

特徴点の局所的配置に基づく物体認識の精度・速度に関する一考察

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 我々は既に、特徴点の局所的配置に基づく高精度かつ高速な文書画像検索方法 (LLAH) を提案している。これは、広く知られた特徴点ベースの物体認識手法である Geometric Hashing を精度・速度の両面において著しく改善する手法である。本報告ではこの手法を Geometric Hashing と比較することにより、どのような要因が高精度化と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, 計算時間, 撮影角度

On Accuracy and Speed of Object Recognition Based on Local Arrangements of Feature Points

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

[†] Graduate School of Engineering, Osaka Prefecture University

1-1 Gakuencho, Naka, Sakai, Osaka, 599-8531 Japan

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

Abstract The geometric hashing is a well-known object recognition technique based on the arrangements of feature points. We have proposed “locally likely arrangement hashing (LLAH)” which outperforms the geometric hashing in both retrieval accuracy and processing time. In this report, by comparing both methods, we consider the major factors which bring the improvement. We also consider the relationship between a picture angle and the accuracy of the LLAH because the accuracy of the LLAH depends on the picture angle.

Key words geometric hashing, locally likely arrangement hashing, retrieval accuracy, processing time, picture angle

1. ま え が き

本報告では、文書など、平面状の物体から特徴点 (例えば英単語の重心) 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持っていないため、検索対象の特徴点、登録された特徴点のどれと対応する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が必要となる。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

影歪みのような幾何変換がかけられているため、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 [1], [2] が広く知られている。Geometric Hashing は従来法の計算時間を大幅に短縮するものであったが、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。Geometric Hashing 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ものの、一般に計算時間やメモリ使用量と、検索精度はトレード

オフの関係にあり、実用的な精度を保持しつつ計算時間を削減することは困難であった。

それに対して、我々は Geometric Hashing が持つ計算時間と検索精度のジレンマを解決した手法である “Locally Likely Arrangement Hashing (LLAH)” を提案している [3], [4]。この手法は非常に高速であり、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 20,000 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である [5]。

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 Geometric Hashing 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 LLAH の検索精度の関係についても考察する。

2. Geometric Hashing と LLAH

2.1 Geometric Hashing [1], [2]

2.1.1 登録

Geometric Hashing 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、参考にさせていただきたい。まず、登録したい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 を相似変換不変な座標系に射影する。相似変換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 (Image #) と基底の取り方の番号 (Basis #) 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

図 1 に示す例を使って検索処理について説明する。まず検索したい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変換不変な座標系に射影する。相似変換不変な座標系は量子化されているので、点が存在する領域に登録されて

いる、画像と基底の通し番号を基に投票し、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。上記の処理を全ての座標系について行い、得票数が最大の画像を検索する。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 [3], [4]

2.2.1 登録

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。ここで連続した 3 点^(注1)を 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。

上記の説明では省略したが、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、Geometric Hashing 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。

(注1): この場合は m が小さいためにたまたま連続した 3 点になっているが、 m が大きい場合は連続しない。連続する、しないに関わらず、常に同じ 3 点が取れるということが重要である。点の選び方の詳細については 3.2.2 で述べる。

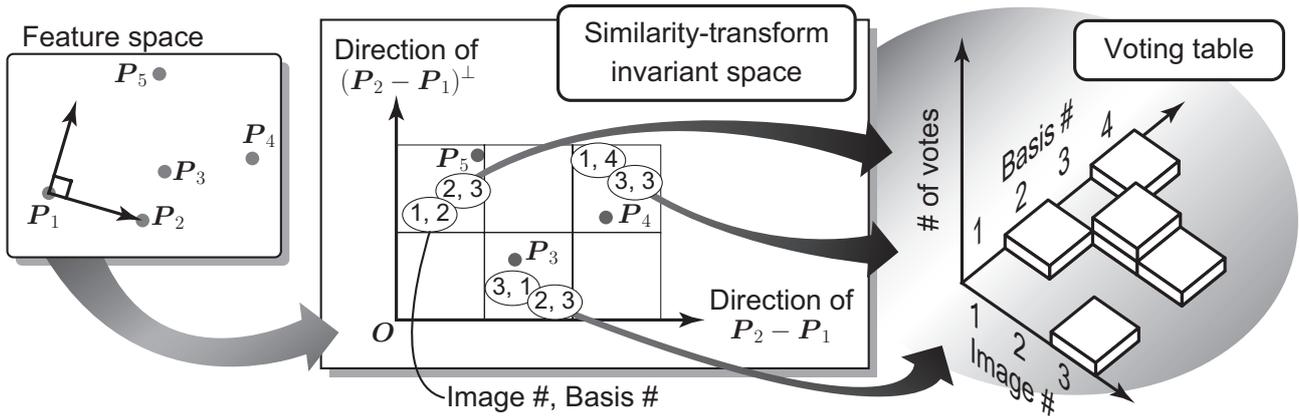


図 1: Geometric Hashing の検索処理 .

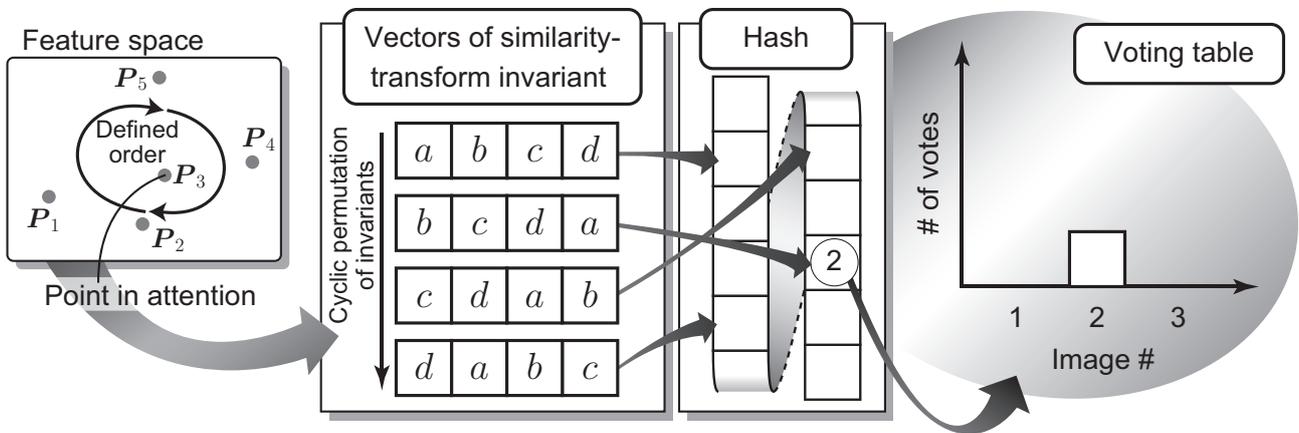


図 2: LLAH の検索処理 . 図中の不変量はそれぞれ , $a = \frac{\|P_4 - P_1\|}{\|P_5 - P_1\|}$, $b = \frac{\|P_2 - P_5\|}{\|P_4 - P_5\|}$, $c = \frac{\|P_1 - P_4\|}{\|P_2 - P_4\|}$, $d = \frac{\|P_5 - P_2\|}{\|P_1 - P_2\|}$ である .

2.2.2 検 索

図 2 に示す例を使って LLAH の検索処理について説明する . 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し , ハッシュ表に登録されている画像番号に投票する^(注2) . この処理を , 全ての特徴点を注目点にして行った結果 , 投票度数が最大の画像が検索結果となる . 図 2 の例では , 画像 2 が登録され

(注2): 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 [5] 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 .

ている要素が 1 度 , 何も登録されていない要素が 3 度引かれ , 画像 2 が検索結果となる . これは比較的高次元な特徴ベクトルを用いてハッシュを引いているためである . 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

LLAH の検索時間とメモリ使用量は Geometric Hashing と比べると飛躍的に改善されている . 本節では , LLAH において検索時間とメモリ使用量が削減できた要因について考察する .

3.1 Geometric Hashing の問題点

LLAH について述べる前に、まず Geometric Hashing について考える。Geometric Hashing は計算時間とメモリ使用量の制約が大きく、使用できる状況が限られる。この原因を二つの観点から考察する。以下では画像 1 枚当たりの点数を N 、画像の枚数を M とする。また、Geometric Hashing 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、相似変換では $b = 2$ 、アフィン変換では $b = 3$ 、射影変換では $b = 4$ である。

計算時間を大きくする一つ目の要因は、特徴点数を基準とした、通常よく用いられる意味の計算量である。文献 [1] に記載されているオリジナルの Geometric Hashing 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 [2] 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り、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もう一つの要因はハッシュの衝突である。Geometric Hashing 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、ハッシュのビン一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り、検索時間が膨大になる。衝突を減少させるには当然 N と 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方法も考えられる。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。

3.2 Geometric Hashing の改良法としての LLAH

Geometric Hashing と LLAH の関連性や相異点を明らかにするため、LLAH を Geometric Hashing の改良

法と位置付ける。すなわち、Geometric Hashing 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 Geometric Hashing [1] は画像中に存在する特徴点を全て用いる。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られない。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。これにより、計算量は大幅に削減され、登録時の計算量は $O(m^{b+1}NM)$ 、検索時の最悪計算量は $O(m^{b+1}N)$ となる。この処理においては、登録時と検索時で同一の m 点が正しく求められるかどうか重要となる。当然画像を斜めから撮影した場合には同一の m 点が正しく求められない場合も考えられるため、検索性能が制限される。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」基底の削減

基底を削減し、3.1 で述べた一つ目の問題点を解決する。既に何度も述べているように、確率的ではない基底の削減方法が必要であるので、LLAH では常に同じ基底が選択されるという再現性を重視して基底を削減する。そのために、次の法則を利用する。

ある特徴点に注目したとき、その近傍の点を時計回りに並べたときの順序は幾何変換に不変である^(注3)。

この法則から「時計回り」の順序を基底の計算に利用することを考えると、登録時と検索時で一致した「順序の起点」を定める必要があることに気付く。しかし、近傍点は m 点しか用いないため、登録時には起点を適当に選択し、検索時には全ての近傍点を起点として試してみても計算量は m 倍にしかならない。したがって、起点をどう選ぶのかはそれほど重要な問題ではない。

以下、このように順序を導入することで、どの程度計算量が削減できるのかについて考察する。LLAH では、 mC_{b+1} 個「相当」の基底を計算している。ここで「相当」と述べる理由は、実際には LLAH では基底を計算せず、その代わりに mC_{b+1} 個の幾何学的不変量を計算

(注3): 画像を裏返せば「時計回り」という順序は保存されないが、その場合は検索時に「反時計回り」も試せばよく、計算量は 2 倍になるものの、計算量のオーダーは変わらない。

しているからである。幾何学的不変量は、相似不変量の場合は、同一平面上に3点あれば計算できる。同様にアフィン不変量は4点、射影不変量は5点あれば計算できる。つまり、幾何学的不変量は同一平面上の $b+1$ 点あれば計算できる。このことを Geometric Hashing 流に解釈すると、同一平面上の $b+1$ 点のうち、 b 点を用いて幾何変換に不変な座標系を作成し、残りの1点を作成した座標系に射影している。LLAHではこのような操作を mC_{b+1} 回行っているので、幾何変換不変な座標系を mC_{b+1} 回計算していることに相当する。

それでは LLAH のように mC_{b+1} 個の不変量を計算したときの計算量を求めてみると、登録時の計算量は $O(mC_{b+1}NM)$ となる。これは、画像中の N 点全てについて mC_{b+1} 個の不変量を計算するためである。また、検索時の最悪計算量は $O(mC_{b+1}mN)$ となる。これは、画像中の N 点全てについて mC_{b+1} 個の不変量を m 通りの起点で計算するためである。このような確率的でない「基底の削減」を行うことで、認識精度を低下させることなく計算量を大幅に減少させることが可能となる。

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で述べた二つ目の問題点を解決する。Geometric Hashing においてハッシュの衝突が多い理由は明白で、幾何変換に不変な座標系の次元が低いからである。3.1で述べたように、 $N^{b+1}M$ 個^(注4)もの登録すべき情報を k^b 個しかないハッシュのピンに押し込めているために多量の衝突が起こる。そこで、ハッシュ表の大きさを大きくすることにより、この状況を解決することを考える。

ハッシュ表の大きさを大きくするためには、特徴量が十分な判別能力を持つ必要がある。ここで特徴量とは、Geometric Hashing においては幾何変換に不変な座標系での特徴点の位置(座標)を k^b 個に量子化した値であり、LLAH においては幾何学的不変量を k 個に量子化した値である。

LLAH では、高い判別能力を実現するために複数の特徴量を組み合わせている。すなわち、 mC_{b+1} 個の「順序を持った」不変量を mC_{b+1} 次元ベクトルとみなし、各次元を k 個に分割することにより、 $k^{\{mC_{b+1}\}}$ の特徴

(注4): 3.2.2 において基底を削減したため、登録すべき情報の数は $mC_{b+1}NM$ に減少している。しかし、それでもハッシュ表の大きさに対して著しく多い。

量空間を利用可能にしている^{(注5)(注6)}。このような高次元の特徴量空間を用いることで、文献[3]のアフィン不変量を用いる場合^(注7)、1,000ページの文書画像を登録したときに2.95%、10,000ページで19.7%しか値を持たない非常に粗なハッシュが構築されている。ハッシュの衝突がほとんどないため、投票の計算量はほぼ $O(1)$ である。

このようにハッシュが粗であるということは衝突によって計算時間が増加するのを防ぐ以外に、検索の頑健性にも貢献している。前述のように、複数の特徴量を組み合わせることで判別能力は向上するが、正しく検索するためには mC_{b+1} 個の特徴量が全く一致しなければならない。このような確率は個々の特徴量が一致する確率よりも大幅に低くなるため、誤投票によって認識精度を低下させる恐れがある。しかし、ハッシュが粗であれば、特徴ベクトルが誤ることによる誤投票を防ぐ効果が期待できる。

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で述べたように、特徴量を組み合わせて作成した高次元ベクトルは個々の特徴量よりもノイズに弱い。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 について、注目点の近傍 n 点から m 点を選ぶ可能な組み合わせを全て試し、特徴ベクトルを作成する。すなわち、登録、検索処理において特徴ベクトルを nC_m 倍作成する。この操作により、特徴点の出現、消失によ

(注5): 実際に用いるハッシュの大きさはメモリ容量の制約を受けるため、 $k^{\{mC_{b+1}\}}$ よりも小さくなる場合がある。

(注6): 以下に述べる理由で、この高次元の特徴量空間の各次元は独立ではないと考えられる。2次元空間で m 点を持つ自由度は $2m$ である。画像中に特徴点が m 点存在する場合の Geometric Hashing を考えてみると、 b 点を基底に使うので、座標を記述するために使用できるのは $m-b$ 点であり、その自由度は $2(m-b)$ である。LLAH は幾何学的不変量を用いるものの、3.2.2で述べたように Geometric Hashing のように基底を使用しているとも解釈できる。つまり、LLAH においても座標を記述するために使用できるのは $m-b$ 点であり、自由度は $2(m-b)$ である。したがって、特徴点の配置から得られる本質的な特徴量空間の大きさは $k^{\{2(m-b)\}}$ 程度であると考えられる。ここでほとんどの m と b について、 mC_{b+1} は $2(m-b)$ よりも大きい。したがって、 mC_{b+1} 次元ベクトルから作成される特徴量空間は、本質的な特徴量空間よりも大きく、 mC_{b+1} 次元ベクトルから作成した特徴量空間は縮退していると考えられる。

(注7): ハッシュ表の大きさを約 2^{27} とし、パラメータとして $n=7$, $m=6$, $k=15$ を用いた。

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 n と m の選び方によってはそれほど大きくない。例えば、 $n = 8$ 、 $m = 7$ を用いれば ${}_8C_7 = 8$ であり、 $n = 10$ 、 $m = 8$ を用いれば ${}_{10}C_8 = 45$ である。計算量を求めてみると、登録時の計算量は $O({}_n C_m {}_m C_{b+1} NM)$ となる。また、検索時の最悪計算量は $O({}_n C_m {}_m C_{b+1} mN)$ となる。

4. 撮影角度と LLAH の検索性能

LLAH は登録時と検索時に同一の m 点を選択されることを前提としている。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、斜めから撮影すると、注目点からの距離が大きくなる特徴点と小さくなる特徴点が存在するため、必ずしも同一の m 点を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に、LLAH では近傍 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。

以下では、登録時と検索時に同一の特徴ベクトルが計算できる確率を求める。そのために、同一の特徴点が目点の近傍 n 点に含まれる確率を最初に求める。

4.1 近傍 n 点に同一の特徴点が含まれる確率

注目点の近傍 n 点の中に登録時と検索時に同一の特徴点が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（紙面など）を正面から見たときが $\theta = 0$ になるように定義する^(注8)。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による特徴点の出現と消失はないものとする。

「登録時と検索時に同一の特徴点が目点の近傍 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) をプロットしたものが図 3 である。式 (1) の導出は付録 A に記す。

(注8)：画像の撮影角度 θ の定義が文献 [3], [4] と異なることに注意する。

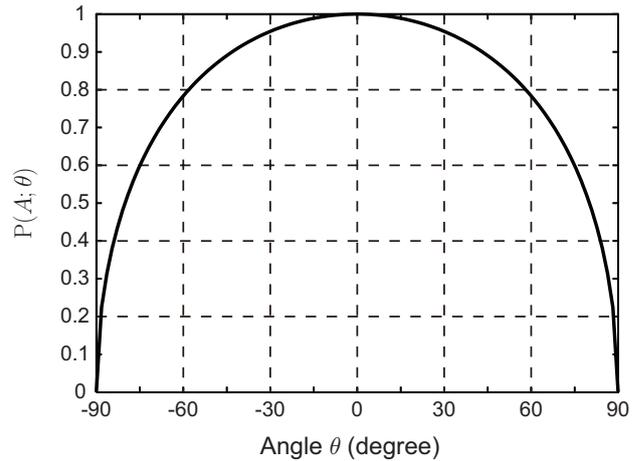


図 3: 注目点の近傍 n 点の中に登録時と検索時に同一の特徴点が含まれる確率 $P(A; \theta)$

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に同一の特徴点である」という確率事象を C とする。LLAH では、 C が起これば、登録時と検索時に同一の m 次元特徴ベクトルが計算できる。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、まず 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、

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

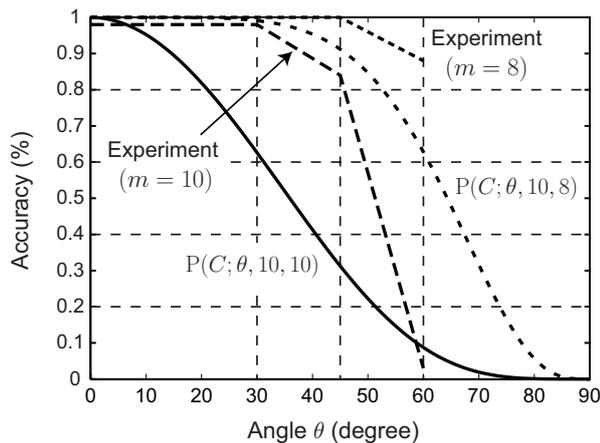
と書ける。これを用いれば、 C が起こる確率は

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

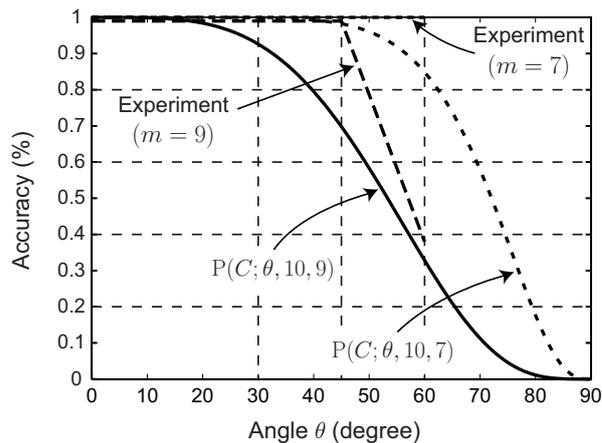
となる。

4.3 近傍点の数 n , m と画像の傾きに対するロバスト性

4.2 で求めた $P(C; \theta, n, m)$ の妥当性を検証する。本来であれば、特徴ベクトルが正しく計算できた割合を実験で求めて $P(C; \theta, n, m)$ と比較するべきであるが、今回はこれができなかったため、代わりに LLAH の認識精度と比較することにする。LLAH の認識精度とし



(a) $m = 10, m = 8$ の場合 .



(b) $m = 9, m = 7$ の場合 .

図 4: 撮影角度 θ で特徴ベクトルが正しく計算できる確率 $P(C; \theta, 10, m)$ と文献 [3] の図 10 の認識率

ては、文献 [3] の図 10 に示したアフィン不変量の検索性能を用いる .

撮影角度 θ で特徴ベクトルが正しく計算できる確率 $P(C; \theta, 10, m)$ と文献 [3] の図 10 の認識率を図 4 に示す . 実験における撮影角度は 0 度 , 30 度 , 45 度 , 60 度であり , パラメータは $n = 10, m = 10, 9, 8, 7$ を用いた . LLAH の検索性能と $P(C; \theta, n, m)$ を簡単には比較できないが , 図から両者の相関がうかがえる . LLAH の実験結果は全ての m , 全ての撮影角度 θ において特徴ベクトルが正しく計算できる確率 $P(C; \theta, n, m)$ を上回っている . これは投票により , 全ての特徴ベクトルが求まらなくても認識できるという効果が検索性能を押し上げていると考えられる . また , 撮影角度が大きくなることによる特徴抽出の失敗などが考えられるため , 撮影角度が大きくなるほど LLAH の検索性能と $P(C; \theta, n, m)$ の相関が弱くなると考えられる .

5. ま と め

本報告では , 我々が既に提案している , 特徴点の局所的配置に基づく高精度かつ高速な文書画像検索方法 (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 .

参 考 文 献

- [1] Y. Lamdan and H. J. Wolfson, Geometric hashing: a general and efficient model-based recognition scheme, Proc. International Conference on Computer Vision, pp.238–249, 1988.
- [2] 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 .
- [3] 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 .
- [4] 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, Sept., 2006 . To appear .
- [5] 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,” 画像の認識・理解シンポジウム (MIRU2006) 論文集, pp.1379–1380, July, 2006 .

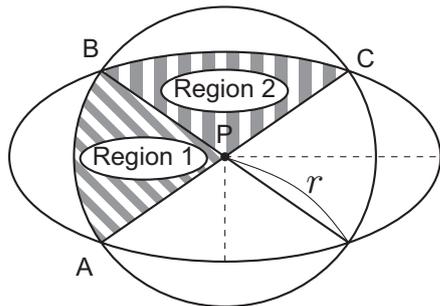


図 A.1: 求める面積

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 (以下、紙面) 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

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

で与えられる。この円は図 A.1 の円である。

次に傾き θ 度で紙面を撮影したときについて考える。紙面の単位面積当たりの特徴点数を一定と仮定すると、紙面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' とおくと、 $r' \leq r$ となる。そして、傾いた紙面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、

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

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、特徴点は紙面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

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

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 $(-r\sqrt{\frac{\cos \theta}{1+\cos \theta}}, -r\sqrt{\frac{1}{1+\cos \theta}})$ 、 $(-r\sqrt{\frac{\cos \theta}{1+\cos \theta}}, r\sqrt{\frac{1}{1+\cos \theta}})$ であるため、 $\angle APB$ が成す角度を ϕ とすれば、

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

である。半径 r の円の面積は πr^2 で与えられるので、

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

が得られる。

次に領域 2 の面積 $S_2(\theta)$ を求める。図形の面積は横軸方向に $\frac{1}{a}$ 倍、縦方向に a 倍しても変化しない。そこで領域 2 を x 軸方向に $\frac{1}{\sqrt{\cos \theta}}$ 倍、 y 軸方向に $\sqrt{\cos \theta}$ 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、

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

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、

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

が得られる。