解　説

フルテキスト・データベースの技術動向†

小川隆一††　菊地芳秀††　高橋恒介††

1. はじめに

近年，出版工程の電子化，CD-ROM などの大容量記憶媒体の普及などにともない，大量の文書が電子化され，これを検索するためのフルテキスト・データベースが増えている．フルテキスト・データベースとは，文書検索において，いわゆる二次情報（題名，著者名，発行日などの書誌情報やキーワード，抄録）だけでなく，一次情報（本文全体）を記録し，これを参照できるデータベースのことをいう．現行のフルテキスト・データベースでは，検索者は二次情報（キーワードや書誌情報）を指定して抄録や本文を参照する，という検索方式をとるのが普通である．しかし，こうした方式では次のような問題が生じる．

① 大量の全文データにキーワードを付加する手間が大変である．これを簡略に行うため，抄録や本文から索引語を切り出す自動インデクシング方式[踏橋 84]やハードウェアが開発されているが[福島 91]，自動インデクシングに用いる辞書の質が高いことが必要であり，大規模システムでないと適用しづらい．また，辞書の不断の保守も必要である．

② キーワードなどの二次情報について，データベースの更新・保守作業が必要である．特に本文情報の更新が頻繁で，かつ大量の索引語を本文から抽出して利用する文献検索システムなどではこの負担が大きくなる．

これに対し，二次情報をあらかじめ準備せず，検索者が自由に指定するキーワードをもとに，本文を直接参照して照合を行い，一致したものを提供する方式（フルテキストサーチ方式）が注目されている．フルテキストサーチ方式は，理論的にはすでに 20 年前から検討されているが，近年の高速サーチエンジンの LSI 化など，ハードウェア技術の進歩により，がぜん現実味を帯びてきた．一方また，ハイパテキストのような，一次情報に直接アクセスする文書検索インタフェースが注目されており，フルテキストサーチ技術の応用の可能性が広がりつつある．

本稿では，フルテキスト・データベースの意味を，通常考えられているものよりも多少狭めて「フルテキストサーチ機能を備えたデータベースシステム」ととらえ，フルテキスト・データベースの要素技術とその応用について解説する．**2.** ではまず，フルテキストサーチ技術の流れについて簡単に述べる．**3.** では，ソフトウェア技術，特にテキスト照合アルゴリズムについて解説する．**4.** では，ハードウェア技術，特にサーチエンジンについて解説する．**5.** では，これらを統合したシステム化技術について述べる．**6.** では，フルテキスト・データベースの今後の発展方向として，近年注目を集めているハイパテキストシステムへの応用について述べる．なお，フルテキスト・データベースとして，テキストだけでなく，それに付随する図表などを検索する事例もあるが[根岸 89]，本稿ではひとまず検索対象をテキストに限定する．

2. フルテキストサーチ技術の流れ

フルテキスト・データベースは，関係データベースのように論理レベルで厳密に定義された概念ではなく，その要素技術も必ずしも系統的に発展してきたものとはいえない．むしろ，フルテキストサーチの実現に際しては通常のデータベース技術のほかに，ハードウェア／アルゴリズム／情報検索／テキスト処理など，異なる分野で発展した種々の要素技術が総合的に必要とされてきた．

キーワードとして指定された文字列と一次デー

† Recent Developments in Full Text Database Technologies by Ryuichi OGAWA, Yoshihide KIKUCHI and Kousuke TAKAHASHI (NEC Corporation).

†† 日本電気(株)

タ（テキスト）を高速に照合することは，フルテキスト・データベースとほかのデータベースの最大の相違点であり，技術的な特徴でもある．このための要素技術としては，①テキスト照合を高速に行うサーチアルゴリズム，②サーチアルゴリズムを実装した高速なハードウェア（LSI 化）の二つが重要である．歴史的には，サーチアルゴリズムの研究がまず発展し，1970 年代後半には，現在広く利用されている照合アルゴリズムが確立した[竹田 89]．1980 年代に入り，上記アルゴリズムを並列，高速に行う LSI の開発が進み[高橋 89]，高速なフルテキストサーチが現実に可能になった．この LSI を含むフルテキストサーチの処理系はサーチエンジンと呼ばれることが多い．近年はサーチエンジンをベースとするシステム化技術の開発が進められている．

一方，フルテキストサーチとキーワードサーチの中間に位置する検索手段として，全文テキストを一定の方式で圧縮したシグネチャファイルを作成し，これを対象としたパターン照合を行う方式も検討されている[Fal 85]．シグネチャファイルは必ずしも効率的な検索方式ではないが，フルテキストサーチの負荷を軽減するため，全文データを圧縮した一種のシグネチャファイルを利用する検索方式（たとえば [加藤 91]）が今後増えてくることが予想される．

3. テキスト照合アルゴリズム

ここでは，2. で述べたテキスト照合アルゴリズムのうち，代表的な 3 種のアルゴリズムについて紹介する．

3.1 Knuth-Morris-Pratt アルゴリズム[Knuth 77]

長さ n のテキスト τ に対して，τ の第 i 番目から第 j 番目までの文字で構成される部分文字列を $\tau[i:j]$，第 k 番目の文字を $\tau[k]$ で表す．これと照合したい m 文字の文字列（パターンと呼ぶ）を π で表す．また，τ と π が一致することを，$\tau[i:j]=\pi[p:q]$ $(j-i=q-p)$ で表す．単純素朴に考えると，τ と π の照合アルゴリズムは以下のようになる．

```
begin
  for i=1 to n−m+1 (i をパターン照合のピ
  j=1                ボット位置とする)
    while τ[i+j−1]=π[j] かつ j<(m+1)
      j=j+1
    do
    if j=(m+1)
      i はパターン出現位置である
    end
  end
```

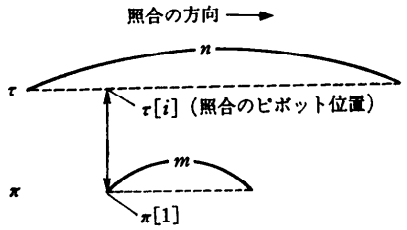


図-1 τ と π の比較

ピボット位置 i の変化は，図-1 に示すように，τ に対して π を相対的にシフトして照合することと等価である．計算中の文字列比較回数は，最悪の場合 $m\cdot(n-m+1)$ で，計算量は $O(n\cdot m)$ のオーダとなる．これはもちろん，各照合位置での照合において，τ と π の同じ文字への参照が繰り返されることに起因する．

最悪の場合でも上記の計算が n または m の線形時間で行えることを最初に示したのが Knuth-Morris-Pratt のアルゴリズム (KMP 法) である．KMP 法の基本的なアイデアは，照合において文字列の不一致，あるいはパターンの出現を検出した場合に次のピボット位置 i をどれだけずらすかを前もって計算し，一致しないことが分かっている照合処理を省く，というものである．KMP 法では，τ,π いずれの文字列も 1 方向の走査だけで参照処理がすみ，最悪時の計算量を $O(m+n)$ におさえることができる．KMP 法は実用的にはそれほど重要とされていないが，テキスト照合問題が理論的に線形時間で解けることを最初に示した意義は大きく，以後のテキスト照合アルゴリズム研究の基礎ともなる重要な貢献であった．

3.2 Boyer-Moore アルゴリズム[Boyer 77]

KMP 法が最悪時の計算量を線形時間に減少させたのに対し，最良の場合の計算量を削減できるのが Boyer-Moore のアルゴリズム (BM 法) である．BM 法では τ,π の文字比較において，文字列の右から左へ向かって照合を行う．文字列の不一致が起きた場合には，以下の二とおりの方法で

照合位置をシフトする.

① τ, π の最初の照合文字が不一致である場合，不一致文字 $\tau[k]$ と同じものが π の左側にあれば，その文字どうしを合わせるように照合位置をシフトする．すなわち，$\tau[k]=\pi[m-s]$ であるような最小の s を求め，シフトの大きさとする.

② τ, π の右から $(j-1)$ 文字が一致し，j 文字目が不一致の場合，一致した部分文字列と同じものが π の左側にあれば，それが重なるように照合位置をずらす．すなわち，$\pi[m-j+1:m]=\pi[m-j+1-s:m-s]$，かつ $\pi[m-j-s]\neq\pi[m-j]$ となる s のうち最小の値を求め，シフトの大きさとする(**図-2** 参照).

①②いずれの場合も，一致する文字（文字列）が π の左側にない場合には，π が周期性をもつと考えて $s=m$ とする．s の値は文字列照合を行う前に一括して求めればよく，計算量は $O(m)$ である．BM 法全体の計算量は，テキスト中の文字がパターン中に現れる確率が小さいときは平均的に $O(n/m)$ となり，パターン長が大きくなれば計算時間が短くなるという特徴をもつ．BM 法は最も高速な照合アルゴリズムとされ，種々の改良が試みられている[竹田 91].

3.3 Aho-Corasick アルゴリズム[Aho 75]

Aho-Corasick のアルゴリズム (AC 法) は，有限オートマン (FSA) をベースに KMP 法を拡張したもので，1回の文字列走査で複数のパターンを照合できる．AC 法は，①検索キーとなるパターンをもとにパターン照合用のオートマトンを生成するアルゴリズム，②これを用いてテキストを走査する照合アルゴリズムから構成される．オートマトン生成において，AC 法では各状態間の遷移を表す goto (照合に成功)，failure (照合に失敗) の2種の関数を作成する．たとえばパターンの集合 {he, she, his, hers} が与えられたとき，

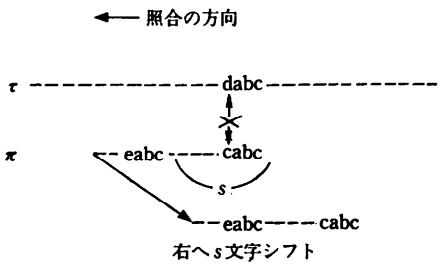


図-2 照合位置のシフト

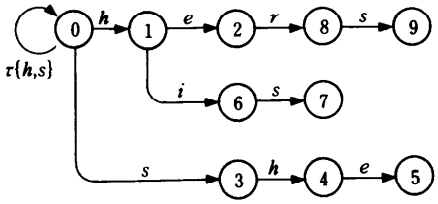


図-3 パターン照合オートマトン

goto 関数により**図-3** に示すようなオートマトンが生成される．failure 関数は，オートマトンのある状態で不一致が起こった場合，次の照合処理を最初からでなく，途中の状態から行えるように遷移を決定するもので，KMP 法，BM 法の照合位置シフトと同等の処理である．これらの計算量は，各検索キーの長さの和のオーダである.

パターン照合においては，入力テキストを goto 関数で一文字ずつ比較していき，たとえば状態4で不一致となると状態1へとび，そこから照合を続ける．あるパターンの照合に成功すると，output 関数により成功したパターンを出力する．AC 法では状態が分岐する際，二つの状態遷移を交互に追跡する手間が増えるが，どちらかは必ず遷移に失敗するので1文字比較が余分に増えるだけで，計算量はテキスト長のオーダ $O(n)$ である.

AC 法は，有川などが実用レベルのフルテキスト・データベースに採用している[有川 89]．有川はまた，1バイト文字と2バイト文字とが混在する日本語テキスト照合のために本アルゴリズムを拡張し，各文字を4ビット単位で分割して照合する方法を提案している．また，これを一般化し，複数の文字種を含むテキスト照合アルゴリズムも提案されている[竹田 89].

4. テキスト照合ハードウェア

フルテキストサーチにおけるハードウェアアーキテクチャの研究は，記号処理技術の研究に関連して古くからなされていた[山本 82]．このころ，すでに知られていた方式として，並列照合方式，セルラアレイ方式，有限オートマトン方式などがある[Hol 79]．最近では，LSI 製造技術の進歩にともない，これらを組み合わせたプログラマブル順序論理方式やダイナミックプログラミング方式が新たに提案され，VLSI 化されている.

4.1 並列照合方式

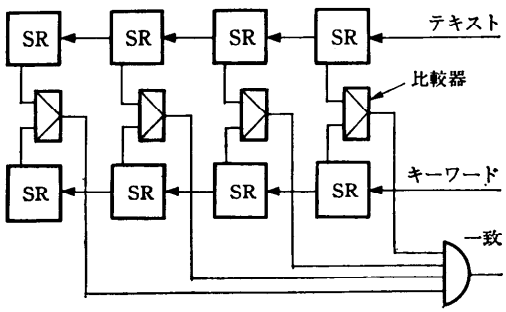
比較器を多数並べて，テキストと検索文字列の照合を並列に行う方式である．**図-4** に動作原理を示す．各セルの一致信号は同時に積が取られて全体の一致結果となる．一般的には回路規模を小さくするために，レジスタと比較器の代わりに連想メモリ (Associative Memory) が用いられる．

並列照合方式はハード化しやすいという利点がある反面，①VLDC (Variable Length Don't Care) 文字* が扱えない，②使用ハード量が多い，などの課題も抱えている．

4.2 セルラアレイ (CA) 方式

セルラアレイ (Cellular Array) 方式は，1文字単位の比較器とロジックが対になったセルをアレイ状に配した照合方式である．各セルから出力される一致信号とそれまでの一致結果との積を順番にとることでまとまった文字列の照合を行う．この方式は各セルに供給するテキストの与え方の違いにより，さらにブロードキャスト型とシストリックアレイ型とに分かれる．

ブロードキャスト (Broadcast) 型の動作原理を**図-5** に示す．入力されるテキストは1文字ごとに全てのセルへ一斉に入力され，比較が行われる．



SR: Shift Register

図-4 並列照合方式の動作原理

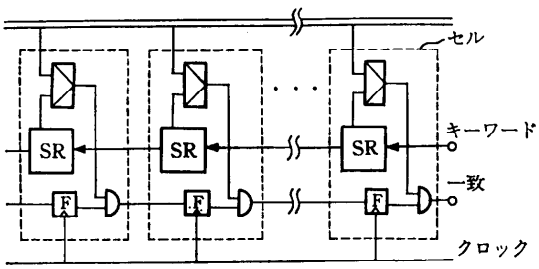


図-5 ブロードキャスト型の動作原理

* VLDC 文字：任意長の文字列と一致する機能を持つ文字．

この一致信号と前のセルの一致信号との積がとられ，次のセルに伝搬される．最後のセルから出る一致信号が全体の一致結果を表す[Mul 79]．

一方，シストリックアレイ (Systolic Array) 型では，一致信号のセル間の伝搬がない．そのかわり，テキストおよび検索文字列の双方が向かい合ってセル間を移動しながら照合が行われ，一致信号は前回の一致信号との AND が取られて再び各セルに蓄えられる．**図-6** に動作原理を示す．テキストおよび検索文字列は一つおきに入力され，テキストと検索文字が交差したときに照合が行われる[Fos 80]．

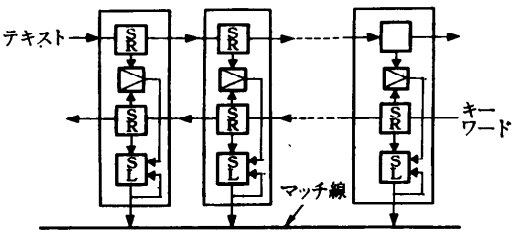
4.3 有限状態オートマトン (FSA) 方式

有限状態オートマトン (Finite State Automaton) 方式は，テキストの入力によって状態を遷移させながら照合を行う方式である．状態遷移をエミュレートする機構としては，状態番号と入力文字コードで表された2次元の状態遷移表を格納するRAMと，RAM アクセスのための周辺回路があればよい[速水 86]．2次元の状態遷移表をRAMに格納した場合，膨大なメモリ量が必要になるが，連想メモリを使うことによりハードウェアの量を抑えることができる．

また，FSA で VLDC 文字を扱う場合，そのままだと非決定性 FSA になるため状態遷移先が複数個となる．Haskin はこれに対し，状態遷移先が一つになるように状態遷移表を分割する方法[Has 80]を提案している．

FSA 方式を用いると，正規表現を用いた複雑なパターンの検索が可能となるが，反面，遷移関数を格納する領域が大きい，前処理時間がかかる，VLSI 化しづらい，などの欠点もある．

これに対し，伊藤らは Aho-Corasick 法[Aho 75]において goto 関数を用いない方式を提案し，メモリ使用量を抑えているほか，input_control 関



SL: Sequential Logic

図-6 シストリックアレイの動作原理

表-1 主な検索ハードウェア

名称	PF 474[Yia 83] (Proximity Tec.)	SHP[速水 86] (NTT)	ISSP[高橋 87] (日本電気)	S-SE[加藤 89] (日立)	SDP[吉野 91 b] (松下)	DISP[本村 90] (日本電気)	GESCAN (GE)
規模	VLSI	ボード	VLSI	ボード	ボード	VLSI	VLSI
方式	DP 法	有限状態オートマトン	連続メモリ＋有限状態オートマトン	有限状態オートマトン	有限状態オートマトン	連想メモリ＋セルラオートマトン	不明
速度	約 50 Kbyte/秒	不明	10 M 文字/秒	20 Mbyte/秒	16.7 Mbyte	33 M 文字/秒	60 Mbyte/秒
特徴	曖昧一致，商品化		曖昧検索	1000単語同時検索	正規表現	曖昧検索	商品化済

数，compare 関数により VLDC 文字列検索時の効率化を図っている[伊藤 88].

4.4 プログラマブル順序論理方式

これは，連想メモリと有限状態オートマトンを組み合わせ，オートマトンの状態遷移を連想メモリの照合結果で制御する方式である[高橋 87]. 状態遷移表は，限定してハードウェア化されている. このため正規表現のような複雑なパターンは扱えないが，FSA 方式で問題となる前処理時間をなくしているほか，連想メモリの課題であった VLDC 文字の取扱いを可能にし，一文字の欠けや混入も扱えることが特徴となっている.

4.5 ダイナミックプログラミング (DP) 方式

あいまいな文字列の検索を行う方法の一つに，文字列間の距離を計算する方法があり，この距離計算にダイナミックプログラミング (Dynamic Programming) 法を用いる方式が知られている. DP 法のハードウェア化には，2次元のシストリックアレイを利用する方法[Che 87] と，セルラオートマトンを利用する方法[本村 90]が提案されている.

これまでに発表された主な検索ハードウェアの概略を表-1 に示す. なお，表には載せていないが，マルチプロセッサシステムに並列パターンマッチングアルゴリズムをインプリメントしたものも報告されている[堀口 89]. 計算機のマルチプロセッサ化が進む中，現実的なアプローチの一つであると言える.

5. フルテキストサーチのシステム化技術

前章までで，フルテキストサーチの基本となる要素技術について述べたが，フルテキストサーチ技術が使いやすい検索システムとなるには，検索ハードウェアのほかに，①システムとしての検索速度の向上，②検索精度の向上，③ユーザインタフェースの向上，などが必要となる.

①に関しては，検索範囲の絞り込みと検索テキストの読み出し速度の向上による高速化が検討されている. ②に関しては，必要な情報をなるべく網羅し，不必要な情報（いわゆるゴミ情報）をなるべく削減する，という二つの方向からの精度向上が検討されている. 情報検索において上記二つの評価尺度は，それぞれ再現率（必要な情報のうち，実際に検索されたものの割合），適合率（実際に検索された情報のうち，必要なものの割合）と呼ばれる. ③に関しては，自然言語による検索条件文から検索を行う試みがなされている.

これらを検索の手続き順に整理すると，次のような技術となる.

（1） 検索条件文からの検索式の生成
（2） 検索式の展開（再現率の向上）
（3） 検索の加速化（検索対象の絞り込み）
（4） 被検索テキストの高速入力
（5） 検索結果の適合性のチェック（適合率の向上）

5.1 検索条件文から検索式の生成

従来の検索では，大部分が検索条件として検索式を与える方式を採っている. 理想的には自然言語で入力した検索条件に対し，計算機が検索の意図を理解した上で検索を行ってくれることが望ましいが，自然言語理解や検索意図の適切な表現などの面で課題が多く，実用に至ってはいない. 現在，最も実用的なアプローチの一つとして，自然言語で書かれた検索条件文を形態素解析し，検索式と呼ばれる特殊な記号や文法で書かれた検索条件に変換する方法が試みられている[吉野 91 a].

5.2 検索式の展開（再現率の向上）

検索式の中で与えられたキーワードだけでは，検索の意図が十分に反映されないことも多い. フルテキストサーチの場合，たとえば，「にほん」をキーワードとして検索しても「にっぽん」が検索できないなど言葉のゆらぎに起因する問題や，

「計算機」をキーワードとして検索しても「コンピュータ」が検索できないなど表現の違いによる再現率の低下の問題がある．この問題を解決する手段として，シソーラスと呼ばれる単語の同義語・下位語などが記述された辞書を用いてキーワード展開を行ったり，変換ルールを用いてキーワードの展開を行うなどの方法が試みられている[島山 89],[菅野 91a],[巌寺 90]．

5.3 検索の加速化

現在のフルテキスト検索ハードウェアの速度は最大でも数十Mbyte/秒であるため，大容量のテキストを始めから終わりまで検索するとかなりの時間がかかってしまう．このため，検索部分の絞り込みに関する研究がなされている[加藤 89]．加藤らは，あらかじめ「文字成分表」と「凝縮本文」とを作成しておく方式を提案している．「文字成分表」とは，検索対象となる文書に含まれる文字の有無を示す表であり，キーワード中の文字が含まれない文書を検索対象から外すのに用いる．「凝縮本文」は本文から助詞などの不要語を除いたものであり，キーワードが含まれるか否かを判別するのに用いる．この二つの方式により，検索速度を等価的に 100 Mbyte/秒まで加速している．ただし，これらは本文と別に作成する必要があるため，作成にかかる時間や保管のための記憶容量の増加などの課題もある．

5.4 被検索テキストの高速入力

フルテキストサーチの実験システムではテキストをメモリ上において検索することが多いが，実用規模のシステムでは大容量蓄積装置からの検索が必要になる．現在，大容量蓄積装置としてはハードディスクが一般的に用いられているが，読み出し速度が遅く，開発されているフルテキストサーチハードウェアの検索速度に比べ，数分の1から数十分の1以下である．この問題に対し，①シリコンファイルなどの大容量高速ファイルメモリの利用（メモリの大容量化による高速化）[高橋 89]，②ディスクのアレイ化（ディスクの並列化による高速化）[金子 89],[加藤 89],[菊地 90]，などが検討されている．①に関してはコストの面から実用に至ってはいない．一方，②に関しては1980年代以降，ディスクの高速化／大容量化技術の進歩に歩調を合わせて進展しており，すでに実用化の時期に入ってきている．

ディスクアレイと検索ハードウェアの結合には，ホスト計算機のシステムを有効利用するためにディスクアレイとフルテキストサーチハードウェアを直結する方式[加藤 89]と，ディスクの制御のしやすさを優先させてシステムバスを介す方式[菊地 90]とがある．

5.5 検索結果の適合性のチェック（適合率の向上）

フルテキストサーチでは単純に文字列の照合を行うだけなので，その検索結果にはかなりの「ゴミ」が含まれることがある．この問題に対し，検索結果を形態素解析することで，誤った照合を削除する例が菅野らによって報告されている[菅野 91a]．これにより，次のような「ゴミ」が削除可能となっている．

- 単語になっていない場合
- 品詞が異なる場合
- 順序が異なる場合
- パターンが異なる場合

適合率の向上に関しては，言語処理技術だけでなく，知識処理技術を併用することにより，今後のブレークスルーが期待されるところである．

6. フルテキストサーチの応用とその将来

6.1 ハイパテキスト技術への接近

ここでは，全文データのような非構造情報を管理するもうひとつのパラダイム「ハイパテキスト」に焦点をあて，フルテキストサーチとの統合の可能性を探ってみる．ハイパテキストシステムの実用化に関連する最近の動向として，

① Dexter[Molin 90]，HyTime[Bert 90]などの標準参照モデルの提案

② マークアップ言語（たとえば SGML）によるリンク記述

③ リンク自動生成／ハイパテキスト自動生成の試み[Raym 87],[Salton 89],[Sarre 91]

などがある．これらは，ツール先行で個々ばらばらに発達してきたハイパテキストの仕様をある程度標準化し，ハイパテキスト間，あるいはハイパテキストと通常ドキュメントとのコミュニケーションをより簡単にしよう，という傾向を示している．この結果として，フルテキストサーチとハイパテキストの自動生成やリンク検索が，技術的に等質なものになりつつある．

6.2 統合の可能性

フルテキスト・データベースでハイパテキストを管理する場合，最も大きなメリットは，一次情報検索の高速化，およびリンク生成／管理の負荷軽減である．たとえば，特定の文字列をリンクキーとして，これを含む文章を高速に全文検索できれば，あらかじめリンクを定義していなくても関連ドキュメントをすぐに検索できる．これは仮想リンクと呼ばれ[Halas 88]，データベースの問合せを仮想リンクとしたり[平山 90],[Foun 90]，重み付けされたキーワード（キーワードコネクション）検索によりリンクをたどる方式[小川 91]などが試みられている．フルテキストサーチを用いれば，さらに高速で柔軟な仮想リンク検索が実現できるだろう．

一方，ハイパテキストシステムが提供するナビゲーション（リンクとなる文字列を指定して，関連する文書を次々と検索する）機能やすぐれたWYSIWYG (What you see is what you get) インタフェースは，全文データ利用に欠かせないものであろう．この意味で，フルテキスト・データベースとハイパテキストシステムの発展は表裏一体をなしていると言ってもよく，今後のハイパテキストアプリケーションの普及に期待がかかる．

6.3 SGML とオーサリング

ハイパテキストとの融合において重要な鍵となるのが，標準マークアップ言語 SGML である[芝野 89]．SGML は，文書フォーマット記述用の通常マークアップに参照関係(リンク)を規定するマークアップ仕様を加え，ハイパテキストと通常文書とを統合した標準化仕様をめざしている．マークアップは一種の仮想リンクであり，フルテキストサーチでこれを高速に検索できれば，標準ハイパテキストの普及に大きな力となるであろう．

しかしながら，標準化仕様を決めるだけでは実際の普及は難しい．SGML ドキュメントを簡単に作成／管理できるオーサリング環境が必要である．一方，ハイパテキスト自動生成やリンク自動生成の試みは，現状では評価があまり芳しくなく，ハイパテキスト作成も結局は人手を介さざるをえない．この点でも，オーサリング環境の充実が急務である．オーサリング環境はまた，フルテキスト・データベース自体の構築にも重要であることは言うまでもない．

6.4 マルチメディア化への対処

マルチメディアを含むハイパメディアドキュメントにフルテキスト・データベースがどう対処するかも，今後問題であろう．SGML を拡張した HyTime のように，時間に依存するメディア（音声／動画）までマークアップで統一的に記述すれば，フルテキスト・データベースの土俵でマルチメディアを扱う，という可能性も考えられなくはない．あるいは PostScript の記述をテキストとみなして，全文検索してもよいわけである．しかし，マルチメディアデータの標準化やオーサリングが大きなネックとなり，この方式が急速に普及するとは思われない．フルテキスト・データベースは，当面はハイパメディアの中でもかなり抽象化／標準化されたドキュメントを扱う管理システムとして機能していくと思われる．

7. むすび

フルテキスト・データベースの技術動向について，ソフトウェア／ハードウェアの両面から概観し，今後の応用について，特にハイパテキストシステムとの統合の可能性を検討した．ドキュメントのような非定型情報を管理／検索する枠組みとしては，このほかにもオブジェクト指向データベースや非正規関係データベースがある．マルチメディアドキュメントや，ほかの定型情報を同時に扱いたい場合にはこれらが有効だろう．一方で，すでに電子化された，あるいはワードプロセッサで量産される全文データをてっとり早く検索したい，という要求を満たすには，フルテキスト・データベース／ハイパテキストシステムの普及が待たれる．将来的には，ハイパメディアの名のもとに，これらのデータベースが統合的に利用できるようになるであろう．フルテキスト・データベースがそこでどのような位置づけになるか，ほかのデータベースとの協調をどうするか，また新たな議論が必要となるだろう．

参考文献

[有川 89] 有川，篠原，宮原，武谷，宮野，竹田，大島，白石，酒井，山本：テキストデータベース管理システム SIGMAM とその応用，情報処理学会研究会報告，Vol. 89, No. 66, FI-14-7 (July 1989).

[伊藤 88] 伊藤，早川他：ストリームデータプロセッサ SDP (1)，(2)，第 37 回情報処理学会全国大会

予稿, pp. 113-116 (1988).
[巌寺 90] 巌寺, 木本: 動的シソーラスを用いた連想検索, 自然言語処理研究会報告, 76-9 (1990).
[小川 91] 小川, 森田, 金矢: 動的リンク機能を有するハイパーテキストシステム, 第 42 回情報処理学会全国大会予稿, pp. 4-179〜180 (1991).
[加藤 89] 加藤, 藤澤, 大山, 川口, 畠山: 大規模文書情報システム用テキストサーチマシンの研究, 情報処理学会研究会報告, Vol. 89, No. 66, FI-14-6 (July 1989).
[加藤 91] 加藤, 藤澤, 大山, 川口, 畠山, 兼岡, 秋沢: 大規模文書データベース用テキストサーチマシンの開発, 1991 年情報学シンポジウム予稿集, pp. 97-106 (1991).
[金子 89] 金子: 磁気ディスク高速化技法, 信学技報, Vol. 89, No. 335, DE 89-36, pp. 1-7 (1989).
[菊地 87] 菊地, 宮井: ISSP を用いたテキスト検索システムの試作, 第 35 回情報処理学会全国大会論文集, pp. 1285-1286 (1987).
[菊地 90] 菊地, 杉本, 辻澤: 高速直接検索システム, 1990 年度信学会秋期全大, D-175 (1990).
[芝野 89] 芝野: SGML と全文データベース, 情報処理学会研究会報告, Vol. 89, No. 66, pp. 2.1-2.8 (1989).
[菅野 91 a] 菅野, 安藤, 伊藤他: フルテキストデータベースの技術動向, 信学技報, Vol. 90, No. 478, DE 90-34, pp. 31-40 (1991).
[菅野 91 b] 菅野他: ワークステーション内蔵型フルテキストデータベースプロセッサ SDP, 情報処理学会研究会報告, Vol. 91, No. 86, ARC-90-8, pp. 55-62 (Oct. 1991).
[高橋 87] 高橋, 永井, 山田, 平田: ストリング・マッチング・ハードウェア, 信学技報, CPSY 86-57 (Feb. 1987).
[高橋 89] 高橋, 山田, 本村: フルテキストサーチのハードウェア技術について, 情報処理学会研究会報告, Vol. 89, No. 66, FI-14-5 (July 1989).
[竹田 89] 竹田: 固定文字列と文字種の混在するパターンを対象とした Aho-Corasick 型パターン照合機械の構成法, 九州大学大型計算機センター, 計算機科学研究報告第 6 号, pp. 29-51 (1989).
[竹田 91] 竹田: 全文テキスト処理のための高速パターン照合アルゴリズム, 1991 年情報学シンポジウム予稿集, pp. 85-96 (1991).
[根岸 89] 根岸: フルテキスト・データベースの実用化における諸問題, 一学術情報センターでの事例を踏まえて一, 情報処理学会研究会報告, Vol. 89, No. 66, FI-14-1 (July 1989).
[畠山 89] 畠山, 川口, 加藤他: 自由語検索のための同義語異表記展開方式, 第 39 回情報処理学会全国大会論文集, pp. 1077 (1989).
[速水 86] 速水, 井上: サーチプロセッサの設計と評価, データベースシステム研究会報告, 51-2 (1986).
[平山 90] 平山, 西川, 難波, 田中: リンク定義言語を有するハイパーテキストシステム: TexlLink-Ⅲ, 情報処理学会データベースシステム研究会, No. 78-7, pp. 63-70 (1990).
[福島 91] 福島: 形態素抽出マシン MEX-II, 情報処理学会研究会報告, No. 81, 91-NL-81-1 (1991).
[堀口 89] 堀口, 小川, 木村: 並列ストリングパターンマッチングアルゴリズムとインプリメンテーション, 情報処理学会研究会報告, アルゴリズム 5-2 (1989).
[本村 90] 本村, 豊浦, 平田, 大岡, 山田, 榎本: 120 万トランジスタ辞書検索プロセッサ, 信学技報, ICD 90-2 (Apr. 1990).
[諸橋 84] 諸橋: 自動索引付け研究の動向, 情報処理, Vol. 25, No. 9, pp. 918-925 (Sep. 1984).
[山田 87] 山田, 平田, 永井, 高橋: 文字列検索 LSI, 信学技報, CAS 87-25 (May 1987).
[山本 82] 山本, 梅村, 小長谷, 横田: 文字列処理とアーキテクチャ, 情報処理, Vol. 23, No. 8, pp. 719-729 (Aug. 1982).
[Aho 75] Aho, A. and Corasick, M.: Efficient String Matching: An Aid to Bibliographic Search, Communications of the ACM, Vol. 18, No. 6, pp. 333-340 (June 1975).
[Bert 90] Bertsche, S. Editor: X 3 V 1. 8 M SD-6 Hypermedia/Time-based Document (HyTime) and Standard Music Description Language (SMDL) User Needs and Functional Specification, ANSI V 1. 8 M SD-6 (Apr. 1990).
[Boyer 77] Boyer, R. and Moore, J. S.: A Fast String Searching Algorithm, Communications of ACM, Vol. 20, No. 10, pp. 762-772 (Oct. 1977).
[Che 87] Cheng, H. D. and Fu, K. S.: VLSI Architectures for String Matching and Pattern Matching, Pattern Recognition, Vol. 20, No. 1, pp. 125-141 (1987).
[Fal 85] Faloutsos, C.: Access Method for Text, Computing Surveys, Vol. 17, No. 1, pp. 49-74 (Mar. 1985).
[Fos 80] Foster, M. J. and Kung, H. T.: The Design of Special Purpose Chips, IEEE Computer, Vol. 13, No. 1, pp. 26-40 (Jan. 1980).
[Foun 90] Fountain, A., Hall, W. et al.: MICRO COSM: An Open Model for Hypermedia with Dynamic Linking, Hypertext: Concepts, Systems and Applications, Cambridge University Press, pp. 298-311 (Nov. 1990).
[Halas 88] Halas, F.: Reflections on Note-Cards: Seven Issues for the Next Generation of Hypermedia Systems, CACM, Vol. 31, No. 7, pp. 836-852 (July 1988).
[Has 80] Haskin, R.: Hardware for Searching Very Large Text Databases, SIGIR, Vol. 15, No. 2, pp. 49-56 (Mar. 1980).
[Hol 79] Hollaar, L. A.: Text Retrieval Computers, Computer, Vol. 12, No. 3, pp. 40-50 (Mar. 1979).
[Knuth 77] Knuth, D., Morris, H. J. and Pratt, V. R.: Fast Pattern Matching in Strings, Technical Report STAN-CS-74-440, Sci. Dep., Stanford University (Aug. 1974).
[Molin 90] Moline, J., Benigni, D. and Baronas, J. Eds.: Proceedings of the Hypertext Standardization Workshop, National Institute of Standards

and Technology (Jan. 1990).
[Mul 79] Mulkhopadhyay, A.: Hardware Algorithms for Nonnumeric Computation, IEEE Trans. Comp., Vol. C-28, No. 6 (June. 1979).
[Raym 87] Raymond, D. and Tompa, F.: Hypertext and the New Oxford English Dictionary, Proc. of Hypertext '87. pp. 143-153 (Nov. 1987).
[Salton 89] Salton, G.: Automatic Text Processing, Addison Wesley, Reading, Mass. (1989).
[Sarre 91] Sarre, F. and Guntzer, U.: Automatic Transformation of Linear Text into Hypertext, Proc. of International Symposium on Database Systems for Advanced Applications, pp. 498-506 (Apr. 1991).
[Yia 83] Yianilos, P. N.: A Dedicated Comparator Matches Symbol Strings Fast and Intelligently, Electronics, Dec. 1 1983, pp. 113-117 (Dec. 1983).

(平成3年10月14日受付)

小川 隆一 (正会員)

1957年生. 1981年東京大学理学部地球物理学科卒業. 1983年同大学院修士課程修了. 同年日本電気(株)入社. 画像情報システム, ハイパーメディアシステム, 英語ヒアリング学習システムなどの研究開発に従事. 1989～1990年米国メリーランド大学計算機科学科訪問研究員. 現在日本電気(株) C&C情報研究所情報応用研究部主任. 電子情報通信学会会員.

菊地 芳秀 (正会員)

1961年生. 1984年東京工業大学電子物理工学科卒業. 1986年同大学院理工学研究科(電気電子工学専攻)修士課程修了. 同年日本電気(株)入社. C&C システム研究所を経て, 現在機能エレクトロニクス研究所勤務. マルチメディア, 検索システム等の研究開発に従事.

高橋 恒介 (正会員)

1940年生. 1964年慶應義塾大学理工学部計測工学科卒業. 1966年同大学院修士課程修了. 工学博士. 同年日本電気(株)入社. 磁性膜メモリデバイス, 高速ファイルメモリシステム, テキスト検索, VLSI アルゴリズムの研究. 現在は同 C&C システム研究所主管研究員. IEEE, 電子情報通信学会, 電気学会, 応用磁気学会など各会員.