

意味の数学モデルによる意味的メディア連想検索のための 実時間並列アルゴリズム

宮原 隆行[†] 清木 康^{††} 北川 高嗣[†]

[†]筑波大学 電子・情報工学系 ^{††}慶應義塾大学 環境情報学部

大量メディアデータを対象としたデータベースシステムにおいて、検索者が求めるメディアデータを検索者が要求する時間内に抽出する実時間検索アルゴリズムの実現が重要である。本稿では、文脈あるいは状況に応じて動的に変化するデータ間の意味的な関係を計算するモデルである意味の数学モデルによる意味的連想検索を対象とした実時間並列アルゴリズムを提案する。本稿で示すアルゴリズムは、検索者の要求に応じた時間内に、優先度に基づいた意味的なメディアデータ連想検索を並列に行い、応答するものである。提案アルゴリズムを適用した意味的メディア連想検索実験を行い、提案アルゴリズムの実現可能性、および、有効性を明かにする。

A Realtime Parallel Algorithm for Semantic Media Data Search by a Mathematical Model of Meaning

Takayuki Miyahara[†], Yasushi Kiyoki^{††} and Takashi Kitagawa[†]

[†]Institute of Information Sciences and Electronics, University of Tsukuba

^{††}Faculty of Environmental Information, Keio University

It is important to realize a realtime search method to create multi media database systems. On the situation that user needs any results of the retrieval in some limited time, It is important to retrieve data and put the retrieval results to user within the time. In this paper, we present a parallel algorithm for realtime semantic media data search by a mathematical model of meaning. This parallel algorithm provides the facility of the semantic media data search based on priority control in the limited time. To clarify the feasibility and effectiveness of the algorithm, we have performed several experiments in which the realtime and parallel algorithm is applied to semantic media data search. Those experimental results show the effectiveness the realtime parallel algorithm.

1 はじめに

近年、画像データベースなどのマルチメディア・データベースシステムの実現の重要性が高まってきている[1, 4, 6, 8, 10]。大量のメディアデータを対象としたデータベースシステムの実現において、実時間制約を持った検索方式の実現が重要である。検索者が限られた時間内に検索結果を必要とする場合において、検索者が求めるメディアデータをその時間内に抽出する方式およびその実

現が必要である。文献[2]では、本意味的連想検索方式を画像メディアに適用した場合について示しているが、本方式はメディアに依存するものではないので、ここでは意味的メディア連想検索方式とよぶ。

メディアデータを対象としたデータベースシステムの検索方式は、部分的なパターンマッチングによる直接的な方法と、メディアデータの抽象情報による間接的な方法に大別できる。本稿では、後者の方法に分類され、メディアデータの抽象情

報を使用して検索を行う意味的メディア連想検索方式 [2] の実時間検索のための並列化アルゴリズムを提案する。この意味的メディア連想検索方式は、文脈あるいは状況に応じて動的に変化するデータ間の意味的な関係を計算するモデルである意味の数学モデル [3] を拡張したメディア（例えば画像）データベースの検索方式であり、利用者の印象、および、内容を文脈として与えることによりメディアデータの意味的連想検索を実現する。

この意味的メディア検索方式では、検索者が与える文脈に応じて対象メディアデータ群をその文脈との相関の強い順に並べたリストを生成する。意味的メディア連想検索の実時間性を実現するためには、優先度の高い対象メディアデータを検索の初期の段階で抽出することが重要である。意味的メディア連想検索の実時間処理のための並列化アルゴリズムにおいては、優先度を考慮した並列化を行うことが重要になる。本稿で述べるアルゴリズムは、検索者の要求に応じた時間内で検索結果を返すものである。

意味的メディア連想検索方式では、正規直交空間を形成し、その空間上に、検索者の印象やメディアデータの内容を指定するためのキーワード群、および、メディアデータ自身を表現するメディアメタデータ群を各々座標点として写像する。そして、キーワード群を用いて検索者の印象あるいはメディアデータの内容を指定する文脈を与え、その文脈に対応する正規直交空間内の部分空間を選択する。その部分空間におけるメディアメタデータの位置により、与えた文脈と関連の強いメディアデータを動的に検索する。

本方式は、多変量解析による空間生成を用いた情報検索手法 [7] とは、次の点で本質的に異なる。本方式では、直交空間における部分空間の選択を行う演算（意味射影）を定義し、その演算により、言葉の意味を文脈に応じて、曖昧性を排除して解釈する機構を実現している。この機構により、検索キーワードと検索対象データの間の意味的な関係を、与えられた文脈に応じて動的に計算することを可能としている。

画像データベースにおいて、検索者の印象による画像データ抽出に関する研究として、[4]において提案された方法がある。この方法は、正準相関分析により、各検索者の印象語と画像特徴との相関関係を学習し、印象による画像の検索を可能としている。この方法は、画像の特徴を表す静的に固定されたベクトルと検索者の印象語群を表すベクトル間の相関の強さに関する計算により画像

を抽出する方法である。この方式との比較において、本方式の特徴は、各画像について、検索者が与える印象語群に対応する特徴に関する要素群を部分空間の選択という操作により抽出し、その要素群に対応する部分においてのみ強い相関をもつ画像データを抽出する点にある。これにより、本方式では、検索者によって与えられる印象語群に対応する画像の特徴を動的に抽出し、それ以外の特徴を排除することによって、与えられた印象語群に応じた特徴に関する要素群に対応する部分だけに着目した画像の抽出を行うことが可能となる。

2 意味の数学モデルによる意味的メディア連想検索

意味の数学モデルは、本来、言葉の意味を扱うためのモデルである。ここでは、印象やメディアデータの内容を表現する文脈に相関の強いメディアデータを検索するために拡張した意味の数学モデルの基本構成を示す。

1. メタデータ空間 *MDS* の設定：

m 個の基本データ（“空間生成用メタデータ”）が、 m 行 n 列の行列（以下、“データ行列”）の形で与えられる。 m 個の基本データは、それぞれ n 個の特徴 (feature) によって特徴づけられている。このデータ行列から正規直交空間（以下、“メタデータ空間”）*MDS* を生成する。

2. メディアデータの n 次元ベクトル表現：

(1) で用いた n 個の特徴と同一の特徴を用いて、個々のメディアデータを n 次元ベクトルで表現する。これをメディアメタデータとよぶ。また、検索者の印象やメディアデータの内容を表すキーワードあるいは文脈語を n 次元ベクトルで表現する。これを検索語のメタデータとよぶ。

3. メタデータ空間 *MDS* 上への写像：

n 次元ベクトルで表現された 3 種類のメタデータをメタデータ空間上へ写像する。これにより、同じ空間上に言葉とメディアデータが配置されることになり、言葉とメディアデータの動的な関係を空間上での距離として計算することが可能になる。

4. 意味的連想検索：

印象やメディアデータの内容を決定づける文脈が検索語の並びとして与えられる。このとき、文脈として与えられた検索語の列により、メタデータ空間から動的に固有部分空間（以下、“意味空間”）を選択する。そして、この部分空間上で、文脈に対応する（最も関連の深い）メディアデータを選び出す。

3 意味的メディア連想検索のための実時間並列アルゴリズム

3.1 意味的メディア連想検索のための実時間アルゴリズム

意味的メディア連想検索方式は、検索者が与える文脈に応じて対象メディアデータ群をその文脈との関連の強い順に並べたリストを生成する。意味的メディア連想検索のための実時間アルゴリズムは、検索者が限られた時間内に検索結果を必要とする場合において、優先順位に基づいた順序で関連の計算を行い、文脈との関連の強いメディアをその時間内に抽出することを可能とするものである。

意味の数学モデルによる意味的メディア検索方式は、文脈に対応した部分空間を選択し、その部分空間内で大きなノルムを持つ検索対象データを関連の強い検索データとして抽出する。部分空間を構成する各軸には、与えられた文脈との関連の強さに応じて重みがかかっており、重み付きの部分空間においてノルムの計算を行う。

重み付きの部分空間において、大きな重みを持つ軸上で大きな値を持つ検索対象データは、ノルムが大きくなり、検索者が与えた文脈と関連の強い検索対象データとして選択される可能性が高い。そこで、意味的メディア連想検索のための実時間アルゴリズムでは、次の2つの優先度に基づいてノルム計算を行い、指定された時間内に検索結果を抽出する。

- (1) 軸上で大きな値を持つ検索対象データを優先する
- (2) (1) において、大きな重みを持つ軸を優先する

上記2つの優先度のうち、どちらが強い優先度であるかは、検索対象データが登録され、検索者によって文脈が与えられて初めて決定される。な

ぜなら、(1)は検索対象データに依存し、(2)は検索者が与えた文脈に依存するからである。

そこで、実際のデータに応じて動的に優先度を定めることが可能な意味的メディア連想検索アルゴリズムを以下に示す。

本アルゴリズムでは、あらかじめ、検索対象データは、各軸ごとに、各検索対象データの軸上での値の大きさに応じて、ソートされていることを前提とする。文脈 s_ℓ が与えられた時に、 i 番目に重みが大い軸上において j 番目に軸上での値が大い検索対象データを $r_{ij}(s_\ell)$ とする。

STEP 1: 軸の重みに関する優先度に対応して増加する変数 y を 1 に初期化する。 i 番目に大きな重みを持つ軸上での値による優先度に対応して増加する変数 $x(i)$ を 1 に初期化する。 i 番目に大きな重みを持つ軸の優先度を示す変数 $A(i)$ を 0 に初期化する。STEP 2 と STEP 5 を実行するために必要な時間を見積り、その結果が制約時間を超えるのであれば、制約時間内に解を提示できないものとし、実行を中止する。

STEP 2: $r_{11}(s_\ell)$ のノルムを計算し、 $A(1)$ に代入する。ノルムを計算結果リストに追加する。

STEP 3: STEP 5 と STEP 8 を実行するために必要な時間を見積り、その結果が制約時間を超えるのであれば、STEP 8 に行く。

STEP 4: $A(1)$ から $A(y)$ の中から、最大の値 $A(k)$ を持つ $k, (1 \leq k \leq y)$ を探す。

STEP 5: $r_{k(x(k)+1)}(s_\ell)$ のノルムを計算し、 $A(k)$ に代入する。ノルムを計算結果リストに追加する。 $x(k)$ の値を 1 増やす。 $k \neq y$ であれば、STEP 3 に行く。

STEP 6: STEP 7 と STEP 8 を実行するために必要な時間を見積り、その結果が制約時間を超えるのであれば、STEP 8 に行く。

STEP 7: $r_{(y+1)1}(s_\ell)$ のノルムを計算し、 $A(y+1)$ に代入する。ノルムを計算結果リストに追加する。 y の値を 1 増やす。STEP 3 に行く。

STEP 8: 計算結果リストを用いて、ノルムの大きい順にソートし、検索者が与えた文脈との関連の強い順として、検索者に提示する。

このアルゴリズムを適用することにより、2つの優先度に動的に従った、意味的メディア検索を行うことが可能になる。

3.2 意味的メディア連想検索のための実時間並列アルゴリズム

意味的メディア連想検索のための実時間アルゴリズムでは、逐次処理の場合でも、良好な解を得ることができるとされる。並列処理により、実時間内で計算個数を増やせれば、より上位のものを獲得する確率は高くなる。

本並列アルゴリズムでは、近い優先度を持つ検索対象データ群をブロックとして扱い、ブロック内の検索対象データに同一の優先度を与える。ブロック内のノルム計算を並列に処理することで、優先度に基づいた並列化を実現する。

意味的メディア連想検索のための実時間並列アルゴリズムを次に示す。

並列処理の単位となるブロックを $bx \times by$ 個の要素で構成するものとする。本アルゴリズムでは、あらかじめ、検索対象データは、各軸ごとに、各検索対象データの軸上での値の大きさに応じて、ソートされていることを前提とする。文脈 s_e が与えられた時に、 i 番目に重みが大きい軸上において j 番目に軸上での値が大きい検索対象データを $r_{ij}(s_e)$ とする。 $r_{((i-1) \times by + 1)((j-1) \times bx + 1)}(s_e)$ から $r_{(i \times by)((j \times bx)}(s_e)$ までの $by \times bx$ 個の検索対象データを含むブロックを $r'_{i'j'}(s_e)$ とする。

STEP 1: 軸の重みに関する優先度に対応して増加する変数 y' を 1 に初期化する。軸の重みの大きい順で i' 番目のブロックの軸上での値による優先度に対応して増加する変数 $x'(i')$ を 1 に初期化する。軸の重みの大きい順で i' 番目のブロックの優先度を示す変数 $A'(i')$ を 0 に初期化する。STEP 2 と STEP 5 を実行するために必要な時間を見積り、その結果が制約時間を超えるのであれば、制約時間内に解を提示できないものとし、実行を中止する。

STEP 2: ブロック $r'_{11}(s_e)$ に含まれる $bx \times by$ 個の検索対象データのノルムを並列に計算する。計算されたノルムを計算結果リストに追加する。ブロック内の最大ノルムを $A'(1)$ に代入する。

STEP 3: STEP 5 と STEP 8 を実行するため

に必要な時間を見積り、その結果が制約時間を超えるのであれば、STEP 8 に行く。

STEP 4: $A'(1)$ から $A'(y')$ の中から、最大の値 $A'(k')$ を持つ k' 、 $(1 \leq k' \leq y')$ を探す。

STEP 5: ブロック $r'_{k'(x'(k')+1)}(s_e)$ に含まれる $bx \times by$ 個の検索対象データのノルムを並列に計算する。計算されたノルムを計算結果リストに追加する。ブロック内の最大ノルムを、 $A'(k')$ に代入する。 $x'(k')$ の値を 1 増やす。 $k' \neq y'$ であれば、STEP 3 に行く。

STEP 6: STEP 7 と STEP 8 を実行するために必要な時間を見積り、その結果が制約時間を超えるのであれば、STEP 8 に行く。

STEP 7: ブロック $r'_{(y'+1)1}(s_e)$ に含まれる $bx \times by$ 個の検索対象データのノルムを並列に計算する。計算されたノルムを計算結果リストに追加する。ブロック内の最大ノルムを、 $A'(y'+1)$ に代入する。 y' の値を 1 増やす。STEP 3 に行く。

STEP 8: 計算結果リストを用いて、ノルムの大きい順にソートし、検索者が与えた文脈との関連の強い順として、検索者に提示する。

本並列アルゴリズムは、複数のノルム計算を並列に行う。並列に実行される各ノルム計算は、完全に独立しているので、本アルゴリズムによる並列性は非常に高い。しかし、ブロック内の検索対象データに同一の優先度を与えることにより、本方式で並列性を減少させる原因となる現象は次の通りである。

(1) 関連の強いデータの抽出数の減少

本並列アルゴリズムは、ブロック内の検索対象データに同一の優先度を与え、同一の優先度を持つ検索対象データのノルム計算を並列に計算する。逐次アルゴリズムにおいて低い優先度を持つ検索対象データがブロックに含まれていた場合、関連の小さい検索対象データを抽出してしまうことがある。この場合、同一制約時間において、逐次アルゴリズムを適用した場合よりも多くのノルム計算が行われたとしても、関連の強いデータを多く抽出することができない。

(2) ノルム計算が行われるデータ数の減少

1つの検索対象データは、各軸ごとに1回ずつ

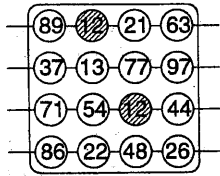


図 1: ブロック内に同一の検索対象データが複数存在する例

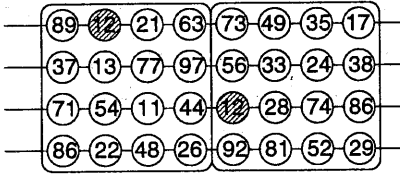


図 2: 他のブロックで処理されたデータが存在する例

つ現れる。本アルゴリズムによって、複数の軸を対象として、ノルムを計算する優先順位を決定すると、既にノルム計算を行った検索対象データが、再びノルムの計算対象になる場合がある。並列化を行わない検索アルゴリズムでは、過去に計算したノルムの値を使用して処理を進めるだけで良いが、ブロック単位による並列化を行う本アルゴリズムでは、あるブロック中に、既にノルム計算が行われた検索対象データが含まれている場合でも、そのブロック全体の計算を省略することができない。具体的には、ブロック内に同一の検索対象データが複数存在する場合（図 1）と、他のブロックにおいて既にノルムの計算を行った検索対象データが、ブロック内に存在する場合（図 2）である。

このような場合では、同一制約時間において、並列アルゴリズムを適用し、より多くの検索対象データのノルム計算を行う見積りをしても、ノルム計算が行われる非同データ数が少なくなってしまう、関連の強いデータを多く抽出することができない。

3.3 実時間アルゴリズムを適用したスケジューリング

実時間並列アルゴリズムでは、検索対象データのノルム計算の処理と、ノルム計算が行われたデータを関連の強い順にソートする処理の 2 つを実時間内に行う。ノルム計算を行う検索対象データ数を増やすと、与えられた文脈と関連の強い解（上位の解）が抽出される可能性も高くなるが、ソートに必要な時間も増えてしまう。ソートに必要な時間を見積り、ソートを開始する時間までに、ノルム計算処理を打ち切ることで、利用者への解の提示を実時間内に行う。

実時間性を実現するためには、各処理について、処理時間の最大時間でスケジューリングを行う必要がある。

1つのノルム計算に必要な時間を T_N とする。 N_B 個の検索対象データのノルム計算を行うと仮定する。ソートに必要な基本単位時間を T_S とすると、ソートに必要な最大時間を $N_B^2 T_S$ と見積もることができる。

与えられた制約時間を T_D とすると、逐次アルゴリズムにおいて、与えられた制約時間内に解を提示するためには、以下の式を満たす必要がある。

$$N_B T_N + N_B^2 T_S \leq T_D$$

この式に基づく逐次アルゴリズムによる実時間検索システムのスケジューリングを図 3 に示す。優先度に従った順序でノルム計算を行い、制約時間からソートに必要な時間分を引いた時刻 $T_D - N_B^2 T_S$ の直前にノルム計算を打ち切り、ソート処理に移行する様子を示している。ソート処理は最大でも $N_B^2 T_S$ 時間で終了するので、制約時間 T_D 内に解を抽出することができる。

並列アルゴリズムでは、1回のノルム計算の時間 (T_N) と、ブロック内の要素数 (P) \times 1 要素の通信時間 (T_C) の和が 1 ブロックの処理時間になる。 N_B ブロックの処理を行った後に、ソートに必要な最大時間は、 $(PN_B)^2 T_S$ となる。

並列アルゴリズムにおいて、与えられた制約時間内に解を提示するためには、以下の式を満たす必要がある。

$$N_B(T_N + P \times T_C) + (P \times N_B)^2 T_S \leq T_D$$

この式に基づく並列アルゴリズムによる実時間検索システムでのスケジュールを図 4 に示す。

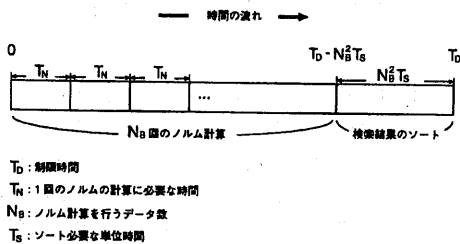


図 3: 逐次アルゴリズムを適用したスケジューリング

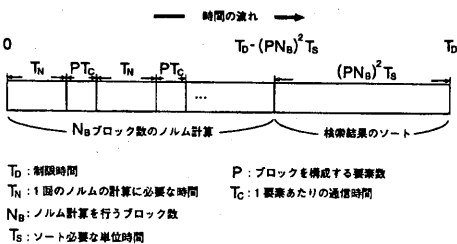


図 4: 並列アルゴリズムを適用したスケジューリング

優先度に従った順序でノルム計算を行い、制約時間からソートに必要な時間分を引いた時間 $T_D - (PN_B)^2 T_S$ の直前にノルム計算を打ち切り、ソート処理に移行する様子を示している。ソート処理は最大でも $(PN_B)^2 T_S$ 時間で終了するので、制約時間 T_D 内に解を抽出することができる。

1 ブロックあたりの要素数を多くすると、多くのノルム計算を行うことが可能になるが、ソートに必要な時間も増加する。また、必要な通信時間も増加する。

ブロックの要素数によっては、 N_B を 1 に設定しても、制約時間に間に合わないことがある。

3.4 スケジューリングに従った処理の流れ

1つのノルム計算に必要な時間を T_N 、通信に必要な時間を T_C 、ブロックの大きさを P 、ソートに必要な基本単位時間を T_S とする。制約時間 T_D が与えられた時に、次のスケジューリングに従って制約時間内に意味的メディア連想検索を行う。

STEP 1: 計算を行うブロック数 N_B を 0 に初期化する。もし、 $(T_N + PT_C) + P^2 T_S$ が T_D を超えるならば、制約時間内に検索を行うことはできない。

STEP 2: N_B を 1 つ増やす。 $N_B(T_N + PT_C) + (PN_B)^2 T_S$ が T_D を超えるならば、STEP 4 に行く。

STEP 3: 優先度の高いブロックから 1 ブロック分のノルム計算を行う。この時に、ブロック内のノルム計算を並列に行う。このステップの実行には、 $T_N + P \times T_C$ 時間必要である。

STEP 4: ノルムの大きい順に検索対象データをソートする。このステップは $(P \times N_B)^2 T_S$ 時間以内に実行される。

STEP 5: 結果を検索者に提示する。

4 実験

提案アルゴリズムの実現可能性、および、有効性を調べるために、意味的メディア連想検索の実験を行った。本実験では、意味的数学モデルによる意味的メディア連想検索の実時間並列アルゴリズムを適用したメディアデータ検索システムにおいて、仮想的に想定した画像メタデータを対象とした意味的メディア連想検索を行い、ブロックサイズを変化させて、制約時間内に抽出された相関の強い検索対象データの個数を調べた。

実験では、並列アルゴリズムにおける 1 ブロックを 1×1 (逐次処理に対応) から 10×10 (並列度 100 に対応) の大きさのブロックで構成した。本実験では、並列処理をシミュレートすることによって、提案アルゴリズムの有効性を検証した。

4.1 実験環境

本実験では、Longman Dictionary of Contemporary English[9] (以下 LD) において基本語とされている英単語 2328 単語を LD の説明語群を用いて特徴付けを行い、 2328×2328 の行列を作成し、メタデータ空間を構成した。

画像メタデータの作成には、“カラーイメージスケール [5]” に記述されている印象語 182 語を用いた。1 枚の画像メタデータに対して 3 語の印象語を対応させ、5339 枚の画像メタデータを作成した。印象語の特徴付けには LD での説明語を用いた。

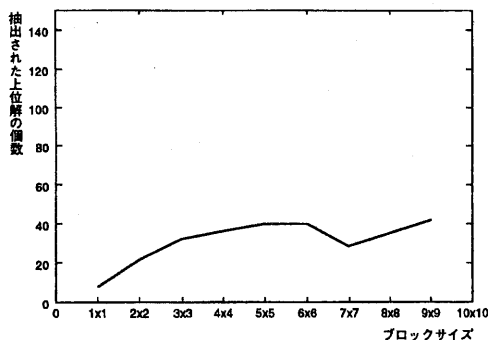


図 5: ブロックサイズと上位解の個数 (制約時間 $10T_N$)

ノルム計算に必要な時間 T_N を単位時間としてパラメータを設定した。部分空間の平均次元数が約 1000 であることから、ソートの単位時間 T_S を、 $T_N/1000$ に設定した。通信に必要な時間 T_C を $0.01T_N$ に設定した。

4.2 実験方法

182 語の印象語中の 1 語を文脈として与え、182 回の提案アルゴリズムを適用した意味的画像検索を行った。また、5339 検索対象全てについて相関の強さを計算した時における、上位 300 画像データを文脈との相関の強い画像データ (上位解) とした。制約時間とブロックを構成する要素数 P を変化させ、抽出された上位解の平均個数を調べた。

4.3 実験結果

検索者が、制約時間として $10T_N$ を与えた時のブロックサイズ P と上位解の個数の関係を図 5 に示す。 1×1 のブロックサイズは、並列化を行わない場合の結果を示している。通信時間 T_C を $0.01T_N$ に設定した場合、ブロックサイズを大きくすると制約時間内に抽出される上位解の個数が増えることがわかる。しかし、 10×10 のブロックサイズでは、制約時間内に結果を提示することができなかった。

また、制約時間として $100T_N$ を与えた時のブロックサイズ P と上位解の個数の関係を図 6 に示す。並列化により、抽出された上位解の個数は増えているが、制約時間を $10T_N$ に設定したときよりも効果は小さい。これは、計算を行うブロッ

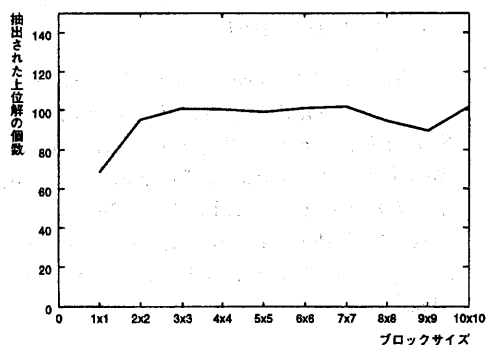


図 6: ブロックサイズと上位解の個数 (制約時間 $100T_N$)

ク数の増加に従い、3.2 (1),(2) に示した現象がより多く発生していることが原因と考えられる。また、実際に得られた解の個数ではなく、理論上の最大値の解の個数を使用してソートの時間の見積りを行っていることも原因の 1 つと考えられる。

並列化アルゴリズムを適用すると、並列化アルゴリズムを適用しない場合との比較において、同一制約時間内でより多くの相関の強い画像データを抽出できたことを示している。

5 おわりに

本稿では、意味の数学モデルによる意味的メディア連想検索のための実時間並列アルゴリズムを提案した。提案アルゴリズムを適用した実験システムを作成し、意味的メディア連想検索実験を行い、提案アルゴリズムの有効性を確認した。

今後は、本稿で述べた意味的メディア連想検索のための実時間並列アルゴリズム、および、意味的メディア連想検索のための学習機構の実現を行っていく予定である。

参考文献

- [1] 有澤 博, 由井 仁, 富井 尚志, “画像データベースシステムの構成の一方式,” Proc. Advanced Database System Symposium '93, pp.181-190, 1993.
- [2] Kiyoki, Y., Kitagawa, T., and Hayama, T., “A Metadatabase System for Semantic Image Search by a Mathematical Model of Meaning,” ACM SIGMOD Record, (Special issue

- on metadata for digital media), Vol.23, No. 4, pp.34-41, Dec. 1994.
- [3] Kiyoki, Y., Kitagawa, T. and Hitomi, Y., "A Fundamental Framework for Realizing Semantic Interoperability in a Multidatabase Environment," Journal of Integrated Computer-Aided Engineering, Vol.2, No.1(Special Issue on Multidatabase and Interoperable Systems), pp.3-20, John Wiley & Sons, Jan. 1995.
- [4] Kurita, T., Kato, T., "Learning A Cognitive Schema for Multimedia Indexing -Experiments on Image Database-, " Technical Reports of IEICE, DE93-3, May 1993.
- [5] 小林重順, "カラーイメージスケール", 講談社, 1984.
- [6] 坂内正夫, "画像検索技術," 信学誌, Vol.71, No.9, pp.911-914, 1988.
- [7] S. Deerwester, S. T. Dumais, T. K. Landauer, G. W. Furnas, and R. A. Harshman, "Indexing by Latent Semantic Analysis," Journal of the American Society for Information Science, Vol.41, No.6, pp.391-407, 1990.
- [8] 平田恭二, 原良憲, "概略画像を用いた画像検索," 信学技報, DE92-2, 1992.
- [9] "Longman Dictionary of Contemporary English," Longman, 1987.
- [10] Y. Masunaga, "A Temporal Expression to the Multimedia Object Model in OMEGA," Proc. of DASFAA'95, pp.430-440, April 1995.