

特徴点の配置に基づく画像検索手法の解析——Geometric Hashing と LLAH の比較——

岩村 雅一^{†a)} 中居 友弘[†] 黄瀬 浩一[†]

Analysis on Image Retrieval Method Based on Arrangements of Feature Points
——Comparison of Geometric Hashing and LLAH——

Masakazu IWAMURA^{†a)}, Tomohiro NAKAI[†], and Koichi KISE[†]

あらまし 画像が複数の特徴点のみで記述されているとき、同一の配置をもつ画像を検索するにはどうすればよいだろうか。画像は射影ひずみによってひずんでいるかもしれないし、特徴点の配置が完全に同一ではなく、特徴点の挿入や消失があるかもしれない。この問題を解決する手法として、Geometric Hashing (GH) がよく知られている。一方、我々は Locally Likely Arrangement Hashing (LLAH) という手法を提案している。いずれの手法も全探索より効率的に画像を検索できるが、計算時間とメモリ使用量においては LLAH が優れている。本論文では GH から LLAH に段階的に変化させることにより、効率的な探索に寄与する要因を考察する。

キーワード Geometric Hashing, LLAH, 検索精度, 計算時間, メモリ使用量

1. ま え が き

本論文では、平面状の物体から特徴点（例えば英語の重心）を抽出したとき、特徴点の配置が、あらかじめ登録された特徴点の配置のどれと同じであるかを発見する検索問題を考える。この問題では特徴点自身は座標以外の情報をもたないため、検索対象の特徴点が、登録された特徴点のどれと対応するのかを単一の特徴点から判断できない。そのため、必然的に周囲の点の配置を用いることになる。しかし、検索対象には、検索対象を斜めから見たときに生じる射影ひずみなどの幾何変換が施されているため、唯一の情報である特徴点の座標も登録されているものと全く同一でなく、周囲の点との位置関係も幾何変換の影響を受ける。このように、この問題では幾何変換の影響を排除しつつ、いかに正確かつ高速に対応する特徴点の配置を捜し当てるかが重要になる。

この問題を解決する有効な手法として Geometric Hashing (GH) [1] ~ [3] が広く知られている。しかし、

GH の計算時間やメモリ使用量は非常に大きく、GH が使用可能なアプリケーションは限定されている。GH の計算時間やメモリ使用量を削減する方法も提案されているが、検索精度は計算時間やメモリ使用量とトレードオフの関係にあることが知られており [4]、精度を保持しつつ計算時間を削減することは困難である。

それに対して、我々は GH のジレンマを解決する Locally Likely Arrangement Hashing (LLAH) を提案している [5], [6]。この手法は GH の欠点であった登録画像数や特徴点数の増加に伴う検索速度やメモリ使用量の増加を抑制する。これにより、カメラで撮影された文書画像が、データベースに登録されている約 10,000 ページの文書画像のどのページのどの部分であるのかを実時間で検索することが可能になった [7]。

本論文の目的は、検索速度とメモリ使用量における LLAH の優れた性質がどの処理に起因するかを明らかにすることである。そのために、これまで LLAH との関連性が十分議論されてこなかった GH を段階的に発展させることで LLAH が得られると考え、GH と LLAH の差異を段階的に吸収する中間的な手法を三つ作成する。これらの手法の性能を理論的、実験的に比較、検証することで、前述の目的を達成する。

[†] 大阪府立大学大学院工学研究科，堺市

Graduate School of Engineering, Osaka Prefecture University, 1-1 Gakuencho, Naka-ku, Sakai-shi, 599-8531 Japan

a) E-mail: masa@cs.osakafu-u.ac.jp

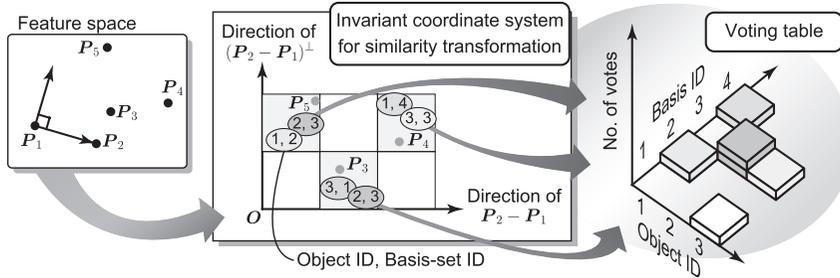


図 1 Geometric Hashing の検索処理 (相似変換の場合)
 Fig.1 Retrieval process of Geometric Hashing for similarity transformation.

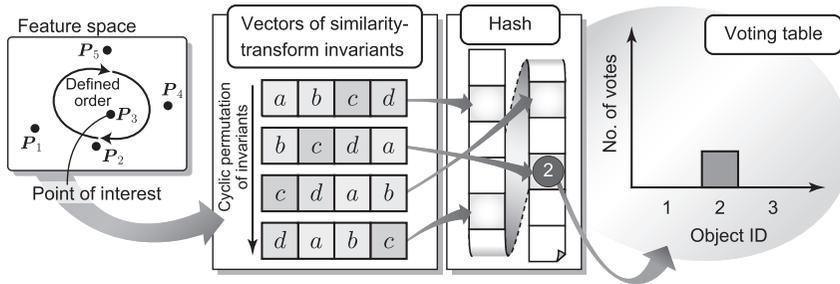


図 2 LLAH の検索処理 (相似変換の場合). 図中の不変量はそれぞれ, $a = \frac{\|P_4 - P_1\|}{\|P_5 - P_1\|}$,

$$b = \frac{\|P_2 - P_5\|}{\|P_4 - P_5\|}, c = \frac{\|P_1 - P_4\|}{\|P_2 - P_4\|}, d = \frac{\|P_5 - P_2\|}{\|P_1 - P_2\|} \text{ である.}$$

Fig.2 Retrieval process of LLAH for similarity transformation.

2. Geometric Hashing と LLAH

GH と LLAH の登録及び検索処理を概説する.

2.1 Geometric Hashing

2.1.1 登録処理

GH は幾何変換に不変な座標系を用いて画像を記述する. ここでは相似変換に対応する場合の登録処理について簡単に述べる. アフィン変換や射影変換の場合については, 差異を脚注に記す. 図 1 は検索処理について説明した図であるが, 登録処理との共通点が多いので, 参考にして頂きたい.

登録画像から特徴点が抽出されたとき, まずその特徴点から 2 点を選び, 図 1 のように 2 本の基底ベクトルを作成する^(注1). 図 1 ではベクトル $P_2 - P_1$ とそれに直交するベクトル ($(P_2 - P_1)^\perp$ と表記する) が基底である. そして, 残りの点 P_3, P_4, P_5 を図 1 中央に示すように相似不変な座標系に射影する^(注2). この座標系の特徴点の座標は, 基底が同じであれば画像の相似変換によらず一意に定まる. 相似不変な座標系はあらかじめ小領域 (二次元ハッシュ表のピンに相

当) に分割されており, 登録画像番号 (Object ID) と基底番号 (Basis ID) をその領域に関連づける. この処理を, $P_2 - P_1$ と $(P_2 - P_1)^\perp$ を基底とする座標系だけでなく, すべての座標系について行い, 更に登録するすべての画像について行うことで登録処理が完了する.

2.1.2 検索処理

図 1 を例に検索処理を説明する. まず登録処理と同様に特徴点から 2 点を選び, 図 1 のように 2 本の基底ベクトルを作成する. そして残りの点を相似不変な座標系に射影する. 座標系は登録時に小領域に分割され, 画像と基底の番号が登録されているので, 各特徴点が射影された領域の情報を読み出し, Object ID と Basis ID の組に投票する. 上記の処理をすべての座標

(注1): アフィン不変な場合は 3 点, 射影不変な場合は 4 点を選び, それぞれ基底ベクトルを 3 本ないし 4 本作成する.

(注2): 不変座標系の次元数は, 相似不変, アフィン不変の場合は基底ベクトルが 2 本なので, 明らかに二次元である. 射影不変な場合, 基底ベクトルが 3 本なので三次元とすることもできるが, 基底ベクトルの作成に用いる 4 点を作る四角形を正方形に変形することで, 二次元の射影不変座標系を作成することもできる. 本論文ではこの方法を用いて射影不変座標系を作成した.

系について行い、投票度数が最大となった画像と基底の組が検索結果となる。図 1 の例では画像 2 と基底 3 の組が最大の票数を得たので、画像 2 が検索結果となる。なお、すべての基底を用いる前に検索結果が確定した場合は途中で処理を中止できる。

2.2 LLAH

2.2.1 登録処理

相似変換に対応する場合の LLAH の登録処理について説明する。図 1 同様、図 2 も検索処理を説明する図であるが、登録処理の説明においても参考にして頂きたい。

特徴点が抽出されたとき、LLAH ではある特徴点（注目点）の近傍 m 点の配置を利用して注目点の特徴ベクトルを定義する。図 2 を用いて具体的に説明すると、 P_3 が注目点であり、近傍 m 点（図 2 では 4 点 P_1, P_2, P_4, P_5 ）の配置から下記の手順で注目点 P_3 の特徴ベクトルを計算する。まず、近傍点を時計回りに並べると、 $P_5, P_4, P_2, P_1, P_5, \dots$ となる。ここである点を起点に時計回りに 3 点を選び、 A, B, C とおけば、2 点間の長さの比 $\frac{AC}{AB}$ は相似不変量であるので、 A となる点を時計回りに移動しながら mC_3 個の相似不変量（図 2 における a, b, c, d ）を計算する^(注3)。次に、これらの不変量を並べて mC_3 次元（図 2 では四次元）の特徴ベクトルを作成する。その際、どの点を起点にするかで m 個のベクトル（図 2 における $abcd, bcda, cdab, dabc$ ）を作成できるが、登録時にはそのうち一つだけを適当に選択して登録する。そして、作成した特徴ベクトルからハッシュ値を計算し、ハッシュ表に画像番号を登録する^(注4)。この処理をすべての特徴点を注目点にして行い、更に登録するすべての画像について行うことで登録処理が完了する。

LLAH は近傍点の情報を組み合わせて特徴ベクトルを作成するため、特徴点の変動や消失、あるいは画像を斜めから撮影することにより生じる近傍点の変化に弱く、このままでは検索精度が GH よりも低下しやすい。そのため、実際には最初に近傍 n 点（ $n \geq m$ ）を選び、その中から m 点を選ぶ。つまり、特徴ベクトルは上記の説明の nC_m 倍作成する。このように特徴ベクトルを冗長化することで登録時と検索時で同じ m 点が選択される確率が上昇し、頑健性が増す。

2.2.2 検索処理

LLAH の検索処理を説明する。特徴点が抽出されたとき、登録時と同様に注目点の特徴ベクトルを計算す

る。図 2 を例にすると、不変量 a, b, c, d を計算するまでは登録処理と同じ処理である。異なるのは、これらの不変量を並べて mC_3 次元の特徴ベクトルを作成するとき、作成可能な m 個のベクトル ($abcd, bcda, cdab, dabc$) をすべて作成して用いる点である^(注5)。そして、作成した特徴ベクトルからハッシュ値を計算し、ハッシュ表に登録されている画像番号に投票する^(注6)。この処理を、すべての特徴点を注目点にして行い、投票度数が最大の画像を検索結果とする。図 2 では画像 2 が 1 度、何も登録されていない要素が 3 度引かれたため、画像 2 が検索結果となる。GH 同様に、1 位と 2 位の得票数に大きな差が生じた場合は LLAH でも途中で処理を中止することができる。なお、検索時も登録時と同様に、近傍 n 点（ $n \geq m$ ）から m 点を選ぶことで頑健性を向上させている。

3. GH の改良による LLAH の導出

登録画像数や特徴点数が多いときの LLAH の検索時間とメモリ使用量は GH に比べて飛躍的に改善されている。本章ではこの理由を考察する。

3.1 Geometric Hashing の問題点

LLAH について述べる前に、GH の計算時間とメモリ使用量が大きい理由を三つの観点から考察する。以下では画像 1 枚当りの特徴点数を N 、画像の枚数を M とする。また、GH の基底の数を b とする。 b の値は幾何変換のクラスで決まり、相似変換で $b = 2$ 、アフィン変換で $b = 3$ 、射影変換で $b = 4$ である。

一つ目の要因は、特徴点数を基準とした計算量である。文献 [1] に記載されているオリジナルの GH では、登録時に $O(N^{b+1}M)$ 、検索時には最悪 $O(N^{b+1})$ の計算量が必要となる^(注7)。メモリ使用量は、登録時の計算時間と同じオーダが必要である。したがって、現在の一般的な計算機では、数百点程度の特徴点を扱うことさえ困難である。文献 [3] では特徴点や基底を確率

(注3): アフィン不変の場合は二つの三角形の面積比をアフィン不変量として、射影不変の場合は四つの三角形の面積の複比を射影不変量として本論文で使用する。

(注4): 具体的な方法は 3.2.3 で述べる。

(注5): 文献 [8] ではベクトルを 1 本に削減する手法が提案されているが、本論文では文献 [6] のオリジナルの LLAH に基づき、 m 本のベクトルをすべて用いる。

(注6): ハッシュ表に画像番号だけでなく、特徴点の番号も登録しておくことにより、検索している注目点がデータベース中のどの画像のどの点に対応するかまで実時間で特定することができる。文献 [7], [9] はこうして得られた特徴点の対応を拡張現実に応用した例である。

(注7): N^{b+1} は、 N 点から b 点を選択して作成する不変座標系の数 N^b と、一つの不変座標系に射影される点の数 N の積で算出される。

的に選択する（間引く）ことによる計算量削減に言及しているが、このような確率的な処理の場合、検索精度は計算量やメモリ使用量とトレードオフの関係にあることが知られ [4]、実用的な精度を保持しつつ計算量を削減することはやはり困難である。

二つ目の要因はハッシュ値の衝突である。GH においてハッシュ表に登録される情報（画像番号と基底番号の組）の総数は $N^{b+1}M$ である。二次元の不変座標系が K^2 個の均等な大きさの小領域に分割されているとすると、ハッシュ表のピン一つ当りに登録される情報の数（すなわち、衝突の数）は $N^{b+1}M/K^2$ である。つまり、検索時にハッシュ値を 1 回計算するたびに $N^{b+1}M/K^2$ 回の投票をすることになる。具体的な数値を当てはめてみると、 $N = 100$, $M = 100$, $b = 3$, $K = 1000$ の場合、投票回数は 10000 回にも及ぶ。衝突を減少させるには当然特徴点数 N と画像の枚数 M を減少させればよいのだが、それが困難な場合には前述の領域分割のパラメータ K を大きくして衝突を少なくするしかない。しかし、 K が大きいということはハッシュ表が細かく区切られているということなので、特徴点の位置の変動によって本来引かれるべきハッシュ表のピンとは異なるピンが引かれる可能性が増加し、結果として検索精度が低下する。

三つ目の要因は投票テーブルの大きさである。GH は投票テーブルに N^bM 個のピン^(注8)をもつため、得票が最大のピンを発見するには $O(N^bM)$ の計算量が必要である。

3.2 Geometric Hashing の改良法としての LLAH

GH と LLAH の関連性や相異点を明らかにするため、LLAH を GH の改良法と位置づける。すなわち、GH に以下の四つの改良を順に施すことにより、3.1 で述べた問題点を解決し、LLAH を導出する。以下で述べる手法 1~4 を 4. で実験により比較する。

3.2.1 手法 1: 注目点の導入と近傍 m 点の計算

オリジナルの GH [1] は画像中の全特徴点を用いる。それに対して、用いる点数を減らすことによる高速化を行い、手法 1 とする。3.1 で述べたように、確率的な特徴点や基底の削減では検索精度の減少を免れない。そこで特徴点の一つずつ選んで注目点とし、その近傍 m 点のみを用いた処理を考える。すなわち、 m 点のうち b 点を基底の作成に用い、残りの $m - b$ 点を幾何変換に不変な座標系に射影し、この処理を繰り返す。これにより、計算量は大幅に削減され、登録時の計算量

は $O(m^bNM)$ 、検索時の計算量は $O(m^bN)$ となる。

検索が成功するためには、登録時に使用した特徴点と検索時に使用する特徴点が同一である（対応する）必要がある。そのため、画像を斜めから撮影した場合は、対応する特徴点が近傍点でなくなることが原因で検索精度が低下する可能性がある。この問題については 3.2.4 で議論する。

3.2.2 手法 2: 順序の導入による「確率的でない」不変量の削減

3.1 で述べた一つ目の問題点を解決する。LLAH は、常に同じ特徴点を選択されるという再現性を利用することにより、検索精度を保持しつつ計算量を削減する。そのために、同一平面上にある特徴点満たす次の法則を利用する。

ある特徴点に注目したとき、その近傍点を時計回りに並べた順序は幾何変換に不変である（ただし、鏡像変換を除く）。

この法則を用いて「時計回り」の順序で特徴点を選択すれば、真正面から見ても斜めから見ても同じ順序に並んだ特徴点が得られる。そして、特徴点を一定の順序で選択することで、不変量を常に同じ順序で選択することができる。ただし、「時計回り」の順序を利用しても、「順序の起点」は一意に定まらないため、起点は全点を試すことにする^(注9)。

GH や手法 1 では不変座標系の特徴点の座標を不変量として用いたが、手法 2 では順序を利用したものに変更する。具体的には、まず注目点の近傍 m 点を選択し、適当な点を起点に m 点を時計回りに並べる。その中から $b + 1$ 点を選択し、幾何学的不変量（線分の長さの比、面積比、面積複比等）を計算する。 m 点から $b + 1$ 点を選択する組合せは ${}_m C_{b+1}$ 通りあるので、 ${}_m C_{b+1}$ 個の不変量が計算できる。これを画像中の N 点すべてを注目点として行くと、不変量は合計 ${}_m C_{b+1}N$ 個になる。

手法 2 では、特徴点の情報が通常用いられる二次元のハッシュ表に格納される。ハッシュ値 H_{index} は不変量を k 段階に離散化した値 r を用いて次式で算出される。

$$H_{\text{index}} = r \bmod H_{\text{size}} \quad (1)$$

(注8): ピンの数 N^bM は画像の枚数 M と基底の数 N^b の積である。

(注9): 既に述べたように、文献 [8] では開始点を一意に決定する方法が提案されているが、本論文ではオリジナルの LLAH のみを考察の対象とする。

ここで H_{size} はハッシュ表の大きさである．ハッシュ表を変更することで前節で述べた三つ目の問題点も解決する．すなわち，投票テーブルが図 1 から図 2 のように変更になり，投票テーブルから基底番号の軸がなくなるため，得票が最大のピンを発見するのに必要な計算量が $O(N^b M)$ から $O(M)$ に減少する．

順序を導入した結果，登録時の計算量は $O(m C_{b+1} N M)$ ，検索時の計算量は $O(m C_{b+1} m N)$ となる．これは画像中の N 点すべてについて $m C_{b+1}$ 個の不変量を m 通りの起点で計算するためである．

3.2.3 手法 3：特徴量の組合せによる高次元特徴ベクトルの導入

ハッシュ値の衝突を回避して，3.1 で述べた二つ目の問題点を解決する．GH においてハッシュ値の衝突が多い理由はハッシュ表が小さいからである．それではただハッシュ表を大きくすればいいのかというと，今度は特徴量の識別能力が問題になる．つまり，大きなハッシュ表を用いただけで衝突が減らせるわけではなく，ハッシュ表の大きさに見合った特徴量の識別能力が必要である．

LLAH では，高い識別能力を実現するために複数の不変量を組み合わせている．すなわち， $m C_{b+1}$ 個の「順序をもった」不変量を $m C_{b+1}$ 次元ベクトルとみなし，各要素を k 個に離散化するとすることで，最大で $k^{\{m C_{b+1}\}}$ 通りの情報が識別可能な巨大な特徴量空間が作られる．そのため，大きなハッシュ表を使用すれば，ハッシュ値の衝突がほとんどなく，投票に要する計算量はほぼ $O(1)$ になる．

手法 3 は手法 2 とほぼ同じであり，唯一不変量をベクトル化することのみが異なる．具体的な処理は， $m C_{b+1}$ 個の不変量をそれぞれ k 段階に離散化して $r_{(i)}$ ，($i = 0, \dots, m C_{b+1} - 1$) を求め，ハッシュ値 H_{index} を次式を用いて算出する．

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

手法 3 の計算量は手法 2 と同じである．

特徴量の識別能力が高いということは衝突による計算時間の増加を防ぐ以外に，投票の信頼性向上にも貢献している．LLAH では特徴ベクトルが完全に一致しなければ投票しないため，正しく検索するためには $m C_{b+1}$ 個の不変量の離散値がすべて一致しなければならない．このようにすべての不変量が一致する確率は個々の不変量が一致する確率よりも大幅に低くなる

ため，1 票当りの信頼度が非常に高い．

3.2.4 手法 4 (LLAH)：近傍 n 点から m 点を選択することによる頑健性の向上

3.2.3 で述べた特徴ベクトルは識別能力に優れる反面，一つの不変量の算出ミスや離散化誤差も許容しないため，ノイズに弱い．したがって，特徴点の出現や消失 (3.2.1 で述べた画像の傾きの影響を含む) によって，検索精度が低下するおそれがある．そこで LLAH では m より大きな値 n を用いて，注目点当り，手法 3 の $n C_m$ 倍の特徴ベクトルを冗長に作成する．具体的には，注目点の近傍 n 点から m 点を選ぶすべての組合せを求め，各 m 点について手法 3 の手順で特徴ベクトルを作成する．この操作を行うと，特徴点の出現や消失によって $n - m$ 点を失ったとしても正しく検索を行うことができる．

この操作によって新たに生じる計算量は一見膨大に思えるが， n と m の選び方によってはそれほど大きくならない．例えば， $n = 8$ ， $m = 7$ を用いれば ${}_8 C_7 = 8$ であり， $n = 10$ ， $m = 8$ を用いれば ${}_{10} C_8 = 45$ である．最終的な LLAH の計算量は，登録時に $O(n C_m \cdot m C_{b+1} N M)$ ，検索時に $O(n C_m \cdot m C_{b+1} m N)$ となる．

4. シミュレーション

GH と前節で述べた 4 手法 (LLAH を含む) を比較するシミュレーション実験を行った．手法 1 は GH の近傍点のみを使用したもの，手法 2 は順序を導入し，不変量の種類を変えたもの，手法 3 は不変量をベクトル化したもの，手法 4 は LLAH である．実験では，平面物体である検索対象画像を斜めから撮影して射影変換を受けたときの各手法の検索精度，画像 1 枚当りの平均計算時間，メモリ使用量を比較した．検索には射影不変量である面積複比を使用した．検索対象として，人工的に生成した画像 100 枚を用いた．画像 1 枚当り，一様乱数に従う特徴点を 100 点生成し，画像上に配置した．そして，画像に対するカメラの角度を計算機上で 10 度ずつ変化させることに相当する射影変換を画像に施した．その際，特徴点に挿入や消失が全くない場合と，消失する点を等確率で 10% 選択した後，一様分布に従って新たに 10% の特徴点を生成した場合の 2 通りを検証した．

手法 1~3 と LLAH については，近傍点の範囲を表すパラメータである m を $m = 6$ とし，LLAH については更に特徴ベクトルの冗長度を表すパラメータであ

る n を $n = 7$ とした．手法 2 と 3 と LLAH で使用するハッシュ表の大きさは $H_{\text{size}} = 2^{27} - 1$ とした．表 1 に、実験に用いた離散化パラメータを示す．これは、GH と手法 1 については K に、手法 2 と 3 と LLAH については k に相当する．離散化パラメータである K と k は、予備実験により検索精度が最大となるように選択した．

計算機は AMD Opteron 2.4 GHz を搭載したものを使用した．図 3 に角度を変化させたときの検索精度を、表 2 に画像を正面から撮影した場合 (0 度) の検索精度、画像 1 枚当りの平均計算時間、メモリ使用量を示す．数値はいずれも 10 回の実験の平均値である．図 3 (a) と表 2 (a) は特徴点の消失と挿入がない場合、図 3 (b) と表 2 (b) は消失と挿入が 10% ずつある場合である．GH については他の手法と同じ条件で計算が終了しなかったため、画像の特徴点数を 10 点に削減して実験した．他の手法と同様に 100 点を用いた場合の平均計算時間とメモリ使用量を計算量の理論値から求め、表中に括弧書きで記す．

メモリ使用量については、GH が最も大きく、手法 1 は次に大きかった．それ以外の手法はほぼ同程度の

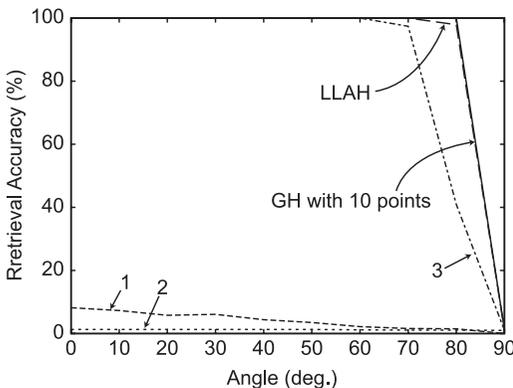
メモリ使用量であった．平均計算時間については、GH が最も大きく、手法 1 と 2 が次に大きく、手法 3 が最も小さかった．LLAH の計算時間は手法 3 の約 3.2 倍であった．検索精度については、特徴点の消失と挿入の有無によらず、GH、LLAH、手法 3 の 3 手法は 100% に近い検索精度を達成したが、手法 1 と 2 の検索精度は非常に低かった．特徴点の消失と挿入の有無で認識精度を比べると、これらがある場合はない場合に比べて検索精度が全体的に低くなる傾向が見られた．

検索精度が高かった手法を見てみると、GH は 80 度の傾きまで検索可能であり、特徴点を 10 点しか使用していないことを考慮すれば頑健といえるが、計算に要する時間が他手法に比べてけた違いに大きかった．手法 3 は近傍点のみで検索を行うため、傾きに対する頑健性は GH より劣っているものの、GH で 100 点を使用したときの理論値に比べて、計算時間は 10 億分の 1 以下、メモリ使用量は 10 万分の 1 以下になった．LLAH は手法 3 より計算時間が若干増加したものの、特徴ベクトルの冗長化によって検索の頑健性が GH と同程度に向上した．すなわち、LLAH は GH と同程度の検索精度をわずかな計算時間とメモリ使用量で実現した．

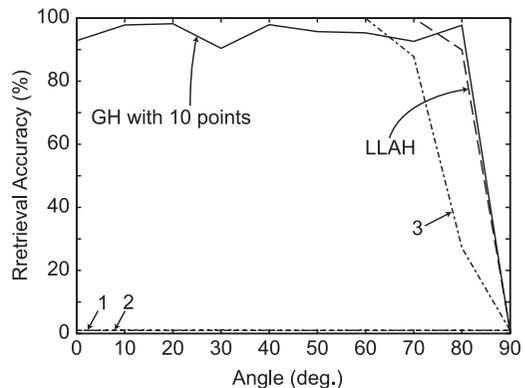
検索精度が低かった手法を見ると、手法 1 は、GH において近傍の特徴点のみを用いる手法であるため、計算時間は GH に比べて大幅に減少したものの、識別能力が失われたものと思われる．手法 2 については、手法 3 のように不変量をベクトル化しないため、特徴量空間が小さく、十分な識別能力が得られなかったものと思われる．また、手法 2 の計算時間が他手法に比

表 1 離散化のパラメータ
Table 1 Parameters for discretization.

Method	No Disappearance & Appearance	10% Disappearance & 10% Appearance
GH with 10 points	30000	30000
Method 1	30000	30000
Method 2	2000	5000
Method 3	10000	10000
LLAH	10000	10000



(a) Without disappearance and appearance.



(b) With 10% disappearance and 10% appearance.

図 3 検索精度
Fig. 3 Retrieval accuracy.

表 2 0 度での検索精度 (%), 画像 1 枚当りの平均計算時間 (ms), メモリ使用量 (GByte) (a) 特徴点の消失と挿入がない場合, (b) 特徴点の消失と挿入がそれぞれ 10% 起こる場合

Table 2 Retrieval accuracy (%), average processing time (ms) and memory amount (GByte) in 0 degrees.

(a) Without disappearance and appearance.

Method	Accuracy	Time	Memory
GH with 10 points	100	1.75×10^4 (1.75×10^9)	6.98 (6.98×10^5)
1	8.2	4.66×10^2	6.98
2	1.3	1.50×10^2	1.12
3	100	1.16	1.12
LLAH	100	3.71	1.13

(b) With 10% disappearance and 10% appearance.

Method	Accuracy	Time	Memory
GH with 10 points	93	1.42×10^4 (1.42×10^9)	6.98 (6.98×10^5)
1	1.0	6.08×10^2	6.98
2	1.0	1.28×10^2	1.12
3	100	1.09	1.12
LLAH	100	3.46	1.13

べて大きくなっているが、これは識別能力に乏しい特徴量を使用することでハッシュ値の衝突が大量に発生したためと考えられる。

以上より、各手法の識別能力と計算時間について、以下の結論が導き出される。まず、GH の不変量（不変座標系の特徴点の座標）は、不変量一つ当りの識別能力が高くない。それゆえ、手法 1 のように、用いる不変量を減少させると極端に検索精度が低下する。したがって、GH のように大量の不変量で検索精度を担保する方法では、高い検索精度と小さい計算時間の両立は望めない。また、LLAH の不変量（面積複比等）についても、手法 2 のように単独で使用する限り、識別能力が不十分である。しかし、手法 3 や LLAH のように不変量を一定の順序でベクトル化すれば、GH のように膨大な計算時間を費やすことなく高精度な検索が実現可能である。

5. む す び

本論文では、特徴点の局所的配置に基づく高精度かつ高速な検索方法“Locally Likely Arrangement Hashing (LLAH)”の検索精度と計算時間について考察した。まず、特徴点の配置に基づく画像検索手法として広く知られている Geometric Hashing (GH) を段階的に変化させることで LLAH が得られると考え、両者の中間的な性質をもつ 3 手法を導入し、GH と LLAH の違いを定性的に明らかにした。次に、シミュレーション実験を行い、各手法の性能を定量的に評価した。実験結果より、GH は頑健であるが膨大な計算時間が必要であること、LLAH は GH と同程度の検索精度を少ないメモリ使用量で高速に実現していることを確認した。また、近傍の特徴点のみを使用する効果、順序の導入により不変量をベクトル化して用いる効果、特徴ベクトルの冗長化の効果を検証した。その

結果、LLAH が小さい計算時間で高い検索精度を実現するために、不変量のベクトル化が最も大きく貢献していることが分かった。

本論文ではオリジナルの LLAH について考察したため、文献 [8] で提案された開始点を一意に定めることによって高速化された LLAH については考慮していない。この改良方法を含めた包括的な実験の評価は今後の課題である。また、画像の一部のみを撮影した場合の頑健性の検証も今後の課題である。

謝辞 本研究は科研費補助金基盤研究 (B) 19300062 並びに挑戦的萌芽研究 21650026 の助成を受けた。

文 献

- [1] Y. Lamdan and H.J. Wolfson, “Geometric hashing: A general and efficient model-based recognition scheme,” Proc. ICCV, pp.238–249, 1988.
- [2] H.J. Wolfson and I. Rigoutsos, “Geometric hashing: An overview,” IEEE Comput. Sci. Eng., vol.4, no.4, pp.10–21, 1997.
- [3] 松山隆司, 久野義徳, 井宮 淳 (編), コンピュータビジョン: 技術評論と将来展望, 新技術コミュニケーションズ, 東京, 1998.
- [4] M. Hoffman and M. Lindenbaum, “Some tradeoffs between complexity and reliability in affine matching,” Technical Report CIS9806, Comp. Sci. Dept., Israel Inst. of Tech., 1998.
- [5] 中居友弘, 黄瀬浩一, 岩村雅一, “デジタルカメラを用いた高速文書画像検索におけるアフィン不変量および相似不変量の利用,” 信学技報, PRMU2005-188, Feb. 2006.
- [6] 中居友弘, 黄瀬浩一, 岩村雅一, “特徴点の局所的配置に基づくデジタルカメラを用いた高速文書画像検索,” 信学論 (D), vol.J89-D, no.9, pp.2045–2054, Sept. 2006.
- [7] 中居友弘, 黄瀬浩一, 岩村雅一, “Web カメラを用いたリアルタイム文書画像検索とその拡張現実への応用,” MIRU2006 論文集, pp.1379–1380, July 2006.
- [8] T. Nakai, K. Kise, and M. Iwamura, “Camera based document image retrieval with more time and memory efficient LLAH,” Proc. CBDAR2007, pp.21–28, Sept. 2007.

- [9] 内山英昭, 斎藤英雄, セルヴィエル ミリアム, モロ ギヨム, “LLAH を用いた交差点マーク付き地図と交差点データベースのマッチングに基づく GIS データの AR 提示システム,” 信学技報, PRMU2009-58, June 2009.

(平成 21 年 10 月 13 日受付, 12 月 12 日再受付)



岩村 雅一 (正員)

平 10 東北大・工・通信卒。平 15 同大大学院博士課程了。同年同大学院工学研究科助手。平 16 阪府大大学院工学研究科助手。現在助教。博士(工学)。パターン認識, 文書画像解析, 物体認識, 情報検索などの研究に従事。平 18 本会論文賞, 平 19 IAPR/ICDAR The Best Paper Award 各受賞。IEEE, 情報処理学会, Pattern Recognition Society 各会員。



中居 友弘 (正員)

平 16 阪府大・工・情報卒。平 20 同大大学院博士課程了。同年同大客員研究員。平 19~20 日本学術振興会特別研究員。平 21 (株) 東芝入社。以来, (株) 東芝研究開発センターにて画像認識に関する研究に従事。博士(工学)。平 17 FIT ヤングリサーチ賞, 平 18 本会論文賞, 平 19 情報処理学会山下記念研究賞各受賞。情報処理学会会員。



黄瀬 浩一 (正員)

昭 61 阪大・工・通信卒。昭 63 同大大学院博士前期課程了。同年同大学院博士後期課程入学。平 2 阪府大・工・電気助手。現在, 同大大学院工学研究科教授。博士(工学)。平 12~13 ドイツ人工知能研究センター客員教授。文書画像解析, 情報検索, 画像認識などの研究に従事。平 18 本会論文賞, 平 19 IAPR/ICDAR Best Paper Award 各受賞。現在, IAPR TC11 (Reading Systems) Vice Chair, IAPR ICPR Standing Committee, Conferences & Meetings Committee 委員。情報処理学会, 人工知能学会, 電気学会, IEEE, ACM 等各会員。