コンピュータビジョン 91−10
（1994. 11. 24）

CVCV-WG 特別報告：コンピュータビジョンにおける技術評論と将来展望（I）
— 投票と多数決原理に基づく幾何学的対象の検出と識別 —

和田俊和
岡山大学 大学院 自然科学研究科

「誤りを含む不完全な情報」から安定に図形の検出・識別を行なう一つの方法として，「Hough 変換」,「一般化 Hough 変換」,「Geometric Hashing」 などに代表される「投票」と「多数決」に基づくアルゴリズムがある．本報告では，これらのアルゴリズムの原理について解説するとともに，特長と問題点について述べ，その問題点を解決するための研究を簡単に紹介する．また，現在提案されている３つのアルゴリズムが持つ本質的限界を示し，投票と多数決に基づくアルゴリズムの今後の展望について述べるとともに「 CVCV ワーキンググループ[1]」によって行なわれた議論を掲載する．

CVCV-WG Special Report: Technical Review and View in Computer Vision (I)
— Detection and Recognition of Geometric Objects by Vote and Majority Verdict —

Toshikazu WADA
The Graduate School of Natural Science and Technology, OKAYAMA UNIVERSITY

In this paper, we discuss the advantages and problems of the algorithms based on "vote" and "majority verdict", which are capable of stable detection and recognition of geometric objects in noisy images. First, we describe the basic ideas of three major algorithms ; Hough Transform, Generalized Hough Transform and Geometric Hashing, and the relationship among these algorithms is clarified. Next, we summarize the advantages and problems of these algorithms and introduce some methods to resolve these problems. Finally, the limitation of these three algorithms are pointed out, and the discussion by the cvcv-working group on the breakthrough to the next stage of the "vote"-and-"majority verdict" algorithm are introduced.

[1] 情報処理学会コンピュータビジョン研究会では，ＣＶ技術評論・将来展望ワーキンググループ (略称 CVCV-WG, Current and Vision of Computer Vision, 幹事:井宮淳 (千葉大), 久野義徳 (阪大)) の活動を開始した．このWGでは，１９８０年代からのＣＶ研究の総括を行い,「何ができて何ができないのか，またその理由は何か」を具体的アルゴリズムのレベルから詳細に見直すことによって，今後のＶｉｓｉｏｎ研究の方向を見いだすことを目的とする． ＣＶの新潮流を探るために，若手の方数名位に，批判的なポジションペーパー＋サーベイを作っていただき，それを基に WG で議論をし，今後の方向性を探る予定である．今回の報告は，その第１回目にあたる．なお，本活動についてご意見，ご質問のある方は，cvcv@top.it.okayama-u.ac.jp へ連絡をお願いします．

1 まえがき

画像から現実世界のシーンに対する構造記述を生成するコンピュータ・ビジョンについては現在まで数多くの研究がなされてきたが，多様な入力画像に対して安定かつ正確な対象認識を実現することは未だに困難な問題である．このような困難さが生じる主な原因としては，次の2つが挙げられる．

- 画像に混入した雑音や，解析アルゴリズムの不完全性によって，誤った情報が抽出されてしまう．
- 隠蔽や，解析アルゴリズムの不完全性によって，抽出されるべき情報の一部が欠落してしまう．

このような，「誤りを含む不完全な情報」から安定に図形の検出・識別を行なう一つの方法として「投票」と「多数決」に基づくアルゴリズムがある．

「投票」と「多数決」に基づく図形の検出アルゴリズムの代表的な例としては，Hough 変換が挙げられる．Hough 変換は，画像中の特徴点が与えられたとき，その特徴点を含むすべての図形に関する仮説を能動的に生成し，多くの特徴点（証拠情報）から同一の仮説が生成される時，その仮説を正しいものと考えて図形の検出を行なう手法である．仮説の生成は，可能な仮説集合を表す空間に「投票」することによって実現され，各仮説に対する投票度数を調べることによって図形の検出が行なわれる．このように，Hough 変換は「投票による証拠情報の統合」と「多数決原理」に基づく対象検出法であり，個々の証拠情報に誤りや欠落があっても，全体としては安定な処理を行なうことができるという優れた特長を持っている．

Hough 変換は，1962 年 Hough[1] によって画像から直線を検出するための手法として提案され，その後の研究によって，円，楕円などの解析的な曲線，さらに 1981 年 Ballard[3] によって提案された「一般化 Hough 変換」によって任意の図形を画像から検出するアルゴリズムへと拡張されてきた．

最近では Lamdan ら [5] によって「Geometric Hashing」と呼ばれる幾何学的対象の識別法が提案され，図形の検出だけでなく，対象の識別問題にも Hough 変換と同様の「投票」と「多数決」に基づく計算の枠組が適用できることが示されている．

本稿では，コンピュータ・ビジョンにおける問題解決の方法として，Hough 変換や Geometric Hashing のような「投票」と「多数決」に基づく計算方式が有効であることを述べると同時に，その問題点と今後の展望について述べる．

2 基本原理

ここでは，最も基本的な直線検出用 Hough 変換と，任意図形の検出を行なう一般化 Hough 変換，複数の図形の検出と識別を行なう Geometric Hashing について概略を説明し，それらの特長と問題点について述べる．

2.1 直線検出用 Hough 変換

Hough[1] が提案した直線検出用の Hough 変換は次のようなものである．まず，画像空間中の直線を傾き a，切片 b を用いて

$$y = ax + b \tag{1}$$

と表現し，特徴点 (X_i, Y_i) が与えられたとき，直線のパラメータ a と b によって張られる「パラメータ空間」に

$$b = Y_i - aX_i \tag{2}$$

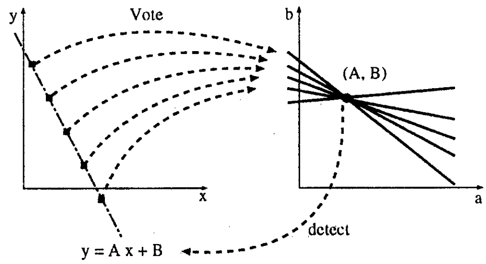


図 1: $a-b$ パラメータ空間を用いた Hough 変換

によって表される軌跡を描く．そして，全ての特徴点について軌跡を描いた後，パラメータ空間中で多くの軌跡が交わる点を抽出することにより，その点に対応する直線が画像空間中に存在するものとみなす（図 1）．

実際の計算機上で Hough 変換を行なう際には，パラメータ空間を「セル」と呼ばれる要素に分解し，

[投票] 軌跡が通過するセルの投票度数を 1 増やす．

[ピーク検出] 全ての特徴点からの投票が終了した後，投票度数が一定値以上の極大値を持つセルを抽出する．

という処理を行なう．

パラメータ (a, b) を用いて画像中の直線を表現する場合，画像の大きさが有限であってもパラメータ空間は有界にはならない．この問題を解決するために，Duda ら [2] は，画像空間中の直線を垂角θと，原点からの符合つき距離 ρ を用いて，

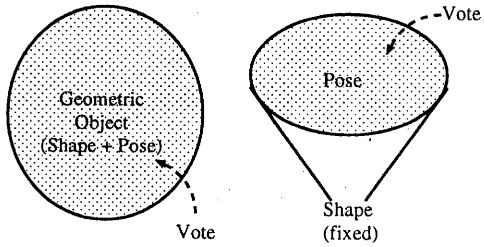
$$\rho = x\cos\theta + y\sin\theta, \qquad (0 \le \theta < \pi) \tag{3}$$

と表す方法を提案している．式 (3) を用いると，角度 θ は有界であり，画像の大きさが有限であるため ρ も有界となる．このため，ほとんどの直線検出用 Hough 変換では $\rho-\theta$ パラメータ空間が用いられている．

2.2 一般化 Hough 変換

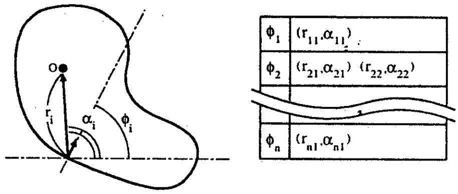
直線や楕円のように，代数方程式によって表現される解析的対象を検出する Hough 変換では，方程式中のパラメータによって張られる空間上で投票と多数決を行なえばよい（図 2(a)）．しかし，パラメータによって表現することができない任意形状の幾何学的対象を検出する場合には，このような方法を用いることができない．Ballard [3] は，「一般化 Hough 変換」を提案し，この問題を解決した．

一般に幾何学的な対象は，「形状」と「ポーズ（位置・姿勢・大きさ）」の組によって表される．解析的な対象を扱う Hough 変換で用いられるパラメータには，どのパラメータが形状を表し，どのパラメータがポーズを表しているのかという区別はない．これに対して，形状とポーズを区別した場合には，「形状をデータとして与え，ポーズに対応する座標変換パラメータによって対象を表す」という対象の表現法が考えられる．この場合には形状のパラメータ表現は不要であり，任意形状の対象を表現することが可能になる．このように“形状を固定し”，座標変換パラメータ空間上で投票と多数決を行なう Hough 変換が一般化 Hough 変換である（図 2(b)）．具体的な変換方法を以下に示す．



(a) Hough Transform for Analytical Object (b) Generalized Hough Transform

図 2: Hough 変換と一般化 Hough 変換における投票空間



(a) モデル記述のための幾何学的配置　(b) R テーブル

図 3: 一般化 Hough 変換における形状記述

[パラメータ空間]
平行移動 (u,v), 回転角 θ, 拡大率 s によって張られる 4 次元空間をパラメータ空間とする．

[形状モデルの表現]
検出対象図形の境界線上の各点において，基準点 O に向かうベクトル $\vec{x}_i = (r_i \cos\alpha_i, r_i \sin\alpha_i)$ と，画像の濃度勾配方向 ψ_i をあらかじめ求めておく．各 ψ ごとに，極座標表示での $\vec{x}$ の座標値 (r,α) を登録した「R テーブル」を作成しておく（図 3）．

[投票方法]
画像上の特徴点 (X_j,Y_j)，およびその点における勾配方向 Ψ_j を求める．パラメータ空間中の全ての (θ,s) の組合せに対して，$\Psi_j-\theta$ に対応する R テーブル中のエントリ $(r(\Psi_j-\theta), \alpha(\Psi_j-\theta))$ を求め，回転角 θ，拡大率 s に対応する平行移動ベクトル (u,v) を以下の式によって求め，点 (u,v,θ,s) の投票度数を 1 増やす．

$$u = X_j + r(\Psi_j-\theta)\times s\times\cos(\alpha(\Psi_j-\theta)+\theta) \quad (4)$$
$$v = Y_j + r(\Psi_j-\theta)\times s\times\sin(\alpha(\Psi_j-\theta)+\theta) \quad (5)$$

[図形の検出]
投票度数の高い座標変換パラメータを抽出し，これを用いて形状モデルを画像空間に写像することにより，図形の検出を行なう．

2.3 Geometric Hashing

前述のように，幾何学的対象は,「形状」と「ポーズ」の組によって表すことが可能である．一般化 Hough 変換では形状を固定し，ポーズ空間への投票を行なった．これとは逆に，図 4に示すように “ポーズ を固定し”，パラメータ表現された形状空間 に対して投票／多数決を行なう方法が考えられる．

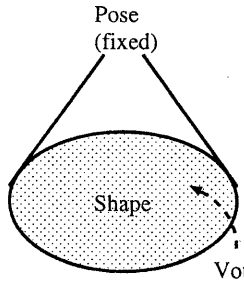


図 4: Geometric Hashing における投票空間

Lamdan ら [5] は，このような考えに基づいて対象の識別を行なう「Geometric Hashing」を提案した．この手法では，画像中に存在する対象の識別と検出を同時に行なうことができる．

Geometric Hashing では，ポーズに対して不変な性質を持つ「不変特徴」の集合によって形状が表現される．識別対象から抽出された不変特徴は，不変特徴によって張られる空間を離散化した「Hash Table」にあらかじめ登録される．そして，“ポーズを固定して” 画像から Hash Table に投票することによって，画像と識別対象とのマッチングを行なう．以下，平行移動，拡大，回転に対して不変な特徴を用いた Geometric Hashing の具体的計算方法について述べる．

[モデルの登録]
まず，識別対象のモデルを Hash Table に登録する方法について述べる．

図 5(a) に示すように N 個の点によって表されたモデル $M = \{\vec{p}_1, \vec{p}_2, \cdots, \vec{p}_N\}$ から，2 点 $\vec{p}_i, \vec{p}_j$ を選択する．これらの 2 点から，

$$\vec{X}_{ij} = \frac{\vec{p}_j - \vec{p}_i}{\|\vec{p}_j - \vec{p}_i\|^2} = (x_{ij}, y_{ij}), \qquad \vec{Y}_{ij} = (-y_{ij}, x_{ij}) \quad (6)$$

なる直交基底を生成する．この基底で表された各特徴点の座標値を求めることにより，図 5(b) に示すような不変特徴の集合

$$\{(x_f, y_f)\} = \{(\vec{X}_{ij}\cdot(\vec{p}_k - \vec{p}_i), \vec{Y}_{ij}\cdot(\vec{p}_k - \vec{p}_i))\} \quad (i,j \neq k) \quad (7)$$

が得られる（・は内積を表す）．この $\{(x_f, y_f)\}$ は，もとの点集合 $\{\vec{p}_i\}$ が平行移動や回転，拡大などの座標変換を受けても変化しないという性質を持っている．

求められた不変特徴は，x_f, y_f によって張られる空間を離散化した Hash Table 中の対応する要素に “（モデル，記述に用いた基底）” の組として登録される（図 5(c)）．

モデル中の全ての 2 点の組合せから生成される基底に対してそれぞれ不変特徴を求め，それらを全て Hash Table に登録しておく．このようなモデルの多重記述により，モデルと相似な図形が与えられたとき，その図形に含まれる任意の 2 点から求められる不変特徴と Hash Table 上に記録された（モデル，基底）の組のどれかが一致することが保証される．Geometric Hashing では，対象とするモデルの全てに関して同様の記述を行なっておくことにより，複数のモデルを同時に扱うことが可能となる．

[投票と識別]
次に，入力画像中の特徴点集合から，Hash Table への投票，識別を行なう方法について述べる．

1. Hash Table の各要素に対する投票数を 0 にする．

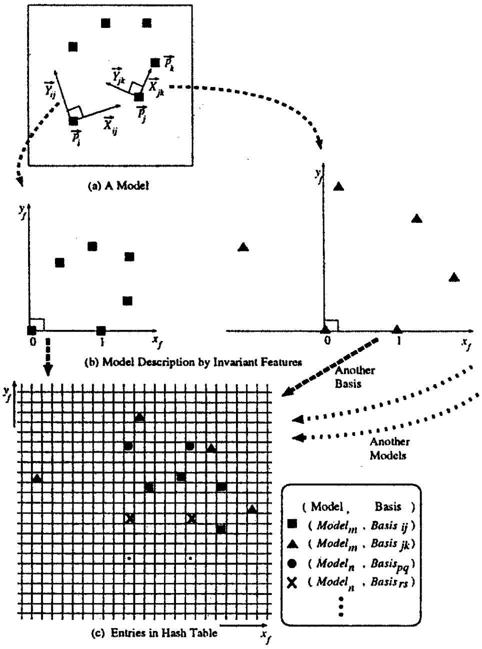


図 5: 不変特徴による形状の記述

2. 画像からまだ選択されていない任意の2点を選択し，式（6）によって直交基底を求める．

3. この基底に対して画像上の全ての点の不変特徴を式（7）を用いて計算し，対応する Hash Table 中の要素の投票度数を各々1増加させる．

4. 投票が終了した後に，(モデル，基底）の組ごとに投票度数を集計し，あらかじめ与えられた閾値を越えるものがあるかどうかを調べる．閾値を越える投票が得られたものがなければ，1の処理に戻る．

5. 高い投票が得られた（モデル，基底）組の基底と，画像から求められた基底を対応付ける座標変換を求める．

6. 求められた変換を用いてモデルを画像上に写像し，モデルにマッチする特徴点が十分存在することを確認する．特徴点が十分存在すれば終了し，そうでなければ1の処理に戻る．

画像中に存在する複数の対象を識別，検出する場合には，モデルにマッチする特徴点集合を取り除きながら，上記の手続きを繰り返し行なう．

2.3.1 Geometric Hashing の意味と性質

Hash Table に登録された形状モデルは，不変特徴によって記述された形状の「テンプレート」と見なすことができる．また Geometric Hashing において，“ポーズを固定する”ということは，画像から不変特徴を求める際に1つの基底を用いることを意味している．「基底によって画像を不変特徴に変換し，Hash Table 上で複数のテンプレートとのマッチングを行なう」という処理を様々な基底について繰り返すことが Geometric Hashing の本質である．

2.4 「投票」と「多数決」に基づくアルゴリズムの特長と問題点

以上に述べた Hough 変換，Geometric Hashing などの投票と多数決に基づく対象の検出／識別アルゴリズムは，以下に示す優れた特長を持っている．

[特長]

- 局所的な特徴のみを用い，特徴点間の連結性を用いないため，隠蔽などによって特徴点が欠落した場合でも対象の検出・認識が可能である．
- より多くの証拠情報によって支持される仮説を求めるという多数決原理に基づく手法であり，雑音などによって生じる偽の特徴点が含まれる場合でも誤りが起きにくい．
- 任意形状の検出･認識に適用できる．特に Geometric Hashing では，画像から3次元物体の識別を行なったり，アフィン変換に対する不変特徴を用いるなどの拡張 [5] も比較的容易に実現できる．
- 投票計算が各特徴点ごとに独立に行なわれるため，計算の並列化が容易である．

このように，投票と多数決に基づく対象の検出／識別アルゴリズムは，「安定性」，「拡張性」の点で，特に優れている．

一方，「精度」，「計算コスト」の点では以下のような問題点がある．［問題点]

- 精度
  - 画像中に複数の図形が含まれる場合，異なる図形からの投票が投票空間上で重畳する．
  - 特徴点の誤差（位置の変動）により，投票空間中での投票位置がずれてしまう．
  - 投票空間の標本化基準が不明確であり，不適切な標本化を行なうと Aliasing が生じるため，投票数に偏りが生じる．
- 計算コスト
  - Hough 変換では，パラメータ空間中のセルは可能なパラメータの組の全てを表しており，その個数は(パラメータの量子化数)^(パラメータ空間の次元数)となり，消費するメモリは，パラメータ空間の次元数に対して指数関数的に増加する．また投票軌跡は，(パラメータ空間の次元数−1）次元の超曲面を形成するため，投票に要する計算は(特徴点の個数)×(パラメータの量子化数)^(パラメータ空間の次元数−1）となり，投票軌跡の次元数に対して指数関数的に増加する．
  - Geometric Hashing では，Hough 変換のようにパラメータを連続的に変化させながら投票を行なうという方式ではなく，点に対する投票を行なうため，1回の投票は高速に計算することができる．しかし，同一の図形に含まれる複数の点から基底を選択する必要があるため，基底の選択を何回も繰り返さなけ

ればならず[2]，図形を構成する特徴点の個数に比べてそれ以外の特徴点の個数が増加すると，投票回数が増加するという問題点がある．また投票数の集計を（モデル，基底）の組毎に行なわなければならないため，投票と集計に時間がかかる．

以下では，これらの問題点を解決するための研究成果について述べる．

3 高精度化

前述の通り「投票」と「多数決」に基づくアルゴリズムは，特徴点の誤りや欠落に対して安定な処理を行なうことができるが，結果として得られる図形や変換のパラメータは必ずしも精度が高いとは言えない．この精度の問題に対する研究を Hough 変換と Geometric Hashing の両者について，整理する．

3.1 Hough 変換における高精度化

Hough 変換において，パラメータの精度が低下する原因は，

- 画像中に複数の図形が含まれる場合，異なる図形からの投票軌跡が重畳する．
- 特徴点の誤差（位置の変動）
- 投票空間の標本化による投票数の変動

などが挙げられる．

図形間の干渉による誤差を防ぐ方法としては，パラメータ空間に側抑制フィルタをかける方法 [15] や，抽出されたピークを通過する軌跡をキャンセルしながらピーク検出を行なう方法（バックマッピング）[11] などが有効である．

近年，Niblack ら [13] は，直線分に対応するパラメータ空間中のピーク形状モデルを求め，このモデルをパラメータ空間中の投票度数分布に当てはめることにより，直線パラメータと同時に直線分の端点を求めるという手法を提案している．この方法は，図形の干渉や雑音に対してロバストであるだけでなく，画像空間を再度走査することなく直線の端点や幅を求めることができるという優れた特長を持っている．

一方，パラメータの量子化誤差に関する問題は古くから指摘されており，多くの研究者によって解析的な研究がなされてきた [14]~[17]．しかし，これらの研究では画像空間が格子状に離散化されていることを考慮していなかったため，現実のディジタル画像からの投票を行なう際に生じる量子化誤差の影響を完全に取り除くことはできなかった．

近年，Maitre[16] によって指摘されていた $\rho-\theta$ パラメータ空間における投票度数の偏りは，画像空間に存在するディジタル直線の間隔と ρ パラメータ軸の量子化間隔の不一致に起因することが，森本ら [21] と和田ら [18] によって明らかにされた．森本らは θ によって帯域幅が変化する可変平滑化フィルタを用いることによって $\rho-\theta$ パラメータ空間の投票度数の偏りを修正する方法を提案している [21]．和田らは，$\rho-\theta$ パラメータ空間を非線形に伸縮することによって，パラメータ軸を均一に量子化しても投票度数に偏りが生じず，投票軌跡が区分的直線となる $\gamma-\omega$ パラメータ空間 [18] を構成するとともに，ディジタル直線の高精度な検出を行なうための Hough 変換アルゴリズムを構成した [19]．また，浅野ら [20] は，セルを用いず，特徴点のソーティングを基本演算としたアルゴリズムにより，$\gamma-\omega$ Hough 変換と同様の高精度な直線検出を行なう方法を提案している．

[2] 正方形の頂点の組が一つだけ含まれる１００個の特徴点から相似変換に不変な特徴を計算し，その正方形を抽出する問題では，${}_{100}C_2/{}_4C_2 = 825$ となり，平均８２５回の基底の選択を行なわなければならない．したがって，$825 \times (100-2) = 80850$ 個の不変特徴を計算することになる．

3.2 Geometric Hashing における高精度化

Geometric Hashing では，以下の理由により，Hough 変換の場合よりも精度は低い．

- 不変特徴の計算によって各特徴点の誤差が拡大される．
- モデル（を構成する不変特徴）の Hash Table 上における密度が不均一である．

これらの問題を解決する手法としては，以下の手法が提案されている．

- 直線を基本特徴とすることにより，個々の特徴の誤差を少なくする方法 [6]

  Geometric Hashing では，アルゴリズムの構造上，多数の特徴点から成る図形に対して登録と認識の処理を行うためには膨大な計算時間とメモリが必要となる．このため，通常は interest operator などによって抽出された比較的数の少ない特徴点に対して Geometric Hashing が適用される．しかし，このような局所的処理によって求められた特徴点の位置は一般に不安定であるため，より安定な特徴として直線を基本特徴とした Geometric Hashing が提案されている．

- 不変特徴によって決まる投票点のみに投票を行なうのではなく，その周辺にも投票を分散させることにより，誤差による投票のずれを軽減する方法 [22]~[23]

  Geometric Hashing では，「個々の特徴点の誤差」×「基底に含まれる誤差」が，基底によって決まるモデルと画像中の対象との間の「座標変換」によって拡大されたり縮小されたりする．この手法では，誤差モデルを導入して誤差の伝搬を解析し，それに基づいて投票の際に用いる分布関数（投票の広がり）を決定している．しかし，この手法では，Hash Table 上に登録されたモデルの「密度」を考慮しておらず，似通った特徴を持つモデルに投票が分散してしまうという問題がある．

このように，Geometric Hashing における高精度化の試みはなされているが，Hash Table 上に登録されるモデルの密度の不均一さに関しては，問題は指摘されているものの解決策は示されていないのが現状である．

4 高速化

投票と多数決に基づく図形の検出／識別アルゴリズムの実用化を図るには，計算の高速化が不可欠である．以下に，Hough 変換と Geometric Hashing の高速化手法について整理する．

4.1 Hough 変換の高速化

現在提案されている Hough 変換の高速化法は，以下の６つの種類に分類することができる．

1. 確率的アルゴリズムの導入 [24]~[29]
2. Hough 変換の並列化 [30]~[35]
3. 特徴点近傍の局所的特徴を用いる方法 [39]~[42]

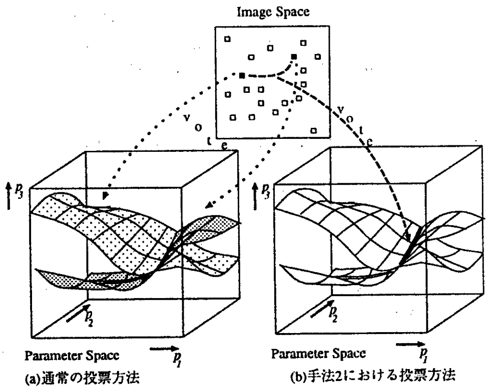


図 6: 確率的 Hough 変換における投票方法

4. 階層的な投票を行なう方法 [43]～[44]

5. パラメータ空間の分解 [45]～[46]

6. 軌跡の高速描画アルゴリズム [47]～[49],[18]

ここでは，1について解説する．2～6については割愛するので文献を参照されたい．

4.1.1 確率的 Hough 変換

情報科学で取り扱う問題の中には，計算は可能であるが現実には天文学的時間やメモリが必要となり解くことができないという問題が数多く存在する．これらの問題は，実用上は近似解が得られれば十分である場合が多い．近似解を求める方法としては，1）問題固有のヒューリスティクスを用いる方法，2）確率的アルゴリズムを導入する方法，などが考えられる．特に確率的アルゴリズムの多くは，試行（計算）の回数が少なければ推定量（解）の精度が低く，試行（計算）を十分な回数繰り返せば，精度の高い推定量（解）が得られるという確率的試行と同様の漸近的性質を持つため，これを用いれば，計算量と解の精度のトレードオフを考慮した問題解決を行なうことができる．

近年，Hough 変換においても，確率的アルゴリズムの導入により投票計算を高速化する方法が提案されている．Hough 変換における確率的アルゴリズムは，投票に用いる特徴点をランダムに選択するものであり，具体的には，

［手法1］投票に用いる特徴点をランダムに選択し，全ての特徴点からの投票を行なわない．

［手法2］複数の特徴点の組をランダムに選択し，これらの特徴点の組から求められる投票軌跡の交わりの部分に対して投票を行なう（図 6(b)）．

という2種類の方法が提案されている．

手法1の投票回数を減らす方法としては，Fischler ら [24] による Hough 変換の Parallel guessing implementation，Kiryati ら [25] による Probablistic Hough などが挙げられる．

手法2の複数の特徴点の組からの投票を行なう方法としては，Bergen ら [50] による Monte Carlo Hough 変換，Xu ら [26][27] による Randomized Hough 変換 (RHT)，塩野 [29] による RCIH 法，Leavers [28] による Dynamic Generalized Hough 変換 (DGHT) などがある．これらの手法では，投票すべきセルの個数が減少するため，投票計算が高速化される．以下に，セルへの投票回数を基準とした場合の，手法2と通常の Hough 変換との計算量の比較を行なう．

まず，画像中の全特徴点の個数を N，検出対象図形に含まれる特徴点の個数を n_f，1回の投票に用いる特徴点の個数を n，投票回数を N_v で表す．この場合，手法2における図形パラメータへの投票度数の期待値は，$(n_f/N)^n \times N_v$となる．通常の Hough 変換と同様に，n_f の度数が蓄積されるために必要な投票回数は，$N_v = n_f \times (N/n_f)^n$ である．パラメータ空間の次元数を n_d，各パラメータの量子化数が同じであるものとしてこれを N_aで表すと，手法2におけるセルへの投票回数 N_iは，

$$N_i = n_f \times (N/n_f)^n \times N_a^{n_d-n} \tag{8}$$

となる．通常の Hough 変換では，$n=1$ であるから，

$$N_i = N \times N_a^{n_d-1} \tag{9}$$

となる．式（8），(9）から，手法2と通常の Hough 変換の計算量の比は，$(N/n_f)^{n-1} : N_a^{n-1}$ となり，$N/n_f < N_a$ が成立する場合には手法2によって高速化が実現できることが分かる．

手法2では，同じ図形上の複数の特徴点を選択しなければ，その図形パラメータへの投票が行なわれないため，正確な検出を行なうためには，特徴点の組を選択する回数を増やさなければならない．したがって，より高速な計算を実現するには同一の図形から特徴点の組が選ばれる確率を高くする工夫が必要である．RHT では，近くに存在する特徴点の組は同一の図形に属する確率が高いと考え，画像中に矩形窓を設け，その中から特徴点を選択するという方法を用いている [10]．また，DGHT では画像の局所領域に対して図形パターンの当てはめを行ない，図形パターンに含まれる特徴点の中から投票に用いる特徴点の組を選択する方法を用いている．

また，確率的 Hough 変換では，プログラムの停止条件が計算の効率と解の精度を大きく左右するが，これに関しては文献 [27],[28] で詳しい検討が行なわれている．

4.2 Geometric Hashing の高速化

Geometric Hashing では，

- 基底のランダムな選択と Hash Table 上の点に対する投票，[3]
- モデルにマッチした特徴点の除去

などの高速化の工夫がアルゴリズムの提案当初から盛り込まれており，単純なアルゴリズムの変更による高速化の余地は少ない．しかし，

- アルゴリズムの並列化による高速化 [36]～[38]

に関しては，盛んに研究が行なわれている．

5 アルゴリズムの限界と展望

以上のように投票と多数決に基づく対象の検出／認識アルゴリズムに関する研究は数多くなされている．しかし，それらはあくまで改良に関する研究であり，アルゴリズムの適用範囲と機能は第2章で述べた通り変わりはない．表1に Hough 変換，一般化 Hough 変換，Geometric Hashing の適用可能な対象と基本機能を整理する．

[3]これは，本質的に RHT と同一の投票方式である．

手法	適用可能な対象	機能
Hough 変換	代数方程式によって表現される幾何学的対象	検出
一般化 Hough 変換	点，もしくは線の集合によって表現された形状に対して線形な座標変換を適用した幾何学的対象	検出
Geometric Hashing	点，もしくは線の集合によって表現された形状に対して線形な座標変換を適用した幾何学的対象	識別と検出

表 1: アルゴリズムの適用範囲と機能

この表から，投票と多数決に基づくアルゴリズムで扱うことができる対象のクラス（モデル）は，代数方程式，もしくは点（線）の集合とそれに対する座標変換といった具体的表現が可能な対象に限られることが分かる．したがって，

- 文字パターンなど，対象の変形に関するモデルを作ることが困難で，統計的な取り扱いをせざるを得ない対象
- 弾性輪郭など形状を代数的に表現することは可能であっても，表現自体が複雑過ぎて仮説の空間を構成することが困難な対象

などは，現状の投票と多数決に基づく対象の検出／認識アルゴリズムでは取り扱うことができない[4]という限界がある[5]．

このような限界を打破し，豊かな表現力を持った対象認識アルゴリズムを構成することができれば，より多くの問題に対して安定な処理が可能になるものと考えられる．このようなアルゴリズムを構成する方法としては，

- 特徴点の座標値のような定量的特徴ではなく，定性的特徴を用いる
- 統計的パターン認識手法を併用する

などが考えられる．

6 議論

以下に CVCV メンバ（敬称略）による議論の要約を掲載する．

久野：大阪大学 HT 等はこれからも注力すべき重要技術で，CV の問題の中心的なものの一つと考えられるのか，それとも単なるツールで，長所短所を明らかにして，それに注意して使う，あるいは短所を改良する部分的研究を行う程度のものなのか？

和田：岡山大学 ここで紹介したアルゴリズムに関しては，今後は「精度」と「計算コスト」に関する改良の研究に重点が置かれることになると思う．「精度」や「計算コスト」という問題を個別に取り上げれば，現状でも完成度の高い計算手法がいくつか提案されており，こういった研究は実用化という観点からは重要な研究であり，今後も続けられるべきだと考える．

本文で取り上げた投票と多数決に基づく対象の検出／識別アルゴリズムというのは，Hough 変換，一般化 Hough 変換，Geometric Hashing に限ったものではない．1で述べたように，こういう枠組で問題を考えることにより，「誤りを含む不完全な情報に対する安定な処理をいかに実現するのか」という CV における課題の一つが解決できるのではないかということを主張したい．

現状で，投票と多数決に基づく対象検出／識別の研究はある程度の成功を収めたが，また一方で5で述べたような限界も見えはじめている．この限界を克服することにより，アルゴリズムの一般性は増し，画像特徴の抽出段階から識別，さらには学習まで，投票と多数決に基づくアルゴリズムによって構成することができるのではないか？これが可能であれば，様々な入力画像に対して安定に動作するシステムが実現できるのではないかということを5で述べたつもりである．

井宮：千葉大 5の限界と展望で述べられている複雑な形状に関しては，形状の線系予測モデルを認識のモデルとして，対象の形状記述パラメータを投票により決定するという筋書きが考えられるのではないか？

栗田：電総研 形状の記述のための自己回帰モデルは，データの順序に依存したモデルであり，フーリエ記述子と同様に，基本的には自己相関（隣との関係）に基づいている．投票と多数決が有効に働くには，たぶん，証拠の独立性が仮定できる必要があると思うが，自己回帰モデルやフーリエ記述子では，その仮定がなりたたないのではないか？

鶴田：九州大学 個人的には，現在ネオコグニトロンに代表されるハイパーコラム（ＨＣ）のモデルによる識別に興味を持っている．ＨＣの基本原理はテンプレートマッチングですので，テンプレートマッチングと Hough 変換との比較に興味を持っている．

ＨＣの利点の１つは，テンプレートが部分特徴を用いて階層的に表現されることにあり，異なる対象のテンプレートの間で部分特徴が共有されるため，全体として効率がよくなっていると考えている．これは，多重解像度を用いない階層的モデルに基づく認識に対応づけて考えることができる．

井宮：千葉大 Hough 変換，Geomtric Hashing では，投票空間における投票の制御が十分ではない．たとえば，Geometric Hashing の学習過程で投票空間上にボロノイ領域を設定すれば，命題の数を減らすことになる．

鶴田：九州大学 命題数を減らしてしまうのは，「高精度化」とトレードオフの関係にあるのではないか？

栗田：電総研 「投票」と「多数決」による方法がどれくらいの範囲の課題をカバーできるかを知るためには，それが適応できる前提条件もう少し明確にする必要があるのではないかと思う．たとえば，証拠は独立で，しかも，評価が加法的である必要があるのではないか．

また，ハフ変換では，全パラメータ空間を離散化して，すべての可能性を調べることにより直線を決定するが，探索という観点から見ると，これは最も原始的で堅実なやり方であるが，その他のもっと効率的な探索方法が考えられないのか？

和田：岡山大学 投票結果に対して Genetic Algorithm や Bug 型探索 のように Multimodal な最適化手法を適用し，ピークを検出する手法もあるが，効率という観点からは投票しながらピークの位置を絞り込んでいくという手法を考えないとあまり意味は無い．

井宮：千葉大 投票しながら候補を絞り込むという投票の制御を Newton 法でおこなう方法が，

E.Oja, L. Xu, P Kultanen, "Curve detection by an extended self-organizing map and the related RHT method" Proc. of INNC 90 Paris, Vol 1.pp.27-30, 1990

で述べられている．

7 むすび

本報告では，投票と多数決原理に基づく対象の検出／識別アルゴリズムに関して近年の研究成果，およびその限界と展望について述べ，それに対するワーキンググループ内での議論の一部を掲載した．

高精度化に関しては，量子化誤差を考慮した高精度なディジタル直線の検出法，ピーク形状の最適な当てはめにより高精度に直線分を検出する方法などが明らかにされている．今後は，直線以外の幾何学的対象とした高精度な Hough 変換，および高精度な Geometric Hashing を構成することが課題が残されている．高速化に関しては，確率的アルゴリズムの導入が成功を収めている．ワーキンググループ内部での議論は，投票と仮説の探索を行なう効率的な方法に関して議論が行なわれ，投票を行ないながら同時に仮説の絞り込みを行なう手法が紹介されるなど，有益な議論がなされた．

投票と多数決に基づく幾何学的対象の検出／認識アルゴリズムの限界は，検出／識別の対象が代数的に表現が可能なものに限られている点であり，この限界を打破することにより，このアルゴリズムがより多くの認識問題に対して適用可能になるものと考えられるが，この問題に関しては今後の研究に期待したい．

[4] Geometric Hashing では，複数のモデルを扱うことが可能であるが，有限個のモデルを用いて対象の統計的性質を表現することは困難である．

[5] 投票と多数決に基づくアルゴリズムでは，隠蔽や雑音に対して安定な対象の検出／識別を行なうことにアルゴリズムの主眼が置かれており，「データに欠落があり得る」「間違ったデータが混入している」ということを考慮する必要がある．このようなデータに対して従来の統計的パターン認識手法をそのまま適用することができないことを考えれば，この限界はある程度必然的なものであることが理解できる．

謝辞

投票と多数決に基づく対象の検出／認識のアルゴリズムを「形状」と「ポーズ」という観点から，Hough 変換，一般化 Hough 変換，Geometric Hashing の３つに整理できるという考えは，岡山大学工学部松山隆司教授のアドバイスによるものであり，この考えがなければ本報告をまとめることができなかった．松山教授に深謝する．

参考文献

[原理]

[1] P.V.C. Hough, "*Method and Means for Recognizing Complex Patterns*", U.S. Patent No. 3069654, 1962

[2] R. O. Duda, and P.E. Hart, "*Pattern Recognition and Scene Analysis*", Wiley, New York, 1973

[3] D.H. Ballard, "*Generalizing the Hough transform to detect arbitrary shapes*", PR 13, 2, pp. 111-122, 1981

[4] G. C. Stockman and A.K. Agrawala, "*Equivalence of Hough Curve Detection to Template Matching*", Com. of ACM 20, 11, pp. 820-822, 1977

[5] Y. Lamdan and H.J. Wolfson, "*Geometric hashing: a general and efficient model-based recognition scheme*", Proc. of ICCV, pp. 238-249, 1988

[6] F. Stein and G. Medioni, "*Structural hashing: Efficient 2-D Recognition*", PAMI 14, 12, pp. 1198-1204, 1992

[サーベイ，解説]

[7] 松山，輿水，"Hough 変換とパターンマッチング"，情報処理 30, 9, pp. 1035-1046, 1989

[8] J. Princen, J. Illingworth, and J. Kittler, "*A formal definition of the Hough transform: Properties and relationships*", JMIV 1, pp.153-168, 1992

[9] V.F.Leavers, "*Which Hough Transform ?*", IU 58, 2, pp. 250-264, Sep. 1993

[10] H. Kälviäinen, L. Xu, E. Oja, "*Recent Versions of the Hough Transform and the Randomized Hough Transform: Overview and Comparisons*", Lappeenranta Univ. of Tech. Dept. of Info. Tech. Research Report, vol. 37, 1993

[高精度化]

[11] G. Gerig, "*Linking Image-Space and Accumulator-space: A New Approach for Object-Recognition*", Proc. of 1th-ICCV, pp. 112-115, 1987

[12] D.J. Hunt, L.W. Nolte, A.R. Reibman, and W.H. Ruedger, "*Hough transform and signal detection theory performance for images with additive noise*", CVGIP 52, pp.386-401, 1990

[13] W. Niblack and T. Truong, "*Finding Line Segments by Surface Fitting to the Hough Transform*", Proc. of MVA'90, pp.237-240, 1990

[14] S.D. Shapiro and A. Iannino "*Geometric Constructions for Predicting Hough Transform Performance*, PAMI 1, 3, pp. 310-317, 1979

[15] C.M. Brown, "*Inherent Bias and Noise in the Hough Transform*", PAMI 5, pp. 493-505, 1983

[16] H. Maitre, "*Contribution to the prediction of performances of Hough transforms*", PAMI 8, pp. 669-674, 1986

[17] N. Kiryati and A.M. Bruckstein, "*Antialiasing the Hough transform*", GMIP 53, pp. 213-222, 1991

[18] 和田，藤井，松山，"$\gamma-\omega$ ハフ変換－可変標本化による$\rho-\theta$パラメータ空間のひずみの除去と投票軌跡の直線化－" 信学論 D-II, J75-D-II, 1, pp.21-30, 1992

[19] 和田，関，松山，"ディジタル直線の幾何学的特性に基づいた$\gamma-\omega$ハフ変換の高精度化"，信学論 D-II, J77-D-II, 3, pp.529-539, 1994

[20] 浅野，加藤，"ディジタル画像における直線成分抽出のためのアルゴリズム"，信学技報，PRU-84-3, pp. 15-22, 1993

[21] 森本，尺長，赤松，末永，"可変フィルタによるハフ変換の高精度化"，信学論 D-II,J75-D-II,9, pp.1548-1556,1992

[22] Y. Lamdan and H.J. Wolfson, "*On the error analysis of* "geometric hashing"", Proc. of CVPR, pp. 22-27, 1991

[23] 川西，出口，森下，" Geometric Hashing による画像マッチングのロバスト性向上について"，情処研報，CV80-22, pp.161-168, 1992

[高速化]

[24] M.A. Fishler and O. Firschein, "*Parallel Guessing: A Strategy for High Speed Computation*", PR 20, 2,pp. 257-263, 1987

[25] N. Kiryati, Y. Eldar, and A.M. Bruckstein, A probabilistic Hough transform, PR 24,pp. 303-316, 1991

[26] L. Xu, E. Oja, and P. Kultanen, "*A new curve detection method: Randomized Hough transform*", PRL 11, pp.331-338, 1990

[27] L. Xu and E. Oja, "*Randomized Hough transform (RHT): Basic mechanisms, algorithms, and computational complexities*", IU 57,pp.131-154, 1993

[28] V.F. Leavers, "*The dynamic generalized Hough transform: Its relationship to the probabilistic Hough transforms and an application to the concurrent detection of circles and ellipses*", IU 56,pp. 381-398, 1992

[29] 塩野，"黒点ランダム抽出と重心を用いたハフ変換による円弧の検出実験"，信学論 D-II,J75-D-II, 7,pp.1195-1201,1992

[30] M. Maresca, M. Lavin and H. Li, "*Parallel Hough Transform Algorithms on Polymorphic Torus Architecture*", Multicomputer Vision , pp.9-21, Academic Press, 1988

[31] C. Guerra and S. Hambrusch, "*Parallel algorithms for line detection on a mesh*", JPDC 6, pp.1-19, 1989

[32] R.E. Cypher, J.L.C. Sanz, and L. Snyder, "*The Hough transform has O(N) complexity on N × N mesh connected computers*", SIAM JC 19, pp. 805-820, 1990.

[33] H.A.H. Ibrahim, J.R. Kender, and D.E. Shaw, "*On the Application of Massively Parallel SIMD Tree Machine to Certain Intermediate-Level Vision Tasks*", CVGIP 36, pp. 53-55, 1986

[34] T.J. Olson, L. Bukys and C.M. Brown, "*Low Level Image Analysis on an MIMD Architecture*", Proc. of 1th-ICCV, pp. 468-475, 1987

[35] 青山 正人，浅田 尚紀，松山 隆司，" 再帰トーラス結合アーキテクチャを用いた並列画像解析アルゴリズム (1)- 並列 Hough 変換アルゴリズムとその性能評価-" 情処研報，CV80-27, pp.201-208, 1992

[36] O. Bourdon and G. Medioni, "*Object recognition using geometric hashing on the Connection Machine*", Proc. of ICPR-D, pp.596-600, 1990

[37] I. Rigoutsos and R. Hummel, "*Massively parallel model matching-Geometric hashing on the Connection Machine*", Computer 25, 2, pp.33-42, 1992

[38] A.A. Khokhar, V.K. Prasanna, and H.J. Kim, "*Scalable geometric hashing on MasPar machines*", Proc. of CVPR,pp. 594-595, 1993

[39] P.L. Palmer, M. Petrou, and J. Kittler, "*A Hough transform algorithm with a 2D hypothesis testing kernel*", IU 58,pp. 221-234, 1993

[40] H.M. Lee, J. Kittler, and K.C. Wong, "*Generalised Hough transform in object recognition*", Proc. of ICPR-D, pp.285-289, 1992

[41] P. Liang, "*A new transform for curve detection*", Proc. of ICCV, pp. 748-751, 1990

[42] J. Princen, J. Illingworth, and J. Kittler, "*A hierarchical approach to line extraction based on the Hough transform*", CVGIP 52, pp. 57-77,1990

[43] H. Li, M.A. Lavin, and R.J. LeMaster, "*Fast Hough transform: a hierarchical approach*", CVGIP 36, pp. 139-161, 1986

[44] J. Illingworth and J. Kittler, "*The adaptive Hough transform*", PAMI 9,pp. 690-698, 1987

[45] S. Tsuji and F. Matsumoto,"*Detection of Ellipses by a Modified Hough Transformation*", IEEE Trans. C-27, 8, pp. 777-781, 1978

[46] L. S. Davis and S. Yam, "*A Generalized Hough-like Transformation for Shape Recognition*", Univ. of Texas, Computer Sciences, TR-134, 1980

[47] 輿水，沼田，"区分的 Hough 直線による高速 Hough 変換法 PLHT について"，信学論 D-II,J72-D-II,1, pp.56 - 65,1989

[48] L.D.F. Costa and M.B. Sandler, "*A binary Hough transform and its efficient implementation in a systolic array architecture*", PRL 10, pp. 329-334, 1989.

[49] I.D. Svalbe, "*Natural representations for straight lines and the Hough transform on discrete arrays*", PAMI 11, pp. 941-950, 1989

[50] J.R. Bergen and H. Shvaytser, "*A probabilistic algorithm for computing Hough transforms*", Journal of Algorithms 12,pp. 639-656, 1991

[5]参考文献中の書名の略称

CVGIP:	Computer Vision, Graphics, and Image Processing
IU:	CVGIP:Image Understanding
GMIP:	CVGIP:Graphical Models and Image Processing
ICCV:	International Conference on Computer Vision
ICPR:	International Conference on Pattern Recognition
CVPR:	IEEE Computer Society Conference on Computer Vision and Pattern Recognition
PR:	Pattern Recognition
MVA:	Machine Vision and Applications
PAMI:	IEEE Transactions on Pattern Analysis and Machine Intelligence
PRL:	Pattern Recognition Letters
IVC:	Image and Vision Computing
IUW:	[ARPA] Image Understanding Workshop
JMIV:	Journal of Mathematical Imaging and Vision
JPDC:	Journal of Parallel and Distributed Computing