コンピュータビジョン 80－22
(1992. 11. 20)

Geometric Hashing による 画像マッチングのロバスト性向上について

川西康之* 出口光一郎 森下巖
東京大学工学部
東京都文京区本郷 7-3-1

Geometric Hashing とは、データの整理方法であるハッシュ法を画像認識に応用したものであり、予め用意したモデル画像を用いて観測画像を認識する Model-Based Matching のその認識過程の処理時間を短縮するのに有効な手段である。しかし、外乱の入った観測画像に対してのパフォーマンスはまだ充分に明らかにされていない。

今回、画像に点集合画像を用い、アフィン変換を施した観測画像について gaussian noise に対する認識結果の影響を解析した。それに基づいて新しい投票プロシージャを考案し、Geometric Hashing のロバスト性の向上に努めた。実験の結果、外乱により通常は認識できなかった観測画像を認識し、更にその認識の信頼性がこの新しい方法によって向上し得ることを確認した。

和文キーワード コンピュータ・ビジョン、マッチング、ジオメトリック・ハッシング、不変量、誤差解析、ロバスト性

An Improvement of Robustness of Geometric Hashing for Model-Based Matching

Yasuyuki Kawanishi* Koichiro Deguchi Iwao Morishita
Faculty of Engineering, University of Tokyo
7-3-1 Hongo, Bunkyo-ku, Tokyo 113 Japan

Geometric Hashing is a Hashing-based method for *Model-based Matching* in image recognition. In *Model-based Matching* , an observed image is compared with a number of models prepared for the recognition. *Geometric Hashing* is an efficient method to reduce the matching time, but, its performance for noisy observed images is still not clearly analyzed.

In this paper, we present a theoretical analysis of effects of noise on the *Geometric Hashing* for the case observed images are deformed with affine transformations. Based on this analysis, we propose a new voting procedure for the improvement of robustness of Geometric Hashing. Experimental results were also shown for point-set observed images with Gaussian noises.

英文 key words Computer Vision, Matching, Geometric Hashing, Invariant, Error Analysis, Robustness

1 はじめに

観測画像の認識において、予め用意された幾つかのモデル画像を観測画像の中から見つけ出す問題はModel-based Matchingと呼ばれている。

モデル画像を見つけ出すのにまず考えられる方法は、観測画像と各モデル画像から比較するに必要な特徴を抽出し各モデルにつき順々に観測画像と比較することを繰り返すというものである。しかしこの方法だとモデル画像のどの特徴が観測画像のどの特徴に対応するか、モデル画像を選択する度に整理する必要があるので認識方法としては効率が悪い。

Geometric HashingはHash Tableをモデル画像のデータベースに用いる。具体的に述べると、全てのモデル画像から特徴を不変量として計算し、それぞれHash Table上のその不変量に対する位置にそのモデル名と不変量を計算するのに用いた基底を保存する。次に観測画像が与えられると同じく不変量を計算し、Hash Table上の位置を割出し、Hash Tableを検索する。不変量はモデルと基底毎に異なるので、検索により得られたデータは観測画像とモデル画像のそれぞれの特徴の正しい対応を示しているはずである。

データベースは一度作っておけば良いので、この方法によって観測画像の認識時間を大幅に短縮できる。

このように、Geometric HashingはModel-based Matchingにおける画像認識の時間を短縮するのに有効な方法であるが、観測画像に誤差が加わった場合、それに基づく不変量もその影響を被るので、Table上の検索位置が実際の箇所よりずれる場合がある。もちろん、その場合、正しい認識が出来ない。

本論文では、観測画像に加わる誤差のGeometric Hashingに与える影響について解析を行ない、誤差に強い認識方法について考察を行なった。

2 Geometric Hashing

2.1 処理プロセス

Geometric Hashingは、大きく分けると次の2つのプロセスに分けられる。

Preprocessing Stage :

1. 各モデル画像から不変量を計算する。
2. 各モデル画像のデータを(不変量, モデル名, 不変量を計算するのに用いたパラメータ（以後、基底と呼ぶ）)と構成する。
3. 不変量をHash Table上の位置に変換し、データをHash Tableに展開する。
4. 全てのモデル画像、考えつく全ての基底について1.～3.を繰り返す。

例えば、後で述べる点画像の場合、3点を基準にした相対座標がアフィン変換における不変量なので、N点からなる画像の場合、基底の選び方はN×（N－1）×（N－2）通りある。従って、1モデル当たりN×（N－1）×（N－2）×（N－3）個のデータがHash Table上に保存される。

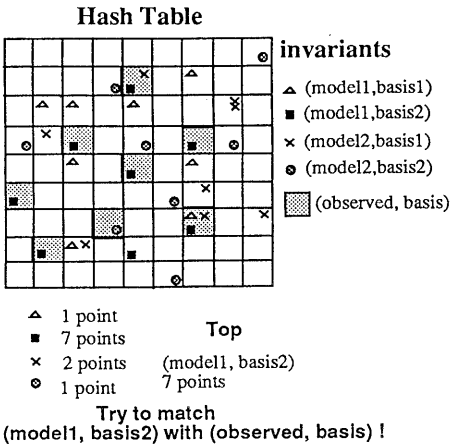

図 1: Hash Tableと Recognition Stage

Recognition Stage :

1. 観測画像について、ある1つの基底に基づいた不変量を計算する。
2. 不変量をHash Tableの位置に変換し、それぞれの位置においてTableを検索する。
3. 検索先にあったデータの内、最も沢山あった（モデル名, 基底）の組を認識対象の候補として選ぶ。具体的に言うと、（モデル名, 基底）の組について、組毎にデータの数のヒストグラムを作り、その中で

$$\frac{\text{検索で得たその組のデータの数}}{\text{（そのモデル画像が所有する点の数－基底の点の数）}}$$

が最も大きい組を次の処理にかける候補として選ぶ。この作業を投票(voting)と呼ぶ。

4. モデル画像と観測画像との実際のマッチングを行なう。

5. マッチングの結果、間違っていると示された場合、観測画像の基底を選び直して1.から繰り返す。

Recognitionにおいては最悪でもN×（N－1）×（N－2）回Recognition Stageを繰り返すだけで良い。順々に1つ1つモデルと画像のマッチングを行なう場合が最悪で$O(N^6)$だけマッチングを繰り返すことを考えると、Geometric Hashingは一度Hash Tableを構築してしまえば認識の処理時間の短縮が期待できる[1]。

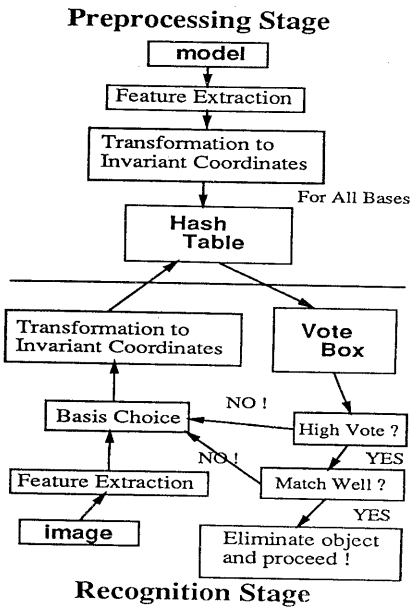

図 2: システム構成

2.2 点画像のアフィン変換での不変量

これより扱うモデル画像及び観測画像は座標を持つ点の集合とする。

観測画像とその元となったモデル画像との間には何らかの座標変換がなされている。この場合、その座標変換についての不変量をハッシュ関数に用いれば、Hash Table上で観測画像とモデル画像の占める位置は同じである。

以後、アフィン変換に対する不変量[3, 4]を用いる。理由としては、

- 全ての2次元平面内の変換を網羅している。
- カメラが対象平面に垂直に近い場合は透視変換の近似になる。
- 透視変換の不変量よりも解析などが簡単である。

などが挙げられる。

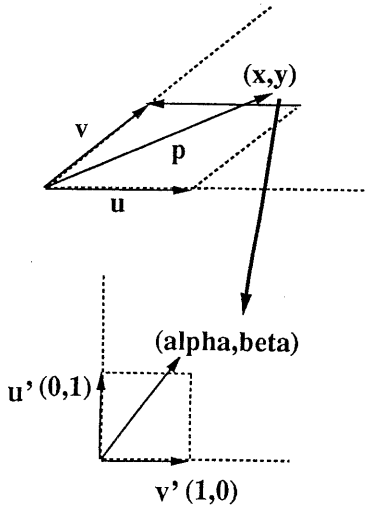

図 3: アフィン変換における不変量

アフィン変換についての不変量は3点を基底として用い、その内の1点を原点ベクトルとした2ベクトルu、v及び基底以外の1点のベクトルpについて、

$$\alpha = \frac{\langle p, v^{\perp} \rangle}{\langle u, v^{\perp} \rangle}, \qquad \beta = \frac{\langle u, p^{\perp} \rangle}{\langle u, v^{\perp} \rangle} \qquad (1)$$

となる。これは図3に示すようにu、vをそれぞれ直交座標系の基底（1，0）、（0，1）に変換した時のpの座標表現である。これは明らかに任意のアフィン変換に関して不変である。

3 画像誤差の振舞い

さて、モデル画像がカメラなりスキャナーなりで取り込まれ、観測画像として認識対象になる場合、レンズの歪みによる座標のずれ、デジタルに変換されたため起こる丸め誤差など、モデル画像と観測画像との間には何らかの誤差が関与すると考えられる（勿論、カメラ撮影は透視変換だから、アフィン変換で近似したことによる値のずれも当然誤差の内に入る）。

このように生じた誤差は、観測画像のHash Table上での検索先を誤らせるおそれがある。それに対する策は以下の2つが考えられる。

（1） 誤差の大きさに合わせてHash binsの刻み幅を大きくする。

（2） 誤差の大きさに合わせて検索箇所の検索範囲を広げる。

いずれの方法も、それに先だって画像の誤差がHash Table上でどの程度影響するかその挙動を想定する必要がある。

3.1 誤差の挙動の想定

観測画像の1点1点に誤差が乗ることを想定する。以下、[4]に基づき、誤差の各不変量に与える影響を計算する。まず、次の仮定を示す。

> 各点の座標（or位置）の読みとり誤差はその点を中心とする半径ϵの円の外にでない。

この仮定の下、不変量のとる値の変化を計算する。

図4を元にして不変量のとるべき値、最大の$(\alpha,\beta)\equiv(\alpha_{max},\beta_{max})$（図の左）、最小の$(\alpha,\beta)\equiv(\alpha_{min},\beta_{min})$を計算する。

$$\alpha=\frac{|p|\sin(\phi-\theta)}{|u|\sin\phi} \tag{2}$$

と表記できるので、誤差半径ϵに基づきα_{min}、α_{max}を計算すると、

$$\alpha_{min}=\frac{|p|}{|u|}\frac{\sin[\phi-\theta-\frac{\epsilon}{|v|}\{\sin(\phi-\theta)+\}-\frac{\epsilon}{|p|}]}{\sin[\phi-\frac{\epsilon}{|u|}(\sin\theta+1)-\frac{\epsilon}{|v|}\{\sin(\phi-\theta)+1\}]} \tag{3}$$

$$\alpha_{max}=\frac{|p|}{|u|}\frac{\sin[\phi-\theta+\frac{\epsilon}{|v|}\{\sin(\phi-\theta)+\}+\frac{\epsilon}{|p|}]}{\sin[\phi+\frac{\epsilon}{|u|}(\sin\theta+1)+\frac{\epsilon}{|v|}\{\sin(\phi-\theta)+1\}]} \tag{4}$$

となる。βについての式も僅かな変更で導き出せる。

従って、正しい票を得るためには、

$$\begin{aligned}\alpha_{min}\ &\leq\ \alpha\ &\leq\ \alpha_{max}\\ &\text{かつ}\\ \beta_{min}\ &\leq\ \beta\ &\leq\ \beta_{max}\end{aligned} \tag{5}$$

の範囲を一つのbinの大きさにする、或いはこの範囲に入っているbinsを検索する必要がある。これが前に述べた対応策の具体例である。

このように、(α,β)の誤差による挙動は大まかに言うとu、vが小さい程不安定になる（αの場合はu、βの場合はv）。従って、正しい票を得るために検索すべき範囲はu、vが小さいほど広いものになることが分かる。また、誤差により要求される検索範囲は各々のpによって同じ（u, v）の組でもことごとく違う値をとる。

3.2 誤差対策

こうしてみると、前述の（1）のやりかたは効率が悪い。Hash Tableの刻み幅は全てのプロセスに共通であり、誤差の挙動は各基底により異なるからである。

従って、（2）のやりかたが望ましいと思われる。Recognition Stageにおいて、観測画像における誤差の挙動を考え、Hash Tableの検索時に各点に検索範囲を与え、その範囲内の全bin内にある全データを各1票ずつとしてヒストグラムを作成するものとする[2]。

しかし、誤差解析が困難な不変量を用いた場合、（1）のやりかたで、経験的にTableの刻み目を操作する方法も考えた方が良い場合もあると思われる。

3.3 実験

以下の実験は、前述の（2）の方法に基づくものである。

実験対象：

> モデル画像：16点で構成される500 × 500 2次元ランダム点画像50種類。
>
> 観測画像：モデル画像の1つにアフィン変換を施し、その後に各点のx、y各軸方向に偏差2 pixelsのガウス分布に基づいた観測誤差を考え各点の位置をずらした。

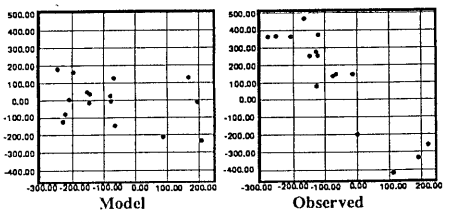

図5: モデル画像と観測画像の例

実験環境：

> *Hash Table*：1000 × 1000 bins。刻み幅はα、β各方向共に0.02均一。
>
> 誤差半径: ϵ= 0.0,2.0,6.0の3通り。つの場合を考えた。

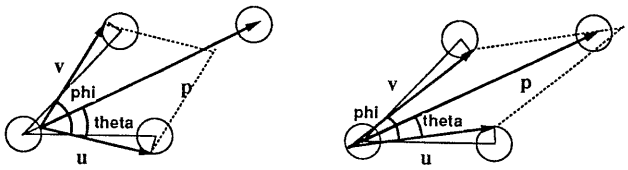

図 4: 誤差の挙動

Recognition Stage における基底の選択:$u, v \geq 100$、ϕ:
$[\pi/6, 5\pi/6] \cup [7\pi/6, 11\pi/6]$。これは、予め誤差の挙動が大きくなると予想される基底を除外するためである。

マッチングの評価基準:

1 6点の自乗誤差総和が5 7 6未満であること(1点平均6 pixels のずれ迄許容)。これ以上モデルの元の点からずれている画像は正解のモデルであっても信頼性に欠けるとみなす。

実験結果:実験結果を表1に示す。観測画像の基底の選び方は多くとも3360通りである。対するモデル画像の(モデル,基底)は対象性を考慮しても83312通りであった。認識プロセスだけを考えると順々にマッチングを行なう場合は最悪3360 × 83312回かかるのに対してGeometric Hashingだと最悪3360回で済むことが分かる。

誤差対策の結果、得票数を増やし、正解数を増やすのに効果をあげている。なお、Falseと出た内、$\epsilon = 2.0$ の全部、及び $\epsilon = 6.0$ の内の50個は正解でありながら、マッチングにおける誤差が大きかったために落されたものとみなされる(これを以降「信頼性の低い正解」と呼ぶ)。これは、モデルの基底と画像の基底によってアフィン変換を計算するのであるが、画像に加えられた誤差のために変換のパラメータが大きく狂ってしまったものである。

マッチングの評価基準は似ているものと区別がつかなくなるので、あまり値を大きくすることができない。従って、信頼性の低い正解も誤りのうちとして除外しなければならない。

これらの多くは画像側で選ぶ基底の長さの制限などである程度減らせるが、これを投票の段階でうまく減らせないだろうか。

3.4 修正策

ここまでは、既存の誤差対策である[2, 4]。

さて、誤差を考慮して検索範囲を増やす試みは誤差に対して効果があることが分かった。しかし、同時に検索範囲を増やすことによってかえって間違いや信頼性の低い候補も選んでしまうことも明らかになった。

これは、検索範囲が広い(間違った票を呼び込みやすい)所も狭い(呼び込みにくい)所も同じ票の価値を与えているのが原因と考えられる。[2]ではHash Tableが一様なデータ密度を持つ場合について数値解析を行なった結果、そのような間違った票が認識結果に悪影響を与える確率が充分小さいことを述べている。

しかし実際のところ、間違った候補や「信頼性の低い正解」は検索範囲を広げるにつれ増加し、Geometric Hashingの信頼性を向上する目的において障害となっている。

そこで、この問題を解決するために検索範囲にフィルタをかけて、元々の検索地点から離れるにつれてヒストグラムに加えるポイントを低くすることにした。票あたりの価値を下げることを試みることにした。また、同時に誤差についての挙動が大きくて多くの間違い票を呼び込みそうな検索範囲については、全体的に総得票ポイントを下げることにした。

この2つの条件に合うものとして、検索範囲に相関なしの2次元ガウス分布関数

$$G(x,y) = \frac{1}{2\pi\sigma_x\sigma_y} \exp(-\frac{1}{2}\{\frac{(x-x_m)^2}{2\sigma_x^2} + \frac{(y-y_m)^2}{2\sigma_y^2}\})$$

ここで、x_m, y_m:元々の検索地点

(検索範囲のx、y各方向のbinの数) $= 6\sigma_x, 6\sigma_y$ (6)

を用いた。1票の価値は各binの中心の座標に於けるこの関数の値を採用した。

実験結果は表2の通りになった。

Falseのうち「信頼性の低い正解」は $\epsilon = 2.0$ の時が13、$\epsilon = 6.0$ の時が2であった。

前の実験と比較して、間違い或は信頼性の低い正解の候補が誤差範囲を広げるにつれ信頼性の高い正解

ϵ	0.0		2.0		6.0	
Votes	True	False	True	False	True	False
14++					10	14
13			2		30	9
12			13	1	69	6
11			30		42	14
10			48	4	22	14
Total	0	0	98	5	173	57
Total 3360 Voting						

表 1: 検索結果のヒストグラム

に比べて低い得票数になるのが分かる。しかし、誤差を大きく想定すると得票数が大幅に下がってしまうので、これは逆に間違った候補が偶然最多得票をとる可能性もはらんでいるようにも思える。

3.5 考察

この2つの検索方法について総得票ポイントの解析を行なった。

まず、誤差（半径 ϵ）によって観測画像の α、β のずれる範囲を求めた（これを ϵ_{af} と呼ぶ）。これは、実際の解析が困難であるため、実際に 500 × 500 pixels の中から任意に4点 o、u、v、p を決め、実験の時の制限 $u, v \geq 100, \phi : [0, 2\pi], \theta : [\pi/6, 5\pi/6] \cup [7\pi/6, 11\pi/6]$ に加え、$p \geq 4\epsilon$ を満たしたものの中からランダムに 30000 回選んで計算したものの平均をとった。その結果

	$\epsilon = 2.0$	$\epsilon = 6.0$
$EX(\epsilon_{af})$	0.0189	0.0550
$VAR^{1/2}(\epsilon_{af})$	0.0189	0.0585

となった。このことにより、観測画像の各点の誤差の偏差を2とした場合でも実際の検索範囲は基底によって 0.0189 を中心にしてかなりばらつくことが分かる。

さて、誤差によるアフィン変換における不変量の挙動が偏差 σ_{af} のガウス分布に従うものとみなすと、一ヶ所の検索範囲での修正策の得票ポイントの期待値の計算式は以下の様になる。

$$Ex = \frac{d^2}{4\pi^2\sigma_{af}^2\sigma_{fil}^2}\{\sum_{i=-N-1}^{N} \exp(-(\frac{d^2}{2\sigma_{af}^2} + \frac{1}{2\sigma_{fil}^2})i^2)\}^2$$

$$N = [\epsilon_{af}], \sigma_{fil} = \frac{N + 0.5}{3},$$

σ_{af} : 不変量の偏差,

$d : HashTable$の刻値 (7)

ϵ	2.0		6.0	
Votes	True	False	True	False
8++	21			
7.5	4	4		
7	5			
6.5	6	18	3	
6	1	3	2	
5.5	3	268		
5	6	1	7	
4.5	9	1	7	
4	10		10	
3.5	7	19	17	2
3	5	8	22	2
Total	77	399	68	4
Total 3360 Votings				

表 2: 修正策のヒストグラム

分散もこの式を元にして求められる。

σ_{af} が大きくばらつくので、$0.5\sigma_{af} \sim 2.0\sigma_{af}$ についての2つの検索方法のそれぞれについての平均と偏差を計算した。

表3に結果を示した。表中に random とあるものは間違った候補がとり得る平均、偏差を示している。なお、間違った候補のデータは Hash Table 上に一様に分布しているものと仮定し、存在確率は

$$p = \frac{16-3}{1000 \times 1000}$$

とした。

計算結果によると、修正策を施さなかった場合、施した場合共に間違った候補がとり得る得票ポイントの平均及び偏差は極めて小さかった。このことから間違った候補が最大得票ポイントをとることは極めて稀であると考えられる。

次に、2つの検索方法についてアフィン変換の不変量の誤差の偏差別に比較を行なった。$\epsilon = 2.0$ の場合は検索範囲が狭いこともあって平均値も偏差もばらついているが、どちらの ϵ の場合についても、

- オリジナルの検索方法では誤差の比較的小さいものと大きいものとどちらについても正解であれば得票ポイントの値に大きな違いはない。
- 修正策の検索方法では誤差の大きさによって得票ポイントにはっきりと差が表れている。

ということがいえる。

$\epsilon = 2.0$ の場合については検索範囲が狭くそのために検索する bin の数が少ないこともあって、オリジナルでは誤差が大きい時の検索洩れになっている。

修正策では検索洩れに加え、検索する bin の数が少ないために検索時に得票にかけるガウス分布の離散近似が粗くなることもあって偏差が大きくなっている。従って、修正策において高得票の部分に誤差の大きい候補を混ぜないためには検索範囲は Hash Table の bin の刻値に対してある程度大きい必要がある。ただし、あまり検索範囲が大きいのも得票ポイントの期待値を下げるもとになるので、注意が必要である。

4 信頼性の向上

Geometric Hashing を実行する際に、最多得票を得たにもかかわらず、最終確認であるマッチングで「間違い」と判定されてしまうものが存在する。このようなものが投票時に高い最多得票ポイントで出るのを押さえられれば、最多得票ポイントの下限値を設定し、その値以上のポイントが得られた候補を選ぶだけでほぼ間違いなく正解を導き出すことができる。

下限値を設定するということは、実際にマッチングにかける回数を減らすことでもあるので、結果的に正解の候補が見つからなかった場合にかかる手間も減らすことが出来る。

マッチングに失敗する候補として考えられるのは、

1. 間違った候補に対して票が偶然集まって最多得票になってしまったもの
2. 構造がたまたま似ていたもの
3. 実際には合っているのであるが、計算された変換の狂いが大きく最後のマッチングで失敗したもの

の3つが今のところ考えられる。（1）については [2] などによって、このような現象は極めて少ないことが数値解析によって示され、今回の数値解析でもそのことが正しい事を確認した。

（2）は画像の持つ特有の問題であるのでこの方法では対策出来ない。しかし、このような物も得票するポイントが十分多いものを選べばマッチングの前にその多くを回避できると考えられる。

最後の（3）は前述の修正策によって改善し得ると思われる。数値解析の結果でも、全体的に誤差が小さいものと大きいものとでは小さいものの方が他者と比べて高いポイントを獲得するようになっている。

従って、

- （1）、（2）：極めて少ない。
- （3）：改善し得る。

となり、Geometric Hashing の信頼性は（3）についての考察によって更に上がることがいえる。

5 おわりに—全体の考察

実験の結果、Hash Table の検索範囲を増やすことにより、誤差の乗った観測画像においても元モデルの認識が可能であり、検索範囲にガウス関数を掛ける方法は、誤差が大きいと思われる基底の総獲得ポイントを著しく下げ、かつ、検索値地点の中心からはなれた点のポイントも下げるため、自乗誤差総和の比較的小さい（モデル,基底）に高得点を与えるため、正解の候補が取り易いことが示された。

現時点での問題点をまとめてみると次のようになる。まず、修正策について。

- 「これ以上とれば良い」という閾値が決定しにくい。
- 検索範囲に掛ける関数として、他にどのようなものが良いのか？

次に、データの分布方法について。

ϵ	2.0				6.0			
	オリジナル		修正策		オリジナル		修正策	
σ_{af}	平均	偏差	平均	偏差	平均	偏差	平均	偏差
0.5	9.267	0.874	53.10	5.008	13.64	0.180	6.245	0.965
0.75	12.46	0.147	3.173	0.953	13.00	0.000	1.751	0.251
1	10.64	0.593	1.966	0.792	13.00	0.001	0.918	0.110
1.25	8.524	1.005	1.335	0.669	12.94	0.016	0.751	0.122
1.5	6.745	1.250	0.961	0.580	12.97	0.009	0.486	0.072
1.75	9.960	0.738	0.603	0.208	12.98	0.005	0.339	0.047
2	8.726	0.971	0.485	0.192	12.92	0.023	0.298	0.049
random	0.002	0.038	0.000	0.009	0.008	0.091	0.000	0.003

表 3: 得票ポイントの平均と偏差

- 検索範囲を増やすこのやり方について。データが高密度に分布する部分が存在する場合、検索場所によっては正しい票以上に多量の間違った票を抱え込んでしまう。
- その場合の Hash Table 上のデータの密度をうまく均一に近くする方法は？
- また、そうしたとして間違った票を抱え込みにくいようにできないものか？

最後の疑問点が問題である。例え変則的に刻値を変えても、検索範囲は同じであるから、（ガウス関数を掛けると少しは変わるが）事情はあまり変化しない。結局位置関係で間違えるものは誤差の影響を被る。「値が少しは変わる」という点で検索範囲に関数を掛けるのが重要なのかも知れない。

そして、不変量について。

- アフィン近似に代わり、透視変換の不変量の誤差についての挙動の解析。
- また逆に、透視変換をアフィン変換で近似したことにより生じる誤差が Geometric Hashing で充分克服できるならば、これも重要なことになる。

Model-based Matching は、観測画像内に正解が見つけ出せない時に最も計算時間のかかる方法であるから、その「最悪の場合」に費やす時間をできるだけ減らすのも必要なことである。Geometric Hashing において実際にマッチングにかけるか判断するために得票数の閾値を設けるのはそのためでもある。また、認識時間とデータベースに確保するメモリとは trade-off の関係にあるので、認識時間の短縮のためにメモリをあまり大きくするのも問題であるので認識に効率の良いデータの選択も重要な問題である。

参考文献

[1] Y.Lamdan and H.J.Wolfson: "Geometric hashing : A general and efficient model-based recognition scheme", Second Int. Conf. Comput. Vision (Tampa, FL), pp. 238–249 (1988).

[2] J. Y.Lamdan and H.J.Wolfson: "Affine invariant model-based object recognition", IEEE Trans. on Robotics and Automation, **6**, 5, pp. 578–589 (1990).

[3] W.E.L.Grimson and D.P.Huttenlocher: "On the sensitivity of geometric hashing", Third Int. Conf. Comput. Vision (Osaka, JP), pp. 334–338 (1990).

[4] Y.Lamdan and H.J.Wolfson: "On the error analysis of 'geometric hashing"', IEEE Conf. on Comput. Vision and Pat. Recog. (Lahaina, Hawaii), pp. 22–27 (1991).