します．

講義や公聴会などを通じ，懇切なる御指導を頂き，また，本研究に関する貴重な御意見を頂いた大阪大学大学院情報科学研究科マルチメディア工学専攻 薦田憲久教授，下條真司教授，小暮潔教授，片桐恭弘教授，並びに情報システム工学専攻 尾上孝雄教授に厚く感謝申し上げます．

本研究において，貴重な御意見と御指導を頂きました大阪大学大学院情報科学研究科マルチメディア工学専攻 石原靖哲助教授に心より感謝申し上げます．

本研究において，貴重な御意見を頂きました大阪大学大学院情報科学研究科マルチメディア工学専攻 北村喜文助教授に厚く感謝申し上げます．

本研究を進めるにあたって，多大なる御助言，御協力，御支援を頂いた大阪大学情報科学研究科 吉田真紀助手に感謝致します．

また，本論文をまとめるにあたっては，大阪大学大学院情報科学研究科 岡村真吾氏をはじめセキュリティ工学講座の皆様に多大なる御協力を頂きました．ここに厚く御礼申し上げます．

本研究は大阪大学大学院情報科学研究科マルチメディア工学専攻および日本電信電話株式会社 NTT コミュニケーション科学基礎研究所の多くの方々の御指導のもとに行われました．NTT コミュニケーション科学基礎研究所にて研究の機会を与えて頂くとともに，日頃から御指導頂く管村昇所長，片桐滋主席研究員，牧野昭二部長，大和淳司グループリーダに感謝致します．また，共同研究者として研究指導を頂いた名古屋彰主幹研究員，須山敬之主任研究員，松浦昭洋氏（現 東京電機大学講師）に感謝致します．また，共同研究者として多大なる御指導を頂きました村瀬洋氏（現 名古屋大学教授）に深く感謝申し上げます．また，共同研究

者として日頃御指導頂く柏野邦夫主任研究員に心より感謝致します．柏野氏の御指導なしに本研究が生まれることはありませんでした．また，共同研究者として御協力頂く黒住隆行氏に感謝致します．そして，常日頃，御指導と御協力を頂く同研究所メディア認識研究グループの皆様に感謝致します．

大阪大学 嵩忠雄名誉教授には，筆者が研究生活を始めるにあたり御指導頂き，その後も多くのことを御教授頂いております．また，おりにふれ，学位を取得するよう叱咤激励下さいました．ここに深く感謝致します．

最後に，筆者が研究生活を送る上で，暖かい支援と多大な理解を頂く両親と妻，そして，励みになってくれる息子に感謝します．

参考文献

[1] 柏野邦夫, ガビンスミス, 村瀬洋, “ヒストグラム特徴を用いた音響信号の高速探索法 —時系列アクティブ探索法—,” 電子情報通信学会論文誌 D-II, vol.J82-D-II, no.9, pp.1365–1373, Sept. 1999.

[2] K. Kashino, T. Kurozumi, and H. Murase, “A Quick Search Method for Audio and Video Signals Based on Histogram Pruning,” *IEEE Transactions on Multimedia*, vol.5, no.3, pp.348–357, Sept. 2003.

[3] 杉山雅英, “セグメントの高速探索法,” 電子情報通信学会技術研究報告, SP98-141, pp.39–45, Feb. 1999.

[4] 黒住隆行, 柏野邦夫, 村瀬洋, “時系列アクティブ探索法のための特徴ひずみに頑健な確率ディザボーティング,” 電子情報通信学会論文誌 D-II, vol.J84-D-II, no.8, pp.1817–1825, Aug. 2001.

[5] 西村拓一, 水野道尚, 小木しのぶ, 関本信博, 岡隆一, “アクティブ探索法による時系列データ中の一致区間検出 —参照区間自由時系列アクティブ探索法—,” 電子情報通信学会論文誌 D-II, vol.J84-D-II, no.8, pp.1826–1837, Aug. 2001.

[6] 蔭山哲也, 高島洋典, “ハミング歌唱を手掛かりとするメロディ検索,” 電子情報通信学会論文誌 D-II, vol.J77-D-II, no.8, pp.1543–1551, Aug. 1994.

[7] A. Ghias, J. Logan, D. Chamberlin, and B.C. Smith, “Query By Humming: Musical Information Retrieval in An Audio Database,” *Proceedings of ACM Multimedia ’95*, pp.231–236, Nov. 1995.

[8] L.A. Smith, R.J. McNab, and I.H. Witten, “Sequence-Based Melodic Comparison: A Dynamic Programming Approach,” *Melodic Similarity: Concepts, Procedures, and Applications*, W.B. Hewlett and E. Selfridge-Field eds., *Computing in Musicology*, vol.11, pp.101–117, MIT Press, 1998.

[9] 園田智也, 後藤真孝, 村岡洋一, “WWW 上での歌声による曲検索システム,” 電子情報通信学会論文誌 D-II, vol.J82-D-II, no.4, pp.721–731, April 1999.

[10] N. Kosugi, Y. Nishihara, S. Kon’ya, M. Yamamuro, and K. Kushima, “Music Retrieval by Humming —Using Similarity Retrieval over High Dimensional Feature Vector Space—,” *Proceedings of IEEE Pacific Rim Conference on Communications, Computers, and Signal Processing*, pp.404–407, Aug. 1999.

[11] 橋口博樹, 西村拓一, 張建新, 滝田順子, 岡隆一, “モデル依存傾斜制限型の連続 DP を用いた鼻歌入力による楽曲信号のスポッティング検索,” 電子情報通信学会論文誌 D-II, vol.J84-D-II, no.12, pp.2479–2488, Dec. 2001.

[12] A.L. Uitdenbogerd and J. Zobel, “Manipulation of Music for Melody Matching,” *Proceedings of ACM Multimedia ’98*, pp.235–240, Sept. 1998.

[13] T. Crawford, C.S. Iliopoulos, and R. Raman, “String-Matching Techniques for Musical Similarity and Melodic Recognition,” *Melodic Similarity: Concepts, Procedures, and Applications*, W. B. Hewlett and E. Selfridge-Field eds., *Computing in Musicology*, vol.11, pp.73–100. MIT Press, 1998.

[14] W.W. Peterson and E.J. Weldon, Jr., *Error-Correcting Codes*, second edition, MIT Press, 1972.

[15] 宮川洋, 岩垂好祐, 今井秀樹, 符号理論, コンピュータ基礎講座 18, 昭晃堂, 1973.

[16] 今井秀樹, 符号理論, 電子情報通信学会, 1990.

[17] 永野秀尚, 柏野邦夫, 村瀬洋, “多数の小領域スペクトログラムの探索に基づく背景音楽の高速探索法,” 電子情報通信学会論文誌 D-II, vol.J87-D-II, no.5, pp.1179–1188, May 2004.

[18] H. Nagano, K. Kashino, and H. Murase, “A Fast Search Algorithm for Background Music Signals Based on the Search for Numerous Small Signal Components,” *Proceedings of 2003 International Conference on Acoustics, Speech, and Signal Processing (ICASSP ’03)*, vol.V, pp.796–799, April 2003 (canceled due to SARS), also in *Proceedings of 2003 IEEE International Conference on Multimedia and Expo (ICME 2003)*, vol.I, pp.165–168, July 2003.

[19] 永野秀尚, 柏野邦夫, 村瀬洋, “多数の小区間信号の探索に基づく背景音楽の探索,” 電子情報通信学会技術研究報告, PRMU2002-86, WIT2002-29, pp. 37–41, Sept. 2002.

[20] 永野秀尚, 柏野邦夫, 村瀬洋, “2 値多重音響特徴ベクトルを用いた類似音楽探索とその高速化,” 電子情報通信学会論文誌 D-II, vol.J86-D-II, no.11, pp.1657–1667, Nov. 2003.

[21] H. Nagano, K. Kashino, and H. Murase, “Fast Music Retrieval Using Polyphonic Binary Feature Vectors,” *Proceedings of 2002 IEEE International Conference on Multimedia and Expo (ICME 2002)*, vol.I, pp.101–104, Aug. 2002.

[22] 永野秀尚, 柏野邦夫, 村瀬洋, “類似度に基づく曖昧文字列照合法と音楽検索への適用,” 情報処理学会研究報告, 2001-MUS-41, pp.67–72, Aug. 2001.

[23] 永野秀尚, 柏野邦夫, 藤原融, “2 値多重音響特徴ベクトルを用いた類似音楽探索法の頑健性評価,” 情報処理学会研究報告, 2004-MUS-58, pp.39–44, Dec. 2004.

[24] 永野秀尚, 柏野邦夫, 村瀬洋, “メディア探索のための曖昧文字列照合の高速計算法,” 電子情報通信学会 情報・システムソサイエティ大会講演論文集, p.197, Sept. 2001.

[25] H. Nagano, A. Matsuura, and A. Nagoya, “An Efficient Implementation Method of a Metric Computation Accelerator for Fractal Image Compression Using Reconfigurable Hardware,” *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol.E84-A, no.1, pp.372–377, Jan. 2001.

[26] A. Matsuura, H. Nagano, and A. Nagoya, “A Method for Implementing Fractal Image Compression on Reconfigurable Architecture,” *Proceedings of ACM International Symposium on Field-Programmable Gate Arrays (FPGA ’99)*, p.251, Feb. 1999.

[27] H. Nagano, A. Matsuura, and A. Nagoya, “An Efficient Implementation Method of Fractal Image Compression on Dynamically Reconfigurable Architecture,” *Proceedings of 6th Reconfigurable Architectures Workshop (RAW ’99) in IPPS/SPDP ’99 Workshops —Parallel and Distributed Processing*, LNCS 1586, pp.670–678, Springer-Verlag, April 1999.

[28] 松浦昭洋, 永野秀尚, 名古屋彰, “フラクタル画像圧縮の再構成可能アーキテクチャによる実現法,” 電子情報通信学会技術研究報告, CPSY98-83, pp.15–22, Sept. 1998.

[29] H. Nagano, T. Fujiwara, and T. Kasami, "Complexity Evaluation of an MLD Algorithm for a Linear Block Code Using Its Trellis Structure," *Proceedings of 17th Symposium on Information Theory and Its Applications (SITA '94)*, pp.209–212, Dec. 1994.

[30] T. Koumoto, H. Nagano, T. Takata, T. Kasami, T. Fujiwara, and S. Lin, "A New Iterative Soft-Decision Decoding Algorithm," 電子情報通信学会技術研究報告, IT-95-28, pp.19–24, July 1995.

[31] H. Nagano, T. Fujiwara, and T. Kasami, "Average Complexity Evaluation of an MLD Algorithm Using the Trellis Structure for a Linear Block Code," *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol.E78-A, no.9, pp.1209–1214, Sept. 1995.

[32] H. Nagano, T. Fujiwara, and T. Kasami, "A Decoding Algorithm for a Linear Block Code Using the Detailed Structure of its Trellis Diagram with Partial Search," *Proceedings of 18th Symposium on Information Theory and Its Applications (SITA '95)*, pp.449–452, Oct. 1995.

[33] T. Koumoto, H. Nagano, T. Takata, T. Kasami, T. Fujiwara, and S. Lin, "An Iterative Soft-Decision Decoding Algorithm," *Proceedings of 18th Symposium on Information Theory and Its Applications (SITA '95)*, pp.557–560, Oct. 1995.

[34] H. Yamamoto, H. Nagano, T. Fujiwara, and T. Kasami, "Recursive MLD Algorithm Using the Detail Trellis Structure for a Linear Block Code and Its Average Complexity Analysis," *Proceedings of 1996 International Symposium on Information Theory and Its Applications (ISITA '96)*, vol.II, pp.704–708, Sept. 1996.

[35] H. Nagano, T. Suyama, and A. Nagoya, "Soft Decision Maximum Likelihood Decoders for Binary Linear Block Codes Implemented on FPGAs," *Proceedings of 1998 ACM SIG/DA Sixth International Symposium on Field Programmable Gate Arrays (FPGA '98)*, p.261, Feb. 1998.

[36] H. Nagano, T. Suyama, and A. Nagoya, "Acceleration of Linear Block Code Evaluations Using New Reconfigurable Computing Approach," *Proceedings of Asia and South Pacific Design Automation Conference 1999 (ASP-DAC '99)*, pp.161–164, Jan. 1999.

[37] 永野秀尚, 須山敬之, 名古屋彰, “再構成可能なハードウェアを用いた線形ブロック符号の性能評価の高速化,” 電子情報通信学会技術研究報告, ICD98-290, VLD98-144, pp.25–32, March 1999.

[38] 木村昭悟, 柏野邦夫, 黒住隆行, 村瀬洋, “グローバルな枝刈りを導入した音や映像の高速探索,” 電子情報通信学会論文誌 D-II, vol.J85-D-II, no.10, pp.1552–1562, Oct. 2002.

[39] 成田智也, 杉山雅英, “楽曲の高速検索手法の検討,” 電子情報通信学会技術研究報告, SP2000-16, pp.1–8, June 2000.

[40] 新版 音響用語辞典, 日本音響学会 (編), コロナ社, 2003.

[41] 安部素嗣, 西口正之, “背景音楽同定のための自己最適化スペクトル相関法,” 電子情報通信学会技術研究報告, PRMU2001-209, pp.25–30, Jan. 2002.

[42] カーチス ローズ, コンピュータ音楽 —歴史・テクノロジー・アート—, 東京電機大学出版局, 2001.

[43] 片寄晴弘, “自動採譜 (概論),” bit 別冊 コンピュータと音楽の世界 —基礎からフロンティアまで—, 長嶋洋一, 橋本周司, 平賀譲, 平田圭二 (編), pp.74–88, 共立出版, 1998.

[44] 柏野邦夫, “重なり合った音を聞き分ける —音源分離,” bit 別冊 コンピュータと音楽の世界 —基礎からフロンティアまで—, 長嶋洋一, 橋本周司, 平賀譲, 平田圭二 (編), pp.89–99, 共立出版, 1998.

[45] M. Mongeau and D. Sankoff, “Comparison of Musical Sequences,” *Computers and the Humanities*, vol.24, no.3, pp.161–175, 1990.

[46] E. Selfridge-Field, “Conceptual and Representational Issues in Melodic Comparison,” *Melodic Similarity: Concepts, Procedures, and Applications*, W.B. Hewlett and E. Selfridge-Field eds., *Computing in Musicology*, vol.11, pp.3–64, MIT Press, 1998.

[47] 後藤真孝, 村岡洋一, “音響信号を対象としたリアルタイムビートトラッキングシステム —コード変化検出による打楽器音を含まない音楽への対応—,” 電子情報通信学会論文誌 D-II, vol.J81-D-II, no.2, pp.227–237, Feb. 1998.

[48] 柏野邦夫, 音楽音響信号を対象とする聴覚的情景分析に関する研究, 東京大学博士学位論文, Dec. 1994.

[49] Z.L. He, M.L. Liou, and K.W. Fu, "VLSI Architecture for Real-Time Fractal Video Coding," *1996 IEEE International Symposium on Circuits and Systems (ISCAS '96)*, vol.2, pp.738–741, May 1996.

[50] F. Ancarani, A. De Gloria, and M. Olivieri, "Design of an ASIC Architecture for High Speed Fractal Image Compression," *Proceedings of Ninth Annual IEEE International ASIC Conference and Exhibit*, pp.223–226, 1996.

[51] K.P. Acken, H.N. Kim, M.J. Irwin, and R.M. Owens, "An Architectural Design for Parallel Fractal Compression," *Proceedings of International Conference on Application-Specific System, Architectures and Processors*, pp.3–11, 1996.

[52] S. Lee and H. Aso, "A Parallel Architecture for High Speed Fractal Image Coding," *Proceedings of 1999 International Symposium on Parallel Architectures, Algorithms, and Networks (I-SPAN' 99)*, pp.88–93, June 1999.

[53] H. Yamauchi, Y. Takeuchi, and M. Imai, "VLSI Architecture for Real-Time Fractal Image Coding," *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol.E83-A, no.3, pp.452–458, March 2000.

[54] T. Miyazaki, "Reconfigurable Systems: A Survey," *Proceedings of Asia and South Pacific Design Automation Conference 1998 (ASP-DAC '98)*, pp.447–457, Feb. 1998.

[55] D.A. Buell, J.M. Arnold, and W.J. Kleinfelder eds., *Splash 2: FPGAs in a Custom Computing Machine*, IEEE Computer Society Press, 1996.

[56] *XC6200 Field Programmable Gate Arrays*, Xilinx Inc., April 1997.

[57] *AT40K FPGAs*, Atmel, Dec. 1997.

[58] J.R. Koza, F.H. Bennett III, J.L. Hutchings, S.L. Bade, M.A. Keane, and D. Andre, "Evolving Computer Programs using Rapidly Reconfigurable Field-Programmable Gate Arrays," *Proceedings of International Symposium on Field Programmable Gate Arrays (FPGA '98)*, pp.209–219, Feb. 1998.

[59] M.F. Barnsley, V. Ervin, D. Hardin, and J. Lancaster, "Solution of an Inverse Problem for Fractals and Other Sets," *Proceedings of the National Academy of Sciences, USA*, vol.83, pp.1975–1977, April 1986.

[60] A.E. Jacquin, *A Fractal Theory of Iterated Markov Operators with Applications to Digital Image Coding*, PhD thesis, Georgia Institute of Technology, 1989.

[61] Y. Fisher, Fractal image compression, *SIGGRAPH Course Notes*, 1992.

[62] Y. Fisher ed., *Fractal Image Compression: Theory and Application*, Springer, 1996.

[63] A.E. Jacquin, "Fractal Image Coding: A Review," *Proceedings of the IEEE*, vol.81, pp.1451–1465, Oct. 1993.

[64] M.F. Barnsley, *Fractals Everywhere*, AK Peters, 1993.

[65] *Data Book*, Altera Corporation, Jan. 1998.

[66] 今井秀樹 (監修), 誤り訂正符号化技術の要点, エレクトロニクスエッセンシャルズ, no.20, 日本工業技術センター, 1986.

[67] A.J. Viterbi, "Error Bounds for Convolutional Codes and an Asymptotically Optimum Decoding Algorithm," *IEEE Transanctions on Information Theory*, vol.IT-13, no.2, pp.260–269, April 1967.

[68] A.J. Viterbi and J.K. Omura, *Principles of Digital Communication and Coding*, McGraw-Hill, 1979.

[69] 中川聖一, 確率モデルによる音声認識, 電子情報通信学会, 1988.

[70] 古井貞熙, ディジタル音声処理, 東海大学出版会, 1985.

[71] 森健一 (編), パターン認識, 第 3 版, 電子情報通信学会, 1993.

[72] 今井秀樹, 情報理論, 昭晃堂, 1984.

[73] J. Wolf, "Efficient Maximum Likelihood Decoding of Linear Block Codes Using a Trellis," *IEEE Transactions on Information Theory*, vol.IT-24, no.1, pp.76–80, Jan. 1978.

[74] Jr. G.D. Forney, "Coset Codes—Part II: Binary Lattices and Related Codes," *IEEE Transactions on Information Theory*, vol.34, no.5, pp.1152–1187, Sept. 1988.

[75] T. Kasami, T. Takata, T. Fujiwara, and S. Lin, "On Complexity of Trellis Structure of Linear Block Codes," *IEEE Transactions on Information Theory*, vol.39, no.3, pp.1057–1064, May 1993.

[76] T. Kasami, T. Takata, T. Fujiwara, and S. Lin, "On Structural Complexity of the L-section Minimal Trellis Diagrams for Binary Linear Block Codes," *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol.E76-A, no.9, pp.1411–1421, Sept. 1993.

[77] T. Kasami, T. Fujiwara, Y. Desaki, and S. Lin, "On Branch Labels of Parallel Components of the L-section Minimal Trellis Diagrams for Binary Linear Block Codes," *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol.E77-A, no.6, pp.1058–1068, June 1994.

[78] S. Baase, *Computer Algorithms:Introduction to Design and Analysis*, Addison-Wesley, 1978.

[79] 浦谷嘉一, 藤原融, 嵩忠雄, "ある $(64, 40)$ 符号の最尤復号器の回路規模と最大復号遅延について," 第 17 回情報理論とその応用シンポジウム (SITA '94) 予稿集, pp.221–224, Dec. 1994.

[80] Y. Nakamura, Y. Oguri, A. Nagoya, M. Yukishita, and R. Nomura, "High-Level Synthesis Design at NTT Systems Labs," *IEICE Transactions on Information and Systems*, vol.E76-D, no.9, pp.1047–1054, Sept. 1993.

[81] T. Fujiwara, H. Yamamoto, T. Kasami, and S. Lin, "A Trellis-Based Recursive Maximum Likelihood Decoding Algorithm for Linear Block Codes," *IEEE Transactions on Information Theory*, vol.44, no.2, pp.714–729, March 1998.

[82] K. Nagami, K. Oguri, T. Shiozawa, H. Ito, and R. Konishi, "Plastic Cell Architecture," *IEICE Transactions on Electronics*, vol.E81-C, no.9, pp.1431–1437, Sept. 1998.