時系列データ圧縮のための類似部分区間探索手法

小柳 佑介† 范 薇† 朝倉 宏一‡ 渡邉 豊英†

†名古屋大学情報科学研究科 〒464-8603 名古屋市千種区不老町
‡大同工業大学情報学部 〒457-8530 名古屋市南区滝春町 10-3
E-mail: †{koyanagi, fan, watanabe}@watanabe.ss.is.nagoya-u.ac.jp, ‡asakura@daido-it.ac.jp

あらまし 時系列データの圧縮手法として Golomb-Rice 符号化が提案されている．Golomb-Rice 符号化ではデータ値の絶対値が大きいほど符号長が長くなるため，過去のデータの類似箇所と差分を計算し絶対値を小さくする方法が提案されている．しかし，センサデータなどの時系列データでは値の変化が類似していても変化の出現位置にずれがある場合が多く，既存手法ではそのような類似箇所を検出することができない．本稿では，時系列データの変化のずれを考慮して類似構造を検出する手法を提案する．提案手法では，時系列データを増加・減少する箇所毎で分節化する．分節化された部分列の類似判定を行い，類似構造を検出する．評価実験では，周期的なデータをより圧縮することができた．また，既存手法で差分処理を適用できなかった箇所を検出することができ，データ全体で符号長を約 37%削減することができた．

キーワード 類似箇所探索，時系列データ圧縮，タイム・ワーピング

Similar Subsequence Retrieval for the Compression of Time-Series Data

Yusuke KOYANAGI† Wei FAN† Koichi ASAKURA‡ and Toyohide WATANABE†

†Graduate School of Information Science, Nagoya University Furo-cho, Chikusa-ku, Nagoya-shi, 466-8603 Japan
‡School of Informatics, Daido Institute of Technology 10-3 Takiharu-cho, Minami-ku, Nagoya-shi, 457-8530 Japan
E-mail: †{koyanagi, fan, watanabe}@watanabe.ss.is.nagoya-u.ac.jp, ‡asakura@daido-it.ac.jp

Abstract Golomb-Rice coding is a compression method for time-series data. As the value of time-series data is large, the code length based on Golomb-Rice coding becomes long. In order to shorten the code length, calculating a differential signal with the past similar signal. In existing method, subsequence the data value of which is more than threshold is detected. However, the timing of appearance of similar subsequence is different and similar subsequence cannot be detected well. Therefore, we propose the method for detecting two subsequences the change of which is similar. First, the input data is segmented by increase, decrease and constancy. Second, the detected segment is clustered based on the degree of similarity and the similarity of the change of each segment is judged. The respect to which the change is corresponding is detected based on the change obtained by the analysis. Then the relation of the common feature is detected by applying DTW between similar parts of detected change pattern. The proposed method was able to detect periodicity of data and improve compressibility than an existing method. Moreover, for non-periodic data, the proposed method was able to detect a short-length similar subsequence which an existing method was not able to detect, and the code length of the whole data is about 62.7% of the existing method.

Keyword Similar Sequence Retrieval, Compression of Time-Series Data, Time Warping

1. はじめに

センサデータや気象データなど様々な時系列データが，事象の予測や分析に活用されている．事象の予測や分析には大量のデータを収集・蓄積する必要があり，扱うデータ量も時間に従って増加する．そのため，時系列データを圧縮し，データサイズを小さくすることが重要である．

時系列データの圧縮方法として Golomb-Rice 符号化が提案されている[1]．Golomb-Rice 符号化ではデータ値の絶対値が大きいほど符号長が長くなるため，過去のデータにおける類似系列や予測系列との差分を計算し絶対値を小さくすることで符号長を短くする手法が提案されている[2, 3]．しかし，車載センサなどから得られるデータでは，値の変化が類似した箇所が存在しても，変化の出現位置が異なる場合が存在する．このような時系列データでは，類似部分を検出するのが困難である．

本稿では，時系列データの類似部分列を検出するた

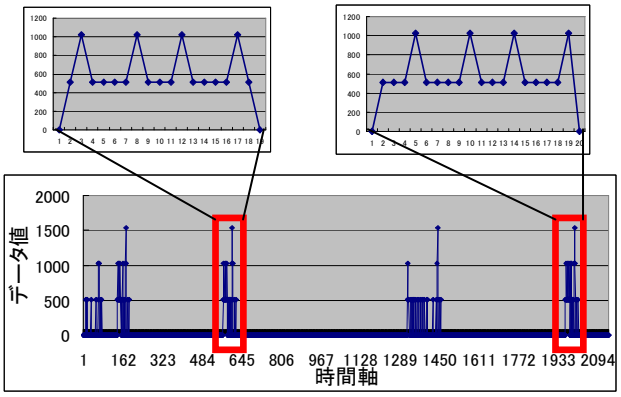


図 1：部分系列における時間軸上のずれ

めに，時系列データ内におけるデータの変動パターンを分析し，変動の一致した部分区間を検出する手法を提案する．提案手法では，まず時系列データを一旦短い部分列に分節化し，その後類似探索時に連続して一致する箇所をまとめて検出する．これにより，短い区間における類似部分列と長い区間連続する類似部分列の両方を検出できる手法が実現可能となる．

2. 関連研究

2.1. 時系列データ圧縮手法

時系列データの圧縮手法として，過去の類似系列との差分を取ることで Golomb-Rice 符号の符号長を抑える手法が存在する[2]．提案手法は時系列データの前後

$$x_i^{'} = \begin{cases} x_i - x_{i-1} & (i \neq 1) \\ x_i & (i = 1) \end{cases}, \quad x_i^{''} = \begin{cases} 2x_i^{'} & (x_i^{'} \geq 0) \\ -2x_i^{'} - 1 & (x_i^{'} < 0) \end{cases}$$

差分信号 **x'**の値を正に変換した信号 **x''**に対して適用される．信号 **x''**は以下の式で定義される．

この手法では，信号 **x''**の移動平均値が連続して閾値を超える範囲を部分系列として検出する．検出された部分系列と類似した系列を，同様にして過去に検出された部分系列から検索し，差分処理を適用する．部分系列間の類似尺度としてはユークリッド距離を用いている．これらの処理を繰り返すことで生成された残差信号に Golomb-Rice 符号化を適用し，復元時に必要となる差分処理の参照情報を付加する．

この手法では，部分系列の区切りが値の変動に対して長くなる可能性がある．そのため，部分系列間に変動のずれが生じ，類似箇所を判定できない．より細かい部分列において類似判定することができない．また，類似尺度にユークリッド距離を用いているため，部分系列同士に時間軸における微小なずれが存在する場合は類似系列として検出することができない．例えば，図 1 では，四角で囲まれた箇所同士において変動が大まかに類似しているように見える．しかし，それぞれ

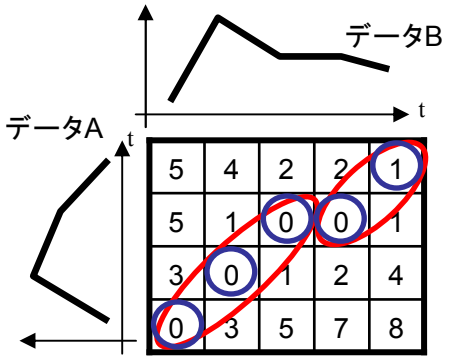


図 2：DTW における点対応

の箇所において既存手法で検出された部分列をみると，左の部分列の 2 点目から 17 点目までの系列が右の部分列の 4 点目から出現し，類似系列の出現するタイミングに違いがある．このように，既存手法での部分列の検出方法では，検出した部分列における値の変動に違いが存在するため，類似した系列として判定されない．時間軸上における類似系列の出現のタイミングによらず，同じ変動系列を類似箇所として検出する方法が必要となる．

2.2. 時間軸のずれを考慮した類似尺度

変化の出現位置のずれを考慮した類似尺度として，DTW（Dynamic Time Warping）が提案されている[4]．DTW は，比較した二つの部分系列間において，最も距離が短くなるような点対応を検出することができる．DTW では，図 2 のようにデータの最初から各データ点間までの最小距離 $a_{i,j}$ を次の式によって計算し，テーブルに格納する．

$$a_{i,j} = D_{i,j} + \min(a_{i-1,j}, a_{i-1,j-1}, a_{i,j-1})$$

次に，二つの系列の最初から最後までの距離の和が最小になる点対応を検出する．検出方法はテーブルを右上から走査し，左，左下，下の三つの値から最小値を選択して辿っていくことで検出が可能である．そのようにして検出されたテーブル上の走査ルートをワーピングパスと呼ぶ．図 2 では円で囲まれた数字の列がそのワーピングパスに当たる．このワーピングパスの内，テーブル上を斜めに進んでいる範囲は，類似点の点対応が 1 対 1 となっている部分である．その部分を Golomb-Rice 符号化の際の差分処理の適応箇所とすることができる．図 2 の例では，楕円で囲まれた部分となる．

DTW において検出されるワーピングパスは，二つの系列全体における距離が最小となるパスである．そのため，お互いに類似部分列を有する二つの系列間に適用しても，類似部分列の出現位置が二つの系列間において大きく異なる場合や，類似部分列以外の部分列

がまったく異なる変動をしている場合は，ワーピングパス上に類似部分列が必ずしも検出されるとは限らない．DTW によって類似部分列を検出したいのであれば，ある程度変動が類似した箇所同士に適用すべきである．

2.3. 類似箇所検出手法のアプローチ

我々は，時系列データの圧縮処理での過去の類似箇所の検索処理において，値の変動に基づいて類似部分列を判定する手法を提案する．DTW を利用して類似点対応をとるため，時間軸におけるずれを検出することで，ずれを考慮した類似箇所の検出が可能となる．提案手法では，DTW を適用する箇所を検索するために，時系列データ内におけるデータの変動パターンを分析し，変動の類似した部分区間を検出する．

提案手法は，次の四つの処理から成る．まず，時系列データを値の変動に基づいて分節化する．次に，区切られた分節を類似度によってクラスタリングし，変動の類似した分節を検索する．次に，分節ごとの変動の類似関係に基づいて変動が連続して一致する区間を検索する．最後に，変動の類似した箇所同士で DTW を適用し，分節ごとの類似の点対応から類似部分列を検出する．

3. 提案手法

3.1. 時系列データの分節化処理

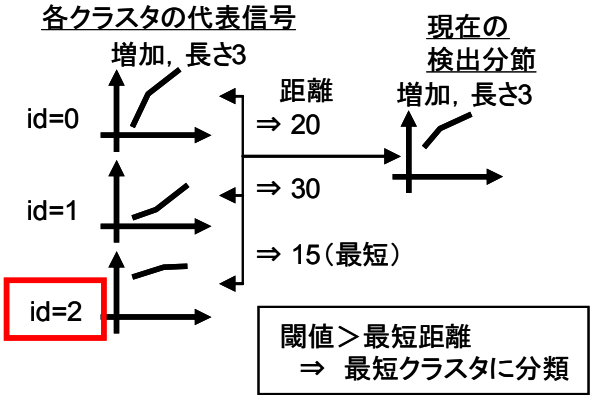
時系列データを値の変動を基に分節化する．文献[2]の手法のように値の変動に比べて部分列の区切りの粒度が大きくならないようにするために，一番細かいデータの形の特徴である，増加・減少・一定の三種類で時系列データを分割する．

変動は信号 **x”**の時間軸上で前後する値の大小を比較して判定する．信号 **x”**の時刻 t における値を x''_t とすると，時刻 t から時刻 t+1 までの変動はそれぞれ，$x''_{t+1} > x''_t$ のとき増加，$x''_{t+1} < x''_t$ のとき減少，$x''_{t+1} = x''_t$ のとき一定となる．信号 **x”**を最初から走査し，連続して変動に変化がない区間を一つの分節とする．

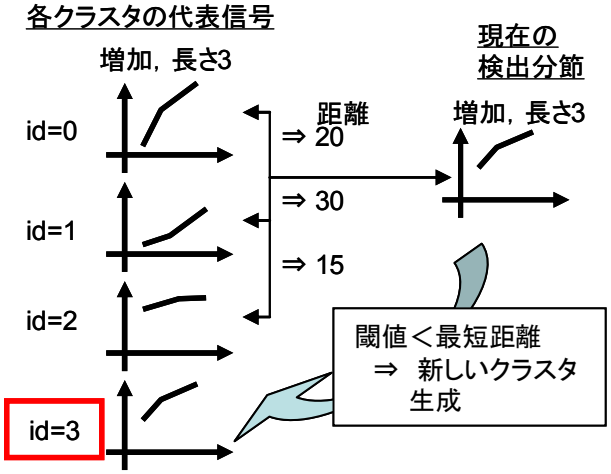
3.2. 分節間の類似判定

分節間の類似判定をクラスタリングによって行い，各分節の情報を時系列に並べた配列，変動配列を出力する．分節の情報は，変動の種類，分節の長さ，類似クラスタの id から構成される．

分節間の類似判定のために，分節を類似度によってクラスタリングする．クラスタリングは，変動の種類，分節の長さ毎に行われる．二つの分節間での残差信号の Golomb-Rice 符号の長さは，それらの分節間のマンハッタン距離と比例関係にある．そのため，部分列間



(a)類似クラスタの選択



(b)新しいクラスタの生成

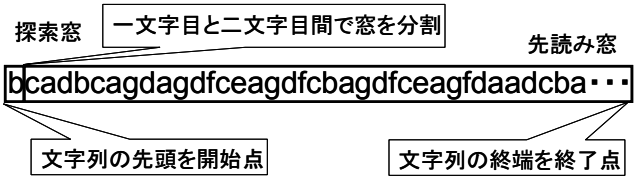
図 3：分節の変動類似クラスタリング

の類似尺度は，マンハッタン距離で算出する．長さ L の二つの分節 a,b のマンハッタン距離は次の式で算出される．

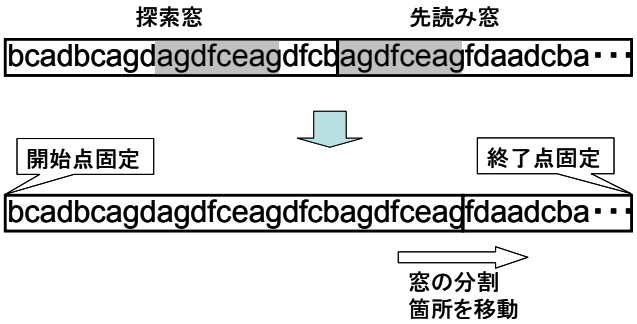
$$D(x,y) = \frac{1}{L}\sum_{i=1}^{L} | x_i - y_i |$$

クラスタリングの類似判定は，類似度の閾値によって判定される．類似判定の閾値はパラメータとして与えられるものとする．

各クラスタとの類似度は，クラスタ内の一つの分節との距離から計算される．距離を計算するクラスタを代表信号と呼ぶ．代表信号は，最初にそのクラスタに分類された信号が選択される．まず，代表信号との距離が一番短いクラスタを検出する．次に，クラスタとの最短距離と閾値の大小関係を判定する．クラスタとの最短距離が閾値より短い場合，検出された分節はそのクラスタと同じ変動であると判定される（図 3(a)）．一番最初に分節が検出された場合や，クラスタの代表信号との距離が閾値を超えている場合は，検出された部分列を代表信号とするクラスタが新しく生成される



(a)窓の初期状態



(b)窓の移動

図 4 ：探索窓と先読み窓の更新方法

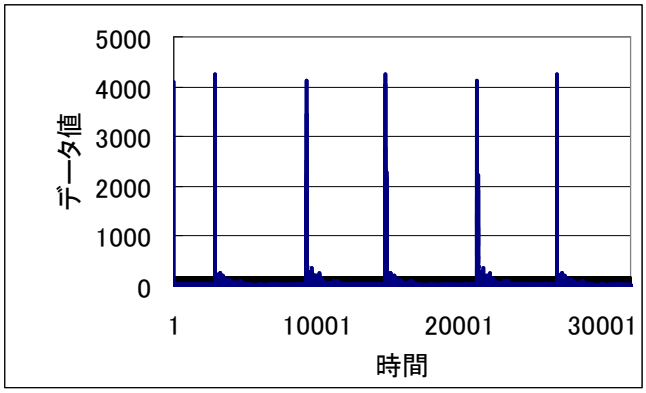
（図 3(b)）. 各クラスタには，変動の種類と長さごとに他のクラスタと識別するための id が割り当てられる. クラスタの id はクラスタ生成時にはそれまでのクラスタ数が割り当てられる. このクラスタ id が変動配列の類似クラスタ id の値となる.

3.3. 変動一致箇所の検出

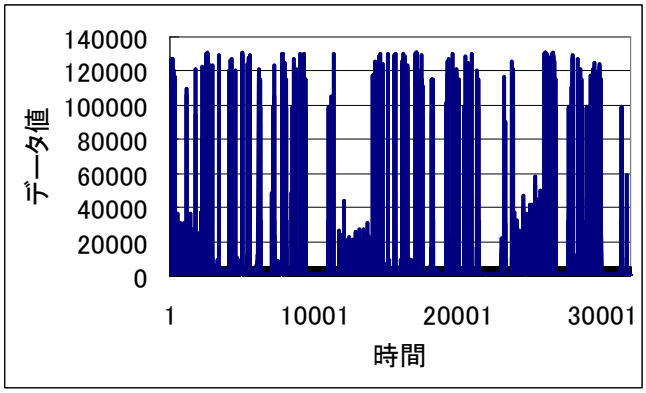
3.2 節の処理で生成した変動配列を走査し，変動が連続して一致している箇所を検索する. 変動配列の各要素の変動の種類，分節の長さ，類似クラスタの id が全て一致する場合は，変動が一致すると判定する. 各分節間の関係は一致か不一致のどちらかとして判定されるため，一致箇所を検索するためにはテキスト中の文字列検索と同様の処理を適用することができる. 本手法では，Zip 等で使われる Sliding Window の方法を用いる[5]. この方法では，探索窓と先読み窓の二つの窓を文字列の最初から順に走査する. 探索窓内を走査して，先読み窓の先頭から始まる文字列に一致する一番長い文字列を検索する. この処理を変動配列に対して行い，検出された変動一致箇所同士で DTW を適用し類似箇所のデータ点対応を抽出する.

説明の簡単化のために，図 4 では変動配列を文字列で表現する. 同じ文字で表現される分節は同じ変動の分節であることを示している.

我々の手法では，窓の大きさを制限せずに変動配列中の全探索を行う. そのため，探索窓と先読み窓の長さは固定しない. まず，変動配列の探索窓と先読み窓を図 4 の初期状態のように配置する. 探索窓の開始点は変動配列の先頭，先読み窓の終了点は変動配列の終端，探索窓と先読み窓の分割点は配列の 1 要素目と 2 要素目の間とする. 次に，窓の範囲の更新において，図 4 のように，探索窓の開始点と先読み窓の終了点はスライドせず，探索窓と先読み窓を分割する点のみスライドする. この処理では探索窓と先読み窓は配列全体を常に含むため，配列全体を配列一致箇所の探索範囲となる.



(a)データグループ A の一データ



(b)データグループ B の一データ

図 5：実験に用いたセンサデータ

4. 評価実験

提案手法の有効性を検証するために，実際のセンサデータを用いて評価実験を行う. センサデータには完全に周期性のあるデータ 15 個から成るデータグループ A（図 5(a)）と，あまり周期性のみられないデータ 115 個から成るデータグループ B（図 5(b)）を用いた. 図 5(a)の前後差分正規化信号では 1 周期中に 2 回大きな値が出現する. また，それぞれの周期におけるデータの数字列は等しい. データの長さは 1,250,000 点である. 図 5(b)の前後差分正規化信号にはおおまかには同じような変動部分が見られるものの，データグループ A のような同じ部分列の繰り返しではない. データの長さは 32,513 点である. 周期性のあるその両データに提案手法と既存手法を適用し，性能評価を行う. 既存手法として，文献[2] の手法のように，データ値

が閾値を超える範囲を部分列として切り出す手法を用いる．

まず，データグループ A のデータに対して既存手法と提案手法のそれぞれを適用した結果を示す．両手法において類似箇所として検出した長さを表 1 に示す．長さ 1,250,000 の 15 個のデータのうち，いずれかの手法で類似箇所が検出されたデータを示した．全てのデータにおいて検出した長さは提案手法の方が長い．また，いずれの検出においても，ちょうど 1 周期前のデータを参照しており，提案手法は正確に類似した箇所を検出できていることが分かる．すなわち，提案手法は周期的な信号の周期を検出して類似判定が可能である．

次に，データグループ A のデータに対して差分処理を適用した場合の符号長を比較する．表 2 は両手法における差分後の信号の符号長である．各手法の差分後のデータの符号長，参照情報の符号長，その両方をあわせた符号長をそれぞれバイト数で示した．

参照情報の符号長は，データ 11 以外では既存手法に比べて短くなっている．この結果は表 1 での検出した類似箇所の長さとは逆の結果であるが，このことは提案手法が参照情報の連続した箇所を一まとまりで参照しているためである．特に，データ 2, 9, 11, 13 においては一つの参照情報のみで類似関係を表現できている．データグループ A のような完全な周期信号の場合，一周期分走査が終わった後では参照は一箇所だけ行うのが一番効率がよい．このような参照の検出は，値が閾値を超える部分列同士でしか類似判定を行わない既存手法では実現不可能である．

また，差分後の信号の符号長は，データ 2, 8, 11, 12, 13, 15 において既存手法より削減することができている．一方，データ 9 においては既存手法ほど削減できていない．これは，提案手法では変動が一致しないと判定された箇所は類似判定を行わないためである．データ 9 においては，参照情報との合計では提案手法は既存手法より符号長を少なくすることができている．

最後に，ほとんど周期性のないデータに対する評価を述べる．データグループ B 中のデータ 115 個に対し，59 個のデータで提案手法の方が既存手法に比べて符号長が短くなり，30 個のデータで提案手法における符号長が既存手法より長くなった．結果が顕著であったものを表 3 に示す．データ 16, 113, 114 は提案手法の方が符号長が短かった例，データ 77, 90, 95 は提案手法の方が符号長が長かった例である．それぞれ，既存手法での符号長に対する提案手法での符号長の比の小さいものから順に 3 個，大きいものから順に 3 個となっている．

データ 114, 115 では，既存手法では検出することができなかった差分処理適用箇所を提案手法において検出することができたことがわかる．また，データ 90 では，検出した類似は提案手法の方が短かった．これは，提案手法において，既存手法で差分を行った部分に差分処理を適用検出することができなかったためであることを示す．また，データ 77, 95 においても，既存手法で差分処理を適用した箇所に対して提案手法が差分処理を実行できていないことがあった．これは，提案手法において変動が一致しない箇所では類似部分列検索を行わないためである．また，連続して一致した箇所を検出したが，差分後の符号長が悪くなったため最終的に差分が行われなかったことも原因であると考えられる．既存手法では値が閾値を超える箇所のみで判定を行っているため，部分列ごとに参照箇所に区切りが生じるため，その区切りごとに差分前後での符号長の変化の判定を行う．一方，提案手法では類似点の 1 対 1 対応が続く限り検出し，その後で差分前後での符号長の変化の判定を行うため，差分適用箇所がう

表1：周期的なデータにおける差分箇所

データ番号	提案手法	既存手法
2	874996	192
8	1182165	29781
9	1023541	11517
11	1063381	0
12	1154822	14040
13	1171885	4992
15	1169795	12726

表2：周期的なデータにおける符号長

データ番号	手法	残差	参照情報	合計
2	既存	953101	128	953229
	提案	859375	16	859391
8	既存	880421	9920	890341
	提案	792217	144	792361
9	既存	781466	6640	788106
	提案	783128	16	783144
11	既存	781819	0	781819
	提案	781363	16	781379
12	既存	809257	3328	812585
	提案	785099	48	785147
13	既存	785814	3328	789142
	提案	781775	16	781791
15	既存	825049	3312	828361
	提案	785261	48	785309
全体	既存	12849447	26928	12876375
	提案	12601762	320	12602082

まく取れていなかったと考えられる．

全体では符号長を既存手法の約 62.9%に抑えることができた．既存手法において符号長が著しく長いデータでは，提案手法において符号長を短縮できている．逆に既存手法において符号長が短いデータでは提案手法において結果が悪くなっているが，符号長が短くなったデータでの削減分よりは少ないため，提案手法において全体での符号長が短くなっている．

5. まとめ

我々は，Golomb-Rice 符号化を行うために，DTW を適用する箇所を検出する手法を提案した．評価実験により，周期性のあるデータにおいて圧縮率を向上させることができた．また，周期性のあまりないデータに対しては，既存手法で差分処理を適用することができなかった箇所を検出することができ，符号長を短くすることができた．しかし，逆に既存手法において検出していた差分処理適用箇所が検出できず，既存手法でうまく圧縮できたデータでは符号長が長くなった．

今後の課題として，提案手法の差分判定処理を改善することがあげられる．DTW でのテーブルの走査時に，テーブルの値から距離を加算し差分判定に用いる．また，変動の類似しない箇所でも類似検索をするように手法を改良することも課題である．リアルタイムな状況にも活用できるように，処理のリアルタイム化を行う必要がある．

表3：非周期的なデータにおける符号長

データ番号	手法	残差	参照情報	合計
16	既存	5762216	256	5762472
	提案	1853943	43936	1897879
77	既存	37037	6368	43405
	提案	50403	2304	52707
90	既存	29430	208	29638
	提案	35061	16	35077
95	既存	31968	384	32352
	提案	42462	96	42558
113	既存	3549560	0	3549560
	提案	1354974	75088	1430062
114	既存	4100814	0	4100814
	提案	1632639	75568	1708207
全体	既存	74992004	36000	75028004
	提案	45551549	1481328	47032877

文　献

[1] S. W. Golomb, “Run-Length encodings”, IEEE Trans. on Information Theory, IT-12, pp.399-401(1966).

[2] 竹澤 哲矢，朝倉 宏一，渡邉 豊英，“近傍平均値の増大に着目した時系列データ可逆圧縮”，電気学会論文誌 C, Vol. 128, No. 2, pp.318-325(2008).

[3] 鎌本 優，守屋 健弘，原田 登，西本 卓也，嵯峨山 茂樹，“ISO/IEC MPEG-4 Audio Lossless Coding(ALS) におけるチャネル内とチャネル間の長期予測”，電子情報通信学会論文誌 B, Vol.J89-B, No.2, pp. 214-222 (2006).

[4] J. B. Kruskal and M. Liberman, ”The Symmetric Time-Warping Problem: From Continuous to Discrete” Time Warps, String Edits, and Macromolecules: The Theory and Practice of Sequence Comparison, pp. 125-161 Addison-Wesley, (1983).

[5] L. Peter Deutsch, ”DEFLATE Compressed Data Format Specification version 1.3”, Internet RFC 1951, (1996).