

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[†] 中居 友弘[†] 黄瀬 浩一[†]

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

あらまし Geometric Hashing は広く知られた特徴点ベースの物体認識手法である．検索したい特徴点の配置をデータベースから全探索よりも高速に検索することができる．一方，我々は特徴点の局所的配置に基づく高精度，省メモリかつ高速な検索手法である“Locally Likely Arrangement Hashing (LLAH)”を提案している．いずれの手法も特徴点の照合を高速に行うという点で共通しているが，検索速度とメモリ使用量の使用量に関しては LLAH が優れている．本稿では LLAH を Geometric Hashing と比較することにより，どのような要因が LLAH の精度を保った高速化に寄与しているかを考察する．また，LLAH の検索性能は検索対象の撮影角度に依存するため，撮影角度と LLAH の検索性能の関係についても考察する．

キーワード Geometric Hashing, LLAH, 認識精度, 計算量, メモリ使用量, 撮影角度

An Analysis of Image Retrieval Methods Based on Arrangements of Feature Points

— Comparison Between Geometric Hashing and LLAH —

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

[†] Graduate School of Engineering, Osaka Prefecture University
1-1 Gakuencho, Naka, Sakai, 599-8531 Japan

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

Abstract The geometric hashing (GH) is a well-known object recognition technique based on arrangement of feature points. Recently, we have proposed “locally likely arrangement hashing (LLAH)” which outperforms GH in both processing time and required amount of memory with little loss of retrieval accuracy. One of two objectives of the current paper is to reveal where the outstanding performance of LLAH comes from. The other objective is to discuss the robustness of LLAH against change of the picture angle which can spoil the performance.

Key words geometric hashing, LLAH, retrieval accuracy, computational cost, amount of memory, picture angle

1. はじめに

本稿では，データベースに登録されている多数の画像から，検索対象の物体が写った画像を検索する問題を考える．この問題では検索したい物体は画像で与えられ，物体の見た目の情報を取得することにより，データベース中の画像を検索する．検索処理の結果，利用者は物体の名前を言い当てたり，物体に関連付けられている情報を取り出ししたりすることができる．登録されている画像と寸分違わぬ画像でしか検索できないのであれば応用範囲が限られるため，撮影角度や撮像過程の違いなどを

許容する，ある程度柔軟な検索が望まれる．当然，可能な限り高速であるほうが使い易い．

このような画像検索問題に対して，128 次元特徴ベクトルの SIFT [1] など，高次元特徴量を用いて画像の局所パターンをインデキシングする手法が近年盛んに研究されている．これらの特徴量は輝度変化などに比較的頑健に設計されているため，特徴量を比較することで画像検索が可能になる．しかし，特徴抽出や検索に必要な計算量は小さくない上，高次元特徴量が効果的でない対象も存在する．例えば，テクスチャなどの繰り返しパターンや文書に対しては同じ特徴量が多数計算されるため，

十分な識別性能が得られない。

そこで本稿では特徴抽出処理が高速に行え、様々な対象に適用可能な、特徴点の位置情報のみから検索する方式について考える。この方式は前述の高次元特徴量を用いる方式の対極に位置し、特徴点の座標以外の情報を利用しない。そのため、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ず、周囲の点の配置を用いる必要が生じる。ところが、質問画像を回転したり斜めから見れば幾何歪みが生じるため、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はないという問題が生じる。すなわち、この問題では幾何変換の影響を排除しつつ、いかに正確に、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) [2], [3] が広く知られている。GH は従来法の計算時間を大幅に短縮したが、それでも実用的な計算時間やメモリ使用量を実現しているとは言い難い。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著しく抑制する。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も可能になる [7]。

本報告では、このような LLAH の優れた性質が、どのような処理に起因するのかを LLAH と GH を比較することによって明らかにする。また、LLAH の処理は検索対象の撮影角度に依存するため、撮影角度と検索精度の関係についても考察する。

2. Geometric Hashing と LLAH

GH と LLAH の登録および検索処理を概説する。

2.1 Geometric Hashing [2], [3]

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。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、参考にしていただきたい。

まず、登録する画像から特徴点を抽出する。次に特徴点から 2 点選び、図 1 のように 2 本の基底ベクトルを作成する。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) を基底にしている。そして、残りの点 P_3, P_4, P_5 を相似不変な座標系に射影する。相似不変な座標系はあらかじめ小領域に分割 (量子化) されており、登録する画像の番号 (Object ID) と基底の取り方の番号 (Basis ID) をその領域に関連付ける。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、全ての座標系について行い、さらに登録する全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。まず検索する画像から特徴点を抽出する。次に登録処理と同様に特徴点から 2 点選び、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。座標系は登録処理で量子化され、画像と基底の通し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基に投票する。上記の処理を全ての座標系について行い、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 の組み合わせが最大の票数を得ているので、画像 2 が検索結果となる。なお、全ての基底を用いる前に正解画像が発見されたと判断される場合は途中で処理を中止することがある。

2.2 LLAH [5], [6]

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させていただきたい。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 $m = 4$ 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。ここである点を基準に時計回りに 3 点を選び、 A, B, C とすると、 $\frac{AC}{AB}$ 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 $mC_3 = 4$ 個の相似不変量 (図 2 における a, b, c, d) を計算する。次に、これらの不変量を並べて $mC_3 = 4$ 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 $m = 4$ 種類のベクトル (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 1 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。この処理を全ての特徴点を注目点にして行い、さらに登録する全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しているため、GH よりも特徴点の変動や消失の影響を受け易い。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。このようにすることで登録時と検索時で同じ m 点を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 $mC_3 = 4$ 次元の特徴ベクトルを作成するとき、作成可能な $m = 4$ 種類のベクトル ($abcd, bcda, cdab, dabc$) を全て作成して用いる点である。そして、作成した特徴ベクトルからハッシュ値を計算

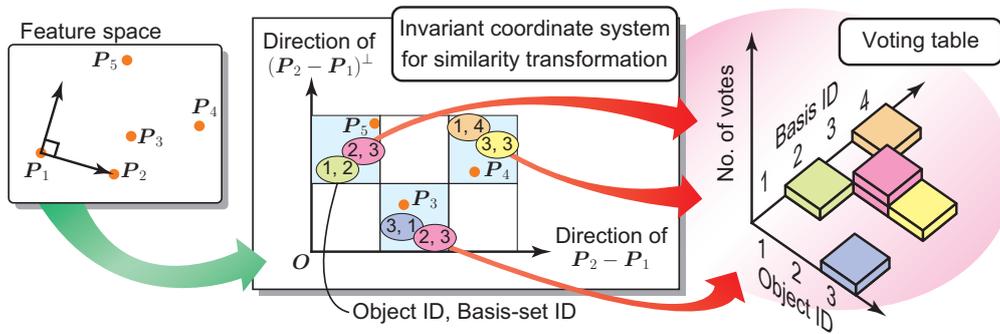


図 1: Geometric Hashing の検索処理 .

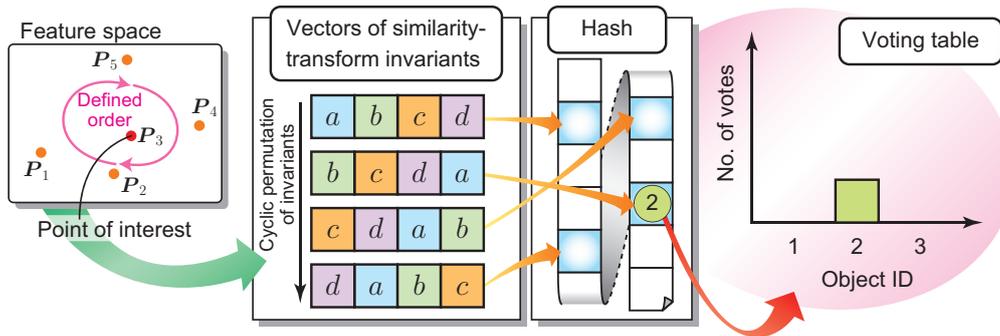


図 2: LLAH の検索処理 . 図中の不変量はそれぞれ, $a = \frac{\|P_4 - P_1\|}{\|P_5 - P_1\|}$, $b = \frac{\|P_2 - P_5\|}{\|P_4 - P_5\|}$, $c = \frac{\|P_1 - P_4\|}{\|P_2 - P_4\|}$, $d = \frac{\|P_5 - P_2\|}{\|P_1 - P_2\|}$ である .

し、ハッシュ表に登録されている画像番号に投票する^(注1)。この処理を全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。LLAH では GH より大きなハッシュを用いており、何も登録されていない要素が比較的多数存在する。LLAH は 1 位と 2 位の得票数に大きな差が生じた場合などに、GH 同様、途中で処理を中止することができる。なお、検索時にも登録時と同様に、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. LLAH の検索時間の改善要因

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。本節では、この理由を主要な 4 つの要因に分けて考察する。

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 3 つの観点から考察する。以下では画像 1 枚当たりの特徴点数を 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。

計算時間を大きくする 1 番目の要因は、特徴点数を基準とし

た計算量である。文献 [2] 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。メモリの使用量は登録時の計算時間と同じオーダーが必要である。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、計算量(メモリの使用量を含む)と検索精度は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。

計算時間を大きくする 2 番目の要因はハッシュの衝突である。GH においてハッシュに登録される情報(画像番号と基底番号の組)の総数は $N^{b+1}M$ である。それに対して、幾何変換に不変な座標系が量子化によって次元当たり k 個に分割されているとすると、ハッシュのビン(小領域)は k^b 個存在する。したがって、ハッシュのビン 1 つ当たりに登録されている情報の量(すなわち、衝突の量)は $N^{b+1}M/k^b$ である。つまり、ハッシュ値を一度計算する度に $N^{b+1}M/k^b$ 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、 $N = 100$, $M = 100$, $b = 3$, $k = 10$ の場合、投票回数は 10^7 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 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のビンとは異なるビンが引かれる可能性が増え、結果として検索性能が低下する。

3 番目の要因は投票テーブルの大きさである。GH は投票テー

(注1): 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と対応するかまで実時間で特定することができる。文献 [7] はこうして得られた特徴点の対応を拡張現実に応用している。

ブルに $N^b M$ 個のピン(注2)を持っているため、得票が最大のピンを発見するためには $O(N^b M)$ 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 の改良法と位置付ける。すなわち、GH に 4 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。

3.2.1 注目点の導入と近傍 m 点の計算

オリジナルの GH [2] は画像中の全特徴点を用いる。それに対して、ここでは用いる点数を減らすことによる高速化を行う。3.1 で述べたように、確率的な特徴点や基底の削減では認識精度の減少を免れない。そこで特徴量を 1 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 $b-1$ 点を、注目点と共に基底の作成に用い、残りの $m-b+1$ 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 $O(m^b NM)$ 、検索時の計算量は $O(m^b N)$ となる。この処理においては、登録時と検索時に同一の特徴点が求められることが重要となる。当然画像を斜めから撮影した場合には同一の特徴点が求められない場合が考えられ、検索性能が低下する可能性がある。この問題については 3.2.4 と 4. で議論する。

3.2.2 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 1 番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重視することにより、検索性能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を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。

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を得られる。ただし、「時計回り」の順序を利用して、「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。

順序を導入した場合の具体的な処理を述べると、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。並べた近傍点から順序を保持したまま $b+1$ 点選択し、幾何学的不変量を計算する。 m 点から $b+1$ 点を選択する組み合わせは ${}_m C_{b+1}$ 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 点全てを注目点として行うので、不変量は合計 ${}_m C_{b+1} N$ 個になる。したがって、順序を導入した結果、登録時の計算量は $O({}_m C_{b+1} NM)$ となる。また、検索時の計算量は $O({}_m C_{b+1} mN)$ となる。これは画像中の N 点全てについて ${}_m C_{b+1}$ 個の不変量を m 通りの起点で計算するためである。

以下では比較のために、GH が幾つの不変量を算出していることに相当するのかを求める。GH の場合、幾何変換に不変な座標系に特徴点を射影することで、点の座標は不変量になる。座標は 2 次元なので、各点から不変量が 2 個ずつ計算される。すなわち、計算される不変量の数は

$$(\text{基底の数}) \times (\text{射影される特徴点数}) \times 2$$

となる。この値は検索に要する計算量に等しく、オリジナルの GH では $O(N^b) \times (N-1) \times 2 = O(N^{b+1})$ であり、3.2.1 で示した近傍 m 点のみを計算に用いるものでは $O(Nm^{b-1}) \times (m-b+1) \times 2 = O(m^b N)$ である。このことから、GH や LLAH の計算量は算出する不変量の数に比例していること、LLAH は順序を導入することで少ない不変量での検索が可能になり、計算量の削減に成功したことがわかる。

3.2.3 特徴量の組み合わせによる高次元特徴ベクトルの導入

次にハッシュの衝突を回避して、3.1 で述べた 2 番目の問題点を解決する。GH においてハッシュの衝突が多い理由は明白で、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、単に大きなハッシュ表を用いれば衝突の量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となる。

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「順序を持った」不変量を ${}_m C_{b+1}$ 次元ベクトルとみなし、各次元を k 個に離散化することにより、最大で $k^{{}_m C_{b+1}}$ 通りもの情報が識別可能な巨大な特徴量空間が作られる。これにより、文献 [5] の LLAH ではハッシュの衝突がほとんどないため、投票に要する計算量はほぼ $O(1)$ 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。文献 [5], [6] の 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ようにしているため、正しく検索するためには ${}_m C_{b+1}$ 個の不変量が全て一致しなければならない。このように全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 票当たりの信頼度が非常に高い。

3.2.4 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個々の不変量よりも識別能力に優れる反面、1 つの不変量の算出ミスも許容しないため、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、認識精度が低下する恐れがある。そこで LLAH では m より大きな値 n を用いて、注目点あたり ${}_n C_m$ 個の特徴ベクトルを作成する。すなわち、注目点の近傍 n 点から m 点を選ぶ全ての組み合わせを求め、各 m 点について特徴ベクトルを作成する。この操作により、特徴点の出現、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 の計算量を求めてみると、登録時の計算量は $O({}_n C_m \cdot {}_m C_{b+1} NM)$ 、検索時の最悪計算量は $O({}_n C_m \cdot {}_m C_{b+1} mN)$ となる。

(注2): ピンの数 $N^b M$ は画像の枚数 M と基底の数 N^b の積である。

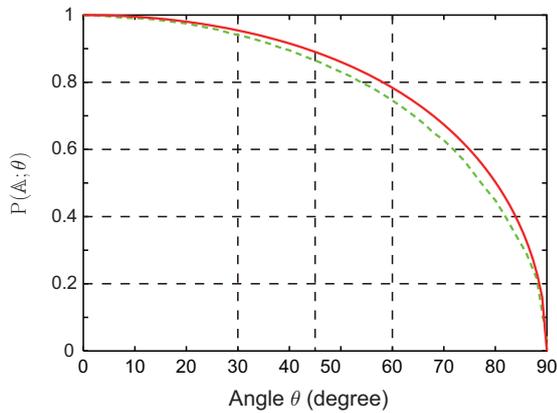


図 3: 確率 $P(A; \theta)$ の理論値 (実線) とシミュレーション (点線)

4. 撮影角度と LLAH の検索性能

LLAH は登録時と検索時で同一の m 点が選択されることを前提とする。ノイズによる特徴点の出現や消失を考えなければ、正面から撮影する限り、この前提は満たされる。しかし、斜めから撮影すれば、必ずしも同一の m 点が選択されるわけではない。このような影響を回避するため、近傍 n 点 ($n \geq m$) を選び、そのうち $n - m$ 点までの不一致を許容する工夫がなされている。本節ではこの「近傍 n 点から m 点を選ぶ」という工夫により、画像の傾きをどの程度許容するのかについて考察する。なお、考察においてはオクルージョンの影響を無視するため、平面上の物体を対象とする。

4.1 近傍 n 点に同一の特徴点が含まれる確率

以下では、登録時と検索時で同一の特徴ベクトルが計算できる確率を求める。「同一の特徴ベクトルが計算できる」という条件は、特徴点の位置ずれを考慮しなければ、「注目点の近傍 n 点中 m 点以上が同一の特徴点である」ということに等しいため、ここではまず、注目点の近傍 n 点に登録時と検索時で同一の特徴点が 1 点含まれる確率を、画像の撮影角度 θ の関数として導く。 θ は検索対象を正面から見たときが $\theta = 0$ になるように定義する^(注3)。検索対象が傾くことによって、検索対象はアフィン変換の歪みを被るものとする。特徴点は検索対象上に等確率で存在する (一様分布) と仮定し、検索対象の傾き以外の要因で特徴点の出現と消失は起こらないものとする。

「登録時と検索時で同一の特徴点 1 点が注目点の近傍 n 点に含まれる」という確率事象を A で表すと、求める確率は

$$P(A; \theta) = \frac{2}{\pi} \cos^{-1} \frac{\cos \theta - 1}{\cos \theta + 1} \quad (1)$$

で与えられる。式 (1) の導出は付録 A に記す。式 (1) をプロットしたものが図 3 の実線である。図 3 の点線は一様乱数を発生させたシミュレーションの結果で、詳細は 4.3 で述べる。

4.2 近傍 n 点から同一の m 次元特徴ベクトルが計算できる確率

「注目点の近傍 n 点中 m 点以上が登録時と検索時で同一の特徴点である」という確率事象を C とする。前述のように、こ

(注3): 画像の撮影角度 θ の定義が文献 [5], [6] と異なることに注意する。

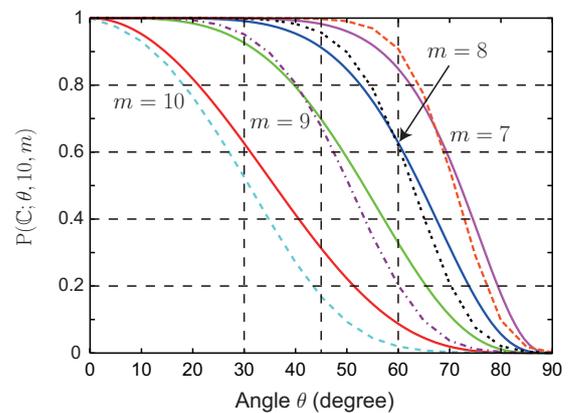


図 4: 確率 $P(C; \theta, 10, m)$ の理論値 (実線) とシミュレーション (点線)

れは「登録時と検索時で同一の特徴ベクトルが計算できる」とほぼ同義である。そこで、 C が起こる確率を撮影角度 θ の関数として導く。

C が起こる確率を求めるために、 C の定義の「 m 点以上」を「 m 点」とした確率事象を B とおき、 B が起こる確率を先に求める。 B が起こる確率を $P(B; \theta, n, m)$ とおくと、これは二項分布を用いて、

$$P(B; \theta, n, m) = {}_n C_m \{P(A; \theta)\}^m \{1 - P(A; \theta)\}^{n-m} \quad (2)$$

と書ける。したがって、これを用いれば、 C が起こる確率は

$$P(C; \theta, n, m) = \sum_{i=m}^n P(B; \theta, n, i) \quad (3)$$

となる。これが同一の特徴ベクトルが計算できる確率の理論値である。

4.3 理論値とシミュレーションの比較

式 (1) の $P(A; \theta)$ と式 (3) の $P(C; \theta, n, m)$ の妥当性をそれぞれ検証するために 2 種類のシミュレーションを行った。

4.3.1 $P(C; \theta, n, m)$

$P(C; \theta, n, m)$ を検証する。100 ページの PDF 文書から特徴点の配置を抽出し、計算機上で θ 度回転させた。回転によって被る歪みは $P(C; \theta, n, m)$ の導出に用いた条件とは異なり、射影歪みである。 θ 度回転させた後、特徴点毎に近傍 n 点から m 点を選び、 $\theta = 0$ のときと同一の m 点を選ばれるかどうかを調べた。各点につき、1 組でも同一の m 点を選ばれば成功、そうでなければ失敗である。これを 100 文書の全ての特徴点について調べ、成功率を算出した。この成功率の理論値に相当するのが $P(C; \theta, n, m)$ である。

$P(C; \theta, n, m)$ の理論値とシミュレーション結果を図 4 に示す。パラメータは、 $n = 10$ とし、 m は $m = 10, 9, 8, 7$ をそれぞれ用いた。図から、理論値とシミュレーション結果は完全に一致しないものの、ある程度の類似性が確認できた。なお、回転による歪みをアフィン歪みとした場合も同様の結果を得た。

4.3.2 $P(A; \theta)$

$P(C; \theta, 10, m)$ の理論値とシミュレーション結果にずれが生じたため、 $P(A; \theta)$ を検証する。 $P(A; \theta)$ の導出に用いたのと同

条件（一様分布）で特徴点を 10,000,000 個生成した．そして， $P(C; \theta, 1, 1) = P(A; \theta)$ の関係を利用して，4.3.1 と同様のシミュレーションを行った．ただし，回転によって被る歪みはアフィン歪みとした．結果を図 3 の点線に示す．理論値である図 3 の実線と比較すると，傾きの傾向はほぼ一致しているものの，若干のずれが生じている．このずれの原因を解明することは今後の課題である．

5. むすび

本報告では，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)”の検索性能と計算時間について考察した．まず，特徴点に基づく画像検索手法として広く知られている Geometric Hashing と LLAH を比較することで，LLAH が高い識別能力を保ったまま高速性を実現できている要因が何であるのかを考察した．そして，検索対象の撮影角度と LLAH の検索精度の関係についても，特徴ベクトルが正しく計算できる確率の理論式を導出することで考察した．導出した確率を検証した結果，理論値とシミュレーションはほぼ一致するものの，若干のずれが生じた．このずれの原因を解明することは今後の課題である．

文 献

- [1] D. G. Lowe, “Distinctive image features from scale-invariant keypoints,” *International Journal of Computer Vision*, vol.60, no.2, pp.91-110, 2004.
- [2] Y. Lamdan and H. J. Wolfson, Geometric hashing: a general and efficient model-based recognition scheme, *Proc. ICCV*, pp.238-249, 1988.
- [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.

付 録

A. 式 (1) の導出

「登録時と検索時で同一の特徴点が注目点の近傍 n 点に含まれる」という確率事象を A とおき， A が起こる確率 $P(A; \theta)$ を導出する．そのために，以下では図 A.1 の円と楕円が重なる面積である $S(\theta)$ を導く．

まず，対象を正面から撮影する場合を考える．注目点 P の近傍 n 点を包含する最小の円の半径を r とおく． P を原点とする座標系を考えると，この円は

$$x^2 + y^2 = r^2 \quad (A.1)$$

で与えられる．この円は図 A.1 の円である．

次に傾き θ 度で対象を撮影したときについて考える．対象の

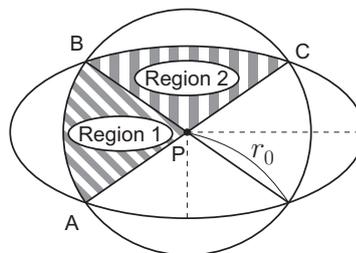


図 A.1: 求める面積

単位面積当たりの特徴点数を一定と仮定すると，対象が傾いた分だけ見かけの特徴点の密度が上昇する．そこで P の近傍 n 点を包含する最小の円の半径を r' とおく ($r' \leq r$)．そして，傾いた対象を正面向きになるように回転すると，この半径 r' の円は図 A.1 の楕円になり，

$$x^2 \cos \theta + \frac{y^2}{\cos \theta} = r'^2 \quad (A.2)$$

で与えられる．ここで式 (A.1) の円と式 (A.2) の楕円の面積が等しいことに注意する．この理由は，特徴点は対象上で一様分布していると仮定しているため，ある領域に含まれる特徴点数の期待値はその領域の面積で決まるからである．このことから，同じ数の特徴点を含む面積の期待値は等しくなる．

$S(\theta)$ を求めるために，図 A.1 の領域 1 (Region 1) の面積 $S_1(\theta)$ と領域 2 (Region 2) の面積 $S_2(\theta)$ を求めることにする．これにより， $S(\theta)$ は

$$S(\theta) = 2 \{S_1(\theta) + S_2(\theta)\} \quad (A.3)$$

で求められる．

まず領域 1 の面積 $S_1(\theta)$ を求める．ここで A, B の座標はそれぞれ $(-r\sqrt{\frac{1}{1+\cos\theta}}, -r\sqrt{\frac{\cos\theta}{1+\cos\theta}})$, $(-r\sqrt{\frac{1}{1+\cos\theta}}, r\sqrt{\frac{\cos\theta}{1+\cos\theta}})$ であるため， $\angle APB$ が成す角度を ϕ とすれば，

$$\phi = \cos^{-1} \frac{1 - \cos \theta}{1 + \cos \theta} \quad (A.4)$$

である．半径 r の円の面積は πr^2 で与えられるので，

$$S_1(\theta) = \pi r^2 \frac{\phi}{2\pi} = \frac{r^2}{2} \cos^{-1} \frac{1 - \cos \theta}{1 + \cos \theta} \quad (A.5)$$

が得られる．

次に領域 2 の面積 $S_2(\theta)$ を求める．図形の面積は横方向に a 倍，縦方向に $\frac{1}{a}$ 倍しても変化しない．そこで領域 2 を x 軸方向に $\sqrt{\cos \theta}$ 倍， y 軸方向に $\frac{1}{\sqrt{\cos \theta}}$ 倍すると，これは領域 1 を回転したものに等しい．つまり， $S_2(\theta) = S_1(\theta)$ であるため， $S(\theta) = 4S_1(\theta)$ となる．したがって，

$$S(\theta) = 2r^2 \cos^{-1} \frac{\cos \theta - 1}{\cos \theta + 1} \quad (A.6)$$

が導かれる．

求める確率 $P(A; \theta)$ は，明らかに $P(A; 0) = 1$ であるため，正規化を施す．結局， $S(0) = \pi r^2$ であるため，

$$P(A; \theta) \equiv \frac{S(\theta)}{S(0)} = \frac{2}{\pi} \cos^{-1} \frac{\cos \theta - 1}{\cos \theta + 1} \quad (A.7)$$

が得られる．