

意味の数学モデルによる高速連想検索方式とマルチメディアデータベースへの応用

著者	宮原 隆行
著者別名	Miyahara Takayuki
内容記述	筑波大学博士（工学）学位論文・平成12年3月24日授与（乙第1621号）
発行年	2000
URL	http://hdl.handle.net/2241/6354

第 5 章

意味の数学モデルによる実時間並列マルチメディア データ検索

5.1 意味の数学モデルによるマルチメディアデータ検索

5.1.1 概略

意味の数学モデルによるデータ検索では、文脈とキーワードを与え、与えたキーワードに最も近い意味を持つデータに関連付けられたデータを抽出する。マルチメディア情報を検索する場合には、マルチメディアデータというものは、キーワードという単語だけではなく、複数の語からなる文脈に相当する情報を持っていると考えられる。そこで、画像などのマルチメディア情報を検索する場合においては、その画像などの雰囲気を表す文脈を与え、その文脈と最も関連が深いデータを抽出することとする。

文脈を与え、部分空間を切り出した後、与えた文脈と相関の強いマルチメディアデータを選択する方法は、いくつか考えられる。ここでは、与えた文脈に対応した部分空間内で、原点からのノルムが最も大きいマルチメディアデータを選択することで、検索者の印象に近いマルチメディアデータを抽出する方法を採用する (図 5.1)。

具体的には、文脈 s_ℓ が与えられた場合において、検索対象マルチメディアデータ \mathbf{x} のノルムの大きさは下式で表される。

$$\bar{\eta}_0(\mathbf{x}; s_\ell) = \frac{\sqrt{\sum_{j \in \Lambda_{s_\ell} \cap S} \{c_j(s_\ell)x_j\}^2}}{\|\mathbf{x}\|_2}$$

ここで、

$$S = \{i | \text{sign}(c_i(s_\ell)) = \text{sign}(x_i)\}$$

で、ある。

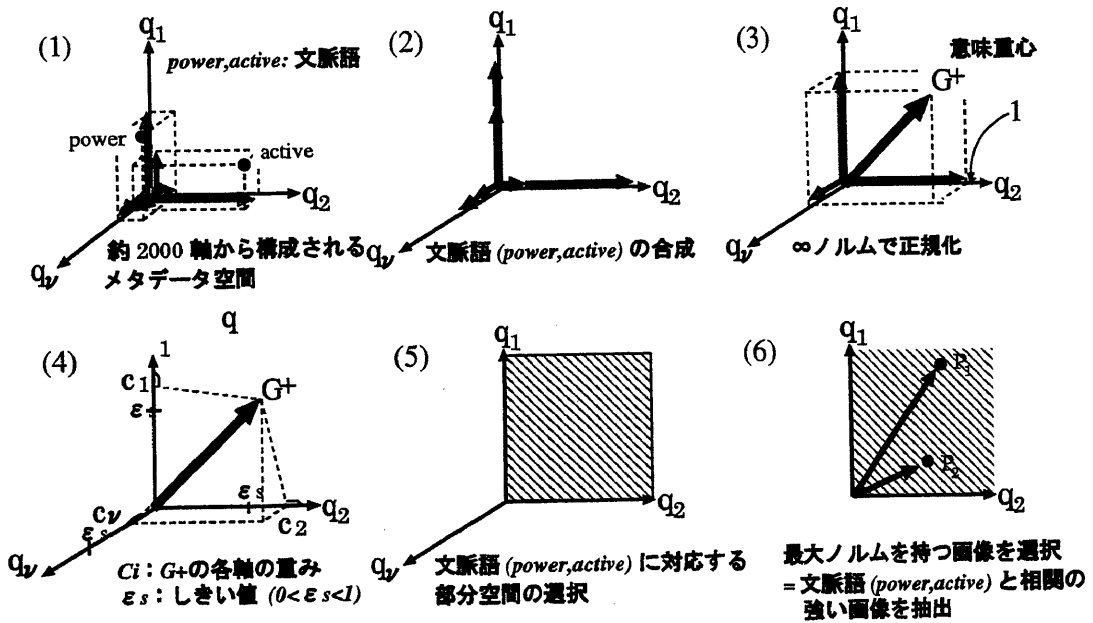


図 5.1: 意味の数学モデルによるマルチメディアデータ検索

5.1.2 定式化

マルチメディアデータの作成方式

マルチメディアデータ P を以下の手順に従って作成する。

- (1) **オブジェクトの抽出**: t 個のオブジェクト (あるいは, t 個の語によって表現される印象) o_1, o_2, \dots, o_t から成るマルチメディアデータ P を次のように定義する:

$$P = \{o_1, o_2, \dots, o_t\}.$$

ここで, 各オブジェクト o_i は, データ行列の特徴と同一の特徴を用いて表現される特徴付ベクトルである:

$$o_i = (o_{i1}, o_{i2}, \dots, o_{in})$$

- (2) **マルチメディアデータ P のベクトル表現**: オブジェクト o_1, o_2, \dots, o_t の和演算子 \oplus を次のように定義し, マルチメディアデータのベクトルを形成する:

$$\begin{aligned} \bigoplus_{i=1}^t o_i &\equiv (\text{sign}(o_{t11}) \max_{1 \leq i \leq t} |o_{i1}|, \\ &\quad \text{sign}(o_{t22}) \max_{1 \leq i \leq t} |o_{i2}|, \\ &\quad \vdots \\ &\quad \text{sign}(o_{tnn}) \max_{1 \leq i \leq t} |o_{in}|). \end{aligned}$$

ただし, $\text{sign}(a)$ は, “a” の符号 (正または負) を表し, $l_k (k = 1, \dots, n)$ は, $|o_{ik}|$ の最大値を与える添字 $i (i = 1, \dots, t)$ を表している.

文脈とマルチメディアデータの相関の強さ

文脈とマルチメディアデータ x との相関の強さを, 2.2.4 節で定めた動的距離を変形させて, 次のように定義する.

$$\bar{\eta}_0(x; s_\ell) = \frac{\sqrt{\sum_{j \in \Lambda_{e_s, nS}} \{c_j(s_\ell)x_j\}^2}}{\|x\|_2},$$

ただし, $S = \{i | \text{sign}(c_i(s_\ell)) = \text{sign}(x_i)\}$.

5.1.3 意味的画像検索アルゴリズム

文脈に対応して選ばれた意味空間 (部分空間) において, 検索対象マルチメディアデータの集合 W から文脈と最も相関の強い検索対象マルチメディアデータを選び出すことによって意味的画像検索を行う. 意味的画像検索アルゴリズムでは, 文脈に応じた部分空間を選び, その中から文脈と最も相関の強いマルチメディアデータを選び出すために, 次に示す2基本機能を実現する.

基本機能 1: 部分空間選択

文脈を確定する文脈語群 $s_\ell = (u_1, \dots, u_\ell)$ が与えられると (u_1 から u_ℓ までの単語ベクトルは M 行列を作成したものと同一特徴で特徴付けられている), 文脈語群の意味重心を計算し, 意味射影と部分空間選択を行う.

Step-1 : 文脈ベクトルを構成する単語ベクトルのフーリエ展開:

u_1 から u_ℓ までの各単語ベクトルについてフーリエ展開を行い, 各々の意味素 (固有ベクトル) との相関を計算する.

Step-2 : フーリエ展開によって得られた値を, 各意味素 (固有ベクトル) ごとに合算する:

文脈語と意味素の相関を計算するために, フーリエ展開によって得られた値を各々の意味素ごとに合算することにより意味重心ベクトルを形成する.

Step-3 : 部分空間の決定:

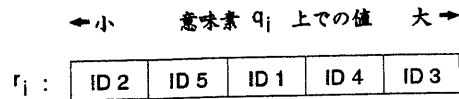


図 5.2: 検索対象画像データの意味素 q_i 上のソートされた画像識別子群の例

Step-2 で求めた意味重心ベクトルの各意味素上での値の絶対値が、与えられたしきい値 ϵ_s より大きい意味素群を選択し、部分空間を構成する。

基本機能 -2: 解の選択

与えられた文脈と最も相関の強いマルチメディアデータを、基本機能 1 によって選択された部分空間に写像されたマルチメディアデータの中から選び出す。具体的には、3.2 節で定義した式を使用して、文脈と各マルチメディアデータとの相関の強さを計算し、マルチメディアデータ群 W の中から次の式によって選び出す。

$$\max_{w \in W} \bar{\eta}_0(w; s_e)$$

5.2 意味の数学モデルによるマルチメディアデータ検索の高速化

前節の相関の強さを求める式から、最も重みの大きい意味素上で大きい値を持つ検索対象画像データほど式の値が大きくなり、文脈との相関が強い画像データとして選ばれる可能性が高いと予想される。本アルゴリズムでは、文脈との相関が強いと予想される画像データから先に相関の強さを求め、計算結果を利用者に提示する。文脈との相関が強い画像データが先に出力されれば、利用者は、全ての検索対象画像データの相関の強さの計算を待つことなく、求める画像データを得ることが可能になるので、検索の高速化を実現できる。

本アルゴリズムでは、あらかじめ、検索対象画像データは、各意味素 $q_i (i = 1 \sim \nu)$ ごとに、各検索対象画像データの意味素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。意味素 q_i 上の検索対象画像データの値の大きさの情報を持つ列を r_i とする。検索対象画像データが n 個存在する時、意味素 q_i 上で最も小さい値を持つ検索対象画像データを r_{i1} で表し、最も大きい値を持つ検索対象画像データを r_{in} で表す。図 5.2 は、検索対象画像データの意味素 q_i 上のソートされた画像識別子群の例である。この例では、“ID 1” から “ID 5” までの 5 つの検索対象画像データが存在

	← 小	意味素上での値					大 →
$r_{4(s)}$:	14	10	6	3	1	↑	大 重 み 小 ↓
$r_{3(s)}$:	17	13	9	5	2	↓	
$r_{2(s)}$:	19	16	12	8	4	↑	
$r_{1(s)}$:	20	18	15	11	7	↓	

図 5.3: 高速な意味的画像検索アルゴリズムにおける相関の強さの計算順序例

し、意味素 q_3 上で最も小さい値を持つ検索対象画像データ r_{i1} は“ID 2”であり、最も大きい値を持つ検索対象画像データ r_{i5} は“ID 3”であることを示している。

文脈 s_l が与えられた時に、 i 番目に重みが小さい意味素上の検索対象画像データの値の大きさの情報を持つ列を $r_{i(s_l)}$ とする。 i 番目に重みが小さい意味素上で最も小さい値を持つ検索対象画像データを $r_{i1(s_l)}$ で表し、 i 番目に重みが小さい意味素上で最も大きい値を持つ検索対象画像データを $r_{in(s_l)}$ で表す。

そして、以下の規則により、相関が強いと予想される検索対象画像データから順に相関の強さを求めることが可能となる。ここで、 i は 1 から部分空間の次元数までの値をとり、 j は 1 から検索対象データの個数までの値をとる。

1. $i_1 + j_1 < i_2 + j_2$ ならば $r_{i_1 j_1(s_l)}$ よりも $r_{i_2 j_2(s_l)}$ を先に計算する。
2. $i_1 + j_1 = i_2 + j_2$ の時、 $j_1 < j_2$ ならば $r_{i_1 j_1(s_l)}$ よりも $r_{i_2 j_2(s_l)}$ を先に計算する。

図 5.3 は、本アルゴリズムを適用した時の相関の強さの計算順序の例である。この例では、検索対象画像データが 5 つ存在し、文脈に対応した意味素群が 4 本選ばれた時の相関の強さの計算順序を表している。

5.3 実験

提案アルゴリズムの実現可能性および有効性を検証するために、提案アルゴリズムの実験システムを Sun SPARC システム上に実現した。

5.3.1 実験環境

本実験では、Longman Dictionary of Contemporary English [22]において基本語とされている 2328 単語を、The General Basic English Dictionary [24] を用いて定義し、 2328×874 のデータ行列を作成し、イメージ空間を構成した。また、 ϵ_s は 0.5 に設定した。

5.3.2 実験方法

30 枚の画像を印象で定義し、検索対象画像データとした (表 5.1)。文脈として印象語を与えて意味的画像検索を行い、提案アルゴリズムを適用した時の計算順序 (出力順序) と、文脈と画像データの相関の強さを調べた。文脈との相関が強い画像データと出力順序の関係を調べ、提案アルゴリズムの有効性を検証する。

5.3.3 実験結果

文脈語として “power”, “active” を与え、この文脈と相関の強い画像を求めた時の実験結果を表 5.2 に示した。表 5.2 は、出力順に応じて次々に選ばれる画像について、その画像の相関の強さの順位 (全画像を対象とした場合の相関の強さの順位) を示している。この実験結果から、出力順の最初の方に相関の強い順位の画像データが多く現れることがわかる。特に、利用者にとって有用な、1 番目、2 番目、3 番目に相関の強い画像が、出力順で 3 番目までに現れているので、利用者は 3 番目の画像データが出力された時点で、問い合わせを終了させることができる。このことから、提案アルゴリズムにより高速な意味的画像検索を実現できることがわかる。

5.4 意味の数学モデルによる実時間マルチメディアデータ検索

意味の数学モデルによるマルチメディアデータ検索において、データのノルムの大きさは下式で表される。

$$\bar{\eta}_0(x; s_\ell) = \frac{\sqrt{\sum_{j \in \Lambda_{\epsilon_s} \cap S} \{c_j(s_\ell) x_j\}^2}}{\|x\|_2}$$

この式から、最も重みの大きな軸上で、最も大きな値を持つデータを、最大の候補とする。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、相関量最大の候補となる検索対象メディアデータから優先的に相関量を計算し、指定された時間で計算を打ち切ることで、文脈と相関の強いメディアデータを制約時間内に抽出する。

r_{ij} を i 番目に重みの大きな軸上で j 番目に大きな検索対象と定義する。

STEP 1: 軸の重みに関する優先度に対応して増加する変数 $imax$ を 1 に初期化する。
 i 番目に大きな重みを持つ軸上で、軸上の値に関する優先度に対応して増加する変数 $x(i), (i = 1 \sim \nu)$ を 1 に初期化する。

STEP 2: 検索対象データの部分集合 $r_{ix(i)}, (i = 1 \sim imax)$ の中から、最も大きなノルムを持つ検索対象データを選択する。検索対象データ $r_{ix(i)}$ のノルムの大きさが計算されていない時は、ノルムの大きさを計算し、結果を保存する。選択された検索対象データを $r_{kx(k)}$ とする。

STEP 3: $x(k)$ を 1 つ増やす (これは、軸上の値の優先度に従って計算を進めることを意味する)。もし、 k が $imax$ と等しいならば、 $imax$ を 1 つ増やす。 (これは、軸の重みの優先度に従って計算を進めることを意味する)

STEP 4: 指定された時間内であれば、STEP 2 へ行く。

STEP A: 上記の動作と並行して、保存されたノルムの情報を用いて、検索者が与えた文脈との相関の強い順 (ノルムの大きい順) に検索結果をソートし、検索者に提示する。

このアルゴリズムの動作例を示したものが、図 5.4 から図 5.6 である。

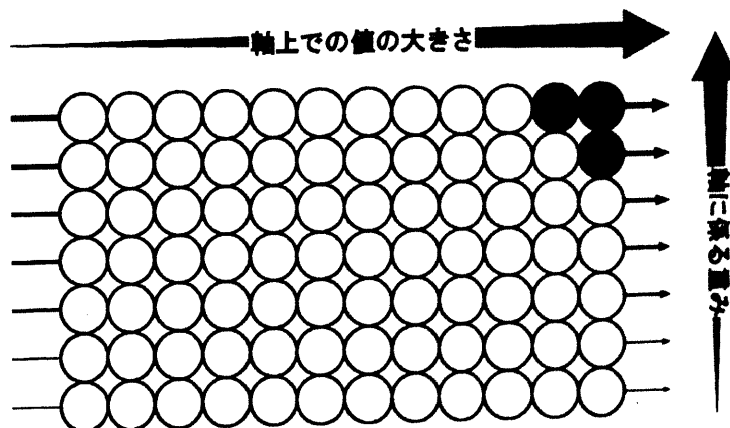


図 5.4: 実時間検索アルゴリズムの動作例 (1)

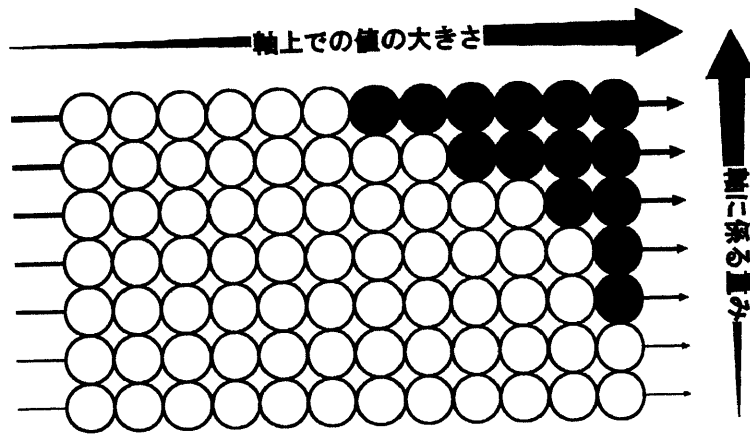


図 5.5: 実時間検索アルゴリズムの動作例 (2)

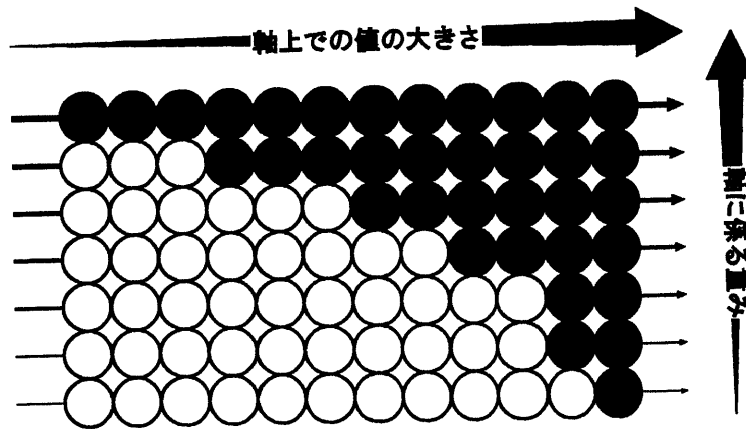


図 5.6: 実時間検索アルゴリズムの動作例 (3)

5.5 意味の数学モデルによる実時間並列マルチメディアデータ検索

前章では、意味の数学モデルによる実時間マルチメディアデータ検索について述べた。本章では、さらに、実時間マルチメディアデータ検索アルゴリズムの並列化を行い、ノルム計算を行うデータの個数を増やし、制約時間内に抽出される解の精度を上げることを目標とする。

前章のアルゴリズムは、今までに計算されたデータを使用して、次に行うべきデータを決定する。すなわち、計算すべきデータは、前回に計算されたデータに依存して決定される。よって、このままでは単純に並列化を行うことはできない。そこで、複数の検索対象データをまとめたものを「ブロック」とし、ブロック内のノルム計算を並列に行うことで、制約時間内に抽出される解の精度を上げる。

「ブロック」の定義は以下の通りである。

- (1) 軸を重みの大きい順に並べて By 本ずつ組にしたものを考える。先頭から i' 番目の組には、 $(i' - 1) \times By + 1$ 番目に重みの大きい軸から $i' \times By$ 番目に重みの大きい軸群までが含まれる。
- (2) また、検索対象データを軸上での値が大きい順に並べて Bx 個ずつ組にしたものを考える。先頭から j' 番目の組には、 $(j' - 1) \times Bx + 1$ 番目に軸上での値が大きい検索対象データから $i' \times Bx$ 番目に軸上での値が大きい検索対象データまでが含まれる。
- (3) $r_{((i'-1) \times By + 1)((j'-1) \times Bx + 1)}$ から $r_{(i' \times By)((j' \times Bx)}$ までの $By \times Bx$ 個の検索対象データを含むブロックを $r'_{i'j'}$ とする。

このような定義を行うと、実時間並列アルゴリズムは以下の通りになる。

STEP 1 : 軸の重みに関する優先度に対応して増加する変数 $imax'$ を 1 に初期化する。 i 組番目に大きな重みを持つ軸の組上で、軸上の値に関する優先度に対応して増加する変数 $x'(i)$, ($i = 1 \sim \nu/By$) を 1 に初期化する。

STEP 2 : 検索対象データブロック $r'_{i'x'(i)}$, ($i = 1 \sim imax'$) の中から、最も大きなノルムを持つ検索対象データを含むブロックを選択する。この時に、検索対象ブロック $r'_{i'x'(i)}$ 内のノルムの大きさが計算されていないければ、そのノルムを計算し、結果を保存する。選択された検索対象データを $r'_{k'x'(k')}$ とする。

STEP 3 : $x'(k')$ を 1 つ増やす。(これは、軸上の優先度に従って計算を進めることを意味する)。もし、 k' が $imax'$ と等しいならば、 $imax'$ を 1 つ増やす。(これは、軸の重みの優先度に従って計算を進めることを意味する)

STEP 4 : 指定された時間内であれば、STEP 2 へ行く。

STEP A : 上記の動作とは並行して、保存されたノルムのデータを用いて、検索者が与えた文脈との相関の強い順 (ノルムの大きい順) に検索結果をソートし、検索者に提示する。

このアルゴリズムの動作例 (2×2 ブロック, 4PE) を示したものが、図 5.7 から図 5.9 である。

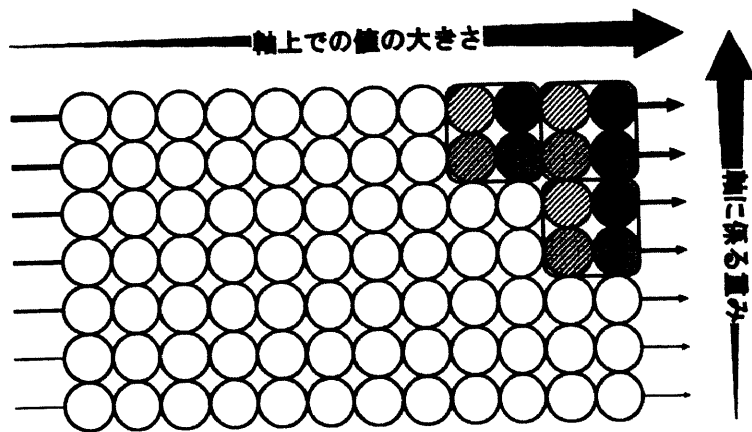


図 5.7: 実時間並列検索アルゴリズムの動作例 (1)

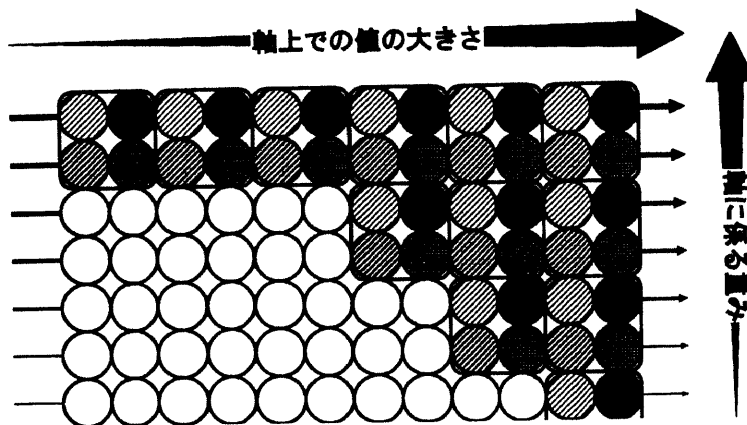


図 5.8: 実時間並列検索アルゴリズムの動作例 (2)

5.6 実験

5.6.1 実験環境

メタデータ空間の作成には“Longman Dictionary of Contemporary English” (基本語 2328 語のみを用いて全ての見出し語を説明している英英辞典) を使用し 2328 × 2328 の行列を作成, また, 検索キーワード (文脈) として使用する単語群, 検索対象メタデータに用いる単語群には, “Longman Dictionary of Contemporary English” の説明語を用いて特徴付けを行った。

また, 実時間アルゴリズムを実現する実験システムを逐次処理システム上に構築し,

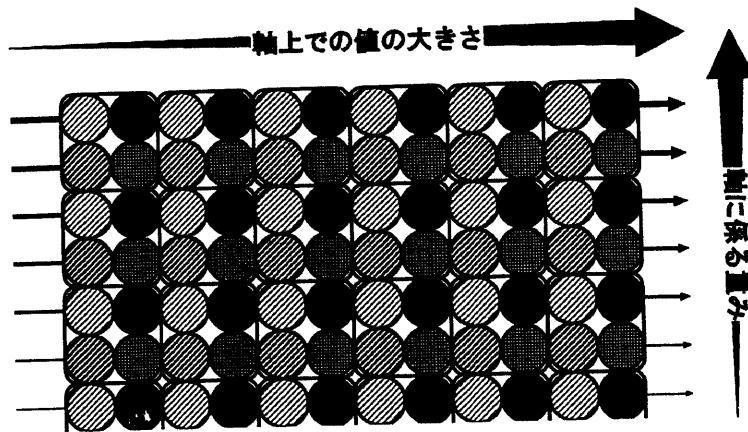


図 5.9: 実時間並列検索アルゴリズムの動作例 (3)

ノルム計算の時間, 通信時間は, 並列処理環境に対応したパラメータとして設定した。

ノルム計算に必要な時間	T_N
ブロックを構成する要素数	P
通信に必要な時間	T_C
1 データあたりに必要なソートの時間	T_S
制約時間	T_D
計算を行ったブロック数	N_B

想定するシステムは, 1 つのセントラルユニット (CU) と, 複数のプロセッシングエレメント (PE), バス型の通信回路から構成される。各プロセッシングエレメントは, 同一のデータを独立に保持しており, 計算対象となるデータ番号さえ与えられれば (図 5.10),

結果をセントラルユニットへ順次返す (図 5.11)。この時の通信量は, わずか 1 WORD である。

結果を受け取ったセントラルユニットは, 次の計算ブロックを決定し, 全てのプロセッシングユニットへブロードキャストを行う。プロセッシングユニットが計算をしている間に B-tree 作成を行い, 上位のものから利用者に提示する。

この様に, 単純なシステム構成を採用したのは, システムの性能を評価するのではなく, 並列アルゴリズムの性能を評価するためである。

上記のシステムにおいては, 以下の式を満たさなくなった時点で計算を中断すれば

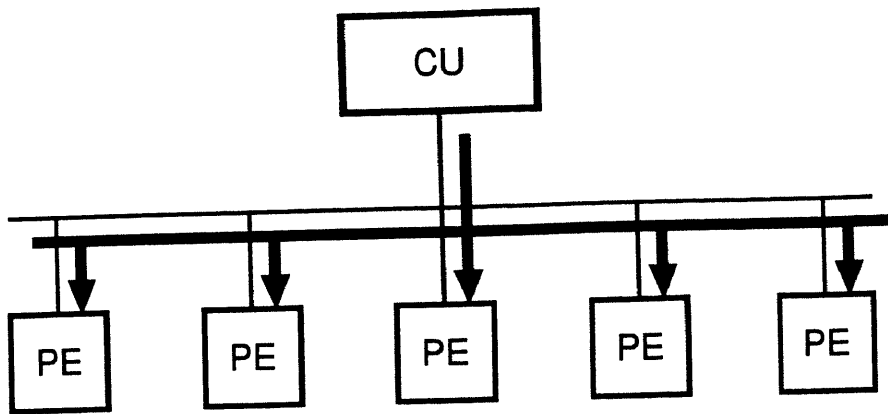


図 5.10: 想定するシステムの動作 (1)

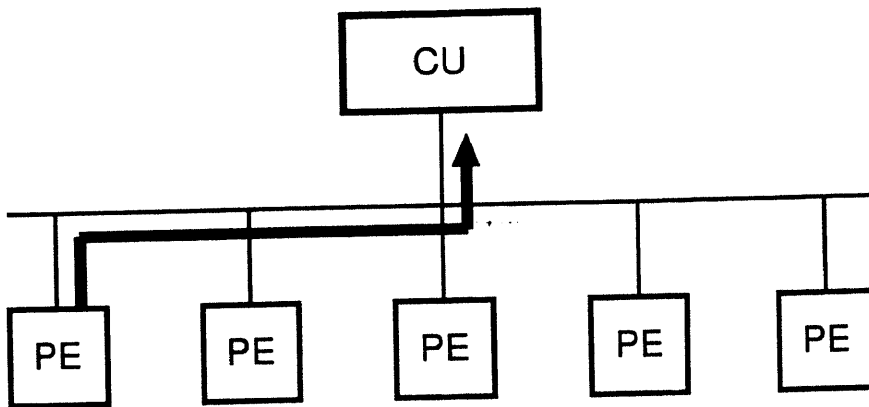


図 5.11: 想定するシステムの動作 (2)

良い

$$\max(N_B(T_C + T_N + PT_C), T_S N_B \log(T_S N_B)) \leq T_D$$

また、本実験のパラメータは、ノルム計算に必要な時間 T_N との相対的な時間を使用し、以下の様に設定した。

ノルム計算に必要な時間 T_N

通信に必要な時間 T_C $0.01 \times T_N$

1 データあたりに必要なソートの時間 T_S $0.001 \times T_N$

このパラメータは、各 PU が同一のベクトル情報を保持しており、かつ、メインメモリ上に全てのデータが存在しているという、並列処理には厳しい条件になっている。

検索対象メディアメタデータは、「カラーイメージスケール」という、「(株)日本カラーデザイン研究所」による、色と印象の対応表を用い、この本の中では、色から受ける印象は、15グループに分けられた183の代表色の組み合わせで決まるとしているを利用し、各グループ内で3色の組み合わせを作成し、5339の仮想画像メタデータを作成した。

また、解の精度の評価であるが、評価指数として下式で表される h_i を適用した。

$$h_i = \frac{\text{上位}i\text{番目までの理想順位の合計}}{\text{取り出された解の上位}i\text{番目までの実順位の合計}}$$

5.6.2 実験目的

本実験では、同一制約時間内において、並列処理方式を適用したシステムにより、プロセッシングユニット数の増加と共に、解の精度が上がることを確認することを目的とする。

5.6.3 実験結果

制約時間 $T_D = 100T_N$

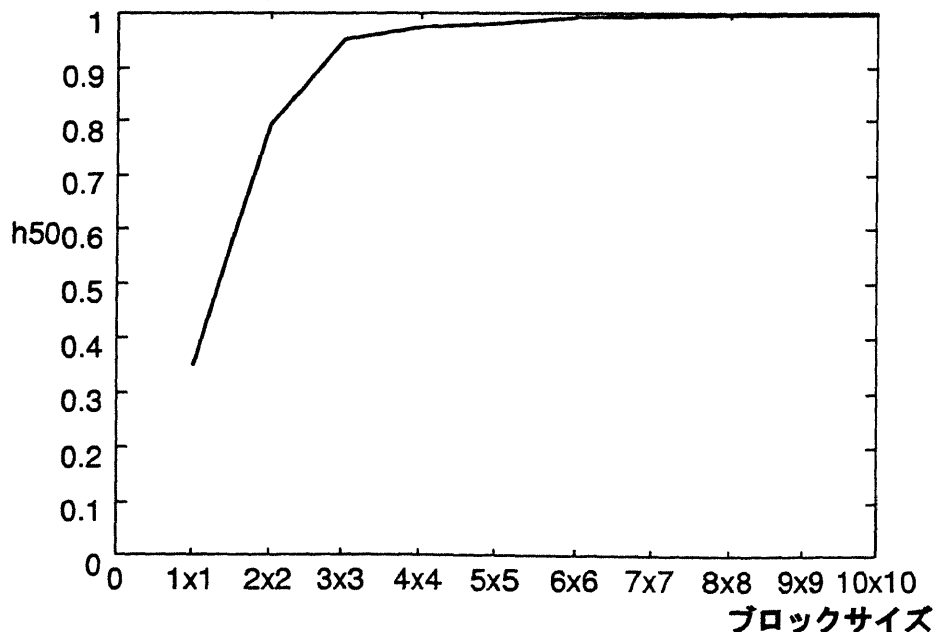


図 5.12: ブロックサイズと解の正当性の関係

図 5.6.3 は、ブロックサイズ (= プロセッシングユニット) の数と、計算精度の関係を表したグラフである。プロセッシングユニットの増加と共に、同一制約時間で得られる解の精度が飛躍的に向上しているのが分かる。

本実験により、提案アルゴリズムの有効性が確認できた。

表 5.1: 検索対象画像データとその定義語群

画像 データ名	定義語群
chagall1	vivid quiet substance
chagall2	grief terrible sombre
chagall3	sober dynamic motion
chagall4	shine tender calm
corot1	beautiful grand calm
corot2	beautiful delicate calm
corot3	grief sombre sober
corot4	shine beautiful calm
gogh1	merry delight shine
gogh2	grief terrible sombre
hiro1	twilight grand quiet
hiro2	cheer dim quiet
hiro3	beautiful quiet calm
hiro4	fine shine beautiful
hiro5	fine beautiful calm
hokusai1	dynamic strong motion
hokusai2	fight motion calm
hokusai3	delicate calm quiet
hokusai4	vivid motion speed
loirand1	shine grand calm
loirand2	delight shine calm
loirand3	delight grand calm
loirand4	quiet substance material
nelson1	grand dynamic motion
nelson2	twilight calm quiet
renoir1	dim tender quiet
renoir2	delight dim calm
renoir3	loud bustle crowd
renoir4	fine1 strong quiet
sarthou1	dynamic motion speed

表 5.2: 出力順と相関の強さの順位の対応表

context: power active

出力順	相関の強さの順位	画像データ名	相関の強さ
1	1	sarthou1	0.229303
2	3	renoir3	0.220917
3	2	nelson1	0.221236
4	4	hokusai1	0.201502
5	18	hiro2	0.109731
6	5	chagall3	0.188423
7	17	gogh1	0.110321
8	15	gogh2	0.111263
9	12	corot1	0.121324
10	16	chagall2	0.111263
11	10	hokusai4	0.124058
12	26	corot3	0.087398
13	7	renoir4	0.132660
14	8	loirand4	0.129607
15	6	hokusai2	0.140624
16	9	loirand3	0.124300
17	11	loirand1	0.123303
18	28	renoir1	0.080252
19	14	renoir2	0.115855
20	21	hiro3	0.097729
21	19	chagall1	0.102361
22	25	hiro4	0.088651
23	22	corot4	0.096555
24	20	chagall4	0.098629
25	23	hiro1	0.094530
26	24	nelson2	0.090299
27	27	hiro5	0.081588
28	30	corot2	0.063832
29	13	loirand2	0.116701
30	29	hokusai3	0.065211