(19)日本国特許庁(JP)　(12)公表特許公報(A)　(11)特許出願公表番号
特表2011-511366
(P2011-511366A)
(43)公表日　平成23年4月7日(2011.4.7)

(51)Int.Cl.		F I			テーマコード（参考）
G06F 17/30	(2006.01)	G06F	17/30	210A	5B075
		G06F	17/30	170A	

審査請求 未請求　予備審査請求 未請求　（全 31 頁）

(21)出願番号	特願2010-545034 (P2010-545034)
(86)(22)出願日	平成21年2月2日(2009.2.2)
(85)翻訳文提出日	平成22年8月2日(2010.8.2)
(86)国際出願番号	PCT/US2009/000691
(87)国際公開番号	WO2009/097162
(87)国際公開日	平成21年8月6日(2009.8.6)
(31)優先権主張番号	61/063,230
(32)優先日	平成20年2月1日(2008.2.1)
(33)優先権主張国	米国(US)

(71)出願人　510210601
ジ・オリバー・グループ・リミテッド・ライアビリティ・カンパニー
The Oliver Group, LLC
アメリカ合衆国０６３７９－２０５５コネチカット州パーカタック、グリーンヘイブン・ロード５９５番
(71)出願人　510210612
ブライアン・オリバー
Brian OLIVER
アメリカ合衆国０６３３９コネチカット州レッドヤード、オーガスト・メドウズ１７番

最終頁に続く

(54)【発明の名称】データの検索および索引付けの方法およびそれを実施するシステム

(57)【要約】
複数のデータとともに含まれている語を特定および検索するための、データ形式が不明の複数のデータを処理するためのシステムおよび方法を提供している。この方法には、データ内の語の識別が含まれ、ここで識別には、語を識別するためにデータを検索前に処理することが含まれる。この方法にはまた、所定の方法での語の保存および語の検索が含まれ、ここで検索には、一致結果を識別するために少なくとも1つの検索語に応答する語の検索、および一致結果のファイルへの保存および一致結果の表示の少なくとも1つを行うことによる一致結果の処理が含まれる。

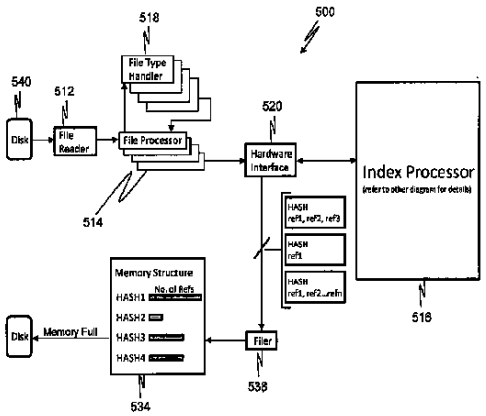


Figure 7

【特許請求の範囲】
【請求項１】
複数のデータを処理して、前記データ形式は不明である前記複数のデータとともに含まれている語の特定および検索をするための方法であって、該方法は、
前記識別ステップが、
語を識別するために前記データを検索前に処理するステップ、および
所定の方法で前記語を保存するステップを含む前記データ内の語を識別するステップ、ならびに、
前記検索ステップが、
一致結果を識別するために少なくとも1つの検索語に応答する前記語を検索するステップ、および
前記一致結果のファイルへの保存および前記一致結果の表示の少なくとも1つを行うことにより前記一致結果を処理するステップを含む方法。
【請求項２】
前記識別するステップはさらに前記データの少なくとも一部分について自然な構成の言語の判別を含む請求項１の方法
【請求項３】
前記識別するステップはさらに、
前記データの文字符号化を判別するステップ、および
前記データ内の語群の場所を判別するステップ、
のうちの少なくとも1つを含む請求項１の方法。
【請求項４】
前記識別するステップはさらに、
データファイルタイプを判別するために前記データの少なくとも一部を調べ、および
前記データが、前記データファイルタイプに応答する特別な取り扱いが必要か否かを判別するステップ
を含む請求項１の方法。
【請求項５】
前記識別するステップはさらに前記データについての第1の組の処理パラメータを判別するステップを含む請求項１の方法。
【請求項６】
前記第1の組の処理パラメータは満足できる結果を生成するかどうかを判別し、
前記第1の組の処理パラメータは満足できる結果を生成しない場合、満足できる結果を生成する第2の組の処理パラメータは存在するかどうかを判別するステップ、
をさらに含む請求項5の方法
【請求項７】
第2の組の処理パラメータが存在するか否かの前記判別が、
テキストのセクションおよび前記テキストの言語を探すための前記データの調査、および
前記データのすべてまたは一部が圧縮データまたは画像データであるか否かを判断するための前記データのエントロピーの調査、
のうち少なくとも1つの実行を含む請求項６の方法。
【請求項８】
前記識別するステップはさらに、
識別された語と語参照とを関連付けるステップ
を含む請求項１の方法。
【請求項９】
線形記憶法および索引付き記憶法のうち少なくとも1つの方法による前記語を保存するステップを含む請求項１の方法。
【請求項１０】
前記語が線形記憶法を使用して保存された場合、前記検索は線形検索法を使用して実施

され、
前記語が索引付き記憶法を使用して保存された場合、前記検索は索引付き検索法を使用して実施される前記検索に以下の通り前記保存に応答する前記語の検索を含む請求項1の方法。
【請求項１１】
データ形式が不明である複数のデータに含まれている語を識別するための方法であって、
前記データの少なくとも一部分の自然な構成の言語を判別し、
前記データを含む語を識別するために、前記自然な言語に応答する前記データを検索前に処理し、および線形記憶法および索引付き記憶法のうち少なくとも1つの方法を使用し前記語を保存するステップを含む方法。
【請求項１２】
前記データの文字符号化の判別、および
前記データ内の語群の場所の判別
のうち少なくとも1つをさらに含む請求項１１の方法。
【請求項１３】
データファイルタイプを判別するために前記データの少なくとも一部を調べ、および
前記データは、前記データファイルタイプに応答する特別な取り扱いが必要か否かを判別するステップ
をさらに含む請求項１１の方法。
【請求項１４】
前記語と語参照とを関連付けるステップをさらに含む請求項１１の方法。
【請求項１５】
データ形式が不明である複数のデータに含まれている識別された語を検索するための方法であって、
少なくとも1つの検索語を受信するステップ、
一致結果を識別するために少なくとも1つの検索語に応答する前記語の検索であって、前記検索が、前記語の完全一致検索またはファジー検索を平行して実行するために構成された複数の検索エンジンによって実行される検索ステップ、および、
前記一致結果のファイルへの保存および前記一致結果の表示の少なくとも1つを行うことにより前記一致結果を処理するステップを含む方法。
【請求項１６】
前記語が線形記憶法を使用して保存された場合、前記検索は線形検索法を使用して実施され、
前記語が索引付き記憶法を使用して保存された場合、前記検索は索引付き検索法を使用して実施される前記検索ステップが前記の識別された語の保存方法に応答する前記識別された語を検索するステップを含む請求項１５の方法。
【請求項１７】
データファイル内に含まれている複数のデータの検索および索引付けをする方法を実施するためのシステムであって、該システムは、
データファイルを受信する手段、
データファイルを保存する手段、および
データ形式は不明である複数のデータとともに含まれている語を識別および検索するために複数のデータを処理する方法を実行する手段であって、該方法は、
前記識別ステップが、
語を識別するために前記データを検索前に処理するステップ、および
所定の方法で前記語を保存するステップを含む前記データ内の語を識別するステップを含み、ならびに、
前記検索ステップが、
一致結果を識別するために少なくとも1つの検索語に応答する前記語を検索するステ

ップ、および
　前記一致結果のファイルへの保存および前記一致結果の表示の少なくとも1つを行うことにより前記一致結果を処理するステップを含む方法
を実施する手段を含むシステム。
【請求項１８】
　データ形式は不明である複数のデータとともに含まれている語を識別および検索するために、前記複数のデータを処理する方法を実施するためのコンピュータ実行可能な命令を持つ、コンピュータ可読記憶媒体であって、該方法は、
　前記識別ステップが、
　語を識別するために前記データを検索前に処理するステップ、および
　所定の方法で前記語を保存するステップ
を含む前記データ内の語を識別するステップ、ならびに、
　前記検索ステップが、
　一致結果を識別するために少なくとも1つの検索語に応答する前記語を検索するステップ、および
　前記一致結果のファイルへの保存および前記一致結果の表示の少なくとも1つを行うことにより前記一致結果を処理するステップを含む
方法を実施するためのコンピュータ実行可能な命令を持つ、コンピュータ可読記憶媒体。
【請求項１９】
　データ形式は不明である複数のデータとともに含まれている語を識別および検索するためのシステムであって、
入力装置、
メモリ装置、
索引処理装置、
前記入力装置および前記メモリ装置との信号通信における処理装置、および
前記メモリ装置に結合された索引プロセッサを含み、
　前記処理装置は、
データを受信し、
前記データをプロセッサへ分配し、
前記プロセッサを使用する前記データのコンテンツ内の語を識別し、
前記語の場所の記録することにより語参照を生成し、
前記語参照についてのハッシュ値を計算し、
前記ハッシュ値を使用し構造的な方法で前記語参照を保存し、
前記参照および前記ハッシュ値を前記メモリ内にある少なくとも1つのテーブルへ転送し、
一致結果を識別するために少なくとも1つの検索語に応答する前記語を検索し、
および、
前記一致結果のファイルへの保存および前記一致結果の表示の少なくとも1つを行うことにより前記一致結果を処理するように構成された処理装置
を含むシステム。
【請求項２０】
　データを読み取るよう構成されたデータリーダー、
　前記データリーダーに結合され、前記データのコンテンツを判別するよう構成されたデータプロセッサ、
　前記データプロセッサに結合され、かつ前記データの索引付けをするよう構成された索引プロセッサであって、前記索引プロセッサは、前記データ内の語を検出し、また前記語からハッシュ値を生成するよう構成されている検索／検出プロセッサを含み、および
　語参照が前記語に応答して生成され、かつ前記語参照および前記ハッシュ値をテーブルに転送するように構成されたメモリに保存される前記索引プロセッサに結合されたメモリを含む、検索および索引付けシステム。

【発明の詳細な説明】
【技術分野】
【０００１】
本出願は、2008年2月1日提出の米国仮特許出願番号61/063,230（代理人整理番号5303.112957）の利益を主張するもので、その全内容を参照により本書に組み込む。この発明は一般に、非常に多様なファイルシステムでの大量のデータの処理に関し、またさらに具体的には非常に多様なファイルシステムからの大量のデータを索引付けおよび検索する方法ならびにシステムに関する。
【背景技術】
【０００２】
多くの事業が、事業運営の実施および／または大量のデータの保存について、コンピュータシステムに依存しつつあるなか、破滅的な出来事が発生した場合、または極端に大量のデータを処理する必要がある場合に、メディアの修復およびデータ変換といったサービスが企業の継続における重要な要素となってきている。各データファイルのコンテンツを読み取り、コンテンツを調べて検索語を検索する従来のユーティリティが存在する。この主題に関する最近なされたバリエーションにより、正規表現として知られるものを使用して検索語の変形を検索できるようになった。
【発明の概要】
【発明が解決しようとする課題】
【０００３】
これらのユーティリティは、小グループのファイルでの少数の検索語の検索を可能にするにあたってはいくらか有効であったが、適正な時間内での大量のデータの検索、または多数の検索語の検索に必要な性能に欠けていた。より早いプロセッサ速度に加え、時間経過に伴うアルゴリズムの向上により、この状況は改善されてきたが、依然として完全一致の検索時の0（log（n））および不完全一致またはファジー一致の検索時の0（n）を必要とする。ここで、nは検索語の数である。テキストが検索される数多くの状況において、不完全一致またはファジー一致が望ましい。また、標準的な事務書類で人がタイプするほとんどのテキストにおいて（厳密な点検および編集がなされる出版される書籍とは異なり）、スペルミスおよび誤字は一般的で、その結果、検索語に対する不完全一致またはファジー一致を行う要望または必要が生じる。ところが大きな問題は、その他の一般的な語を比較的一般的でない検索語への一致として承認する範囲において、あまりにも不完全な一致は望まないことにある。その結果、適正な一致アルゴリズムは、比較的にプロセッサ集約的であり、かつ0（n）であり、結果的に多数の検索語があるときに汎用CPUでは不完全および非常に遅い性能となる。
【０００４】
多数の検索語の検索に使用される別の一般的な方法は、全ての語を収集し、それらをインデックス付きデータベースに保存することにより、各データファイルを大量に処理する方法である。次に、検索語についてデータベースを検索できる。残念ながら、この方法での問題は、データベースの過剰格納を回避する必要があることである。これは、データベースのディスクスペース要件が増大すれば、語数の増加に伴い性能は一般的に低下するためである。これは、ファイル内のテキストフィールドで定義されたファイル形式に由来することが分かっている語のみをデータベースに格納する必要性につながる。これは、処理の対象であるすべてのファイルのファイル形式を知る必要があること、ならびにファイルタイプが不明の場合には、その内部にあるファイルおよび語が保存されなくなることを意味する。さらに、この方法では伝統的なデータベース技術を使用して語が保存されるため、処理は一般的に遅い。
【課題を解決するための手段】
【０００５】
複数のデータを処理して、その複数のデータ（データ形式は不明）とともに含まれている語の特定および検索をするための方法を提供している。この方法には、データ内の語の

識別が含まれ、ここで識別には、語を識別するためにデータを検索前に処理することが含まれる。この方法にはまた、所定の方法での語の保存および語の検索が含まれ、ここで検索には、一致結果を識別するために少なくとも1つの検索語に応答する語の検索、および一致結果のファイルへの保存および一致結果の表示の少なくとも1つを行うことによる一致結果の処理が含まれる。

【０００６】

複数のデータ（データ形式は不明）とともに含まれている語を識別するための語の検索が提供されており、これには、データの少なくとも一部分の自然な構成の言語の判別、データ内に含まれている語を識別するために自然な言語に応答するデータを検索前に処理すること、ならびに線形記憶法および索引付き記憶法のうち少なくとも1つの方法を使用した語の保存が含まれる。

【０００７】

識別された複数のデータ（データ形式は不明）とともに含まれている語を検索する方法が提供されており、これには、少なくとも1つの検索語の受信ならびに一致結果を識別するための少なくとも1つの検索語に応答する語の検索が含まれる。この検索は、語の完全一致検索またはファジー検索を平行して実施するよう構成され、一致結果のファイルへの保存および一致結果の表示の少なくとも1つを行うことによる一致結果を処理する、複数の検索エンジンによって実施される。

【０００８】

データファイル内に含まれている複数のデータの検索および索引付けをする方法を実施するためのシステムが提供されており、このシステムには、データを受信するための装置、データを保存するための装置、ならびにデータ（ここでデータ形式は不明）とともに含まれる語を識別および検索するためにデータを処理する方法を実施する装置が含まれる。この方法には、データ内の語の識別が含まれ、ここでこの識別には所定の方法で語を検索および保存する前に語を識別するためのデータの処理が含まれる。この方法にはまた、一致結果を識別するために少なくとも1つの検索語に応答する語の検索、ならびに一致結果のファイルへの保存および一致結果の表示の少なくとも1つを行うことによる一致結果の処理が含まれる。

【０００９】

データ（ここでデータ形式は不明）とともに含まれている語を識別および検索するために、複数のデータを処理する方法を実施するためのコンピュータ実行可能な命令を持つ、コンピュータ可読記憶媒体。この方法には、データ内の語の識別が含まれ、ここでこの識別には所定の方法で語を検索および保存する前に語を識別するためのデータの処理が含まれる。この方法にはまた、一致結果を識別するために少なくとも1つの検索語に応答する語の検索、ならびに一致結果のファイルへの保存および一致結果の表示の少なくとも1つを行うことによる一致結果の処理が含まれる。

【００１０】

複数のデータ（データ形式は不明）とともに含まれる語を識別および検索をするシステムが提供されており、ここでこのシステムには、入力装置、メモリ装置、索引処理装置、入力装置およびメモリ装置との信号通信の処理装置、ならびにメモリ装置に連結された索引プロセッサが含まれ、ここで処理装置は、データを受信し、データをプロセッサに分配し、プロセッサを使用してデータ内の語を識別し、語の位置を記録することで語参照を生成し、その語参照のためのハッシュ値を計算し、その語参照をそのハッシュ値を使用して構造化された方法で保存し、参照値およびハッシュ値をメモリ内の少なくとも1つのテーブルに転送し、一致結果を識別するために少なくとも1つの検索語に応答する語を検索し、一致結果のファイルへの保存および一致結果の表示のうち少なくとも1つの方法により一致結果を処理するように構成されている。

【００１１】

検索および索引付けシステムが提供されており、これにはデータを読み取るように構成されたデータリーダー、データリーダーに連結されたデータプロセッサ（ここで、データ

プロセッサは、データの内容を判別するように構成）、データプロセッサに連結され、データのインデックス付けをするよう構成された索引プロセッサ（ここで、索引プロセッサには、データ内の語を検出して、その語からハッシュ値を生成するよう構成された検索／検出プロセッサが含まれる）、ならびに索引プロセッサに連結されたメモリ（ここで前記語に応答する語参照が生成され、メモリ内に保存され、そのメモリは語参照およびハッシュ値をテーブルに転送するよう構成されている）が含まれる。

【図面の簡単な説明】

【００１２】

本発明の上述およびその他の機能および利点は、以下の例証用の実施形態の詳細な説明を、下記の添付図面とともに理解することによりさらによく理解される。

【図１】図1は、発明の実施形態に従ってデータを処理するための全体的な方法を図示した演算ブロック図である。

【図２】図2は、図1の全体的方法に従い、ファイルおよび／またはデータストリームについての情報を判別する方法を図示した演算ブロック図である。

【図２Ａ】図2Aは、図1の全体的方法に従う、索引付け記憶装置方法の1つの実施形態を図示した演算ブロック図である。

【図３】図3は、図1の全体的な方法に従う、索引付き検索方法の1つの実施形態を図示した系統ブロック図である。

【図４】図4は、図3の索引付き検索方法を図示した系統ブロック図である。

【図５】図5は、図3の索引付き検索方法を図示した系統ブロック図である。

【図６】図6は、図3の索引付き検索方法を図示した系統ブロック図である。

【図７】図7は、発明に従いデータの検索および索引付けをするためのシステムの1つの実施形態を図示した系統フローブロック図である。

【図８】図8は、発明に従った索引処理装置の1つの実施形態を図示した系統フローブロック図である。

【図９】図9は、発明に従った線形検出プロセッサとして構成された処理装置の1つの実施形態を図示した系統フローブロック図である。

【図１０】図10は、本発明の線形検出方法の1つの実施形態の一例を図示したブロック図である。

【発明を実施するための形態】

【００１３】

本発明に従い、本書で開示したシステムおよび方法は、既存の方法およびシステムとは異なり、ファイル形式を必要とせず、従来のデータベースは置かれた語参照を保存するためには使用されず、また検索語の線形検索が実行された場合に、適切な数の検索語までの０（1）拡張性を備えた超並列ハードウエア実行プロセッサを使用して実行される。

【００１４】

本発明に従い、複数のデータ（1つ以上のデータファイルに保存）を処理するための方法およびシステムが開示されており、この方法およびシステムにより、任意のファイルおよびデータストリームの検索および索引付けがなされる。性能、精度および実施の労力のレベルのバランスのとれた方法により、ファイル内または大量のファイル内に発生する語（これには、限定はされないものの、固有名詞、業界固有の用語、共通の略語および特に定義された用語）が識別される。このタスクを実行する1つのアプローチには、テキストがファイル内のどこかに存在し、および多様な共通の文字符号化により表現されうるという仮定が含まれてもよい。ファイルおよび／またはファイルの部分を、希望に応じた特別な取り扱いのために「タグ付け」または識別しうる。さらに、ファイルは、バイトのストリームとして取り扱うことができ、これには、希望に応じて、またはコードページによる指図により定義しうる文字が含まれることができ、ここでこのコードページは、既知である場合や、ファイルの分析を実施することにより判断される場合もある。文字定義のいくつかの例としては、1バイト（ASCII/EBCDIC数値など）、可変長バイト（Unicode Transformation FormatまたはUTF-8値など）および／または2バイト（UTF-16値など）が含まれう

る。当然ながら、本発明は本書で単一言語の処理に関連して開示しているが、異なる言語でのデータを持つファイルをサポートする言語固有のパラメータを変化させることにより、複数言語の処理も実施しうることは理解されなければならない。これは、大量のファイルのうち一部のファイル（データ）には異なる言語が含まれうるため有用である。

【００１５】

図1では、本発明に従ってデータを処理するための全体的な方法100を図示した演算ブロック図が提供されている。方法100には、演算ブロック102に示すとおり、ファイルおよび／またはデータに関するパラメータ／情報を決定するためのデータファイルの分析の実施、ならびに演算ブロック104（ここで、識別された語が語参照と関連付けられる）に示すとおり、文字および／または語を識別するためにデータの処理が含まれる。この方法100にはさらに、演算ブロック106に示すとおり、あらかじめ定めたおよび構造化された方法でのデータ（つまり、語参照）の保存、ならびに演算ブロック108に示すとおり、希望の検索語について保存したデータの検索が含まれ、ここで検索語には、希望の語および／またはフレーズが含まれうる。その後、結果を、希望に応じて演算ブロック110に示すとおり通信することもできる。演算ブロック102-110を参照して上記で開示したそれぞれの演算は、下記にさらに詳細に考察されている。

【００１６】

本発明に従い、文字および／または語（演算ブロック104に示すとおり）を識別するためにデータを処理する前に、データ／ファイルを正確に分析するにあたり役立てるために、データ／ファイルについてある特定のパラメータを前もって知っておくことは有益である。これらのパラメータには、使用する言語およびコードページ、また特別な取り扱いが必要であるかどうかが含まれる。情報が多いほど、より効率的および正確な検索ができるようになる。情報は、ファイルタイプを正確に識別するために、ファイル内のデータの一部（例えば、ファイルの最初の数百バイト）を調べることにより判別しうることが理解される。ファイルタイプの適切なハンドラーを識別し、文字および／または語を識別するためのその他のパラメータを判別する為に使用することができる。ファイルタイプが識別できない場合には、その他の戦略を使用できる（希望のパラメータを判断するためのファイルデータの各部を調べるなど）ことが理解される。初期パラメータ化（つまり、第1の組の処理パラメータ）により、満足できる結果（例えば、精度）が提供されない場合には、希望のパラメータ（つまり、第1の組の処理パラメータ）を得るために、さらに複雑なファイル分析を実施することもできる。これには、テキストの部分およびそのテキストの言語を調べるためのデータの分析、および／またはデータの全体または一部が圧縮データまたは画像データであるかどうかを判別するファイルデータのエントロピーの分析が含まれうる。

【００１７】

再び図1を参照するが、演算ブロック104に示すとおり、文字および／または語を識別するためのデータの処理は、ファイルをバイトのストリームとして処理する語検出アルゴリズムとして使用して達成することもできる。語を識別するために、アルゴリズムにより定義済みのコードページについて有効なASCIIおよび／またはUnicodeの範囲内の文字を「調べ」、文字が見つかった場合に、アルゴリズムにより文字が有効であるかどうかが判別される。文字が有効な場合には、アルゴリズムにより、UTF-8および／またはUTF-16デコーダを使用してバイトから文字が生成され、文字が後の記憶用にバッファに追加される。上記のとおり、アルゴリズムにより、識別された残りの文字が分析され、有効であるかどうかが判別され、有効な文字がバッファに追加される。アルゴリズムが文字（letter）または数字ではない文字に遭遇すると、アルゴリズムではその文字を区切り記号として扱い、および語の蓄積が終了する。バッファには有効な語が含まれる可能性があるが、語の最初および／または最後に「余分な」文字（つまり、語に「属」さない文字）もありうることが理解される。この場合には、これらの余分な文字は、検索段階で処理されうる。

【００１８】

有効な語が見つかると、アルゴリズムにより、見つかった語がアルゴリズムにより語と

して承認されるべきかどうかを判別するために、見つかった語が調査される。これを達成するために、アルゴリズムにより、見つかった語をすべて大文字に変換し、および句読点があればすべて除外することによって、候補語が生成されうる。次に、候補語は、検査や調査が行われ、候補語が承認されるべきかどうかが判別される。これは、あるグループの語（語群）が、調べているファイルについてアルゴリズムにより既に作成されたかどうかを判別することにより、達成されうる。あるグループの語が作成された場合には、候補語は自動的にそのグループの一部として承認される。ところが、候補語がグループの一部ではない場合には、語として承認されるべきかどうかを判別するために候補語について2回の検定が実施され、ここで候補語中、いずれかの検定結果が通過すると、アルゴリズムにより語として承認される。第1の検定には、その候補語に3文字（またはそれ以上）および少なくとも1つの母音（または外国語の場合、それに値するもの）が含まれているかどうかの判別が関連する。第2の検定には、候補語を24ビットハッシュ値（その他のサイズのハッシュ値も使用しうる）に細分し、そのハッシュ値を辞書テーブルのアドレス付けに使用することを含む。そのハッシュ値が有効な言語依存の語にでくわした場合は、それに応じて辞書テーブル内のアドレス付けされたビットがセットされ、候補語が承認される。

【００１９】

ハッシュアドレス付けされた辞書テーブルは、作成することも、または商業的に入手可能な辞書とすることもでき、また、固有名詞、よく用いられる略語および業界固有の用語（法律用語や医学用語および略語など）が含まれうることが理解される。また、辞書テーブルの使用により、供給された辞書からのすべての語について肯定的な検索結果による判別が可能となることが理解される。さらに、ハッシュ値のビット長は限られているため、2つの異なる候補語が同じ値に細分されることが認められる。これにより、語ではない文字の組み合わせが有効な語として解釈されることが生じうる。本発明はこれを許容し、索引付けされた語が有効な語ではない割合を予想することが意図されている。

【００２０】

ファイル内での語の場所は一般に、単にファイル内のそのバイトオフセットであることが理解される。また、局所的な検索機能は語間のギャップには寛容であるため、これは一般には構わず、問題となることはないが、検索機能が正確に機能しなくなる特性を持つ特定のファイルタイプがある。こうした1つの特性には、フレーズの語間の2つ以上の大規模なギャップが関与し、また他の特性には、適切な順序になっていない語が関与する。これらの問題に対処するために、これらのどれか1つの状況が処理対象のファイルのファイルタイプで発生することが分かっている場合には、特殊なファイルタイプハンドラーを使用することができ、これらの問題に対処することができるようになる。例えば、特殊なファイルタイプハンドラーは、データを再構成するか、および／またはファイルの処理方法についてアルゴリズムに適したパラメータを供給することにより、形式について十分に知ることにより、それに従い動作するように構成することができ、ここで、パラメータはテキストのフィールドを指し示し、およびフィールド内のデータは語として取り扱われることができる。代替的に、文字および／または語を識別するためにデータを処理する前に、データを再注文することもできる。さらにまた、テキストが順序どおりではない場合や、語および／またはフレーズが断片に分かれている場合には、ハンドラーは、テキストをその適切な順序に戻すことができる。データがどのようにパラメータ化されているか、またはデータの順序がどのように変更されたかにかかわりなく、満足できる結果が得られないと判断された場合には、形式固有のソフトウエア実施を語の検索に使用しうることが意図されている。

【００２１】

上記で簡単に考察したとおり、ファイルおよび／またはデータストリーム内の語および／または文字を識別するために、特定の情報が必要となることがある。この情報には、ファイルデータとともに、文字符号化体系（character encoding scheme）、コードページおよび言語が含まれうる。発明の一実施形態に従い、この情報を判別する方法200は図2に図示しており、これには演算ブロック202に示すとおりファイルタイプを判別するための

ファイルの識別または分析が含まれる。これには、ファイルタイプを効率よく識別するためのファイルの基本的な分析の実行が含まれうる、またはこれには、大部分の状況について使用可能な適切なパラメータを判別するためにファイルから十分な情報を見つけることが含まれうる。この分析は、ファイル拡張子およびファイル構造を判別するためにファイルを調査することにより実行することができ、これは、ファイル拡張子およびファイル内の特定位置での特定のバイトの組み合わせを調べることにより達成することができる。例えば、多くの場合、ファイルヘッダーをその型を識別する為に使用できるが、必ずしもそうとは限らない。署名分析ソフトウエアが広く利用できるが、この機能性を提供するため号化体系、コードページおよび言語の判別を試みる基本的な分析の後のファイルの処理が含まれる。この処理には、類似した型を持つと見られるその他のファイルに基づくデフォルトを使用しうる。画像または圧縮データである可能性は、ファイルの多様な部分からのデーに使用しうる。

【００２２】

ファイルタイプが識別できる場合には、演算ブロック204に示すとおり、言語およびコードページが抽出される。言語は判別できるがコードページは判別できない場合には、その言語にとって一般的なコードページが利用できる。一方、言語が判別できない場合には、この型について最近処理したファイルのうち大半を占める言語にデフォルトが設定される。次に、コードページが、その言語に適合した適切なデフォルトに設定される。ファイルタイプが判別できない場合には、演算ブロック206に示すとおり、ファイルの分析が実行され、これにはファイルからサンプルを取り、そのサンプルをインストールした辞書と比較することが含まれる。分析にはまた、文字符タのサンプリングおよびこのデータのエントロピーの調査により、判別しうる。圧縮データおよび非圧縮データは、分析による特性パターンを持つ傾向にある。例えば、圧縮データ（圧縮画像データを含む）は、非常に高レベルのエントロピーを持つ。当然ながら、データ型の分類を、ヒット率閾値の設定に使用することもでき、ここで「ヒット」率を割り当てることができ、画像または圧縮データが見つからない限りそれを高く設定し、見つかった場合には「ヒット」率を低く設定しうることが理解される。

【００２３】

この時点で、演算ブロック208に示すとおり、システムは語を識別するためにファイルを処理するよう構成される。これには、コードページ内で有効な文字範囲を持つ語ファインダーの設定、コードページ内の有効な区切り記号、ならびに言語、言語に依存した母音、スキップ閾値、テキストのバイトオフセット範囲（文字コード化を含む）および辞書が含まれうる。ファイルタイプ識別／分析の結果に応じて、適切な文字デコーダを有効化または無効化することができる。その後、演算ブロック210に示すとおり、語を識別するために、ファイルが処理される。この処理は、文字デコーダによりデータを実行することにより開始することができ、その出力は語を識別するために処理され、ここで語の辞書ヒット数および処理したバイト数の計数値が記録され、またヒット率を計算するために使用される。すべてのファイルデータの処理が終わると、演算ブロック212に示すとおり、ヒット率が評価される。ヒット率が最小値プリセット閾値よりも大きい場合には、結果が承認され、その次のファイルが処理できる。ところが、最小値ヒット率閾値（または、判別されていることや、将来的に定義されることのあるその他の基準）が満たされない場合には、さらなる処理を実行しうる。最小値ヒット率閾値が満たされない状況では、パラメータが正しく判別できない可能性がある（これによりファイル内で語がほとんどまたは全く見つからないことになりうる）。新しいパラメータを判別する1つの方法は、演算ブロック214に示すとおり、より高度なファイルの分析を実行することである。これには、ファイルのサンプルだけではなく、ファイル全体のさらなるエントロピー評価の実施が含まれうる。テキストセクションが識別された場合には、多様な異なる文字符号化、コードページ、および可能性のある言語での一般的な語を見つけるために、より積極的な試行がなされうる。より適切なパラメータが識別された場合には、演算ブロック208に示すとおり、語を識別するためにファイルが処理されるようシステムが再構成でき、新規のパラメータでプ

ロセスが繰り返される。ファイルまたはデータストリームの処理中の任意の時点で、例外またはエラーが発生した場合には、例外処理を呼び出すことができ、ここで例外の詳細はファイルおよび／またはデータストリームのコンテンツとともに保存することができる。

【００２４】

上記で簡単に考察したとおり、言語およびコードページ確認に役立つよう、辞書にヒットした語の統計的カウンターを実施することができる。このカウンターの値は、その値をファイルのバイト数で割るなどにより、希望に応じて語ヒット率に変換しうる。その後、この率は、正しいコードページおよび言語が使用される可能性が高いかどうかを判別するために、予想される最小閾値と比較されうる。辞書の語がほとんどまたは全く見つからない場合には、不正確な言語またはコードページの使用によることも考えられる。ヒット数が期待される最小閾値よりも低い場合には、さらなる分析を続けうる。単一のファイルで、複数の文字符号化体系、コードページ、および言語の採用もしうることが理解される。この場合に該当することを前もって判別できるとき、ファイルまたはデータストリームを、セクションに分割して、各セクションを文字コード化、コードページ、および言語を識別するために、そのセクションについて新しいパラメータ使用して処理することができる。ファイルはまた、所定のサイズ（例えば、合計ファイルサイズが2ギガバイトを超える場合にセクション当たり2ギガバイト）のセグメントに分割することができる。

【００２５】

上記で考察したとおり、方法100には、所定の構造化された方法でのデータ（つまり、語参照）の保存が含まれ、これは多様な方法により達成しうる。こうした1つの方法は、「線形記憶法」と呼ばれ、および単に語参照が単一のテーブルに付加され、テーブルがいっぱいのとき、このテーブルが次の段階に転送される。本発明によれば、参照は早い速度で蓄積されることがあり、およびこれらの参照はメモリに記録され、最終的にディスクに記録されるべきである。一部の場合において、処理されるデータ1ギガバイト当たり数百万の語参照が生成されうる。線形検索では、検索語の検索を試み、および語参照が検索語にどの程度近接しているかについてスコアを作成するために、ハードウエア加速手段が適切なソフトウエアとともに使用されうる。よって、導入される検索エンジンの数によって、効率よく対応できる妥当な数の検索語に制限される。

【００２６】

こうした別の方法は、「索引付き記憶法」と呼ばれ、語参照がテーブル（線形記憶法と類似）に保存されるが、語参照が索引付けされるもので、従来的なソフトウエア技術を使用して効率の高い方法で、効果的に語参照が検索されるようになる。索引付き記憶方法の1つの一実施形態600を図2Aに示すが、ここで使用した頭字語は、凡例602に記載したとおりである。索引付けを促進するために、可変長の語を複数の方法で細分して、綴り違いまたはミスタイプがある場合に、ハッシュテーブル内で語が見つかる可能性を最大化することができる。ハッシュ値の第1の部分は、ハッシュ割当テーブルのアドレス付けに使用することができ、一方ハッシュ値の残りの部分は、それが指し示す語参照とともに、ハッシュテーブル内に保存される。これにより、容認できる一致であるハッシュ値および語参照を検索するために、素早く位置を定めて検索することができる多数のサブテーブルが効果的に生成される。語参照には語と同様に数字も含まれうることが理解されるべきである。また、一般的な語に遭遇するたびに、同一の値に細分され、これにより、ある特定のサブテーブルでは、一般的な語が細分されないサブテーブルよりも早くいっぱいになることになることが理解されるべきである。従って、本発明では、ダイナミックメモリ割当機能を必要とすることなく、効率の高い方法で提供される。メインメモリテーブルがいっぱいになると、これらのテーブルのコンテンツが所定の構造化された方法でディスクに書き込まれた後、テーブルは再初期化されることが理解される。

【００２７】

上記で簡単に考察したとおり、参照は、急速に蓄積でき、また検索用に効率の高い方法で保存される必要がある。これを達成する1つの方法は、参照を索引付けによる方法で保存することである。索引付き記憶法では、ハードウエアベース（および／またはソフトウ

エアベース）の処理が使用され、語についてのこれらの索引が簡単に検索できるように、見つかった語の索引付けがなされる。例えば、1テラバイトのデータが処理された場合、およそ30億の語参照がこの1テラバイトのデータ内に見つかると仮定でき、およそ90GBの出力（3GB×30）が生成されることになる。出力データはすべて索引付けされているが、数百語からなる一般的な検索セッションでは、数百万の索引ルックアップが生成されうる。これはファジー一致のためであり、一般的には、綴り間違いまたは誤植のある語を検索するための試み1回で、多数の順列（permutations）の語をルックアップする場合に望ましい。

【００２８】

語参照は、コンテンツ別にルックアップできるものであるべきである。語参照の1つの構造には、文字当たり6ビットとして表現される語が含まれることができ、これは、2～15文字の語であれば、12～90ビットとなることになる。従って、この語参照を固定ビット幅のハッシュ値に変換して、検索の効率をより良くするために索引として使用できるようにすることが望ましい。語参照が辞書内に見つかった場合、語全体について1つのハッシュ値が作成される。語参照が辞書内に見つからなかった場合には、その語が不正確に綴られているかまたはタイプされている可能性がある。これに対処する1つの方法は、細分することによって語の異なる部分から複数の索引を生成すること（ハッシュ値の生成）である。当然ながら、効果的なハッシュ値を持つためには、値を形成するために語参照内に少なくとも4文字が使用される必要がある。本発明は、5文字以上の長さの語から4個のハッシュ値が生成されうることが意図されるが、希望に応じて、それを超える個数が生成されうる。

【００２９】

一実施形態において、第1のハッシュ値では、語の最初の4文字が使用され得、第2のハッシュ値では最後の4文字が、第3のハッシュ値では第1の文字から始まる1つおきの文字（1、3、5...）が使用され得（5文字の語については最後の3文字、および6文字の語については最後の2文字が使用される）、および第4のハッシュ値は、第2の文字（2、4、6...）から始まる1つおきの文字が使用されうるが、ここで、5文字の語の場合には最初の3文字が含まれうる。最初の2文字は、6文字および7文字の語についてハッシュ値に含まれる。この場合に、ある時点で、ハッシュ値の1つにおいて語内の各文字をスキップして、任意の1文字の綴りまたはタイプミスを回避することになる。ハッシュ値生成の2つにおいて、最初および最後の語を使用することにより、ほとんどの文字の過不足の状況を回避しうる。どの程度近接した一致かについての本当の判断は、検索時になすことができるが、1つ以上の索引が一致しない場合に、参照は見つからない場合がある。

【００３０】

ハッシュ値は、ハッシュ値（32ビットハッシュ値など）を生成するための標準ハッシングアルゴリズムを使用して計算しうることが理解される。これらのハッシュ値は、ファイルまたはファイルの一部およびファイル内でのオフセットを表現する数字とともに、索引付けされたハッシュテーブル内に保存しうる。追加的なテーブルを未並べ換えのサブテーブルとして組織化することもでき、ここでサブテーブルの選択は、ハッシュ値の一部から判別されうる。例えば、32ビットのハッシュバルブについては、ハッシュテーブルは、未並べ換えの100万（1,048,576）個のサブテーブルに組織化でき、またテーブルの選択は、ハッシュプレフィックスとして既知の32ビットのハッシュ値のうち上位20ビットをもとにしうる。このように、テーブル全体内でハッシュ値が検索されるとき、ハッシュプレフィックスに基づき、サブテーブルが選択されうる。最後に、ハッシュ値の残りについてサブテーブルが線形的に検索される。

【００３１】

参照の並べ換えには数多くの方法があるが、参照を組織化された方法で保存するにあたり、最も制限的な要素の1つは、メモリアクセス時間である。コンピュータメモリは順次に非常に高速にアクセスできるが、ランダムメモリの場所へのアクセスには、ずっと長い時間がかかる（場合によっては最大20倍長い）。メインハッシュテーブルの格納は二段階

で実行されうるが、ここで第1段階はハードウエア（および／またはソフトウエア）論理回路に実装することができ、また参照記憶テーブルおよび関連するハッシュ割当テーブルおよびハッシュリンクテーブルを非常に高速なランダムアクセスが可能な高速アクセスのスタティックRAM内に保持できる。第2段階は、このテーブルがいっぱいになったときに発生する。この時点で、そのコンテンツは、コンピュータのメインメモリ内に常駐しうる、類似した構造ではあるがより大きな記憶テーブルに規則正しく移動する。このテーブルへの転送は、ほとんどの場合、順次に実行され、ホストメインメモリにアクセスする際に発生するランダムアクセスの性能ペナルティが回避される。

【００３２】

ハードウエアハッシュアルゴリズムは、ハッシュ記憶テーブル、ハッシュ割当テーブル、ハッシュリンクテーブル、および参照テーブルの4つのテーブルを保持しうることが理解される。ハッシュ記憶テーブルは、最も大きなテーブルであり、上述のハッシュ要素を保持する。例えば、ハードウエアベース版には、524,288個のビンが含まれうるが、ここで各ビンは16個のハッシュ要素を保持でき、またこのハッシュテーブル内の各ハッシュ要素は、32ビットであり、ハッシュ値の残りの12ビットおよび語参照テーブルへの20ビットポインタを保持する。

【００３３】

ハッシュ割当テーブルは、ハッシュ記憶テーブル内のそのビンの最新の割当への多数のポインターの配列で、およびテーブルは、それぞれのハッシュ記憶サブテーブルについての場所を持つ。上記の例を続けるために、32ビットハッシュ値のうち上位20ビットが、このルックアップテーブルおよびそのサブテーブルのエントリーで現在満たされているハッシュ記憶テーブル内のビンへのポイントをアドレス付けするために使用される。この表の各要素は32ビット幅である。ハッシュリンクテーブルは、それぞれのビン（ハッシュ記憶テーブル）の要素を含み、および特定のテーブルに応答するすべてのビンが交差できるようポインターを保持するテーブルである。上記の例を続行するために、ハッシュリンクテーブルは、524,288要素（ハッシュ記憶テーブル内のそれぞれのビンに対して1個）を含むことができ、および特定のサブテーブルに応答するすべてのビンが交差できるようポインターを保持する。そのサブテーブルにいっぱいになったばかりのビンを指し示すようビンが割り当てられるたびに、エントリーがこのテーブルに追加され、それによってそのハッシュプレフィックスのためのすべてのビンへのポインターのリンク付きリストが作成される。この表における一連のポインターは、各サブテーブルに応答するすべてのビンを回収するために使用しうる。繰り返すが、この例において、この表の各要素は32ビット幅である。

【００３４】

図3-6を参照するが、索引付け方法は以下のとおり実施されうる。新規の語参照が見つかったとき、これがまず参照テーブル内のその次の空きスポットに追加される。また語も細分して、32ビットハッシュ値を作成しうる。その次のステップは、20ビットハッシュプレフィックスで、ハッシュ割当テーブルのアドレス付けをする。これがハッシュ記憶テーブル内のある要素をポイントし、かつビンがいっぱいではない場合には、参照がハッシュ記憶テーブルに追加される。ビンがいっぱいの場合には、新規のビンが割り当てられ、ハッシュリンクテーブル内のいっぱいになったビンへのリンクが生成され、新しく割り当てられたビンのアドレスでハッシュ割当テーブルが更新される。この図は、第1のビンへのリンク付けを確立するための新しいリンクテーブル管理を示す。参照テーブルに関連して、参照テーブルへの各新規エントリーに対し、ハッシュ記憶テーブル内には1個または4個のエントリーが存在しうる。これは、語参照が単純に細分されたもの（辞書ヒットが発生した場合）または上記で考察した4通りの異なる方法で細分されたもののいずれかであるためである。よって、ハッシュ割当テーブルは一般に、異なるハッシュプレフィックスの数に基づきまばらに格納され、およびハッシュリンクテーブルは単に、特定の割当についてハッシュ記憶テーブル内の一連のビンを反映する。その結果、見つかったそれぞれの新しいエントリーについて、関連した情報を保存するために、すべてのテーブルについて合

計で24または36バイトの記憶装置が消費される。参照テーブルへの新規のエントリーが数字である場合、その数字は、上記で考察したものと同一のハッシングアルゴリズムを使用して細分され、語参照と同一の方法で保存および索引付けがなされうる。ハッシュ記憶テーブル内または参照テーブル内のすべてのビンがいっぱいであるとき、処理は中止され、ハードウエアベースのテーブルは、ホストコンピュータのメモリ内のメインテーブルに転送される。ただし、この場合において、いっぱいになっているいずれかのテーブルにより、この転送動作が誘発される。最初のステップは、参照テーブルを連続したブロックとして転送し、メイン参照テーブルに加えることでありうる。次に、特定のハッシュプレフィックスサブテーブルについてのすべてのハッシュエントリーが転送され得るが、これは、ハッシュ割当テーブル内の特定のエントリーによりポイントされているハッシュ記憶テーブルのビン内にある格納されたすべての要素を送信することにより達成される。これは、ハッシュリンクテーブル内にそのエントリーについてそれに応答するリンクが存在し、そのビン内のすべての要素が転送される場合に、有利となりうる。このプロセスは、ハッシュプレフィックスに関連したすべてのビンが送信されるまで継続しうる。リンクはそれらが作成された順序とは反対の順序でたどることができるため、記憶装置の順序は検索処理には関係しないものの、ビンは逆の順序で転送・保存されうる。このように、ホストコンピュータにより、すべてのハッシュエントリーが、簡単かつ効率よく保存できる連続したブロックのデータとして受信されうる。すべての転送が終了すると、ハードウエアベースのテーブルが再び初期化され、処理が再開されうる。

【００３５】

従って、ホストコンピュータのメモリ内のテーブルは、ハッシュ記憶テーブル内のビンの数が利用可能なメインメモリをもとにずっと大きいものとなりうることを除き、ハードウエアベースのテーブルと同一の方法で整理されうる。同様に、参照テーブルのサイズは、同様にずっと大きいものとなりうる。例えば、ハッシュ記憶テーブル内の共通のビンの数は、6,400万個のビンとなりうるが、これは、最大で10億個のハッシュ要素となり、また参照テーブルは、最大で2億5,600万個のエントリーを保持しうる。また、参照テーブルは大きめであるため、ハッシュ記憶テーブルの各要素は、より幅の広いハッシュサフィックスについて12ビットのポインターのニーズを満たす48ビット、また参照テーブルへのポインター用に最大36ビットとしうる。数字による参照（語参照と対照に）であるすべての参照へのポインターのリストを含む第5のテーブルをホストコンピュータ上に生成しうる。これにより、1つのファイル内またはすべてのファイル内のすべての数字が素早く見つかり、検索できるようになる（例えば、線形的に）。メインメモリテーブルがいっぱいになったとき、実質的にメモリに含まれている内容の画像として、ディスク記憶装置に転送される。索引付けセッション中に、数多くのこうした画像が生成され得、また検索の実行時に、各画像がディスクからメモリに読み取られることができ、ここですべての語およびその変形が本書で説明したとおりに検索されうる。画像の処理が終了すると、すべての画像が処理されるまで、その次の画像がメモリに読み込まれうる。

【００３６】

前の方法の代替として、これらの2つのデザインを含む別の方法は、見つかった語参照の直接的記憶装置であり、それにハードウエア（および／またはソフトウエア）により加速した語参照の線形検索が続く。この方法における記憶ステップは、単に語ファインダーアルゴリズムの出力でもよく、また単にファイル名および／またはデータストリームへの順序を示す32ビット付加物付の128ビット出力として保存してもよい。これは、それぞれの語参照が160ビットでも、または5個の32ビットの語でもよいことを意味する。この出力は、単にメモリに蓄積され、ホストコンピュータのメインメモリに最も早い機会に転送される。この転送中に処理は停止する必要がないことは理解されるべきである。すべての語参照が見つかり、ディスク上のボリュームに記録された後、処理されたボリュームのそれぞれの語は、基本的に経時的に並べ換えされる。この方法では、ファイルが提示された順序で、また次に各ファイル内のその場所別に、語参照が自然に並べ換えされるようになる。検索は数十もの検索エンジン（つまり、ゲートアレイを経由）を実施することにより開

始することができ、ここで各検索エンジンは、その検索をするようにプログラムされた複数の語の検索能力を持つ。プログラミングには、語参照が、承認されるものとして検索を試みている語にどの程度近接しているかどうかを定義したいくらかの規則が含まれうる。数字についても、検索エンジンは数字による参照と語参照を区別でき、数字による参照を別個の数字検索論理要素を使用して処理できるため、簡単に検索できる。

【００３７】

データが所定の構造化された方法で保存されると、保存されたセグメントがメモリ内に読み戻され、少なくとも1つの検索要求に対応して処理される。検索要求は単一の要求としても実行できるが、各セグメントに数ギガバイトの情報が含まれ、かつ検索に数分間かかりうるように検索要求をバッチ処理することが推奨される。従って、検索語のバッチ全体が、次のセグメントが読み込まれる前に、各セグメントに対して処理できる。一般に、検索要求は、検索語（例えば、語および／またはフレーズ）を、方法100を実施するシステムと対話する検索インターフェース（グラフィカルユーザーインターフェースなど）に入力することにより開始される。ただし、検索要求は、自動化されたプロセスによって開始しうることが意図されている。入力された検索語には、1）綴り間違いおよび／またはタイプミスという点で、一致がどの程度近接して一致する必要があるか、および／または2）語順という点で、フレーズがどの程度近接して一致する必要があるか、および／または3）いくつの語およびどの語が表示される必要があるかを定義するパラメータまたは属性が含まれうる。検索語（およびその属性）のバッチが入力されると、用語が検索されうる。これは、検索語（つまり、語）に含まれるすべての文字を大文字に変換し句読点を除去する、検索語の処理により達成しうる。本発明は、本書では語参照が検索フレーズと一致するかどうかを判別する前に、語検索が実行されるものとして開示しているが、任意の順序での演算の実行を希望の最終結果に適した方法で導入しうる。

【００３８】

本書で下記にさらに考察するとおり、データの検索に使用する方法は、データを保存するために使用する方法に応答するものとしうることは理解されなければならない。例えば、線形記憶法により保存されたデータは、線形検索法を使用して検索しうる。線形検索法では一般に、複数の検索エンジンがハードウエアプロセッサ内に実装されるハードウエアベースの検索技法が採用されるが、それぞれの検索エンジンが検索語の1つに応答する。次に、検索エンジンは、識別された語参照の圧縮を平行して実行し、ここで語参照が希望のパラメータを満たした場合には、一致が発生したことが判別され、語参照が承認される。一方、索引付き記憶法により保存されたデータは、索引付き検索方法を使用して検索ができ、ここで索引付き検索方法には、検索ハッシュ値を生成するための検索語の細分が関与しうる。綴り間違いおよびその他のミスを考慮するための検索語についてその他のハッシュ値を作成しうることが意図されている。検索ハッシュ値はこの後、索引テーブル内のルックアップを行い、該当する任意の語参照が調査され、一致のパラメータを満たすかどうか（つまり、検索語に十分近接しているか？）が判別される。単一語の検索語の場合に、検索語およびその派生語に対するすべての参照が、最も高いランクのものがその用語の一致として割り当てられて戻されることが理解される。その上、多数の一般的な語は、単にフレーズ一致を促進するために検索され、これらの語は別個のエントリーとしては戻されないことがある。

【００３９】

検索語にフレーズが含まれるとき、そのフレーズは初めに、フレーズの最初の語を一致させた後、そのファイルに関連したフレーズ内のそれに続く語を一致させることにより処理しうる。フレーズ内の語の一致が識別されると、フレーズ検索語を伴う属性に応答する評価アルゴリズムが実装されうるが、ここで、評価アルゴリズムは、すべてまたはほとんどの語が互いに近接して、かつ全般的に正しい順序で存在するかどうかを判別するために一致を調査する。次に、この判別をもとに結果がランク付けされうる。例えば、すべての語が近接してかつ正しい順序で存在する場合、高いランクを与えることができ、1つ以上の語が不足している場合、または2つ以上の語の順序が入れ替わっている場合には、ラン

クレベルは低くなる。検索機能の部分は、状況に応じて各種のユーティリティを使用して実装しうることが理解される。例えば、ある特定のファイルタイプは、テキストデータを方法100の効率および精度に影響を及ぼしうる通常的でない方法で保存しうる。これらのファイルタイプの1つに遭遇し、フレーズの可能性を示唆する語が見つかった場合には、本来のファイルは、フレーズを見つけ、およびその語の局在性についてより適切な評価をするために、別の低い性能のコンバーターまたはパーサーにより処理することができる。次に、この評価の結果が、上述のとおり、このフレーズのランク付けに使用されうる。

【００４０】

データが処理されて、結果が得られると、その結果を関係者に通信することができ、またこれには1）検索した内容、2）見つかった語／フレーズ、3）見つかった語／フレーズの場所、4）書類のファイルタイプについて、検索語が現れる場所の文脈を示す出力、および5）検索語を含むファイルのコピーが含まれうる。これは、関係者がアクセス可能なウェブサイトに結果を表示するなど、各種の方法により達成され、そこで結果が表示され、検索語と見つかった語がどの程度一致しているかをもとに（インターネット検索エンジンにより得られた検索結果と類似した方法で）ランク付けされる。代替的に、結果は、標準データベースにより、一致の品質および性能に関する統計とともに通信しうる。従って、結果に対してデータベースの演算を実施し、どのデータが承認され、どのデータが拒否されうるかについての基準を確立しうる。例えば、作成した基準は、希望の日付範囲内に該当する文書のみを承認する、および2つ以上の検索語を含む文書のみを承認するといったものとしうる。あるファイルについて検索結果が「ヒット」したとき、そのファイルを参照し、見つかった語および／またはフレーズの周辺の文脈全体を収集することができることが理解される。これは、ファイルまたは文書のテキストは、語参照からは再構成できないことがあるため、有益である。

【００４１】

本書に記載したアルゴリズムをすべてまたは一部を、専用のハードウエア装置（例えば、ゲートアレイ、ASICなど）の内部の論理回路を使用して実装することにより、高い性能を達成しうることが理解される。この装置は、ランダム（非順次的）な方法で非常に高速（例えば、アクセス当たり8ns未満）に同時進行的にアドレス付けできる2つのスタティックRAMのバンクに電気的に接続しうる。論理回路を実装したアルゴリズムは、UTF-8およびUTF-16シーケンスの構文解析に使用する文字プロセッサ、いくつかの特殊な取り扱い能力を有する語アナライザー、辞書自体を高速アクセスRAMに保存しうる辞書ルックアップ機能、語からハッシュ値を生成するためのハッシュ値生成装置、エントリー当たりおよび／または検索エンジン当たり、複数のランダムアクセスのために第1レベルのハードウエアベースの索引テーブルを維持する能力が含まれうる。その上、本発明の全体または一部は、C/C++など最新の高性能プログラミング言語に実装しうることが意図されている。

【００４２】

図7を参照するが、データの検索および索引付けのためのシステム500の1つの実施形態を図示しており、これには、性能を維持するために制御された方法で、540から複数のファイルを読み取ることができるファイルリーダー512が含まれる。ファイルリーダー512は、ファイル全体（またはファイルの大部分）をメモリバッファに読み取ることができ、メモリバッファがいっぱいのときには、ファイルリーダー512は、その次のファイルが存在すればそれを処理できる（または大きなファイルの次の部分を読み取る）。このように、ファイルリーダー512は、順次ファイル読み取りを実施してディスクの読み取り性能を最適化する一方、システムの残りの部分に、並列に処理しうるデータの同時発生的なファイルストリームを提供しうる。ファイルリーダー512は、非常に多様なファイルシステムから、システム固有のファイルシステムハンドラーなど各種の手段によりファイルを読み取る能力を有する。

【００４３】

ファイルリーダーの出力は、ファイル分析を行い、その型のファイルを適切に処理するために必要である場合、その必要に応じて適切なファイルタイプハンドラー518を呼び出

すファイルプロセッサ514の単一または複数の例を供給する。ファイルタイプハンドラー518はまた、共通の圧縮形式の圧縮解除を処理するように構成しうることも意図されている。よって、ファイルタイプハンドラー518は、ファイルストリームの圧縮解除ができ、また結果的に生じる非圧縮ファイルストリームをファイル処理装置514に戻すよう指図することができる。また、ファイルの型などの状況に応じて、ファイルは、別のファイルタイプハンドラー518に送信して、形式固有の取り扱いを行うか、または索引処理装置516に直接送信して、その後の処理に備えることもできる。一実施形態（図示のとおり）において、ファイルハンドラー518の出力は、オペレーティングシステム固有の装置ドライバー、およびPCI-XまたはPCI Expressなどの高性能バスインターフェースを含むハードウエアインターフェース520を介して、語ファインダーに供給される。

【００４４】

本書でこれまでに簡単に考察したとおり、索引処理装置516は、本書で考察した方法に対応して、データを（部分的または全体的に）処理するように構成されたFPGAとしうる。索引処理装置516は、フィールドプログラマブルゲートアレイ（FPGA）ならびに専用メモリ、複数のインターフェース、および電源サブシステムなどその他の望ましいハードウエアを含むPCI-XまたはPCI-expressボードを介して実装しうる。図8を参照するが、データファイルストリームおよびパラメータ化を、索引処理装置516に入力620を通して導入しうる。索引処理装置516は初め、語または数字の検出の処理、その語または番号の検証（語ファインダーの説明を参照）、およびメモリ制御装置624を介したテーブルへのこの語参照の転送をするよう実装できる検索プロセッサ622を含むように構成される。検索／検出プロセッサ622には複数コア能力を含みうることが意図されている。1つの望ましい実施形態において、複数の検索／検出プロセッサ622を利用して、複数の情報ストリーム（例えば、4つの検索／検出プロセッサについて、それぞれ毎秒125メガバイト、4つのファイルストリームから合計毎秒500メガバイトを処理する能力）を処理しうるアーキテクチャを生成することができる。

【００４５】

当然ながら、メモリ制御装置624は、スタティックRAM（SRAM）など、そのコンテンツへの非常に高速な真のランダムアクセスを許容するメモリに接続しうる。コンピュータに使用されている一般的なダイナミックRAM（DRAM）は、一群の語に非常に高速に、順次アクセスできるが、プロセッサが新しい群の語にアクセスする必要があるとき、メモリは、古い群を保存して、新しい群を引き出すために、著しく長い時間がかかりうる。ただし、索引処理装置516とともに使用されているメモリ制御装置624が、そのコンテンツ（スタティックRAM（SRAM）など）への非常に高速な真のランダムアクセスを許容する場合は、語およびテーブルのエントリーへは、DRAMよりも約10倍高速にアクセスできる。メモリ制御装置624によるほぼすべてのアクセスがランダム（ハッシュ値に基づくアクセスのアドレスを持つ）であるため、本発明による索引付けの実施は、SRAMなどの真のランダムアクセスのメモリのいずれを利用する際に非常に有益である。さらに、1つ（または複数）の別個にアドレス付け可能なメモリのバンクを使用することもできることが意図されており、ここで第1のメモリバンクは、ハッシュテーブルおよび各参照グループ内で次に利用可能なスポットのリストを維持するリストメモリ630とすることができ、および第2のメモリバンクは語参照を維持する参照メモリ632とすることができる。検索方法で索引付け（例えば、線形検索法）を必要としない場合は、リストメモリ630は使用または維持されないことがある。当然ながら、これらのメモリバンクは、同時にアクセス可能でも、そうでなくてもよい。これにより、毎秒5,000万個を超える語参照を保存しうる。

【００４６】

索引処理装置516は、メモリバンクがいっぱいになったときに異なるモードにシフトするように構成することができ、ここで、索引処理装置516は参照メモリ632のコンテンツをホストコンピュータに移動（またはダンプ）できる。これは、参照ダンプユニット636により達成しうる。索引処理装置516によりこのデータが移動またはダンプされるとき、索引処理装置516は、メモリ534に効率よく保存できる並べ換えされたグループとしてタスク

を完了しうる。移動（またはダンプ）されたグループは、ファイラー538により取り扱われ、ここでファイラー538は、ハッシュ値の一致があるとき新しい参照を既存のグループに追加でき、希望に応じて新しいハッシュ値について新しいグループを生成できることが意図されている。ホストコンピュータは、数ギガバイトのメモリ534を持つことができ、また10億個を超える参照を保持する能力を持ちうる。メモリ534によって索引テーブルがいっぱいになると、それが高性能ディスクアレイ550に書き込まれ、また新しい索引テーブルが初期化され、新しい索引テーブルを使用して処理が続行されると、ディスクアレイ550への書き込みが起こる。索引テーブルは、連続した大規模なメモリブロックとして書き込むこともでき、これによりディスクへのこのデータの非常に高速な書き込みが許容される。当然ながら、上記に説明したプロセスは、ソースボリューム540上にあるすべてのデータまたは選択したグループのデータの処理が完了するまで続行される。索引付けが完了すると、複数の大きな索引テーブルをメモリディスクアレイ550上に配置させることができる。

【００４７】

次に、索引付けされた検索を実施する従来的な方法などにより、検索を希望に応じて完了させうる。検索プロセスを効率の高いものにするために、パラメータ化（つまり、ファジーの程度、順序、局所性、など）をするしないにかかわらず、検索語、フレーズ、および／または数字を、入力したり、ファイルから読み込んだりすることができる。次に、索引付けシステムは、現行の索引テーブルの使用中に次の索引テーブルをディスクアレイ550から読み込むことにより、これらの検索語をバッチで処理しうる。検索語に対するハッシュ計算は、随意的に索引処理装置516により加速することもできるが、索引テーブルの処理は、希望に応じて、ソフトウエアおよび／またはハードウエアによって完了することができる。

【００４８】

本書で説明したシステムは、線形検索または語参照もしくはファイルデータの一致検出にも対応できることが理解される。この線形検索能力は基本的に、人に検索語のリストと山積みの文書を与え、検索語を探しながら文書全体を読んで、文書、場所、および見つけたものを識別するといった、文書を見つけ出す旧来の方法を真似たものである。例えば、線形検索では、ユーザーは、前もって何を探すかを知っている場合があり、またそうであるため、線形検索により、まさしくユーザーが探しているものの検索において、ソフトウエア／ハードウエアの検索が積極的なものとなる。これは、それらがどこにあり、またどのように書式設定されているかにもよるが、最終的に検出されるまたは検出できない略語または数字を、ユーザーが探している場合に有用である。よって、索引処理装置716は、図9に示すとおり、従来のCPUよりも実質的に高速で線形検索を実施するよう構成することができる。これは、線形検索で従来型のCPUを使用するとき、検索の速度は、検索語数の増加にともない直線的に遅くなるため有益でありうる。例えば、ファジー一致での単一の用語の検索は、文字当たり10ns近くかかり、最大検索レートが約100MB/sec（100用語を検索する場合、検索時間は約1MB/secまで遅くなる）となるが、ここで線形処理装置716を使用すると、ファジー検索は最高100用語を同時に実施でき、また全体的な処理速度100MB/sec（100倍の速度増加）が維持される。

【００４９】

その上、検出プロセッサ722は、語、フレーズ、および／または数字のいずれかとしうる定義可能な検索語に対して完全一致またはファジー一致のいずれかを試みることで、語参照の入力ストリームの線形検索を実施しうることが意図されている。1つの望ましい実施形態において、それぞれが複数の検索語の検出が可能な複数の検出プロセッサ722を利用できる（並列および／または直列）。例えば、複数の検出プロセッサ722のそれぞれが16個の検索語を検出するよう構成されている場合、これにより、16の検出プロセッサを実施した場合に、最高256個の検索語ができるようになる。それぞれが、1クロックサイクル当たり1文字を処理でき、100MB/secを超える検索を可能にする。線形検索語が供給されている場合には、入力される文字がそれぞれの検索語に対して連続的および同時に比較され

、ヒットが発生すると、用語の識別およびそのヒットの場所を、ホストコンピュータへの転送を待機するために検索バッファ728に保存することができる。このプロセスを、図10に、語「FOOTBALL GAME」を用いて例証する。線形検索の処理装置716には、ヒットまたは参照を構造化された方法で保持・バッファする内部メモリが含まれうることが意図されている。

【００５０】

本書で説明した方法は、ファイルデータのボリューム内で語と見なされているものの検索の結果として見つかり、以前から保存されている語参照に対して適用でき、および／または未加工のファイルまたはデータストリーム入力に対して適用できることが理解される。未加工のファイルまたはデータストリーム入力の場合、候補語は検索語の文字セットに入っていない任意の文字を探すことにより、断片化しうる。文字セットに一致するデータの断片は次に、語参照が処理される方法と非常に類似した方法で、検索語に対して比較されうる。単に語を分割するこの方法は、有効な語の分離を試みる方法とは異なる。その結果、この方法では、語として見なされるすべてを保存する必要があり、よっていくらかのコストがかかる語ファインダーの場合に比べて、ランダムな断片の処理には実際に全くコストが関与しないため、より多くの処理用の文字シーケンスが容認される。

【００５１】

さらに、本書で開示したハードウエアシステムには、高性能のファイルレベルdeduperを実装することができるが、これには、ハードウエア索引処理装置516を異なるモードで利用しうる。この場合には、索引処理装置516は、安全ハッシュアルゴリズム（SHA）ハッシュ計算エンジンとしての役目を果たすよう構成でき、ここでdeduperは約500MB/sec（またはそれ以上）の処理ができ、重複データファイルの処理を避けるために処理システムに転送されうる結果ファイルの生成ができる。代替的に、deduperをスタンドアロンとして実行して、単に重複を削除した（deduped）データを提供することもできる。従って、本発明は、重複の削除、索引付けおよび／または線形検索に利用することができ、ここでシステムはバランスのとれた効率の高いソフトウエアおよびハードウエアの組み合わせを利用する。システムは、ホストマシンのメインプロセッサおよびメモリサブシステムが処理で塞がってしまうことから開放されるよう、その性能を深刻に制限しうる処理のボトルネックを防止するようバランスがとられる。その代わりに、CPUが効率よくI/O、ファイル分析、圧縮の解除、特殊なファイルの処理または変換、および結果管理を、最小の待ち時間で処理し、データのスムーズな流れが維持される。

【００５２】

図9を再び参照するが、線形検索検出プロセッサ716は、語のストリームを受けて、それぞれの語を大きなグループの検索語に対して比較して、一致を検出するように構成されたハードウエア実装システムである。これは、複数の検索検出プロセッサ722を持つことにより達成しうるが、それぞれが複数の語を検索する能力を持ちうる。複数の検索検出プロセッサ722および複数の語の間の関係により、入力語を数百（またはそれ以上）の検索語に対して非常に高速に、一般に数プロセッサクロックサイクルで比較できる拡張可能なシステムを構築できる。この同一のタスクを従来型の汎用CPUを使用して行うと、1語当たり同様に数クロックサイクルがかかるが、この実行時間には、検索語の数を掛ける必要がある。その結果、従来型CPUは、著しく高速のクロックレートで実行されているとしても、同一の結果を達成するには、1桁またはそれ以上長い実行時間がかかることになる。

【００５３】

比較は、必ずしも正確でなくてよく、また、検索語に軽い綴り間違いまたはタイプミスが含まれている場合でさえも、その語を一致として承認できる、いわゆる「ファジー」であってもよい。一致プロセスがどの程度に厳格または寛大であるかは、語単位でなど、希望に応じてパラメータ化することができる。これは、あまり一般的ではない一部の語が、1文字変化すると、非常に一般的な語と一致するようになるときに有用であり、この場合は容認可能な結果とはしたくはない。

【００５４】

本発明によれば、線形検索検出プロセッサ716は、語を組み立てるように構成しうる。注目すべきは、上記に説明したとおり、語は既にフレーム化されていて、語参照として保存されていることである。ところが、処理の対象のデータがファイルまたはデータストリームである場合には、語を探す必要がある。一般に、このプロセッサに実際の語ではない可能性のある任意の文字シーケンスを提示するのには、リソースコストはかからないため、本書で説明したものよりも単純なアルゴリズムを使用できる。アルゴリズムは、単純に、文字または数字ではない任意のキャラクタで区切られた文字または数字を、グループ化して、それらを語としてフレーム化することができる。文字の正規化はなおも適用できるが、その結果、UTF-8およびUTF-16デコーダが依然として必要となる。ほとんどの言語は、256コード内の大文字および数字のセットで表現しうるため、語はその後、文字／数字について16ビットの正規化された文字コードから、8ビットに変換しうる。この時点で、フレーム化した候補語が検索検出プロセッサ716のそれぞれのコアまたはインスタンスに導入されるが、ここでそれぞれのコアが検索エンジンであると考慮しうる。線形検索検出プロセッサ716内で利用できる素材の量によるが、8から最大256の検索エンジンを実装しうる。検索エンジンを増やしても、実際には処理速度は上がらないが、より多くの数の検索語が可能となる。各検索エンジンには、8～32の検索語および各検索語に伴うパラメータ化が読み込まれうる。

【００５５】

各エンジン内の各検索語の始め（最初の4～6文字）が、候補語の初めに対して比較される。この比較は、下記に説明するとおり、軽度のスペルミスおよび誤字を許容するために、非常に多様な異なる組み合わせについて実行しうる。この比較により、一致が全く見つからないまたは1つ以上の一致が見つかる可能性が生じる。一致が全く識別されない場合、任意の検索語に対する一致が全くなく、そのためこのエンジンによる候補語の処理が完了する。示された候補語および検索語以外の一致が1つのみ識別された場合、当初の一致をここでさらに考慮する必要がある。2つ以上の一致が識別された場合、これにより検索エンジンによるさらなる処理が可能ではないということを示す例外が生成される。候補語およびエンジンのインスタンスが例外に記録され、例外情報がホストコンピュータに戻される。その後、複数の一致の状況を処理するために、本書に記載したものと類似した比較的速度の遅いソフトウエアアルゴリズムが使用される。一般に、各検索エンジンに供給された検索語が最初の6文字で互いに異なる場合には、複数の一致およびそれにより生成される例外は、まれである。類似した語が検索語の全体的なグループ内に存在する場合、類似した語は、この潜在的な問題を回避するために、異なる検索エンジン間に分割されるべきである。

【００５６】

一例として、14個の文字比較を表1に詳述したとおり実施される状況を考慮してみる。比較器により、検索語中の指定された文字が候補語中の指定された文字と比較される。各比較には、文字の指定があり、下記の表中のCPは、語内の文字の位置を意味する。これらの14個の比較結果は、一致を判別するためのグループ（表2を参照）に組み合わされる。各グループでは、表1にある4つの比較が考慮され、ここで候補語および検索語が一致していると考慮されるには、4つの比較のうち3つが正しいものとされなければならない。組み合わせの結果は、候補語の開始文字を判別するためにも使用できる。これは、意図された語には属さない語の先頭にある文字となった余分なバイトが、語のフレームに含まれている可能性があるためである。

【表１】

比較	CP 検索語	CP 候補語
A	1	1
B	2	2
C	3	3
D	4	4
E	2	3
F	3	4
G	4	5
H	3	2
I	4	3
J	1	2
K	1	3
L	2	4
M	3	5
N	4	6

【表２】

比較	開始文字
ABCD	1
AEFG	1
ABFG	1
ABCG	1
ABHI	1
ABCI	1
JEFG	2
KLMN	3
JLMN	2
JEMN	2
JEFN	2
JECD	2
KLFG	3

【００５７】

１つの一致が見つかったと仮定して、次のステップに進む。その他のエンジンは、この

エンジンとは独立的に作動し、異なる結果を持ちうることが理解される。最初の一致候補が検出されると、候補語が選択した検索語と十分に近接して一致しているかどうかを判別するために、より高度な分析を実施しうる。これには、検索語内の各文字について文字の属性ペアの読み取り（例えば、高速オンチップメモリから）が関与し、ここでその文字は一致させる適切な文字である。属性により、代用の文字が許容されるかどうかが判別されるが、異なる一般的な語を生成することになる限定数の代用を除外することもできる。同様に、属性により、余分な文字が許容されるかどうかが判別でき、ここでも追加された場合に、異なる一般的な語を生成することになる限定数の代用が除外される。属性は、この文字のスキップが許容されるかどうか、またはこれを実行した場合に異なる一般的な語を生成することになるかどうかを示すこともできる。これらの属性を生成する分析は、（検索語を受け取った後で）ソフトウエア内で、考えられるすべての文字の代用を試行し、文字を追加し、また文字の組み合わせをスキップし、各試行の結果を一般的な言語固有の語の辞書に対して照合するというプロセスを用いて実施しうる。それがこの辞書にヒットした場合には、その代用が許容されないものとしてセットされうる。

【００５８】

単一文字の比較が、属性によって許容された代用を含めてすべて成功した場合には、各カテゴリーで承認され、かつ合計でこの語に対する限度のパラメータを超えていない代用の数を確認するためのチェックを実施しうる。一般に、長めの語は、より多くの代用が許容され、一方、短めの語は、おそらく合計1個の代用しか許容されない。比較が完全かつ首尾よく行われると、候補語が、その場所、検索エンジンの索引、および検索語の索引とともにホストコンピュータに逆供給される。この情報が次に、希望に応じてさらなるアルゴリズムによる分析用に使用されるか、または本書に記載したとおりフレーズ比較の評価のために適切に保存するために使用される。本書で開示した方法は、データストリームにも適用でき、ここでデータストリームは、データファイルに論理的に分割されうることが理解される。

【００５９】

数字は、ちょうど語が検索語に対して比較されるように検索する数字に対して比較でき、および／または検索が数字の一部について実施することもでき（特定の市外局番など）、および／または検索が下限および上限の間にある数字の整数部分について実施できるため、数字は著しく異なる方法で取り扱いうることが理解される。最後の2つの代替物の場合には、これにはこの異なる機能性を備えた異なる種類の検索エンジンが必要である。数字は語よりもずっと頻度が少ない傾向にあるため、数字は分配器内の異なる待ち行列に進むことがあり、これらの数字検索エンジンは少数のみが実装されうるが、例えばこれは一般的に、構造内に実装される語検索エンジンの4分の1である。また当然ながら、本発明は複数の自然な構造の言語（例えば、英語、フランス語、ロシア語など）での語を含むデータおよび／またはデータファイルでも使用できる。この場合には、語は、一語ずつ処理されることも、またはデータ／データファイルが各言語について別個に処理されることもある。

【００６０】

さらに、性能はこのデザインの実施に依存したものとなりうる。ところが、想定内において、これは最新のFPGAの論理回路として実装することができ、このプロセスの第一および第二の部分は連結でき、またその結果、これらがこのプロセスの第三および第四の部分よりも早く達成できるため、実際には性能に対して何の影響もないと考えられる。第三の部分は、検索語に高速のオンチップメモリからアクセスでき、また1クロックサイクルで2個の検索語のうち最初の4文字にあたる最高72ビットの情報にアクセスできる。一実施形態において、メモリから読み取られた2個の検索語は、各検索エンジンにおいて同時に、かつ1クロックサイクルで処理できる。例えば、16個の検索語が読み込まれた場合、比較および結果は、8クロックサイクルで得られうる。平均的な些細でない語は、約8文字であるため、その結果、平均性能は、1クロックサイクル当たり1文字となる。検索語の1つが一致した場合、検索エンジンは異なるモードに入ることがあり、ここでこのプロセスの第

四の部分で記載した内容が実行されうる。これには、選択された検索語に含まれる各文字について72ビットの語（文字に属性を加えたもの）の取り出し、および必要な比較の実施が関与しうる。従って、文字は、クロックサイクルごとに処理され、これは平均8文字の語について平均8クロックサイクルとなる。

【００６１】

ただし、これらの余分な8クロックサイクルは、第三の部分で検索語と一致のあった検索エンジンによってのみ使用されるため、これらの余分な8クロックサイクルは、文字／語当たり必要な平均時間を倍増することはないと考えられる。これは、一般的に、検索エンジンのわずか数パーセントであり、その結果、語当たり追加されるクロックサイクルの平均数は、2未満となる。各入力語は、それぞれの検索エンジンが平行して演算しうるように、それぞれの検索エンジンに導入できることが理解される。従って、それぞれの検索エンジンは、その他の検索エンジンが「見る」ものと同じ語セットを「見る」ことになる。第三の部分での一致は、特に検索語として使用された一般的な語がよく分布している場合に、検索エンジン間で負荷バランスがとられる傾向があることが考えられるため、ほとんどの場合に、どれか1つの検索エンジンが著しく処理を遅くすることはない。ただし、分配器の待ち行列がいっぱいになり、追いつくべき大量のバックログのある検索エンジンを許容するために、残りのすべての検索エンジンについて処理を中止させる必要がある状況は考えられる。検索エンジンの数は実際には、利用できるシリコン構造の量によってのみ制限される。検索エンジン当たり16個の検索語は望ましい実施形態であるが、検索エンジン当たりその16個より多いまたは少ない検索語も導入しうる。利用可能な検索エンジンにセットできる検索語がこれより多い場合には、検索語を、2つ以上のグループに分けて、入力データを妥当な大きさのブロック（数百メガバイト）にバッファすることができ、その後で検索エンジンを第1のグループにセットアップして、バッファのデータを供給しうる。終了したら、第2のグループの検索語を、検索エンジンにセットでき、バッファしたデータが再び処理される。これを、すべてのグループの検索語が使用されるまで続行しうる。次に、新しいバッファのデータを読み込み、同じ方法で処理しうる。これにより、検索語のグループの数の分だけ処理が遅くなる。ただし、多くの状況において、必要とされる検索語は利用可能な検索エンジンに適合できるため、これを必要としない。

【００６２】

本発明に従い、コンピュータ ファイルまたは検索対象の語（つまり検索語）のファイル形式、構造および／またはレイアウトが分からない場合の、任意のコンピュータファイル内の語を探す全体的なプロセスを提供する。語を検索する既存の方法では、検索語が確立された後で、ファイルのうちそれらのセグメント内のすべての語を収集し、および通常は索引付けされた方法で将来的な検索用に保存できるように、検索対象の語の特性が分かっているか、または語のある場所が分かっているかのいずれかが必要である。これらの2つの条件の1つが要求されないことで、システムの複雑さが低減され、柔軟性が向上し、一般に精度が向上する。ファイル形式に関連して実際には語ではない数多くの用語が保存される際に失われた効率が、検索機能のハードウエア実装により補われうる。

【００６３】

本書の技法で提供した記憶装置および索引付けは、ファイル内で見つかった語を標的としており、これは、大量の参照が急速に生成されるその他の用途にも使用できる。この技法には、以下のとおり、いくつかの注目すべき点がある。

・・・この技法は、一致が許容されるように、語中に綴り間違いまたはタイプミスがある場合でさえも、複数のハッシュを生成し、それにより語の索引付けをするために使用された。

・・・索引の一部によりサブテーブルが選択された後、残りの部分がその索引エントリーに保存されて、サブテーブルを高速に選択する検索と、その後の高性能な線形検索が可能となる、部分的索引付けソリューションを、サブテーブル内の要素について実行できる。このアーキテクチャにより、この点について下記に説明した効率、単純性、性能、および拡張性のための環境が創出される。

・・・複数の索引が1つの語に対する多対一対応は、より大きな単一の語参照を指し示す32ビットの値で効率よく保存できる。
・・・メモリ使用量を最適化する参照および索引記憶装置技術の効率は、高速SRAMが非常に高価であるため、重要である。
・・・技法の単純性は、複雑なメモリ管理は一切必要なく、また固定ビット幅の要素を使用するため、これは実現可能な記憶および管理アルゴリズムのハードウエア実装に適している。
・・・高速SRAM内に保持されている小さなテーブルを、ホストコンピュータのメインメモリ内にある同一構造のより大きなテーブルと効率よく結合することができるという点での、この技法の拡張性。より大きいこれらのテーブルは、表の未修正の画像として線形の方法でディスク外に高速に保存でき、それらがディスクに高速保存され、プロセスへの読み込みやプロセスの検索段階での利用できるようにする。
・・・ほとんどの小さなランダム（非順次的）メモリアクセスは、語ファインダープロセッサに接続された高速SRAM内部で発生し、ここで、ランダムメモリアクセスについて性能のペナルティを払う必要がないという点での、プロセスの性能。SRAM内の小さなテーブルがいっぱいになると、必要なランダムメモリアクセスの数を最小化する方法でホストコンピュータのメインメモリに転送される。一般的なデータでは、メインメモリへのメモリアクセスのうち2%がランダムであって、ホストコンピュータメインメモリへのランダムメモリアクセスに関連して重大な性能ペナルティがあるとき、メインメモリテーブルの格納の性能が劇的に最適化される。

【００６４】

開示されているのは、ハードウエアベースの線形検索技法であり、ハードウエアベースの一致検出プロセッサ内に実装できる複数のファジー（完全一致ではない）検索がその概念に含まれていると言う点でユニークである。この実装により、汎用CPUを用いて可能であったものに比べ、これらのタイプの検索が劇的にスピードアップする。この技法の導入により、ゲートアレイ内の論理回路が、チップ上に存在する小ブロックの非常に高速なアクセスの記憶装置と組み合わされ、使用する構造のスペース量および時間の両方において効率の高い実装が達成される。この処理能力の存在により、索引付け検索をする中心的な環境が造成され、データまたは（語ファインダーにより見つかった語のような）データの認識されたサブセットが検索できるため、大量のデータを検索する必要がなくなり、データがディスクの記憶装置から読み取ることができる速度に十分匹敵する速度で複数のファジー検索語の検索ができる。これによって、索引付けに関連する複雑さおよび非効率さが最低限に抑えられる。その上、ファイルまたはファイルを現すセクションに論理的に分割されたデータストリームに対して処理が実行できることが理解される。データストリームは、ネットワーキング装置から、または非常に多様な電子通信装置からのものがある。本書で用語ファイルまたはデータファイルが使用される場合、このファイルまたはデータファイルがデータストリームの適切な論理的セクションであることをも言及しうる。

【００６５】

さらに、本発明の各要素は、部分的に、または全体を、希望の最終目的にとって相応しい任意の順序で実装しうる。模範的な実施形態によれば、本発明の方法を実施するために必要な処理の全体またはその一部は、機械可読コンピュータプログラムに呼応して演算されるコントローラにより全体的または部分的に実装しうる。所定の機能および希望の処理、またそのための演算（例えば、実行制御アルゴリズム、本書に記載した制御プロセス、その他）を実施するために、コントローラには、限定されないものの、プロセッサ、コンピュータ、メモリ、記憶装置、レジスタ、タイミング、割り込み、通信インターフェース、および入出力信号インターフェース、ならびに上述のうち少なくとも1つを含む組み合わせが含まれる。また、当然ながら、本書で開示した実施形態は、例証のみを目的とするもので、本発明により意図された考えられるいくつかの実施形態が含まれるのみである。

【００６６】

その上、本発明は、コンピュータシステムまたはコントローラに実装されたプロセスの

形態で全体的または部分的に実施しうる。いずれのタイプのコンピュータシステム（当技術で既知のもの）および／またはゲーム用システムを使用することができ、また本発明は、限定されないものの、LANおよび／またはWAN（有線または無線）を含むいずれのタイプのネットワーク設定を経由して実装しうることが理解される。本発明はまた、フロッピー（登録商標）ディスク、CD-ROM、ハードドライブ、および／またはその他のいずれのコンピュータ読み取り可能媒体など、有形の媒体に実施された命令を含むコンピュータプログラムコードの形態で実施できるが、ここで、コンピュータプログラムコードがコンピュータまたはコントローラによって読み込み・実行されるとき、コンピュータまたはコントローラは、本発明を実施するための装置となる。本発明はまた、例えば、記憶媒体に格納された、コンピュータまたはコントローラによる読み込みおよび／または実行がなされる、または電線またはケーブル、光ファイバー、または電磁放射など何らかの通信媒体によって転送されるコンピュータプログラムコードの形態で実施できるが、ここでコンピュータまたはコントローラによるコンピュータプログラムコードの読み込み時および実行時に、コンピュータまたはコントローラは、本発明を実施する装置となる。汎用マイクロプロセッサ上に実装するとき、特別な論理回路を生成するために、コンピュータプログラムコードのセグメントによりマイクロプロセッサを構成しうる。

【００６７】

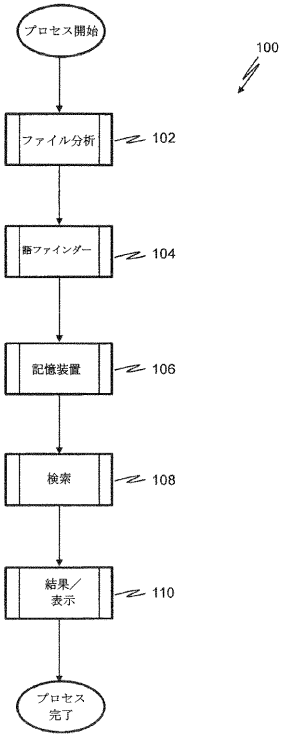
本発明について模範的実施形態を参照しながら説明してきたが、当業者であれば、本発明の範囲を逸脱することなく、その要素について各種の変更をなしうること、および同等物で代用しうることが理解される。さらに、本発明の教示に特定の状況または材質を適応させるために、その範囲を逸脱することなく、数多くの修正をなしうる。従って、本発明は、この発明を実行するために意図された最良の態様として開示された特定の実施形態に限定されず、本発明には、添付した請求項の範囲に該当するあらゆる実施形態が含まれることが意図される。また、第一、第二などの用語の使用は、具体的な記述がない限り、いかなる順序または重要性を示すものでなく、むしろ第一、第二などの用語は、1つの要素を別の要素と区別するために使用している。

【符号の説明】

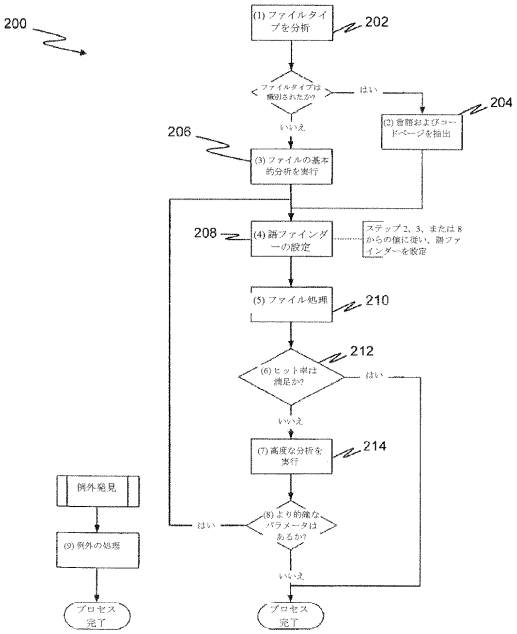
【００６８】

５００・・・システム
５１２・・・ファイルリーダー
５１４・・・ファイル処理装置
５１６・・・索引処理装置
５１８・・・ファイルタイプハンドラー
５２０・・・ハードウエアインターフェース。

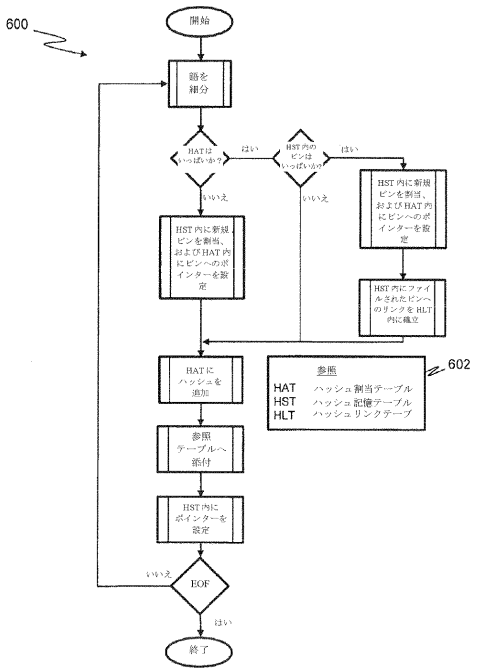
【図１】


プロセス開始
100
ファイル分析
102
語ファインダー
104
記憶装置
106
検索
108
結果／表示
110
プロセス完了

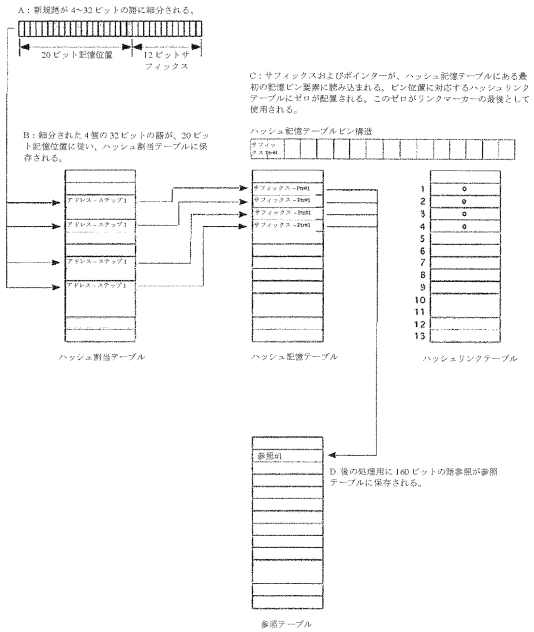
【図２】


200
(1) ファイルタイプを分析
202
ファイルタイプは識別されたか?
はい
いいえ
(2) 言語およびコードページを抽出
204
206
(3) ファイルの基本的分析を実行
208
(4) 語ファインダーの設定
ステップ2、3、または8からの値に従い、語ファインダーを設定
(5) ファイル処理
210
(6) ヒット率は満足か?
212
(7) 高度な分析を実行
214
例外発見
(8) より的確なパラメータはあるか?
(9) 例外の処理
プロセス完了
プロセス完了

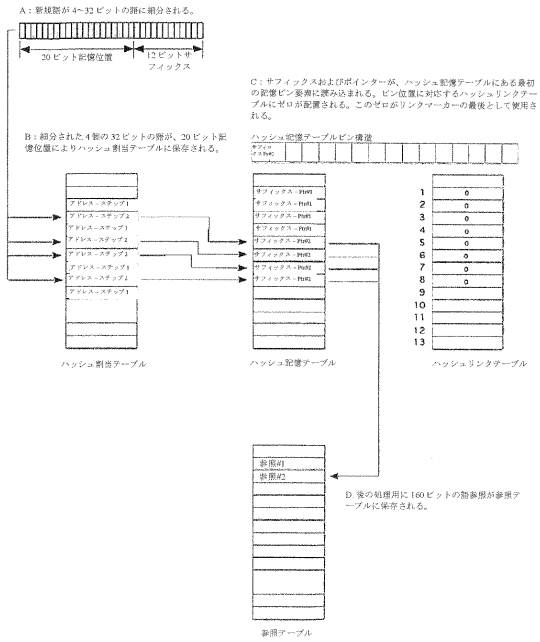
【図２Ａ】


600
開始
語を細分
HAT はいっぱいか?
はい
いいえ
HST 内のビンはいっぱいか?
はい
いいえ
HST 内に新規ビンを割当、および HAT 内にビンへのポインターを設定
HST 内に新規ビンを割当、および HAT 内にビンへのポインターを設定
HST 内にファイルされたビンへのリンクを HLT 内に確立
HAT にハッシュを追加
参照テーブルへ添付
HST 内にポインターを設定
EOF
いいえ
はい
終了
参照
HAT ハッシュ割当テーブル
HST ハッシュ記憶テーブル
HLT ハッシュリンクテーブル
602

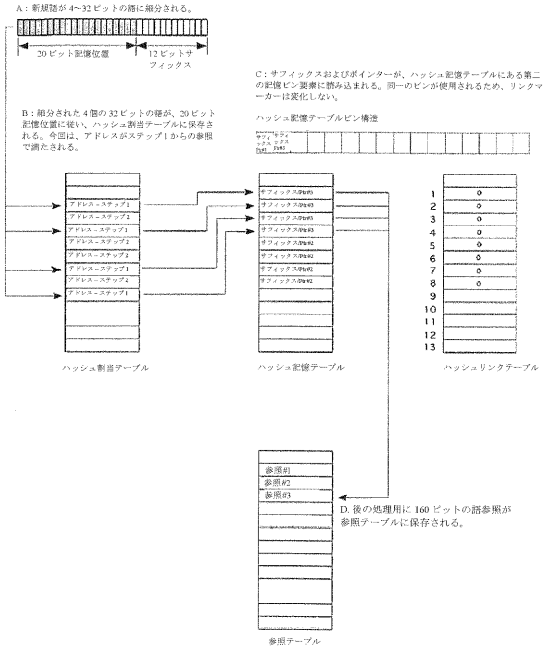
【図３】


A：新規語が 4~32 ビットの語に細分される。
20 ビット記憶位置
12 ビットサフィックス
C：サフィックスおよびポインターが、ハッシュ記憶テーブルにある最初の記憶ビン要素に読み込まれる。ビン位置に対応するハッシュリンクテーブルにゼロが配置される。このゼロがリンクマーカーの最後として使用される。
ハッシュ記憶テーブルビン構造
B：細分された 4 個の 32 ビットの語が、20 ビット記憶位置に従い、ハッシュ割当テーブルに保存される。
アドレス - ステップ 1
サフィックス - Ptr#1
サフィックス - Ptr#2
サフィックス - Ptr#3
サフィックス - Ptr#4
ハッシュ割当テーブル
ハッシュ記憶テーブル
ハッシュリンクテーブル
参照#1
D：後の処理用に 160 ビットの語参照が参照テーブルに保存される。
参照テーブル

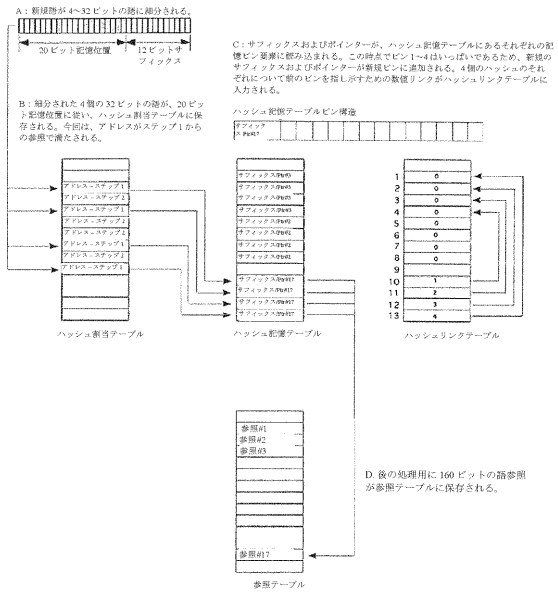
【図４】


A：新規語が 4~32 ビットの語に細分される。
20 ビット記憶位置
12 ビットサフィックス
B：細分された 4 個の 32 ビットの語が、20 ビット記憶位置によりハッシュ割当テーブルに保存される。
C：サフィックスおよびポインターが、ハッシュ記憶テーブルにある最初の記憶ビン要素に読み込まれる。ビン位置に対応するハッシュリンクテーブルにゼロが配置される。このゼロがリンクマーカーの最後として使用される。
ハッシュ記憶テーブルビン構造
ハッシュ割当テーブル
ハッシュ記憶テーブル
ハッシュリンクテーブル
参照テーブル
D. 後の処理用に 160 ビットの語参照が参照テーブルに保存される。

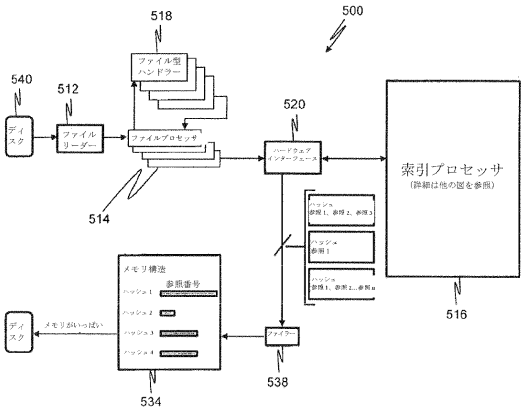
【図５】


A：新規語が 4~32 ビットの語に細分される。
20 ビット記憶位置
12 ビットサフィックス
B：細分された 4 個の 32 ビットの語が、20 ビット記憶位置に従い、ハッシュ割当テーブルに保存される。今回は、アドレスがステップ 1 からの参照で満たされる。
C：サフィックスおよびポインターが、ハッシュ記憶テーブルにある第二の記憶ビン要素に読み込まれる。同一のビンが使用されるため、リンクマーカーは変化しない。
ハッシュ記憶テーブルビン構造
ハッシュ割当テーブル
ハッシュ記憶テーブル
ハッシュリンクテーブル
参照テーブル
D. 後の処理用に 160 ビットの語参照が参照テーブルに保存される。

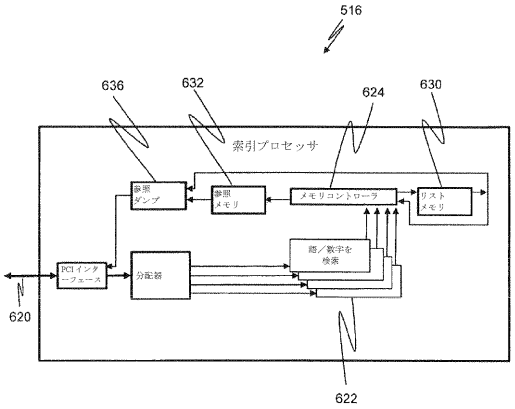
【図６】


A：新規語が 4~32 ビットの語に細分される。
20 ビット記憶位置
12 ビットサフィックス
B：細分された 4 個の 32 ビットの語が、20 ビット記憶位置に従い、ハッシュ割当テーブルに保存される。今回は、アドレスがステップ 1 からの参照で満たされる。
C：サフィックスおよびポインターが、ハッシュ記憶テーブルにあるそれぞれの記憶ビン要素に読み込まれる。この時点でビン 1~4 はいっぱいであるため、新規のサフィックスおよびポインターが新規ビンに追加される。4 個のハッシュのそれぞれについて前のビンを指し示すための数値リンクがハッシュリンクテーブルに入力される。
ハッシュ記憶テーブルビン構造
ハッシュ割当テーブル
ハッシュ記憶テーブル
ハッシュリンクテーブル
参照テーブル
D. 後の処理用に 160 ビットの語参照が参照テーブルに保存される。

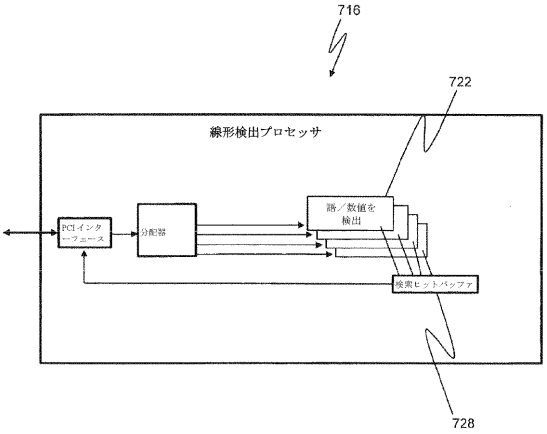
【図７】


500
518
540
512
520
ファイル型ハンドラー
ディスク
ファイルリーダー
ファイルプロセッサ
ハードウェアインターフェース
索引プロセッサ
（詳細は他の図を参照）
514
メモリ構造
参照番号
メモリがいっぱい
メイナー
538
534
516

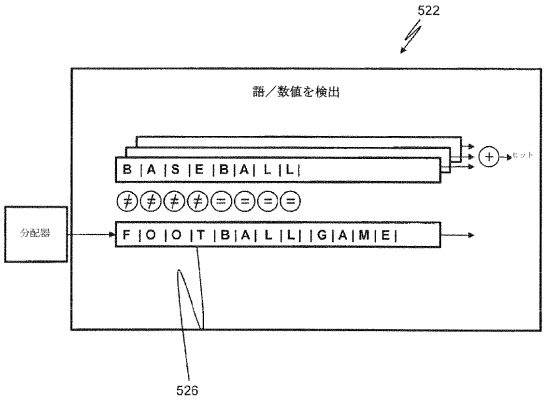
【図８】


516
636
632
624
630
索引プロセッサ
参照ダンプ
参照メモリ
メモリコントローラ
リストメモリ
語／数字を検索
PCIインターフェース
分配器
620
622

【図９】


716
722
線形検出プロセッサ
PCIインターフェース
分配器
語／数値を検出
検索ヒットバッファ
728

【図１０】


522
語／数値を検出
B | A | S | E | B | A | L | L |
ヒット
≠ ≠ ≠ ≠ = = = =
分配器
F | O | O | T | B | A | L | L | | G | A | M | E |
526

【国際調査報告】

PCT/US2009/000691 16.03.2009

PATENT COOPERATION TREATY

PCT

INTERNATIONAL SEARCH REPORT

(PCT Article 18 and Rules 43 and 44)

Applicant's or agent's file reference 5303.112957	**FOR FURTHER ACTION**	see Form PCT/ISA/220 as well as, where applicable, item 5 below.
International application No. PCT/US 09/00691	International filing date *(day/month/year)* 02 February 2009 (02.02.2009)	(Earliest) Priority Date *(day/month/year)* 01 February 2008 (01.02.2008)
Applicant The Oliver Group		

This international search report has been prepared by this International Searching Authority and is transmitted to the applicant according to Article 18. A copy is being transmitted to the International Bureau.

This international search report consists of a total of 2 sheets.

☐ It is also accompanied by a copy of each prior art document cited in this report.

1. **Basis of the report**
   a. With regard to the **language**, the international search was carried out on the basis of:
      ☒ the international application in the language in which it was filed.
      ☐ a translation of the international application into _____ which is the language of a translation furnished for the purposes of international search (Rules 12.3(a) and 23.1(b)).
   b. ☐ This international search report has been established taking into account the **rectification of an obvious mistake** authorized by or notified to this Authority under Rule 91 (Rule 43.6*bis*(a)).
   c. ☐ With regard to any **nucleotide and/or amino acid sequence** disclosed in the international application, see Box No. I.
2. ☐ **Certain claims were found unsearchable** (see Box No. II).
3. ☐ **Unity of invention is lacking** (see Box No. III).
4. With regard to the **title**,
   ☒ the text is approved as submitted by the applicant.
   ☐ the text has been established by this Authority to read as follows:
5. With regard to the **abstract**,
   ☒ the text is approved as submitted by the applicant.
   ☐ the text has been established, according to Rule 38.2(b), by this Authority as it appears in Box No. IV. The applicant may, within one month from the date of mailing of this international search report, submit comments to this Authority.
6. With regard to the **drawings**,
   a. the figure of the **drawings** to be published with the abstract is Figure No. 7
      ☐ as suggested by the applicant.
      ☐ as selected by this Authority, because the applicant failed to suggest a figure.
      ☒ as selected by this Authority, because this figure better characterizes the invention.
   b. ☐ none of the figures is to be published with the abstract.

Form PCT/ISA/210 (first sheet) (April 2007)

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US 09/00691

A. CLASSIFICATION OF SUBJECT MATTER
IPC(8) - G06F 7/00 (2009.01)
USPC - 707/3
According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
USPC: 707/3

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched
USPC: 707/1-3,6

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)
PubWEST(USPT,PGPB,EPAB,JPAB); DialogPRO; WIPO, EPO, CITESEER, Google patents, Google scholar Search Terms Used: search$ near3 index$, data, word, data format near2 unknown, format near2 unknown, identif$ near2 word$, natural near3 language, search word, search term, result, word near group$, group near3 word

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 2005/0198070 A1 (LOWRY) 08 September 2005 (08.09.2005) entire document especially abstract, para [0021]-[0030], para [0077]-[0080], para [0172]-[0174], para [0184], para [0222], [0232]-[0233], para [0279], para [0301]-[0308], para [0329]-[0330].	1-20
A	US 2007/0260450 A1 (SUN) 08 November 2007 (08.11.2007) entire document	1-20

☐ Further documents are listed in the continuation of Box C. ☐

* Special categories of cited documents:
"A" document defining the general state of the art which is not considered to be of particular relevance
"E" earlier application or patent but published on or after the international filing date
"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
"O" document referring to an oral disclosure, use, exhibition or other means
"P" document published prior to the international filing date but later than the priority date claimed
"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
09 March 2009 (09.03.2009)	16 MAR 2009
Name and mailing address of the ISA/US Mail Stop PCT, Attn: ISA/US, Commissioner for Patents P.O. Box 1450, Alexandria, Virginia 22313-1450 Facsimile No. 571-273-3201	Authorized officer: Lee W. Young PCT Helpdesk: 571-272-4300 PCT OSP: 571-272-7774

Form PCT/ISA/210 (second sheet) (April 2007)

フロントページの続き

(81)指定国 AP(BW,GH,GM,KE,LS,MW,MZ,NA,SD,SL,SZ,TZ,UG,ZM,ZW),EA(AM,AZ,BY,KG,KZ,MD,RU,TJ,TM),EP(AT,BE,BG,CH,CY,CZ,DE,DK,EE,ES,FI,FR,GB,GR,HR,HU,IE,IS,IT,LT,LU,LV,MC,MK,MT,NL,NO,PL,PT,RO,SE,SI,SK,TR),OA(BF,BJ,CF,CG,CI,CM,GA,GN,GQ,GW,ML,MR,NE,SN,TD,TG),AE,AG,AL,AM,AO,AT,AU,AZ,BA,BB,BG,BH,BR,BW,BY,BZ,CA,CH,CN,CO,CR,CU,CZ,DE,DK,DM,DO,DZ,EC,EE,EG,ES,FI,GB,GD,GE,GH,GM,GT,HN,HR,HU,ID,IL,IN,IS,JP,KE,KG,KM,KN,KP,KR,KZ,LA,LC,LK,LR,LS,LT,LU,LY,MA,MD,ME,MG,MK,MN,MW,MX,MY,MZ,NA,NG,NI,NO,NZ,OM,PG,PH,PL,PT,RO,RS,RU,SC,SD,SE,SG,SK,SL,SM,ST,SV,SY,TJ,TM,TN,TR,TT,TZ,UA,UG,US,UZ,VC,VN,ZA,ZM,ZW

(71)出願人 510210623
ショーン・テリー
Ｓｈａｗｎ ＴＥＲＲＹ
アメリカ合衆国０６３６５コネチカット州プレストン、リバー・ロード７４番
(74)代理人 100101454
弁理士 山田 卓二
(74)代理人 100081422
弁理士 田中 光雄
(74)代理人 100125874
弁理士 川端 純市
(72)発明者 ブライアン・オリバー
アメリカ合衆国０６３３９コネチカット州レッドヤード、オーガスト・メドウズ１７番
(72)発明者 ショーン・テリー
アメリカ合衆国０６３６５コネチカット州プレストン、リバー・ロード７４番

Ｆターム(参考) 5B075 ND03 NK02 NK31 NK45 QM02