

処理速度とメモリ効率の改善された LLAH による カメラベース文書画像検索

中居 友弘[†] 黄瀬 浩一[†] 岩村 雅一[†]

[†] 大阪府立大学大学院工学研究科

〒 599-8531 大阪府堺市中区学園町 1-1

E-mail: †nakai@m.cs.osakafu-u.ac.jp, ††{kise,masa}@cs.osakafu-u.ac.jp

あらまし 本稿では, Locally Likely Arrangement Hashing (LLAH) を用いた文書画像検索の改善法を提案する. LLAH は高速かつ高精度でロバスト性が高いという長所をもつ一方で, 必要メモリ量が多いという問題がある. また, リアルタイム文書画像検索などへ応用するためにはさらなる高速化が必要である. これらの問題点を解決するため, 以下の 2 つの改良を施す. 1 つは重要性の低い特徴量をデータベースから削除し, データ構造を単純化することで実現される必要メモリ量の削減である. もう 1 つは, 検索処理におけるアルゴリズムの改良による処理時間の短縮である. 実験により, これらの改善によって必要メモリ量の 80%, 処理時間の 60% が削減されることが確認された.

キーワード 文書画像検索, デジタルカメラ, ハッシュ

Camera Based Document Image Retrieval with More Time and Memory Efficient LLAH

Tomohiro NAKAI[†], Koichi KISE[†], and Masakazu IWAMURA[†]

[†] Graduate School of Engineering, Osaka Prefecture University

1-1 Gakuen-cho, Naka, Sakai, Osaka, 599-8531 Japan

E-mail: †nakai@m.cs.osakafu-u.ac.jp, ††{kise,masa}@cs.osakafu-u.ac.jp

Abstract In this paper, we propose improvements of our camera-based document image retrieval method with Locally Likely Arrangement Hashing (LLAH). While LLAH has high accuracy, efficiency and robustness, it requires a large amount of memory. It is also required to speed up the retrieval of LLAH for applications to real-time document image retrieval. For these purposes, we introduce the following two improvements. The first one is reduction of the required amount of memory by removing less important features for indexing from the database and simplifying structure of the database. The second improvement is to reduce exploring alternatives during the retrieval process. From the experimental results, we have confirmed that the proposed improvements realize reduction of the required amount of memory by about 80% and that of processing time by about 60%.

Key words Document image retrieval, Digital camera, Hashing

1. はじめに

文書画像検索とは, 与えられた検索質問に対応する文書画像を, データベースから見つける処理である. 文書画像検索の研究では, さまざまな種類の検索質問が提案されている [1]. デジタルカメラを用いた文書画像検索は, デジタルカメラで撮影された文書画像を検索質問とするものである. このような形式の文書画像検索が実現されれば, 印刷文書を撮影することでさまざまなサービスにつなげるといった応用が可能になる. つまり, サービスを提供するためのメディアとして印刷文書を活用する

ことが可能になる. 例えば, データベースで文書画像に URL を関連付けておき, 利用者が印刷文書を撮影すると, 検索を通じて関連した Web ページにアクセスするというサービスが考えられる.

文書画像検索に関して, Locally Likely Arrangement Hashing (LLAH) というハッシュに基づく検索法が提案されている. LLAH は高速かつ高精度で, 高いロバスト性をもつという特長がある. 実験では, 10,000 ページのデータベースにおいて精度 95%以上, 検索の処理時間約 100ms という結果が得られている [2]. このような高い精度と高速性は, LLAH で用いられる

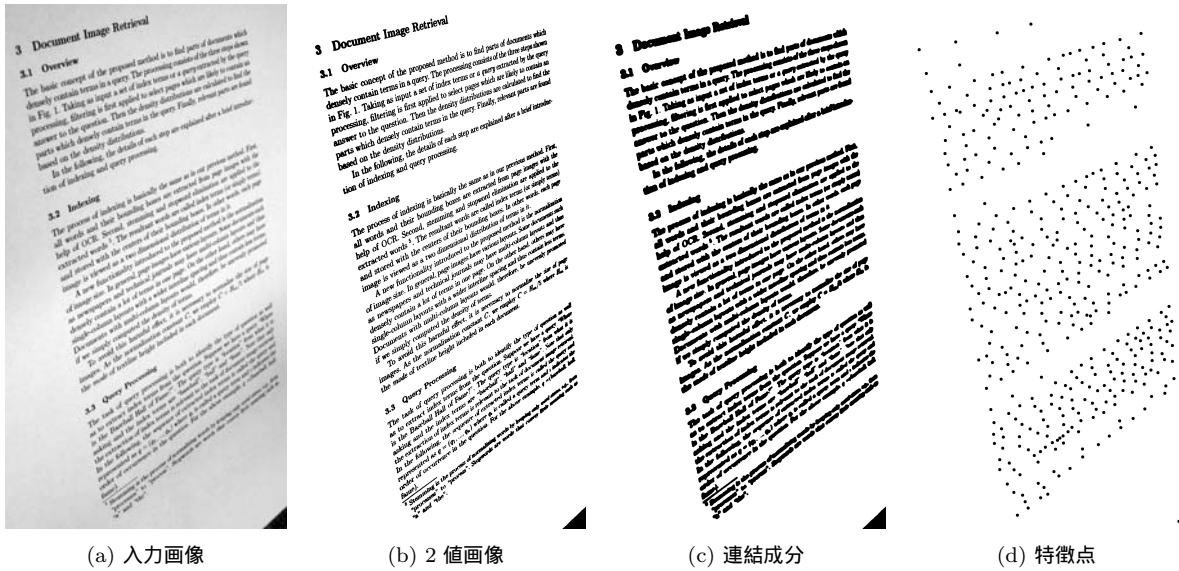


図2 特徴点抽出処理

特徴量が安定性と識別性を兼ね備えているためである。また、LLAHは射影歪みや隠れ、紙面の非線形な湾曲という、デジタルカメラで文書を撮影するときに生じる問題[3],[4]に対してもロバストであることが確認されている。このロバスト性は、局所領域で定義される幾何学的不変量に基づく特徴量によって実現されている。

一方、LLAHには高い精度とロバスト性を実現するために、大量のメモリを必要とするという問題がある。具体的には、10,000ページの文書画像がデータベースに登録されている場合、高精度な検索を実現するためには約2.6GBのメモリが必要となる。このようなメモリ効率の悪さは、LLAHのスケラビリティを制限するものである。また、カメラを用いたリアルタイム文書画像処理は高い利便性をもつ[5]ため、LLAHのリアルタイム文書画像検索への応用についても提案されている。しかし、LLAHを用いてより追従性の高いリアルタイム文書画像検索を実現するためには、検索処理のさらなる高速化が必要である。

本稿では、上記の問題を解決するためのLLAHの改善法を提案する。メモリ消費量の削減のための基本的なアイデアは、信頼性の低い特徴量を取り除き、データベースの構造を単純化するということである。処理の高速化については、特徴量を画像の回転に対して不変なものにし、検索時に総当たりで探索する処理を省略することで実現する。10,000ページのデータベースを用いた実験では、元のLLAHに対して必要メモリ量は1/5、処理時間は2/5となることが確認された。また実験では、改善されたLLAHは高いスケラビリティをもつことも示された。データベースの登録画像数が10,000までは、メモリ消費量と処理時間がほぼ一定であった。

2. LLAHによる文書画像検索

まず、オリジナルのLLAHと、それを用いた検索処理について説明する。

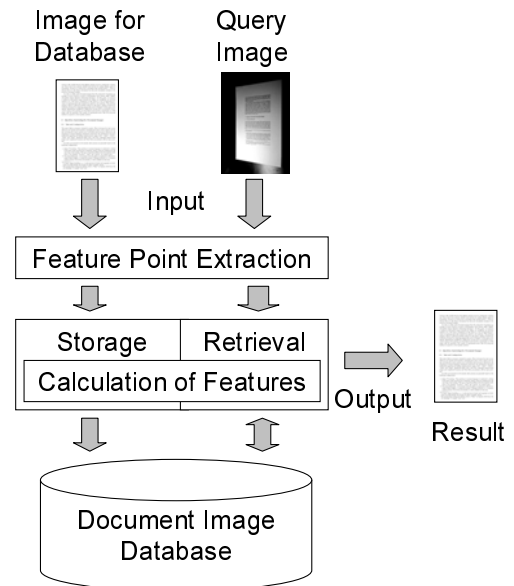


図1 処理の概要

2.1 処理の概要

図1に処理の概要を示す。まず、特徴点抽出処理 (Feature Point Extraction) で文書画像は特徴点の集合に変換される。次に、特徴点は登録処理 (Storage) および検索処理 (Retrieval) に入力される。これらの処理は特徴量計算処理 (Calculation of Features) を共有している。登録処理では、各特徴点は独立に、その特徴量に基づいて文書画像データベース (Document Image Database) に登録される。つまり、文書画像は特徴点を用いてインデキシングされる。検索処理では、検索質問の特徴量を用いて文書画像データベースにアクセスし、投票処理で対応する文書画像を決定する。以下では各処理について説明する。

2.2 特徴点抽出処理

LLAHでは特徴点の配置に基づいて文書画像のマッチングを行う。従って、特徴点抽出処理では、射影歪みやノイズが生じていたり、低解像度の場合でも同一の点を抽出する必要がある。

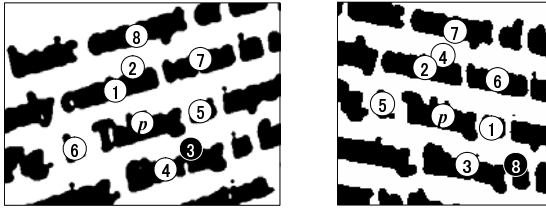


図 3 異なる視点から見た 2 枚の画像における特徴点 p の近傍 8 点 (丸で囲まれた数字は p からの距離の順位, 黒抜きの数字は異なる点が見られているもの)

そのため, 単語領域の重心を特徴点として用いる.

特徴点抽出の手順を以下に示す. まず, 入力画像 (図 2(a)) を適応 2 値化し, 2 値画像 (図 2(b)) を得る. 次に, 2 値画像をガウシアンフィルタでぼかし, 再度適応 2 値化を行うと, 単語ごとに連結された画像 (図 2(c)) が得られる. 最後に連結成分の重心を計算して特徴点 (図 2(d)) とする.

2.3 特徴量計算

特徴量とは, 文書画像の特徴点を表現する値である. 特徴点のマッチングは特徴量に基づいて行われる. そのため, 特徴量は以下の 2 つの条件を満たす必要がある. 1 つは, 同一の特徴点は, さまざまな外乱が生じて画像の見え方が変わったとしても, 同一の特徴量を与えなければならないというものである. 同一の特徴点であるにもかかわらず, 登録処理と検索処理で異なる特徴量を得られた場合, それらの対応付けを行うことはできない. そのため, 正しい検索結果を得ることができなくなる. 本稿では, この条件を“特徴量の安定性”と呼ぶ. もう 1 つの条件は, 異なる特徴点は異なる特徴量を与えなければならないというものである. 異なる特徴点から同一の特徴量を得られた場合, 誤った特徴点と対応付けられてしまう. そのため, 正しいものだけでなく誤ったものも検索結果として得られることになる. 本稿では, この条件を“特徴量の識別性”と呼ぶ. 正確な検索のためには, 安定性と識別性を両立した特徴量が必要である.

(1) 安定性

カメラで文書を撮影する場合, 隠れが生じていたり, 紙面の一部だけが撮影される場合がある. 従って安定性の実現のためには, 特徴量は文書画像の部分から計算されるものが望ましい. LLAH では, 各特徴点についてその近傍の特徴点の配置から特徴量を計算する. 特徴量が文書画像の局所領域から得られるため, 同じ領域が撮影されていれば同じ特徴量を計算することができる.

カメラを用いる場合, 画像が射影歪みを受けることがある. そこで, 射影歪みが生じても安定な特徴量を計算するために, 幾何学的不変量を用いる. カメラで斜め方向から撮影するとき生じる歪みは射影変換であるため, 射影変換の不変量である複比を用いることが考えられる. しかし [2] の結果より, 射影変換より自由度の低い変換であるアフィン変換の不変量を用いる方が検索において高い精度をもたらすことが確認されている. これは, 局所領域においては射影変換はアフィン変換に近似可能であることと, アフィン不変量の方が特徴点の位置の変動に強く安定であることが原因である.

本稿では, 同一平面上の 4 点 ABCD から以下の式で計算されるアフィン不変量を用いる.

$$\frac{P(A, C, D)}{P(A, B, C)} \quad (1)$$

ここで, $P(A, B, C)$ は頂点 A, B, C をもつ三角形の面積である.

近傍点の配置から特徴点 p の特徴量を計算する場合, 最も単純なものは p の近傍 4 点から計算されるアフィン不変量を特徴量とするものである. しかし, 図 3 に示されるように, 射影歪みが生じた場合は近傍点に異なるものが得られることがある. 従って, 近傍 4 点から計算されるアフィン不変量は安定ではない. この問題を解決するため, より広い範囲の局所領域を用いる. 図 3 では, 近傍 7 点のうち 6 点までは共通のものが得られている. このことから, 近傍 n 点のうち $m (\leq n)$ 点まではある程度の射影歪みが生じていても共通のものを得ることができると考えられる. 以上のことから, m 点を用いて安定な特徴量を計算する. 図 4 に示されるように, n 点からのすべての m 点の組み合わせ $P_{m(0)}, P_{m(1)}, \dots, P_{m(nC_m-1)}$ を調べる. n 点のうち m 点が共通であれば, 同じ m 点を得ることができる. LLAH ではこのようにして特徴量の安定性を実現している.

(2) 識別性

m 点から特徴量を計算する場合, 最も単純な方法は $m = 4$ として 4 点から計算されるアフィン不変量を特徴量とするというものである. しかし, 異なる文書から得た特徴点でも, 類似した 4 点の配置をもつ場合があるため, このような単純な方法は識別性に問題がある. そこで, 識別性を向上させるため, $m (> 4)$ の値を大きくしてより多くの点から特徴量を計算する. m が大きくなると, 異なる文書画像から類似した m 点の配置をもつ確率が低くなる. 図 5 に示されるように, m 点の配置は, m 点からすべての 4 点の組み合わせを求め, そこから計算されるアフィン不変量を離散化した値の列 $(r_{(0)}, r_{(1)}, \dots, r_{(mC_4-1)})$ で

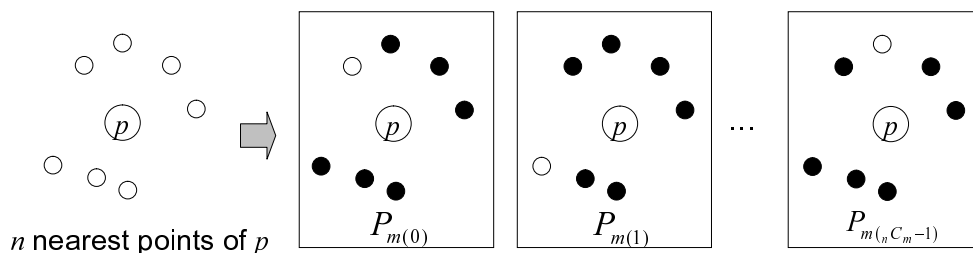


図 4 近傍 $n (= 7)$ 点からすべての $m (= 6)$ 点の組み合わせ

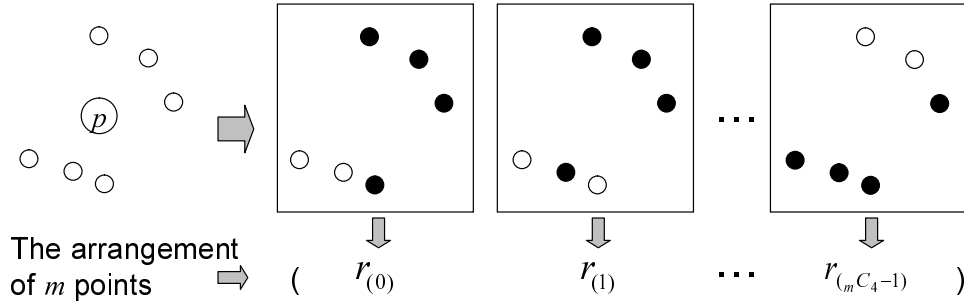


図 5 $m(=6)$ からのすべての 4 点の組み合わせから計算されるアフィン不変量の列

- 1: for each $p \in \{ \text{特徴点} \}$ do
- 2: $P_n \leftarrow p$ の近傍 n 点の集合
- 3: for each $P_m \in \{ P_n \text{ から } m \text{ 点を選ぶすべての組み合わせ} \}$ do
- 4: P_m から適当に 1 点 p_0 を選び, P_m の要素を p を中心として p_0 を先頭に時計回りに並べて特徴点の列 $L_m = (p_0, \dots, p_{m-1})$ を作る. 以後, p_i の順序とは添字の若い順のことをいう
- 5: $(L_4(0), \dots, L_4(i), \dots, L_4(mC_4 - 1)) \leftarrow L_m$ から 4 点を選んでできる部分列 $L_4(i)$ をすべて求めて辞書式順序に並べたリスト
- 6: for $i = 0$ to $mC_4 - 1$ do
- 7: $r_{(i)} \leftarrow L_4(i)$ から計算されるアフィン不変量の離散値
- 8: end for
- 9: $H_{\text{index}} \leftarrow$ 式 (2) で計算されるハッシュ表のインデックス
- 10: H_{index} を用いて文書 ID, 点 ID, $r_{(0)}, \dots, r_{(mC_4-1)}$ をハッシュ表に登録
- 11: end for
- 12: end for

図 6 登録アルゴリズム

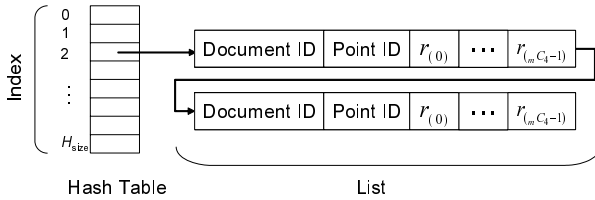


図 7 ハッシュ表の構成

表現される.

2.4 登録

図 6 に登録処理のアルゴリズムを示す. ここで, 文書 ID とは文書の識別番号, 点 ID とは特徴点の識別番号である.

ハッシュ表のインデックスは以下に示すハッシュ関数で計算される.

$$H_{\text{index}} = \left(\sum_{i=0}^{mC_4-1} r_{(i)} k^i \right) \bmod H_{\text{size}} \quad (2)$$

ここで, $r_{(i)}$ はアフィン不変量の離散値, k は離散化レベル数, H_{size} はハッシュ表のサイズである.

図 7 に示されるように, 文書 ID, 点 ID, $r_{(0)}, \dots, r_{(mC_4-1)}$ の組がハッシュ表に登録される. 衝突が生じた場合は, リスト形式で追加される.

- 1: for each $p \in \{ \text{特徴点} \}$ do
- 2: $P_n \leftarrow p$ の近傍 n 点の集合
- 3: for each $P_m \in \{ P_n \text{ から } m \text{ 点を選ぶすべての組み合わせ} \}$ do
- 4: for each $p_0 \in P_m$ do
- 5: P_m の要素を p を中心として p_0 を先頭に時計回りに並べて特徴点の列 $L_m = (p_0, \dots, p_{m-1})$ を作る
- 6: $(L_4(0), \dots, L_4(i), \dots, L_4(mC_4 - 1)) \leftarrow L_m$ から 4 点を選んでできる部分列 $L_4(i)$ をすべて求めて辞書式順序に並べたリスト
- 7: for $i = 0$ to $mC_4 - 1$ do
- 8: $r_{(i)} \leftarrow L_4(i)$ から計算されるアフィン不変量の離散値
- 9: end for
- 10: $H_{\text{index}} \leftarrow$ 式 (2) で計算されるハッシュ表のインデックス
- 11: H_{index} を用いてハッシュ表にアクセスし, リストを得る
- 12: for each リストの各要素について do
- 13: if 誤対応を防止するための条件 (文献 [2]) が成立する then
- 14: 文書 ID に投票
- 15: end if
- 16: end for
- 17: end for
- 18: end for
- 19: end for
- 20: 最大の得票数をもつ文書を検索結果とする

図 8 検索アルゴリズム

2.5 検索

検索処理のアルゴリズムを図 8 に示す. LLAH では, 投票テーブルを用いた登録文書への投票を通じて検索を行う.

まず, 7~10 行目で登録処理と同様にハッシュ表のインデックスを求める. 得られたインデックスを用いて, 11 行目で図 7 に示されるリストを得る. リストの各項目について $(r_{(0)}, \dots, r_{(mC_4-1)})$ が一致するか調べ, 一致していたら投票テーブルの文書 ID の項目をインクリメントする. 最後に, 最大の得票数を得た文書を検索結果として出力する.

3. 必要メモリ量の削減

LLAH では, 安定性の実現のために多くの特徴量を計算し, それらすべてをリスト形式でハッシュ表へ保存する. そのため, 必要メモリ量が多くなる. そこで, 重要性の低い特徴量をデータベースから削除し, データ構造を改めることで記憶容量の削減を図る.

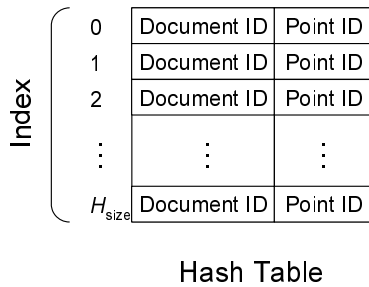


図 9 単純化されたハッシュ表の構成

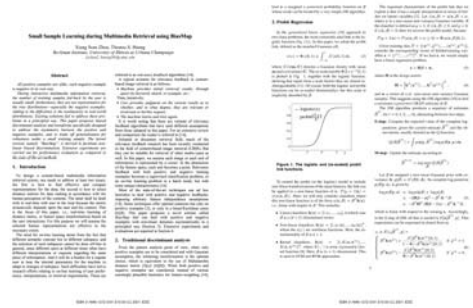


図 11 データベースの登録画像の例

メモリの削減のためには、重要性の低いデータを削除することが望ましい。重要性の低いデータとして、リストにおいて衝突を起こしているものが挙げられる。これは、衝突を起こしているものは識別性に欠ける可能性があり、また頻繁に生じる特徴量であるため、検索処理において速度の低下を引き起こすからである。また、実験では衝突を起こしているものはハッシュ表でデータの登録されているもののうち 28%程度であり、削除しても影響は限られると考えられる。以上の理由から、ハッシュ表において衝突の生じているリストを削除する。これにより、メモリ容量の削減が期待できる。

衝突のあったリストを削除すると、データ構造の変更によるさらなる容量の削減が可能になる。リストにおける特徴ベクトルは、衝突している場合に適切なものを見つけるためにあった。そのため、衝突がないのならば特徴ベクトルを保存する必要がなくなる。また、衝突がないのならそもそもリスト形式にする必要もない。そこで、図 9 に示すような単純なハッシュ表にすることで、記憶容量を削減する。

4. 検索処理の高速化

次に、検索処理の高速化のための登録および検索アルゴリズムの改良について述べる。

図 8 に示される検索アルゴリズムでは、4~5 行目で L_m のすべての巡回置換を試すために、 P_m のすべての点が開始点 p_0 となるように繰り返し処理を行う。これは、カメラで画像を撮影する際に回転が生じるため、検索時の L_m と登録時の L_m が必ずしも一致しないためである。

しかし、登録時と検索時に同じ開始点 p_0 を選ぶことができ

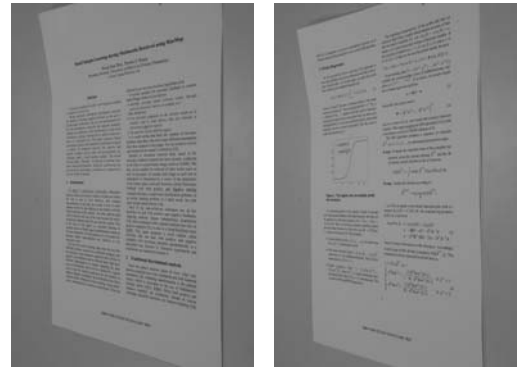


図 12 検索質問画像の例

れば、巡回置換を試す必要はない。以下では、開始点の選択規則を導入し、検索処理の効率化を図る。

開始点の選択規則を図 10 に示す。特徴点 p に属する m 点の各点 i について、後続の 3 点と合わせてアフィン不変量 $s_{(i)}$ を計算する。 $(s_{(0)} \cdots s_{(m-1)})$ の中で最大のものが $s_{(j)}$ であれば、点 j を開始点として選択する。図 10 の例では、 $s_{(1)}$ が最大であるため点 1 が開始点となる。最大値を取るものが複数あった場合、次の $s_{((i+1) \bmod m)}$ の最大値を調べて開始点を定める。例えば、 $s_{(i)} = s_{(j)}$ がともに最大値であれば、 $s_{((i+1) \bmod m)}$ と $s_{((j+1) \bmod m)}$ を比較する。 $s_{((i+1) \bmod m)}$ の方が大きければ、点 i が開始点として選ばれる。ここで $s_{((i+1) \bmod m)} = s_{((j+1) \bmod m)}$ であれば、さらに次の値 $s_{((i+2) \bmod m)}$ と $s_{((j+2) \bmod m)}$ を比較し、開始点を選択する。

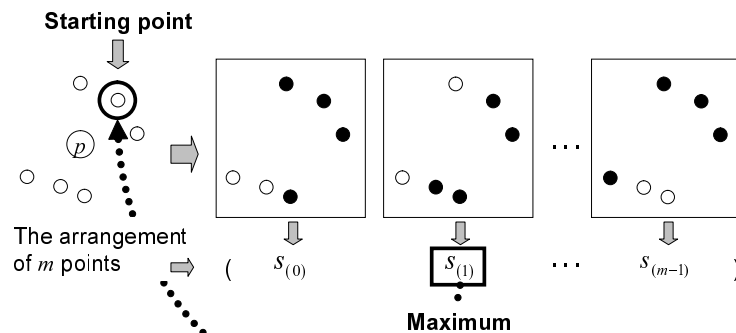


図 10 開始点の選択規則

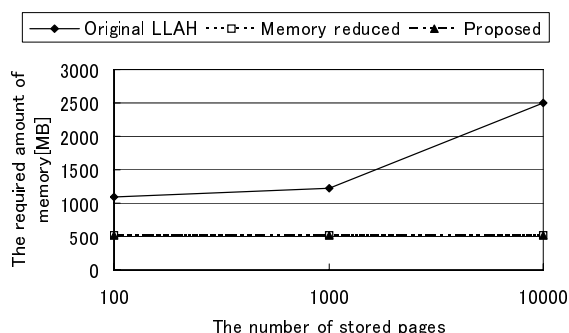


図 13 登録ページ数と必要メモリ量の関係

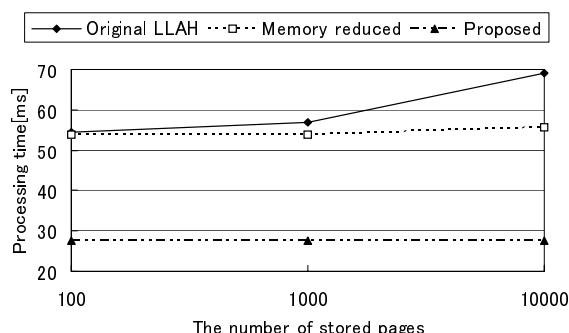


図 14 登録ページ数と検索処理時間の関係

5. 実験結果

本稿で提案された改善法の効果を調べるため、元の LLAH と改善された LLAH で検索実験を行い、性能を比較した。3. で述べられたメモリ削減と 4. で述べられた高速化の効果を明確にするため、以下の 3 つのバージョンの LLAH でメモリ消費量、処理時間、精度を調べた。

- (1) オリジナル
 - (2) メモリ削減版
 - (3) メモリ削減・高速化版
- (3) が本稿における提案手法である。

データベースに登録された文書画像は、主に予稿集の CD-ROM から集められた、1 段組および 2 段組の英語論文の PDF ファイルを 200dpi で画像に変換したものである。登録された文書画像の例を図 11 に示す。検索質問の文書画像は 630 万画素のデジタルカメラで撮影されたものである。図 12 に示されるように、検索質問は紙面に対して斜め方向（約 45 度）から撮影されたものである。検索質問の撮影角度（45 度）と登録画像の撮影角度（90 度）は異なるため、これらを用いた実験によって、手法の射影歪みに対するロバスト性が示すことができる。なお、検索質問画像の受ける射影歪みは文献 [2] のものより強いものである。実験に用いた計算機は、AMD Opteron 2.8GHz の CPU と 16GB のメモリをもつものである。パラメータは $n = 7, m = 6, k = 15, H_{\text{size}} = 1.28 \times 10^8$ とした^(注1)。

5.1 必要メモリ量

図 13 に、登録ページ数を 100, 1,000, 10,000 と増加させたときの 3 つのバージョンの LLAH におけるメモリ消費量を示す。オリジナルの LLAH は、登録ページ数 10,000 ではメモリの削減されたものに比べて 5 倍のメモリを消費した。さらに、登録ページ数の増加に伴ってメモリの消費量も増加した。これは、オリジナルの LLAH ではデータベースでリストを用いており、登録ページ数が増加するとメモリ消費量も増加するためである。一方、メモリ削減の改善がなされた LLAH では、登録ページ数が増加してもメモリ消費量は一定であった。これは、データベースが単純なハッシュ表であり、ハッシュ表のサイズが固定されているためである。

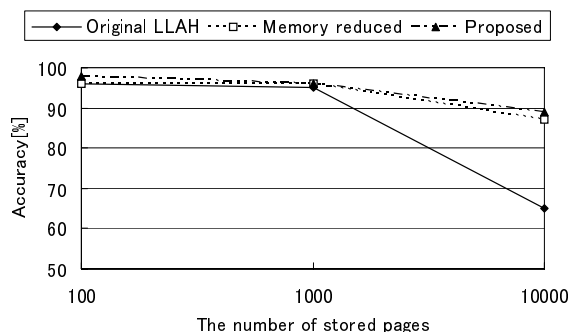


図 15 登録ページ数と精度の関係

5.2 処理時間

処理時間について調べた結果を図 14 に示す。このように、高速化バージョンはそれ以外に比べておよそ 60% の処理時間の削減を実現した。これは、巡回置換を試すための繰り返し処理を避けることで不変量の計算回数やハッシュへのアクセス回数を減らすことができたためと考えられる。

また、メモリ削減版はオリジナルに比べて登録ページ数の増加に対する速度の低下が見られなかった。これは、オリジナルではリスト構造のためデータ量が増えたとリストをたどる処理時間が増加する一方、単純なハッシュ表である削減版は常に一定の処理時間しか要しないためと考えられる。

5.3 精度

それぞれのバージョンの各データベースサイズにおける精度を図 15 に示す。処理時間とメモリ効率に加えて、精度に関しても改良版はオリジナルより高い性能を示した。オリジナルが登録ページ数の増加に伴って精度を低下させた一方で、改良版は登録ページ数 10,000 では精度がやや低下したものの、オリジナルよりも高い精度を維持した。

本稿で導入した改良が精度の向上を意図したものではないにも関わらず、精度の向上をもたらした理由は、メモリの削減のために識別性の低い特徴量を削除した結果、誤投票を減らすことができたことと考えられる。改良版では衝突を生じた項目は削除される。そのような項目は誤った投票を生じやすいため、それらを削除することで精度の向上が実現されたと考えられる。誤投票の減少による精度向上の例として、図 16 を示す。これは、オリジナルの LLAH で検索に失敗し、提案手法では検索

(注1): 異なる n と m の場合の実験結果については文献 [2] を参照のこと。

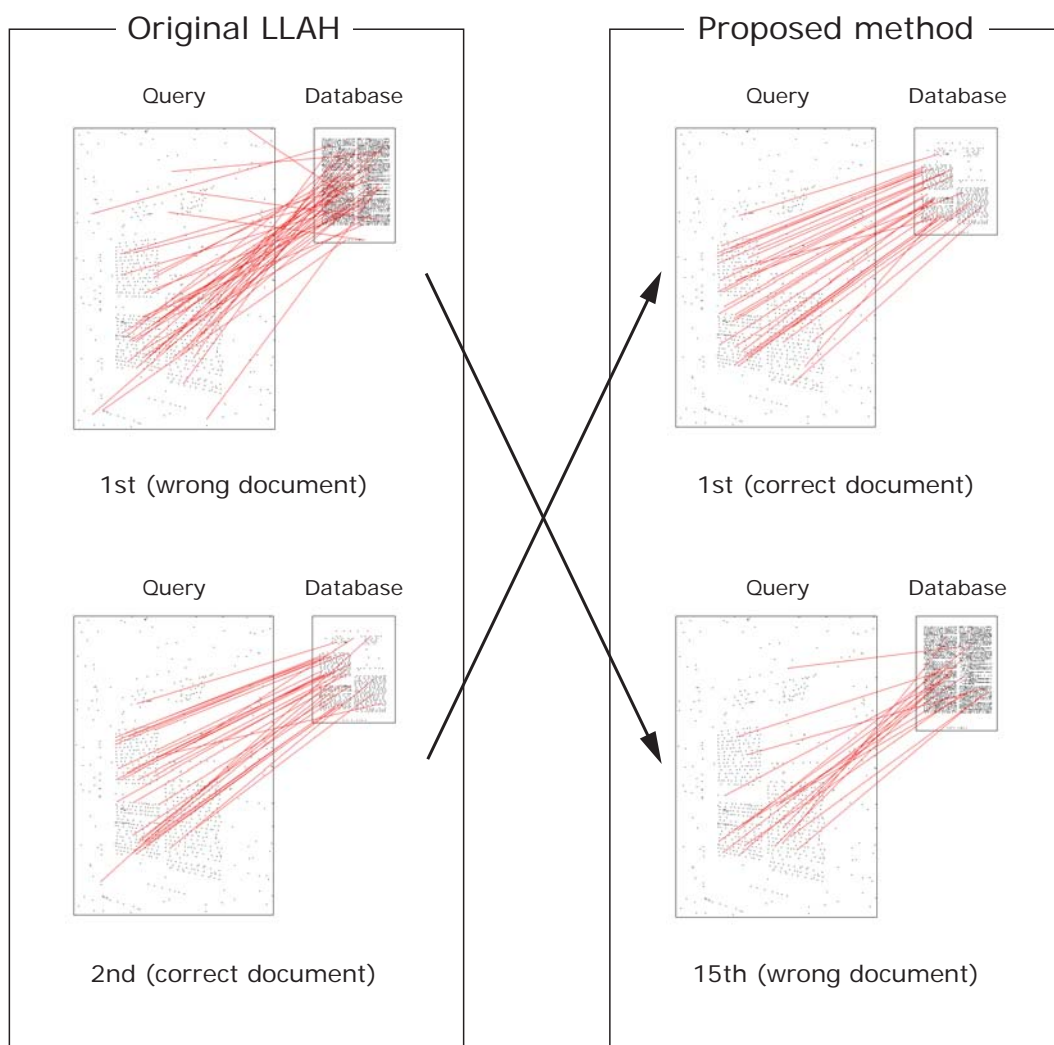


図 16 誤対応の減少による精度向上の例

に成功した検索質問での特徴点の対応関係を示したものである。図 16 には、4 つの特徴点の対応関係が示されている。左側の 2 つはオリジナルの LLAH を用いた場合のものであり、右側の 2 つは提案手法によるものである。それぞれの対応関係では、左側に検索質問の特徴点、右側にデータベース文書の特徴点が描画されており、検索での投票処理によって対応付けられたものが直線で結ばれている。左上の対応関係は、オリジナルの LLAH を用いたときに、最も多くの得票数を得たデータベース文書のものである。このデータベース文書は検索質問に対応しないものであり、従って対応付けられている特徴点はすべて誤対応によるものである。オリジナルの LLAH を用いたときの正解文書の対応関係は左下に示されている。正解文書は得票数は 2 位だった。右上に示されているものが、提案手法を用いた場合で最も多くの得票数を得たデータベース文書の対応関係である。正解文書が最も多くの得票数を得ており、検索は成功した。なお、正解文書では、オリジナルの LLAH の場合と比べて得票数の大きな変化はなかった。右下に示されているものは、オリジナルの LLAH において得票数が 1 位となったデータベース文書の、提案手法での対応関係である。誤投票の減少により対応関係が減少し、得票数は 15 位となった。このよう

に、提案手法では誤投票の減少により誤った文書は得票数が減少し、正解文書は得票数が相対的に増加することで検索精度が向上することが示された。

6. ま と め

本稿では、LLAH の改良手法を提案した。不必要な特徴量の削除とデータ構造の単純化により、必要メモリ量の削減を実現した。また、検索アルゴリズムを改良することで処理時間を削減することができた。実験により、必要メモリ量の 80%、処理時間の 60% が削減されたことを確認した。実験ではさらに、改良手法では精度についても向上することが確認された。以上のことから、LLAH のスケーラビリティや、リアルタイム処理などの高速性を要求するアプリケーションへの拡張性が向上したといえる。

今後の課題としては、今回の改良では触れなかった特徴点抽出処理の改良が挙げられる。実際に LLAH を用いる場合には、画像処理を必要とする特徴点抽出処理の処理時間の長さがボトルネックとなる場合があるため、この点についても高速化が必要である。また、ポスターや標識といった一般の平面物体への検索対象の拡張も必要である。

謝 辞

本研究の一部は、科学研究費補助金(基盤研究(B)19300062)、
(特別研究員奨励費19-7621)の補助による。

文 献

- [1] D. Doermann, “The indexing and retrieval of document images: a survey”, *Computer Vision and Image Understanding*, vol. 70, no. 3, pp.287–298, 1998.
- [2] T. Nakai, K. Kise, and M. Iwamura, “Use of affine invariants in locally likely arrangement hashing for camera-based document image retrieval”, *Lecture Notes in Computer Science (7th International Workshop DAS2006)*, vol. 3872, pp.541–552, 2006.
- [3] J. Liang, D. Doermann, and H. Li, “Camera-based analysis of text and documents: a survey”, *IJDAR*, vol. 7, pp.84–104, 2005.
- [4] P. Clark, and M. Mirmehdi, “Recognising text in real scenes”, *IJDAR*, vol. 4, pp. 243–257, 2002.
- [5] C. H. Lampert, T. Braun, A. Ulges, D. Keysers, and T. M. Breuel, “Oblivious document capture and real-time retrieval”, *Proc. CBDAR2005*, pp.79–86, 2005.