

デジタルカメラによる文書画像検索 — 1万ページから0.1秒で検索する —

Document Image Retrieval with Digital Cameras — Retrieval in 0.1 Second from 10,000 Pages —

中居 友弘 † 黄瀬 浩一 † 岩村 雅一 †
Tomohiro Nakai Koichi Kise Masakazu Iwamura

1 はじめに

デジタルカメラの高機能化、一般化に伴い、その情報デバイスとしての利用が注目されている。最も素朴な利用法の一つは検索である。すなわち、デジタルカメラで撮影した対象物の画像をキー（検索質問）として、その対象物に関する情報を検索することである。本稿では、このようなデジタルカメラを用いた検索の一形態として、文書画像検索を考える。

デジタルカメラを用いた文書画像検索とは、手元の文書を撮影した画像をキーとして、データベースから同一文書（PDF など）を検索する処理である。この技術は、撮影画像からの関連情報の取得にも応用できる。文書画像検索によって対応する文書が得られるのであれば、あらかじめデータベース中の文書の関連情報を定義しておくことで印刷文書の撮影画像から関連情報を得ることができるからである。具体的には、学术论文の撮影による関連した別の論文の取得や、商品広告の撮影による商品についてのより詳しい情報の取得といったサービスが考えられる。

デジタルカメラを用いた文書画像検索が、従来のスキャナベースの検索 [1] と大きく異なる点は、対象が2次元統制環境で撮像されるか、3次元非統制環境で撮像されるかにある [2, 3]。

この問題に対して、我々は既に複比の投票による検索手法を提案している [4]。この手法の特徴は、複比という射影変換の不変量を用いた文書画像の索引付けと、ハッシュを用いた投票による効率的検索にある。

本稿では、この手法を検索速度の面で大幅に改良した手法を提案する。提案手法の特徴は、索引付けにおいてインデックスの計算方法を改良することにより、ハッシュの衝突を大幅に抑える点にある。また本稿では、1万ページの文書画像を対象とした検索実験の結果から、本手法の検索速度と精度を検証する。

2 提案手法

2.1 基本方針

デジタルカメラを用いた文書画像検索を実現するためには、デジタルカメラで撮影された文書画像と、データベース中の文書画像を照合する必要がある。この際、照明・フォーカス・手振れなどの画像劣化の要因、低解像度、幾何歪みなど様々な問題に対処しなければならない [3]。提案手法では、このうち特に画像が受ける幾何歪

み（射影変換歪み）への対処に着目し、以下の方針を立てる。なお、処理対象は現在のところ英文文書とする。

- 特徴量 … 射影変換歪みによらない文書画像の特徴量を得るため、射影変換の不変量である複比を用いて特徴量を計算する。複比の計算に用いる特徴点には、射影変換やノイズに対する安定性を得るため、単語の重心を用いる。また、部分画像からの検索を可能にするため、文書画像の部分ごとに複比を用いて特徴量を計算する。
- 登録・検索 … 特徴点に基づく画像検索の単純な方法は、特徴点の可能な対応付けをすべて試すことであるが、組み合わせ爆発が生じるため現実的ではない。この問題点は、複比を含む幾何学的不変量の実利用を妨げるものである [5]。本手法ではこの問題点を解決するため、ハッシュを用いた投票を導入し、特徴点の対応を陽には求めずに検索を行うことを考える。登録時には、まず文書画像から特徴点を求めて特徴量を計算し、そこからインデックスを求めてハッシュに登録する。検索時には、検索質問から同様の処理で特徴点、特徴量、インデックスを順に求めてハッシュにアクセスし、登録されている文書に対して投票を行うことで検索する。

2.2 複比

次に本手法で用いる複比について述べる。複比は、射影変換の不変量であり、同一平面上の5点 $ABCDE$ の座標から以下の式で計算される [6]。

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

ここで、 $P(A, B, C)$ は頂点 A, B, C で囲まれる三角形の面積である。複比は射影変換の不変量であるため、点 $ABCDE$ の座標が射影歪みによって変化しても、その値は同一に保たれるという性質をもつ。

さて、画像上の特徴点から得られる複比の値は連続値だが、インデックスに用いる際には離散化する必要がある。その際には、頻度の高い値に対しては細かく、頻度の低い値に対しては粗く離散化すると効果的である。本手法では、予備実験により特徴点から得られる複比のヒストグラムを作成し、各階級における複比の頻度が等しくなるように離散化する。

2.3 処理の流れ

提案する文書画像検索システムの構成を図 1 に示す。文書画像は、特徴点抽出により点の集合に変換された後、登録の場合は登録処理へ、検索の場合は検索処理に入力される。登録処理では、特徴点を用いて複比を計算

† 大阪府立大学大学院工学研究科, Graduate School of Engineering, Osaka Prefecture University

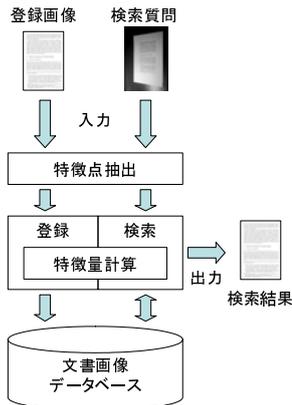


図1 文書画像検索システム

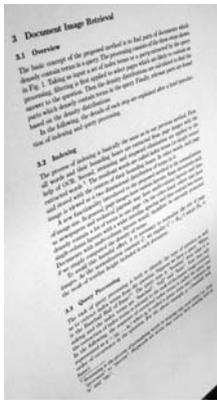


図2 入力画像

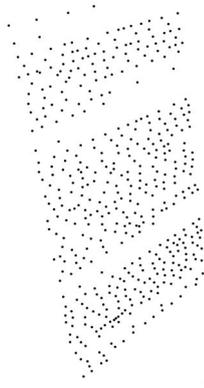


図3 特徴点

し、それをインデックスに変換したのち、ハッシュに文書を登録する。一方、検索処理は、特徴点から同様にインデックスを計算し、投票を行うことで所望の文書画像を検索する。以下、各々について述べる。

2.4 特徴点抽出

特徴点抽出では、複比の計算に用いる特徴点を求める。ここで重要なことは、特徴点の再現性、すなわち射影歪み、ノイズ、低解像度の影響下でも同一の特徴点が得られることである。英文文書における単語の重心は、この条件を満たす特徴点の1つである。これは、英文文書では単語と単語の間に空白があり、分離が比較的容易なためである。

処理の概要は以下の通りである。まず、入力画像を適応2値化 [7] により2値画像に変換する。次に、2値画像に対してガウシアンフィルタを適用し、再度2値化を行うことで単語領域を得る。ここで、ガウシアンフィルタのパラメータは文字サイズの推定値に基づいて自動的に定める。最後に、各単語領域の重心を求め、これを特徴点とする。図2に示す入力画像からは、図3のような特徴点が抽出される。

2.5 インデックスに用いる特徴量計算

登録、検索の鍵は、複比を用いていかにハッシュのインデックスを計算するかにある。ここでは、登録、検索の詳細について述べる前に、インデックスに用いる特徴

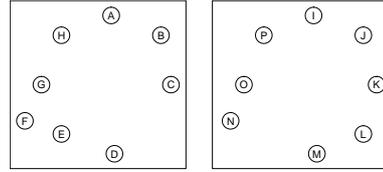


図4 8点

量の計算方法についてまとめる。

文献 [4] では、可能な特徴量として、以下の3つを比較考察している。

1. 近傍5点の複比
2. 近傍 n 点からの5点の複比
3. 近傍 n 点からの $m (< n)$ 点の配置、および m 点からの5点の複比

その結果、正しい検索を行うためには最も複雑な3.が必要であることを示している。この主な理由は、複比を計算するために必要な近傍5点が射影変換歪みによって変化することである。

本稿では紙面の都合から、現在用いている3.の手法について図4の具体例を用いて説明する。ここで ABCDEFGH と IJKLMNOP は、同一文書の異なる画像において、対応する点の近傍8点の組とする。8点の組のうち異なるものは E, L の1点のみであり、残り7点は共通である。従って、それぞれの8点からすべての7点の組み合わせを取り出せば、そのうち ABCDFGH と IJKLMNPO の1組はまったく同じものになる。このように完全に共通な7点の組から、ある定まった順序ですべての5点の組み合わせを生成すれば、順序も含めて完全に一致する。つまり、最初は ABCDFG から得られた複比と IJKLMN から得られた複比、次は ABCDFG から得られた複比と IJKMO から得られた複比というように、共通な7点から同じ順序で複比を計算すれば、同じ複比が同じ順序で得られる。異なる7点の組から同じ複比がまったく同じ順序で現れるということはまれであるので、この順序付けられた複比を特徴量として用いることが考えられる。

特徴量の計算方法についてまとめる。まず、与えられた点の近傍 n 点を取り出す。次に、 n 点からすべての m 点の組み合わせを作成する。 m 点から定まった順序で5点の組み合わせを作成し、複比を計算する。ここで、複比の値が順序も含めて完全に一致したとき、特徴量は一致すると判定する。

2.6 登録

以上の準備に基づき、図5に示す登録アルゴリズムについて説明する。登録処理では、登録する文書の各点について、その近傍 n 点からすべての m 点の組み合わせを生成し、そこから複比を用いたインデックスを求めて図6に示すハッシュに登録する。

以下、アルゴリズムに沿って説明する。1行目で特徴点の集合から1点を取り出して p とし、2行目で p の近傍 n 点を時計回りに取り出して P_n とする。3行目では P_n から m 点を取り出して P_m とし、 P_m から得られるすべての5点の組み合わせについて5行目で複比の値を計算し、離散化する。この値を $r_{(i)}$ とする。 m 点から得

- 1: for all $p \in \{ \text{特徴点の集合} \}$ do
- 2: $P_n \leftarrow p$ の近傍 n 点 (時計回り)
- 3: for all $P_m \in \{ P_n \text{からのすべての } m \text{ 点の組み合わせ} \}$ do
- 4: for all $P_5 \in \{ P_m \text{からのすべての } 5 \text{ 点の組み合わせ} \}$ do
- 5: $r_{(i)} \leftarrow P_5$ から計算される複比
- 6: end for
- 7: $H_{\text{index}} \leftarrow r_{(0)}, \dots, r_{(mC_5-1)}$ から作成したハッシュのインデックス
- 8: H_{index} を用いて文書 ID, 点 ID, $r_{(0)}, \dots, r_{(mC_5-1)}$ をハッシュに登録
- 9: end for
- 10: end for

図5 登録アルゴリズム

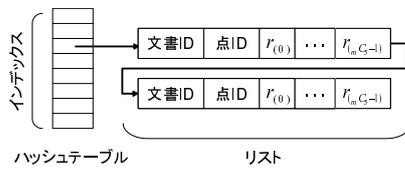


図6 ハッシュの構造

られる5点の組み合わせの数は mC_5 であるので, i は $0, \dots, mC_5 - 1$ の値をとる. このようにして得られた複比 $r_{(i)}$ より, ハッシュ関数を用いて7行目でハッシュのインデックス H_{index} を求め, 8行目で H_{index} を用いて文書 ID (登録文書の識別番号) と点 ID (点の識別番号), $r_{(i)} (i = 0, \dots, mC_5 - 1)$ をハッシュに登録する. ハッシュ関数を以下に示す.

$$H_{\text{index}} = \left(\sum_{n=0}^{mC_5-1} r_{(n)} k^n \right) \bmod H_{\text{size}} \quad (2)$$

ここで, k は複比の量子化レベル数, H_{size} はハッシュのサイズである. また, 登録時に衝突が生じた場合, データは図6のようにリストとして付け加えられる.

このような処理を P_n 中のすべての P_m およびすべての p について繰り返し, 文書の登録を行う.

2.7 検索

次に検索について述べる. 図7に検索アルゴリズムを示す. 同様に, アルゴリズムに沿って説明する. まず5~8行目で登録時と同様の手法によりハッシュのインデックスを求め, 9行目でハッシュを引いて図6にあるようなリストを得る. リストの各要素について, 登録されている複比と検索質問の複比が完全に一致しているかどうか確認し, 一致しているものについては文書 ID の1次元配列である投票テーブルの該当するセルに投票を行う.

投票は検索質問の特徴点と登録文書の点を対応付けることに相当する. ここで, 点ごとに独立に対応付けると, 図8のように1つの点と複数の点に対応するという誤った対応が生じる. このような誤対応による投票を得票数に含めると正しい対応による投票の評価が下がり, 検索が失敗する原因となる. そこで本手法では, 検索質問と登録文書の点の対応を記録しておき, すでに対応付

- 1: for all $p \in \{ \text{特徴点の集合} \}$ do
- 2: $P_n \leftarrow p$ の近傍 n 点 (時計回り)
- 3: for all $P_m \in \{ P_n \text{からのすべての } m \text{ 点の組み合わせ} \}$ do
- 4: for all $P'_m \in \{ P_m \text{の要素のすべての巡回置換} \}$ do
- 5: for all $P_5 \in \{ P'_m \text{からのすべての } 5 \text{ 点の組み合わせ} \}$ do
- 6: $r_{(i)} \leftarrow P_5$ から計算される複比
- 7: end for
- 8: $H_{\text{index}} \leftarrow r_{(0)}, \dots, r_{(mC_5-1)}$ から作成したハッシュのインデックス
- 9: H_{index} を用いてハッシュを引き, リストを得る
- 10: for all リストの各要素について do
- 11: if 登録されている複比と $r_{(0)}, \dots, r_{(mC_5-1)}$ が一致 and p が文書 ID の点として投票されていない and 文書 ID の点 ID で投票されていない then
- 12: 投票テーブルの文書 ID に投票
- 13: end if
- 14: end for
- 15: end for
- 16: end for
- 17: end for
- 18: 投票テーブルで検索結果を判定する

図7 検索アルゴリズム

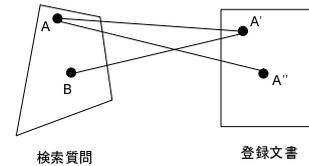


図8 A' と A'' が同時に A と対応したり, A と B が同時に A' に対応したりする誤った点対応

けられている点については投票を行わないことにより誤った対応による投票を制限する. このような制限を行うと, 先に対応付けられるものが優先されるため, 対応付けられる点が必ずしも正しく対応したものになるとは限らない. しかし, 本稿の目的である検索処理を行う上では, 必要な情報は文書ごとの対応点の数であり, 点ごとの対応先が正しい必要はないため, 問題は生じない.

このような処理をすべての点について行い, 最終的に投票テーブルで得票数が最大の文書を検索結果とする.

なお, 4行目で P_m から開始点を変えた P'_m を作成し, すべての P'_m について処理を行っているのは, P_m として ABCDEFG が与られたとき, P'_m として BCDEFGA, CDEFGAB のように巡回させたものを m 通り作成することに相当する. これは, 画像が回転している場合に対応するためである.

3 実験

3.1 実験概要

提案手法の有効性を検証するために, データベースのサイズと検索精度および検索速度の関係について調べた. 検索質問の作成には, CANON EOS Kiss Digital

