# Extended Boyer-Moore 法による古い新聞画像からの高速全文検索

寺沢 憲吾[†,††] 島 貴宏[†] 川嶋 稔夫[†] 田中 譲[†††]

† 公立はこだて未来大学システム情報科学部
〒 041–8655 北海道函館市亀田中野町 116–2
†† 独立行政法人 科学技術振興機構 さきがけ
〒 332–0012 埼玉県川口市本町 4–1–8
††† 北海道大学知識メディアラボラトリー
〒 060–8628 北海道札幌市北区北十三条西八丁目
E-mail: †{kterasaw,g2109021,kawasima}@fun.ac.jp, ††tanaka@meme.hokudai.ac.jp

**あらまし** Boyer-Moore 法は，あるテキストストリングの中から与えられたキーワードと完全に一致する部分を探し出す非常に高速なアルゴリズムとしてよく知られている．本研究では，検索の対象をテキストストリングから一般の実ベクトル系列に拡張した，Extended Boyer-Moore 法を提案する．実ベクトルを複数の整数の組からなる擬似コード表現に変換し，さらに semiequivalent という新しい二項関係を導入してその上で skip 関数の構築を行うことで，Boyer-Moore 法のうち Bad Character Rule を実ベクトル系列の類似検索問題に対して再現した．本手法はさまざまな時系列データに対して高速な類似検索を提供しうるアルゴリズムであるが，その応用の一例として，OCR によるテキスト化が困難な古い新聞画像に対する全文検索実験を行い，従来法によるものよりも検索コストを削減できることを確認した．

**キーワード** 情報検索，全文検索，Boyer-Moore 法，時系列解析，文書解析・文書理解

# The Extended Boyer-Moore Algorithm for Fast String Matching in Old Newspaper Images

Kengo TERASAWA[†,††], Takahiro SHIMA[†], Toshio KAWASHIMA[†], and Yuzuru TANAKA[†††]

† School of Systems Information Science, Future University-Hakodate
116–2 Kamedanakano, Hakodate, Hokkaido, 041–8655 Japan
†† PRESTO, Japan Science and Technology Agency,
4–1–8 Honcho, Kawaguchi, Saitama, 332–0012 Japan
††† Meme Media Laboratory, Hokkaido University
West8. North13, Kita-ku, Sapporo, Hokkaido, 060–8628 Japan
E-mail: †{kterasaw,g2109021,kawasima}@fun.ac.jp, ††tanaka@meme.hokudai.ac.jp

**Abstract** Boyer-Moore algorithm is known as a very efficient algorithm that finds a place where a certain string specified by the user appears within a longer text string. In this study, we propose the Extended Boyer-Moore algorithm that can retrieve a pattern in the sequence of real vectors, rather than in the sequence of the characters. We reproduced the Bad Character Rule of Boyer-Moore algorithm to the sequence of real vectors by transforming the vectors into pseudo-code expression that consists of multiple integers and by introducing a novel binary relation called 'semiequivalent.' We confirmed the practical utility of our algorithm by applying it to the string matching problem in old newspaper images to which the optical character recognition does not work well. In the experiment, our algorithm ran faster than the naive method and reduced the computational cost.

**Key words** Information Retrieval, String Matching, Boyer-Moore Algorithm, Time Series Analysis, Document Analysis and Recognition

## 1. はじめに

ネットワーク上には大規模な情報が蓄積されており，その量は増加の一途をたどるばかりである．それに伴い，こうした情報の中から利用者が必要とする情報を効率よく取り出すための情報検索技術はますますその重要さを増している．テキストストリングに対する全文検索技術はそのうちの最も基本的なものであると言え，古くから多くの研究が行われている．中でも，Boyer-Moore 法（BM 法）[1] は非常に高速なアルゴリズムとしてよく知られている．文字列照合に基づくアルゴリズムである BM 法は，キーワードの末尾から照合を行うことで，テキストストリングのうちのいくつかの文字については１回もアクセスすることなく，すなわち，サブリニア時間で検索を行うことができる．

本研究は，この Boyer-Moore 法を拡張し，検索の対象をテキストストリングから一般の実ベクトル系列に拡張する．これは，さまざまな時系列データ，あるいは擬似的に時系列と見なせるデータに対して高速な類似検索を提供するための基盤となり得る技術である．

この拡張における主な要素技術は次の 2 点である．１つは実ベクトルを擬似コードとして離散化表現する技術である LSPC [2] である．この擬似コード LSPC は，通常のベクトル量子化に比べて，もとの実ベクトルの記述力を極力損なわないままベクトルを離散化することができる．ただしこの擬似コードにおいて中心的な役割を果たす二項関係 semiequivalent は同値関係ではなく，推移律を満たさないものであるため，既存の文字列検索アルゴリズムがそのまま適用できるわけではない．そこで第 2 の要素技術として，本研究では，既存の文字列検索アルゴリズムのうち最も有力なものの１つである BM 法について，これを LSPC に適用可能なようにアレンジした．これが本論文の表題にも示した Extenden Boyer-Moore 法である．

アルゴリズムの詳述に先立ちあらかじめ本手法の限界点を述べると，LSPC による擬似コード変換は確率的アルゴリズムであるため，それに基づく検索アルゴリズムでは結果が確実に正確であることは保証されない．試行回数やその他のパラメタの設定によって正確な出力が得られる確率を高めることはできるものの，それは計算量とトレードオフである．この点は他の確率的アルゴリズムと同様である．

本論文の構成は以下の通りである．まず 2. 章で本研究の重要な要素技術である擬似コード LSPC について述べる．次に 3. 章で BM 法を説明してから，4. 章でこれを LSPC 向けに拡張した Extended Boyer-Moore 法について述べる．5. 章では本手法の応用の一例として，OCR によるテキスト化が困難な古い新聞画像に対する全文検索実験を行い，従来法によるものよりも検索コストを削減できることを示す．最後に 6. 章で結論と今後の展望を述べ，結びとする．

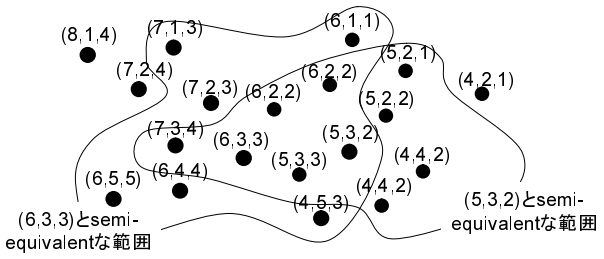


図 1　擬似コード LSPC の例

## 2. 擬似コード LSPC

### 2.1 LSPC の概要

LSPC（Locality-Sensitive Pseudo-Code）は，Terasawa and Tanaka [2] により提唱された，実ベクトルを擬似コード表現に変換する技術である．通常のベクトル量子化とは異なり，１つのベクトルに対して複数の整数を割り当てることで，通常のベクトル量子化よりも豊かな記述力を残したままベクトルを離散化する．

LSPC の例を図 1 に示す．図中，黒丸が $\mathbb{R}^d$ 空間に分布しているベクトルを表し，黒丸につけられた (8,1,4) などの数字の 3 つ組がそのベクトルを離散化した結果の擬似コードを表している．このように，１つのベクトルに複数の整数を割り当てることが擬似コード LSPC の特徴である．そして，LSPC のもう１つの特徴が，そこに導入される二項関係 semiequivalent である．2 つの擬似コードに割り当てられた複数の整数のうち，１つでも一致していればそれらのコード間で二項関係 semiequivalent が満たされると定義する．

以上をまとめると以下のようになる．

**定義 1** (擬似コード LSPC の定義)．$d$ 次元実ベクトル $p$ に対し，擬似コード $C(p)$ は，$d'$ 個の整数の組として与えられる．すなわち，

$$p \in \mathbb{R}^d \mapsto C(p) = (c_1(p)\ c_2(p)\ \cdots\ c_{d'}(p))^T \in \mathbb{N}^{d'}.$$

**定義 2** (二項関係 semievuivalent の定義)．擬似コード LSPC において $C(p)=\{c_i(p)\}$ と $C(q)=\{c_i(q)\}$ が二項関係 semiequivalent を満たすとは，$c_i(p) = c_i(q)$ を満たすような $i \in \{1,\ldots,d'\}$ が存在することである．

LSPC では，通常のベクトル量子化よりも多様な表現が可能である．例えば，通常のベクトル量子化においては，境界面付近では似たベクトルに別な符合が割り当てられてしまうケースがどうしても発生する．一方 LSPC では，図 1 のように，ある擬似コードと semiequivalent な範囲と，別のある擬似コードと semiequivalent な範囲とがオーバーラップすることが許されている．これにより，近いベクトル同士に割り当てられた擬似コードが semiequivalent になる確率を極めて高くなるように擬似

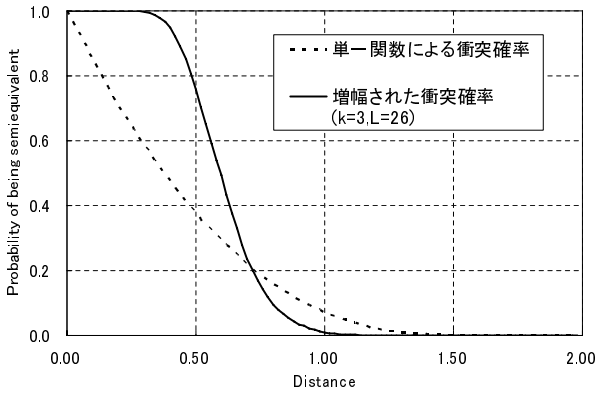


図 2　2 点間の距離と，二項関係 semiequivalent が満たされる確率のグラフ

コードを割り振ることが可能となる．

二項関係 semiequivalent が満たされるか否かは，2 つのベクトルの間の距離に依存している（その例を図 2 に示した）．この性質を用いて，2 つのベクトル間の距離が一定の閾値以上であるか否かを，擬似コードの二項関係だけから推定することができる．例えば図 2 における実線が表す曲線の例であれば，2 つのベクトル間の距離が 0.9 以上であればそれらのベクトルに対応する擬似コード同士が semiequivalent である確率はきわめて小さく，一方で 2 つのベクトル間の距離が 0.4 以下であればそれらのベクトルに対応する擬似コード同士が semiequivalent である確率はきわめて大きい．この性質を用いることで，2 つの擬似コードが semiequivalent であるかどうかを調べるだけで，もともとのベクトル間の距離が大きいのか小さいのかをおおよそ推測することができる．

ここで重要な点は，擬似コード $C(p), C(q)$ が二項関係 semiequivalent を満たすかどうかを照合するためのコストは，2 つのベクトル $p, q$ 間の距離を計算するコストよりも小さいということである．この差は特にもとのベクトル空間が高次元である場合に顕著である．パターン認識で用いられる特徴ベクトルの空間は非常に高い次元数を持つ場合がしばしばあるので，LSPC を使用することによる計算コスト削減が大きな効果を持つことが期待できる．

## 2.2　LSPC の構成法

前節では，擬似コード LSPC の持っている性質について述べた．この節では，そのような性質を持つ擬似コードをいかにして構成することができるかについて述べる．

擬似コード LSPC の各要素には，Locality Sensitive Hashing (LSH) [3]〜[5] のハッシュ関数のハッシュ値を利用する．LSH は，近似的最近傍探索問題の解法として非常に有力なものとして近年注目されているアルゴリズムである．以下，まず LSH の概略を述べる．

LSH アルゴリズムにおいて中心的役割を果たすのは，以下で定義される LSH family と呼ばれる関数族である．

**定義 3** (LSH family)．ベクトル空間 $S$ から適当な離散空間 $U$ への関数族 $\mathcal{H} = \{h : S \to U\}$ は，以下を満たすとき $(r_1, r_2, p_1, p_2)$-sensitive であると呼ばれる．

任意の $v, q \in S$ について，
$d(v,q) \leqq r_1$ のとき $\Pr_{\mathcal{H}}[h(q) = h(v)] \geqq p_1$,
$d(v,q) > r_2$ のとき $\Pr_{\mathcal{H}}[h(q) = h(v)] \leqq p_2$

ここで $d(v,q)$ はベクトル $v, q$ 間の距離を表し，また，$p_1 > p_2$，$r_1 < r_2$ とする．

なお，このような関数族の存在は $L_1$ 空間において [3]，$s \in (0, 2]$ の任意の $L_s$ 空間において [4]，任意の次元の単位球上において [6]，それぞれ保証されている．

関数族 $\mathcal{H}$ に属する関数は，ベクトル $v \in \mathbb{R}^d$ を整数 $h(v) \in \mathbb{N}$ に写像するという意味で，ハッシュ関数であると言える．近傍探索問題の解法としての LSH は，データ集合内のすべての点についてあらかじめハッシュ値を計算してハッシュ表を作成しておいて，クエリ点と同じハッシュ値を持つ点だけをハッシュ表から探し出して探索の対象にする．こうすることによって，クエリ点との距離を計算されるべき点の数を大きく削減する，というのが LSH の基本アイデアである．

LSH における優れた工夫の一つは，ハッシュ関数を複数使うことによって，False positive を減らしつつ，False negative を減らすようにできることである．例えば，ある LSH family があって，その中の関数は図 2 の点線で示されるような衝突確率を持っていたとする．LSH は，この関数 $k$ 個を組み合わせた関数を $L$ 個作ることで，その確率の差を増幅する．つまり，$h_{ij}$ を LSH family $\mathcal{H}$ からランダムに選ばれたハッシュ関数として，

$$g_i(p) = \{\ h_{i1}(p), h_{i2}(p), \ldots, h_{ik}(p)\ \} \qquad (1)$$

を $g_1(p)$ から $g_L(p)$ まで $L$ 個作成することによって，実効的な衝突確率を図 2 の実線のように変換するのである．

LSH の近傍探索問題の解き方は次の通りである．まず，すべての点 $p$ に対する $g_i(p), g_2(p), \ldots, g_L(p)$ をあらかじめ計算し，ハッシュ表に格納しておく．クエリ点 $q$ が入力されたら，$g_1(q), g_2(q), \ldots, g_L(q)$ を計算し，$\exists i, g_i(p) = g_i(q)$ となるような点 $p$ だけをハッシュ表から見つけてきて，探索の対象とする．これにより，クエリから遠い点との距離計算を減らしつつ，クエリに近い点の取りこぼしを減らすということを実現している．

擬似コード LSPC の基本アイデアは，この LSH のハッシュ値は擬似コードとして使えるのではないかという着想である．LSH は近傍探索問題を解くためのハッシュ表を作成するための手段としてハッシュ関数を使っていたが，LSPC ではベクトル系列のマッチングのための疑似コード表現として同じハッシュ関数を用いるのである．すなわち，

**定義 4** (擬似コード LSPC の構成法). $p \in \mathbb{R}^d$ に対し，擬似コード LSPC $C(p)$ を以下のように定義する．

$$C(p) = \{\ g_1(p), g_2(p), \ldots, g_L(p)\ \} \tag{2}$$

$$g_i(p) = \{\ h_{i1}(p), h_{i2}(p), \ldots, h_{ik}(p)\ \} \tag{3}$$

ここで $h_{ij}$ は LSH family $\mathcal{H}$ からランダムに選ばれた関数である．

上記の定義では $C(p)$ は $kL$ 個の整数から構成されるが，実用上これを次の方法で $L$ 個の整数として表すことにする．すなわち，$h_{ij}$ の値域は有限の整数であるので，その上限を $M$ とおく．このとき，$g_i(p) = h_{i1}(p)M^{k-1} + h_{i2}(p)M^{k-2} + \cdots + h_{ik}(p)$ とすることで，$g_i(p)$ は 1 つの整数で表すことができる．従って，$C(p)$ は $L$ 個の整数として表せる．

LSH における近傍探索は，クエリ $q$ に対して，ハッシュ表から $\exists i, g_i(p) = g_i(q)$ となる点 $p$ をすべて拾ってくるのであった．それに対応する，擬似コード LSPC における二項関係が次に述べる semiequivalent である．

**定義 5** (二項関係 semievuivalent の定義). LSPC $C(p) = \{g_i(p)\}$ と $C(q) = \{g_i(q)\}$ が semiequivalent であるとは，$g_i(p) = g_i(q)$ となるような $i$ が存在することである．

この二項関係が満たされているか否かは，前述の図 2 の実線に示すとおり，もとのベクトル $p, q$ の間の距離に依存している．

ここで改めてパラメータ $k$ と $L$ の意味を考えてみる．$p(c)$ が単一のハッシュ関数で，間の距離が $c$ である 2 つの点が衝突する（同じハッシュ値を持つ）確率，$P_{semieq}$ がそれらの点に対応する擬似コードが semiequivalent である確率を表すとする．LSPC では，単一のハッシュ関数を $k$ 個組み合わせた物を $L$ 個作る，という操作で，$P_{semieq}$ を

$$P_{semieq} = 1 - \left(1 - p(c)^k\right)^L \tag{4}$$

のように増幅している．$k$ と $L$ の値を設定することにより，図 2 における曲線の位置を変えることができる．$k$ を増やせば曲線は左に移動し，$L$ を増やせば曲線は右に移動する．$k$ と $L$ をともに増やせば，おおよその位置を変えずに曲線の勾配を増すことができる．

## 3. Boyer-Moore 法

この章では，Extended Boyer-Moore 法の前提となる，Boyer-Moore 法について述べる．Boyer-Moore 法は文字列検索アルゴリズムとしては古典的でかつきわめて有力な方法であり，多くの解説書が出版されている．以下の説明では，Bad Character Rule の名は文献 [7] に，skip 関数の名は文献 [8] によった．

なお，文献 [7] で説明されている Boyer-Moore アルゴリズムは Good Suffix Rule と Bad Character Rule の 2 つの柱から構成されているが，このうち LSPC のような推移律を満たさないコードに適用可能なものは後者のみである．今日 BM 法と呼ばれるものには微妙なバリエーションがあるが，ちょうど文献 [8] の BM 法の説明が Bad Character Rule のみに焦点が置かれたものとなっており，Extended Boyer-Moore 法の前提とするのに都合がよいので，ここでは文献 [8] に従い，Boyer-Moore アルゴリズムの Bad Character Rule について，その概要を述べる．

まず，問題を正確に定式化する．String Matching 問題とは，次のような問題である．

**定義 6** (String Matching 問題とは：). 長さ $n$ の文字列 $P$（パターンあるいはキーワードと呼ぶ）と，長さ $m$ $(\geq n)$ の文字列 $T$（テキストと呼ぶ）とが与えられたとき，テキスト中のキーワードの出現開始位置，すなわち $P(\xi) = T(i + \xi - 1)$ for all $\xi = 1, 2, \ldots, n$ となるような $i$ をすべて見つけること．

なお，文字列 $P$ に対し，$P(\xi)$ は $P$ の $\xi$ 番目の文字を表す．たとえば，$T = abxababababxababx$，$P = abab$ の時，String Matching 問題の解は $\{4, 6, 11\}$ である．また，以下では文字列 $S$ の $i$ 番目から $j$ 番目の文字 $(j \geq i)$ までで構成される部分文字列を $S[i:j]$ と書くことにする．

String Matching 問題の素朴な解法（naive method）は，パターン $P$ に対し，まず $T$ の部分文字列 $T[1:n]$ と照合を行い，次に $T[2:n+1]$，その次に $T[3:n+2]$……という具合に，$T$ の部分文字列の始点を 1 つずつずらしながらすべての部分文字列と照合を行う方法である．この方法の計算コスト（文字照合回数）は最小で $m$ であり，最大で $O(mn)$ である．

ここから BM 法について述べる．素朴な解法が T の部分文字列の始点を 1 つずつずらしながら文字列の照合を行っていくのに対し，BM 法は始点を 1 つよりも大きくずらすことを試みることによって計算量を縮減する．

なお，文献 [7] によれば，BM 法は Bad Character Rule と Good Suffix Rule からなっており，この両方を用いることで計算量が $O(m)$ で抑えられることが保証される．一方で文献 [8] では，上記 2 つのルールのうち Bad Character Rule のみを使ったものを BM 法と呼んでおり，この場合は $O(m)$ の計算量が保証されない（最悪の場合 $O(nm)$ になってしまう）．しかし，極端な場合の入力（例えば $T = aaaaaaa$，$P = aaa$）を考えなければ，通常の場合はこれだけでも十分に高速である．

例として，$T = abcbaxabacabbc$，$P = abac$ という場合を考える（図 3）．BM 法も素朴な解法と同様に，まず $P$ と $T[1:4]$ とを照合する．ただしこの照合にあたって，BM 法は文字の照合を $P$ と $T[1:4]$ の末尾（すなわちこの場合は 4 文字目）から順に開始するというところに特徴がある．今の例の場合，$T[1:4]$ の 4 文字目は $b$，$P$ の 4 文字目は $c$ であるから照合結果は「不一致」であ

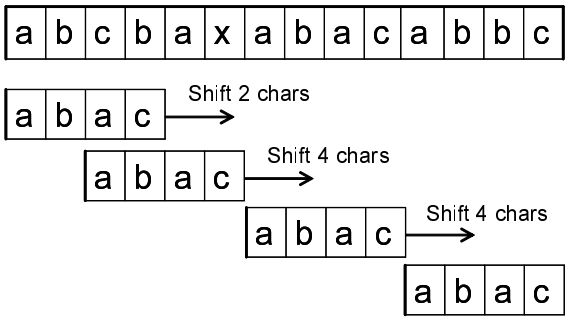


図 3　BM 法によるマッチングのプロセス

| | |
|---|---|
| skip(a) | 1 |
| skip(b) | 2 |
| skip(c) | 4 |
| skip(else) | 4 |

図 4　図 3 の例における skip 関数

り，$T$ の部分文字列の始点をずらして次の照合へ進む．

さてここでポイントは，$T$ の部分文字列の始点を何文字分ずらすかである．いま，$T[1:4]$ の 4 文字目は $b$ であることはすでに調べた．ということは，仮に 1 文字ずらした場合，次は $T[2:5]$ の 3 文字目が $b$ であることはわかっているので，これが $P$ と一致しないことは調べるまでもなく明らかである．仮に 2 文字ずらすとすると，$T[3:6]$ の 2 文字目である $b$ は，$P$ の 2 文字目と一致する．よってこれは調べる価値がある．従って $T[1:4]$ を不一致と判断した次に調べるべき $T$ の文字列は $T[3:6]$ であることがわかる．

では次に，$T[3:6]$ と $P$ の照合を行う．$T[3:6]$ の 4 文字目は $x$, $P$ の 4 文字目は $c$ であるから照合結果は不一致であり，$T$ の部分文字列の始点をずらして次の照合へ進む．さて今の場合，$T[3:6]$ の 4 文字目である $x$ は，$P$ の中に一度も出てこない．ということは，$T(6)=x$ を含むような $T$ の部分文字列と $P$ とは決して一致しないことはこの時点で明らかである．よって次に照合すべき $T$ の部分文字列は，一気に 4 つ飛んで $T[7:10]$ であるということになる．

続いて，$P$ と $T[7:10]$ との照合を行う．ここも末尾から順に照合を行う．照合を 4 回行ったところで，$P$ と $T[7:10]$ の全体が一致していることが判明するので，始点位置 “7” が出力される．そして，末尾の $c$ に着目すると，$T[7:10]$ を 1 文字ずらした場合も 2 文字ずらした場合も 3 文字ずらした場合も $P$ とは一致しないことが明らかであるので，次に照合すべき $T$ の部分文字列は，4 つ飛んで $T[11:14]$ となる．最後に $T[11:14]$ と $P$ の照合を行い，後ろから 2 文字目で不一致が判明して，アルゴリズムは終了する．

上記の BM 法のアルゴリズムでポイントとなるのは，$T[i:i+3]$ の 4 文字目が何であった場合に，次に照合する $T$ の始点を何文字分ずらせばよいか，ということである．ここではこの「何文字分ずらせばよいか」の値を「skip 関数」で表すことにする．すなわち，skip($x$) は，$T[i:i+3]$ の末尾の文字（4 文字目）が $x$ であった場合に照合の始点をずらすべき数として定義され，最小は 1 であり，最大は $P$ の長さ（この場合は 4）である．

skip 関数は次のように構成される．skip 関数は，定義域として想定しているすべての文字（アルファベット $\Sigma$ と呼ぶ），値域として 1 以上 $n$ 以下の自然数を持つ関数である．まず，すべての文字 $x \in \Sigma$ について，skip($x$) を $n$ に初期化する．次に，$P$ の $(n-i)$ 文字目が $x$ であったら，skip($x$) を $i$ に更新する．この手続きを $i = n-1, n-2, \cdots, 1$ の順に繰り返せば，所望の skip 関数が得られる．

BM 法の特徴は，パターン文字列 $P$ が長ければ長いほど，また，文字列中から適当に 2 つ選んだ文字が適合する確率が低いほど（すなわち文字種が多いほど），効率が良くなるということである．

## 4. Extended Boyer-Moore 法

本研究の目的は，BM 法を擬似コード LSPC に拡張することである．そこで，定義 6 で述べた問題を LSPC に拡張した問題を次のように定義する．

**定義 7** (LSPC に拡張された String Matching 問題とは:). 長さ $n$ の擬似コード列 $P$ と，長さ $m$ $(\geqq n)$ の擬似コード列 $T$ とが与えられたとき，$P(\xi) \sim T(i+\xi-1)$ for all $\xi = 1, 2, \ldots, n$ となるような $i$ をすべて見つけること．

ここで，$S(i) \sim T(j)$ とは，$S(i)$ と $T(j)$ が二項関係 semiequivalent を満たす関係にあることを意味する．

Extended BM 法のアルゴリズムは，基本的には BM 法と同じである．すなわち，

> [Extended BM 法のアルゴリズム]
> まず，$i=1$ として，$P$ と $T[i:i+n-1]$ が対応するか，すなわち $P(\xi) \sim T(i+\xi-1)$ for all $\xi = 1, 2, \ldots, n$ が成り立つかどうかを調べる．オリジナルの BM 法と同じく，この照合は右から左の順，すなわち $\xi = n, n-1, n-2, \ldots$ の順に行う．照合が失敗したらそこで照合を打ち切り，照合が $\xi = 1$ まで成功した場合は，キーワードを発見したとして出力する．次に，次の $i$ 位置に対して照合を行うため，$i$ を skip($T(i+n-1)$) だけ動かす．$i+n-1 > m$ となったら照合を終了する．

Extended BM 法の BM 法との違いは，skip 関数の構成法である．理論的には，skip 関数は次の方法で構成される．すなわち，今回想定される skip 関数の定義域は，擬似コードの取り得るあらゆる値（文字列の例にならい，アルファベット $\Sigma$ と呼ぶことにする）であり，値域とし

text : (3,5) (1,3) (1,1) (4,2) (1,2) (3,1) (3,4)
pattern: (3,2) (3,5) (2,4)

skip($h_1$,$h_2$)=

| $h_1$ \ $h_2$ | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| 1 | 3 | 2 | 3 | 3 | 1 |
| 2 | 3 | 2 | 3 | 3 | 1 |
| 3 | 1 | 1 | 1 | 1 | 1 |
| 4 | 3 | 2 | 3 | 3 | 1 |
| 5 | 3 | 2 | 3 | 3 | 1 |

図 5　LSPC における skip 関数（配列サイズ $\mathcal{M}^L$）

skip($h_1$,$h_2$)= min( $skip_1(h_1)$, $skip_2(h_2)$)
where

| $h_1$ | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| $skip_1(h_1)$ | 3 | 3 | 1 | 3 | 3 |

| $h_2$ | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| $skip_2(h_2)$ | 3 | 2 | 3 | 3 | 1 |

図 6　LSPC における skip 関数（配列サイズ $\mathcal{M}L$）

図 7　実験に用いた新聞画像（一部）

図 8　切り出された文字の例

て 1 以上 $n$ 以下の自然数を持つ関数である．まず，すべての $x \in \Sigma$ について，skip($x$) を $n$ に初期化する．次に，擬似コード $P(n-i)$ に対し，それと semiequivalent になるようなすべての擬似コード $x \in \Sigma$ について，skip($x$) を $i$ に更新する．この手続きを $i = n-1, n-2, \cdots, 1$ の順に繰り返せば，所望の skip 関数が完成される（図 5）．

理論上は上記で可能だが，現実的にはこのアルゴリズムは問題がある．この方法では，skip 関数を格納するためにアルファベット $\Sigma$ のサイズに比例した配列サイズが必要だが，擬似コード LSPC は $L$ 個の整数を要素として持っている．各要素の値域は 1 から $\mathcal{M}$ までであるとすると，この場合必要な配列のサイズは $\mathcal{M}^L$ となってしまい，実際に $L$ の値として 20 程度から数百程度のものを用いようとすると，このサイズの配列を確保するのは現実的に不可能である．そこで次の定理が重要となる．

**定理 1.** 擬似コード $C(p) = (c_1(p)\ c_2(p)\ \cdots\ c_{d'}(p))$ に対する skip 関数は，次の形で表現できる．

$$\mathrm{skip}(C(p)) = \min(\mathrm{skip}_1(c_1(p)), \mathrm{skip}_2(c_2(p)), \ldots, \mathrm{skip}_{d'}(c_{d'}(p)))$$

ここで，$\mathrm{skip}_i(c_i(p))$ は，通常の BM 法におけるのと同じ方法で構成される skip 関数である．

**（証明）** skip($C(p)$) とは，$P(n-\xi)$ と $C(p)$ が semiequivalent になるような最小の整数 $\xi > 0$ である．

$\lambda < \min_{i=1,\ldots,d'}(\mathrm{skip}_i(c_i(p)))$ のとき，$P(n-\lambda)$ と $C(p)$ は semiequivalent にならない．なぜなら，$P(n-\lambda) \sim C(p)$ とすると，ある $\delta \in \{1, \ldots, d'\}$ について $P(n-\lambda)$ の第 $\delta$ 要素と $C(p)$ の第 $\delta$ 要素が一致するはずであるが，これは $\lambda < \mathrm{skip}_\delta(c_\delta(p))$ に反するからである．逆に $\lambda = \min_{i=1,\ldots,d'}(\mathrm{skip}_i(c_i(p)))$ のとき，ある $\delta$ について $\lambda = \mathrm{skip}_\delta(c_\delta(p))$ であり，これは $P(n-\lambda)$ の第 $\delta$ 要素と $C(p)$ の第 $\delta$ 要素が一致することに他ならないので，$P(n-\lambda)$ と $C(p)$ は semiequivalent である．よって $\mathrm{skip}(C(p)) = \min_{i=1,\ldots,d'}(\mathrm{skip}_i(c_i(p)))$ である．　□

この工夫により，記憶容量は $\mathcal{M}^L$ から $\mathcal{M}L$ に削減され，実装が現実的に可能となる（図 6）．

## 5. 新聞画像による実験

Extended Boyer-Moore 法の応用の一例として，OCR によるテキスト化が困難な古い新聞画像に対する全文検索実験を行い，従来法によるものと検索コストを比較する．

### 5.1 実験の概要

明治期の「函館毎日新聞」を対象とする．新聞資料はマイクロフィルムによって保存されており，これをスキャナで取り込みディジタル化したデータを用いる（図 7）．解像度は 1 文字あたりおおむね 70 × 70 ピクセル程度である．

なお，このような古い新聞画像に対する全文検索というのは決して容易なタスクではない．OCR によるテキスト化ができれば話は簡単だが，OCR は言語や書体に依存した技術であるため，現代とはフォントも語法も異なる明治期の画像に対して現代の OCR をそのまま適用しても十分な精度は得られない．実際，新聞画像に対して，OCR によらず画像検索技術を用いて全文検索を行うための研究は他にも行われている [9].

### 5.2 文字切出し

文字切出しは，段落の縦方向射影ヒストグラムによる行切出しと，段落の横方向射影ヒストグラムによる文字切出しとを組み合わせるという方法で行った．このような方法が採用可能なのは，本実験では新聞画像を対象としたため，段落画像における文字の配置がほぼ格子状になっているためである．

ただし，単純な方法をそのまま適用しただけではさま

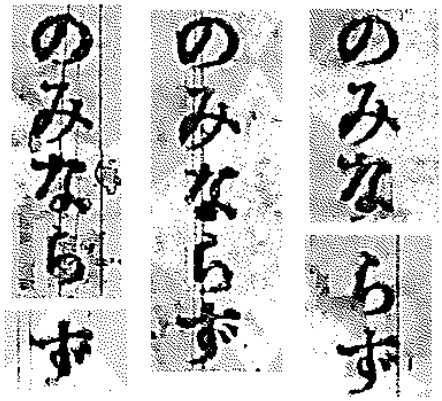

図 9　検索対象とした文字列の画像

ざまな問題が発生する．1つは，段落の最後の行に2,3文字しかない場合に行切出しが失敗しやすいということと，もう1つは，振り仮名（ルビ）が漢字の右側に多く振られており，これは文字のマッチングには用いないため，除去する必要があるということである．

そこで，自己相関関数を用いて本文の幅と振り仮名の幅を特定できるような形で行幅を推定し，上記2点の問題に対処することとした．

切り出された文字画像の例を図8に示す．

### 5.3　特徴量ベクトルおよび擬似コードの構成

特徴量は，勾配分布特徴量 [10] を用いた．文字切出しされた画像を $4 \times 4$ の小領域に分割し，あとはSIFT [11] と同じような方法で128次元の特徴量ベクトルを構成する．

次いで，この各特徴量ベクトルを擬似コードLSPCに変換する．使用する関数族（LSH family）としては，今回使用する特徴量ベクトルはノルムが1に正規化されているものであるので，SLSH [6] の orthoplex 法を用いた．今回使用する特徴量ベクトルはその要素がすべて非負であるという性質を持っているため，この場合1つのハッシュ関数の値域は $1 \sim 128$ となる．LSPC構成の際に設定するパラメタ $k$ および $L$（2.2節参照）については，Recallを優先する設定とPrecisionを優先する設定ということで，$(k, L) = (3, 26)$ と $(k, L) = (3, 24)$ の2種類を試した．前者は特徴量ベクトルがやや離れていても擬似コードがsemiequivalentになりやすいかわりに拾い漏らしが少ない設定（すなわちRecall優先の設定）であり，後者はその逆（すなわちPrecision優先の設定）である．

以上のような方法で，文字切出しされた各文字画像が128次元ベクトルに変換され，次いで擬似コードに変換されることにより，新聞画像が擬似コード列に変換されたこととなる．これに対してExtended Boyer-Moore法を適用することで，新聞画像に対する全文検索を行うことができる．

表 1　実験結果（$(k, L) = (3, 26)$）

| No. | Naive | Extended BM | | Recall | Precision |
|---|---|---|---|---|---|
| | (#comp) | (# comp) | (#skip) | | |
| 1 | 1342 | 713 | 379 | 2/2 | 2/6 |
| 2 | 1452 | 876 | 417 | 2/2 | 2/7 |
| 3 | 1430 | 871 | 426 | 2/2 | 2/9 |

表 2　実験結果（$(k, L) = (3, 24)$）

| No. | Naive | Extended BM | | Recall | Precision |
|---|---|---|---|---|---|
| | (#comp) | (#comp) | (#skip) | | |
| 1 | 1101 | 434 | 289 | 1/2 | 1/1 |
| 2 | 1151 | 545 | 322 | 1/2 | 1/2 |
| 3 | 1186 | 542 | 319 | 2/2 | 2/6 |

### 5.4　実　　験

今回実験で用いた新聞画像は，全755文字からなる．これに対し，手作業でテキスト化したデータを作成した後にn-gram解析にかけ，3回以上登場する文字列のうち最長の長さであった「のみならず」を実験対象に用いた（図9）．3箇所の「のみならず」に対し，それぞれをクエリとして残りの2つを探すという実験を行い，素朴な解法（naive method）と，Extended BM法の2つで，それぞれ文字照合回数を数え上げた．なおここでは比較を明確にするため，素朴な解法においても文字照合はパターンの末尾の文字から順に行った．

### 5.5　実 験 結 果

結果を表1および表2に示す．表中，Naiveは素朴な解法，Extended BMはExtended BM法（提案法）を表し，(#comp) は文字照合回数，(#skip) はExtended BM法における skip 関数の評価回数を表している．また，Recallは再現率，Precisionは適合率であり，それぞれ以下の式で定義される．

$$\text{再現率} = \frac{\text{検索された適合文字列の数}}{\text{全文中に存在する適合文字列の数}} \tag{5}$$

$$\text{適合率} = \frac{\text{検索された適合文字列の数}}{\text{検索された文字列の数}} \tag{6}$$

なお，表に掲げた素朴な解法よりもさらに素朴な方法として，文字列の照合中に不一致が見つかっても照合を打ち切らないアルゴリズムというものが考えられる．この場合の計算コスト（文字照合回数）は，いずれをクエリに用いた場合においても $751 \times 5 = 3755$ である．

表からわかるとおり，すべてのクエリに対して Extended BM 法 の文字列照合回数は，素朴な解法のものを下回っている．中でも，表1における1つめのデータと，表2におけるすべてのデータにおいては，文字列照合回数が全文字数 (755文字) を下回っており，サブリニア時間での検索を達成している．

ただし，Extended BM法においては，素朴な解法に対する追加コストとして skip 関数の評価回数を考えな

ければならない．文字パターン照合は $L$ 個の自然数の一致／不一致を判定する処理であり，skip 関数評価は $L$ 個の自然数の最小値を取る処理であるため，これらをの計算コストを完全に同一視することはできないが，仮に同じだとすると，Extended BM 法のコストは (#comp) に (#skip) を加えたものとして評価できる．この場合も，表のすべての場合において，Extended BM 法の計算コストは素朴な解法よりも削減できていることになる．

## 6. おわりに

本研究では，Boyer-Moore 法を拡張し，一般の実ベクトル系列に対して検索を行うことができるアルゴリズムとして Extended Boyer-Moore 法を提案した．また，その応用の一例として，OCR によるテキスト化が困難な古い新聞画像に対する全文検索実験を行い，従来法によるものよりも検索コストを削減できることを確認した．

今後の予定としては，今回実験に用いたテキストでは，3 回以上登場する文字列のうち最長のものでも 5 文字長でしかなかったので，実験にも長さ 5 のキーワードしか使えなかったが，BM 法は検索キーワードの文字数が長くなるほど効率的になるということを考えると，より長いキーワードで実験を行えばさらに大きい計算量の削減が示せる可能性がある．そのことを確認するため，実験対象の画像を増やし，より長いテキストに対して検索を行うことを予定している．また，それに伴い，検索パターンの文字列の種類も増やし，実験の精度も高める．さらに，必要なパラメタ調整に関する知見を蓄積することも今後の課題のひとつである．

また，BM 法は完全一致（Exact Matching）を見つける手法であるため，今回は応用対象として新聞画像をターゲットとしたが，今後はさらに不完全一致（Inexact Matching）を見つけるための効率的なアルゴリズムの開発を進めていく予定である．これを用いて，手書き文書の全文検索のようなより難しい課題に対しても，擬似コード LSPC による高速アルゴリズムが構築できることを示していく予定である．

## 謝　辞

本研究の一部は，独立行政法人科学技術振興機構さきがけ「擬似コード変換と統計解析による文書画像からの知識抽出」により進められている．

## 文　献

[1] R. S. Boyer and J. S. Moore, “A fast string searching algorithm,” Communications of the ACM, vol. 20, pp. 762–772, Oct. 1977.

[2] K. Terasawa and Y. Tanaka, “Locality Sensitive Pseudo-Code for Document Images,” Proc. 9th Int. Conf. on Document Analysis and Recognition, ICDAR2007, vol. 1, pp. 73–77, 2007.

[3] A. Gionis, P. Indyk, R. Motwani, “Similarity Search in High Dimensions via Hashing,” Proc. 25th Int. Conf. on Very Large Data Base, VLDB1999, pp. 518–529, 1999.

[4] M. Datar, P. Indyk, N. Immorlica, V. Mirrokni, “Locality-Sensitive Hashing Scheme Based on p-Stable Distributions,” Proc. 20th ACM Symposium on Computational Geometry, SoCG2004, pp. 253–262, 2004.

[5] A. Andoni, P. Indyk, “Near-Optimal Hashing Algorithms for Approximate Nearest Neighbor in High Dimensions,” Proc. Symposium on Foundations of Computer Science, FOCS’06, pp. 459–468, 2006.

[6] K. Terasawa and Y. Tanaka, “Sperical LSH for Approximate Nearest Neighbor Search on Unit Hypersphere,” Proc. 10th Workshop on Algorithms and Data Structures, WADS2007, LNCS4619, pp. 27–38, 2007.

[7] D. Gusfield, “Algorithms on Strings, Trees, and Sequences: Computer Science and Computational Biology,” Cambridge University Press, 1997.

[8] 北研二，津田和彦，獅々堀正幹，“情報検索アルゴリズム，” 共立出版株式会社，東京，2002 年 1 月

[9] Yue Lu and Chew Lim Tan, “Word spotting in Chinese document images without layout analysis,” Proc. 16th Int. Conf. on Pattern Recognition, ICPR2002, vol. 3, pp. 57–60, 2002.

[10] 寺沢憲吾，長崎健，川嶋稔夫，“勾配分布特徴量による高精度手書き文字検索，” 画像の認識・理解シンポジウム (MIRU)2006 講演論文集，pp. 1325–1330，2006.

[11] D.G. Lowe, “Distinctive image features from scale-invariant keypoints,” International Journal of Computer Vision, vol. 60, no. 2, pp. 91–110, 2004.