LI-003

大規模データベースに適用可能なバイオメトリクス検索アルゴリズム

Efficient Identification Algorithm for Large-size Biometric Database

阪田 恒次　前田 卓志　松下 雅仁　笹川 耕一†
Sakata Koji　Maeda Takuji　Matsushita Masahito　Sasakawa Koichi

1. はじめに

2001 年 9 月 11 日にアメリカで発生した同時多発テロを契機として，各国のセキュリティ意識は大きく向上し，バイオメトリクス認証技術も注目されるようになった．特に，空港でのテロリストの発見や，警察における自動指紋識別システム(AFIS)などに代表される不審者・犯罪者のスクリーニングに関する要求が強まっている．

スクリーニングには，データベースに登録された犯罪者と被検者のバイオメトリクスデータを比較する必要があり，登録された犯罪者の数が膨大であればそれだけデータ検索に時間が掛かる．この種の用途ではデータベースの登録者数は膨大でありながら，応答時間は短時間でなければならず，従来技術では実現が難しかった．

バイオメトリクスデータの検索では，データベースを順番に検索していく線形検索か，バイオメトリクスデータを幾つかのクラスに分類し，分類したクラス内で線形検索を行う方法が用いられるのが一般的である［1~3］．線形検索は低速な検索法であり，分類による検索は，各バイオメトリクス（指紋・顔など）に応じた分類法の確立が必要な上，分類を間違った場合に低速な検索法となってしまう．

一方，筆者らが開発した「マトリクス検索法」は，データベースに登録されているデータ間の全ての組合せについて照合度を算出してマトリクス化しておき，検索時にそのマトリクスを利用して次に照合すべき登録データを決定する方法である[4]．この方法は高速な検索が可能なだけでなく，登録データ同士の照合度のみを利用するため，バイオメトリクスの種類を問わない汎用性を持っている．しかし，この検索法は登録データ同士の照合度をマトリクス状に保持するため，登録データ数の二乗に比例したメモリを必要とする．このため，マトリクス検索法を大規模なデータベースに適用することが難しかった．そこで本稿では，照合度マトリクスを分割して保持することでメモリ使用量を低減させ，分割したマトリクスを逐次検索することで，大規模なデータベースにも適用可能な拡張型マトリクス検索法について述べる．2節ではベースとなるマトリクス検索法について概要を説明する．3節でマトリクスを分割して登録・検索するアルゴリズムについて詳細に述べる．また，マトリクスの分割によるメモリ使用量低減の効果について述べ，シミュレーション実験から検索回数への影響について説明する．最後に検索性能分析として，マトリクスの分割数と検索回数の関係をモデル化し，それらの関係を明らかにする．

† 三菱電機株式会社 先端技術総合研究所
〒661-8661 尼崎市塚口本町 8-1-1

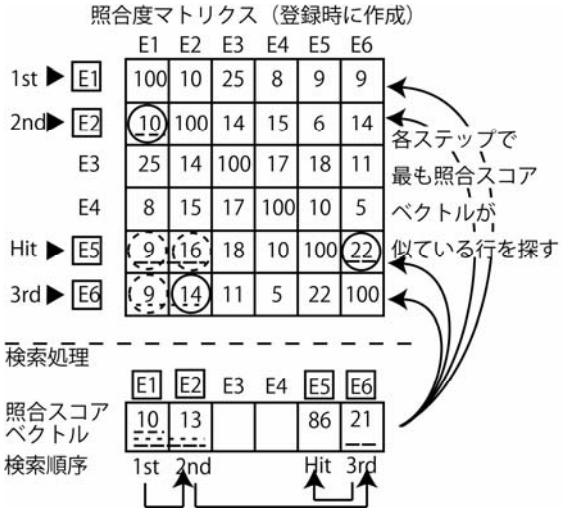

図 1 マトリクス検索法（登録データ E1~E6 の例）

2. マトリクス検索法

マトリクス検索法は，図1のように照合すべき登録データの検索の順序を適切に制御することで，1対1照合の回数を減らし，全体の処理時間の短縮を実現させている．その処理手順の概略を以下に示す．

(ア) 登録時に登録データ間のすべての組合せについて照合度を算出し，事前にマトリクスとして保持する．
(イ) 検索時には上記の照合度マトリクスを利用して照合候補を決める．入力データと今までに照合された登録データとの照合スコアセット（ベクトル）に最も似ている行をマトリクスから探し，それに対応する登録データを照合候補と決定する．そして，選択された登録データと入力データとの１対１照合を行う．
(ウ) 照合度が所定の閾値を超えると，その時点で本人の登録データと一致したとして処理を終了する．照合度が閾値を超えない場合は(イ)へ戻る．

この様にマトリクス検索法では，高速な検索照合を実現するために照合度マトリクスを利用している．このマトリクスは，登録データ数の2乗に比例してそのサイズが大きくなるため，それだけ多くのメモリを消費する．このことがマトリクス検索法を大規模なデータベースに適用する際の障害となっている． 3節ではこのメモリ使用量を低減させ，大規模なデータベースに適用可能な拡張型マトリクス検索法について説明する．

	E1	E2	E3	E4	E5	E6
E1	100	10	25			
E2	10	100	14			
E3	25	14	100			
E4				100	10	5
E5				10	100	22
E6				5	22	100

図 2 分割した照合度マトリクス（2 分割の例）

3. 拡張型マトリクス検索法

2 節で示したように，マトリクス検索法は登録データ数の 2 乗に比例したメモリを使用するため，大規模なデータベースへ適用することが難しい．そこで，照合度マトリクスを複数のマトリクスに分割する(図 2)．拡張型マトリクス検索法では，そのような分割したマトリクスを取り扱う．本節では拡張型マトリクス検索法の登録・検索アルゴリズムを述べ，マトリクスの分割によるメモリ使用量低減の効果と検索回数について説明する．

3.1 登録処理

マトリクスを d 個に分割したとし，現在の登録人数を $n-1$ とする．新規のデータを k 番目の分割マトリクス $\mathbf{X}_k$ に登録データ番号 n として登録する状況を前提とする．

1. 新規登録データを E_n とする．
2. 分割マトリクス $\mathbf{X}_k$ は，

$$\mathbf{X}_k = \begin{pmatrix} x^k_{1,1} & x^k_{1,2} & \cdots & x^k_{1,n_k} \\ x^k_{2,1} & x^k_{2,2} & \cdots & x^k_{2,n_k} \\ \vdots & \vdots & \ddots & \vdots \\ x^k_{n_k,1} & x^k_{n_k,2} & \cdots & x^k_{n_k,n_k} \end{pmatrix}$$

と表され，$\mathbf{X}_k$ に既に登録されている n_k 個のデータ E^k_j と照合し，その照合度を求める．

$$x_{n,j} = f(E_n, E^k_j), (j = 1,2,\cdots,n_k) \qquad (3.1)$$

3. マトリクス $\mathbf{X}_k$ は，(3.1)式と E_n 同士の照合度 x_{nn} から次式で更新される．

$$x^k_{n_k+1,j} = x^k_{j,n_k+1} = x_{n,j}, (j = 1,2,\cdots,n_k)$$
$$x^k_{n_k+1,n_k+1} = x_{nn} = f(E_n, E_n)$$

4. n , n_k の数を一つ増やし，k を更新してステップ 1 へ．

$$k = \begin{cases} 1, (k = d) \\ k+1, (k < d) \end{cases}$$

3.2 検索処理

各分割マトリクス $\mathbf{X}_k$ を $k = 1,2,\cdots,d$ の順番で検索する．検索カウンタ m_k とする． $\mathbf{X}_k$ における m_k 回目の検索の照合候補を $r_k(m_k)$ とし，初回 $m_k = 1$ の時は，各 $\mathbf{X}_k$ の登録データの先頭を照合候補とする $r_k(1) = 1$．

1. 未知の入力データ V_u と現在の照合候補データ $E^k_{r_k(m_k)}$ との照合度 $y_{u,r_k(m_k)}$ を次式で求める．

$$y_{u,r_k(m_k)} = f(V_u, E^k_{r_k(m_k)})$$

2. もし，$y_{u,r_k(m_k)}$ が閾値 T 以上なら，未知のユーザ u は $r_k(m_k)$ で示されるユーザであると判断される $(u = r_k(m_k))$．この場合は，検索処理を終了する．照合度が閾値未満であれば次のステップに進む．
3. もし，$m = \sum_k m_k$ が検索打切回数 D に達していれば，V_u は非登録という結果で検索処理を終了する． m が D 未満であれば，次の照合候補を決定する．
4. 次の照合候補を決定するために，(3.2)式で定義される現在の m_k 回目の照合までの照合度から構成されるベクトル $\mathbf{y}^k_u(m_k)$ と(3.3)式で定義されるマトリクスの i 行目部分ベクトル $\mathbf{x}^k_i(m_k)$ との相関値 $z^k_i(m_k)$ を，既に照合した i を除く全ての i について(3.4)式で求める．

$$\mathbf{y}^k_u(m_k) = (y_{u,r_k(1)}, \cdots, y_{u,r_k(m_k)}) \qquad (3.2)$$

$$\mathbf{x}^k_i(m_k) = (x_{i,r_k(1)}, \cdots, x_{i,r_k(m_k)}) \qquad (3.3)$$

$$z^k_i(m_k) = \frac{2\mathbf{x}^k_i(m_k)\cdot\mathbf{y}^k_u(m_k)}{\mathbf{x}^k_i(m_k)\cdot\mathbf{x}^k_i(m_k) + \mathbf{y}^k_u(m_k)\cdot\mathbf{y}^k_u(m_k)} \qquad (3.4)$$

5. $z^k_i(m_k)$ が最大となる i を求め， $i_{\max}$ とする．
6. $m_k = m_k + 1$ とし， $i_{\max}$ を分割マトリクス $\mathbf{X}_k$ の次の照合候補とする($r_k(m_k) = i_{\max}$).
7. k を更新して，ステップ 1 に戻る．

$$k = \begin{cases} 1, (k = d) \\ k+1, (k < d) \end{cases}$$

(3.4)式の $z^k_i(m_k)$ で次の照合候補を決定するが，この計算に時間が掛かると検索応答時間が遅くなってしまう．しかし $z^k_i(m_k)$ の計算には， $z^k_i(m_k - 1)$ の結果を用いるため，実質， $(x^k_i(m_k))^2, (y^k_u(m_k))^2, 2x^k_i(m_k)\cdot y^k_u(m_k)$ の積演算 4 回と，(3.4)式の和演算 3 回，商演算 1 回であり，データ照合 $f()$ の計算量に比べて無視できるほど微小である．

3.3 メモリ使用量と平均検索回数

上記の拡張型マトリクス検索法を適用した場合のメモリ使用量と平均検索回数について述べる．

まず，照合度マトリクスに使用されるメモリについて考える．照合度 1 つにつき 1byte のメモリが必要とすると，マトリクスサイズ n (マトリクスに登録された人数)におけるマトリクス検索法のメモリ使用量 $M(n)$ は，

$$M(n) = n^2 \text{ [byte]} \qquad (3.5)$$

となる．次に，全ての分割マトリクスのサイズが同じであると仮定すると，任意の $k = 1,2,\cdots,d$ に対して，

$$n = d \cdot n_k$$

となり， k 番目の分割マトリクスのメモリ使用量 O_k は，

$$O_k = {n_k}^2 = (n/d)^2 \text{ [byte]}$$

となる．拡張型マトリクス検索のメモリ使用量 $O(n,d)$ は，

$$O(n,d) = d \cdot O_k = n^2/d = M(n)/d \text{ [byte]} \qquad (3.6)$$

となり，マトリクス検索法に比べてメモリ使用量を $1/d$ に低減できることが分かる．例えば $n = 10000$ の場合に，$d = 10$ とすると，メモリ使用量は 100Mbyte から 10Mbyte に低減される．

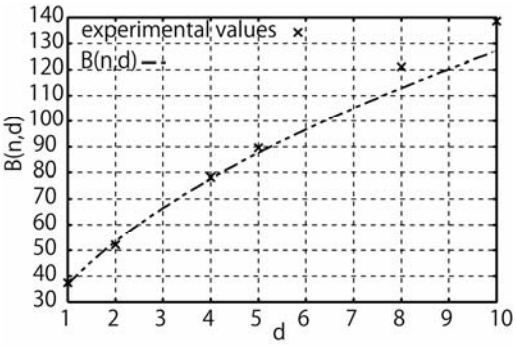

図 3 平均検索回数の実験値とモデル値

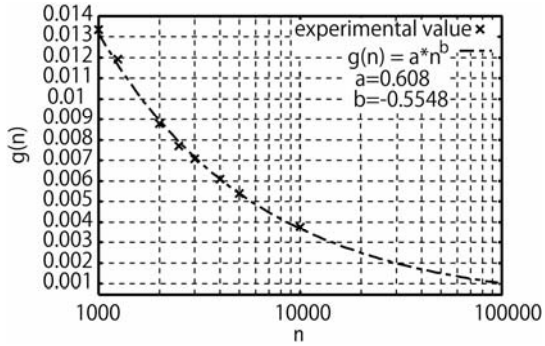

図 4 照合度マトリクスサイズと検索効率

次にマトリクスの分割数 d が検索回数に与える影響を調べるために，参考文献[5]に基づいて生成した指紋データを用いてシミュレーション実験を行った． $n=10000$ とし，d を 1～10 に変化させた結果を図 3 に×印で示す．図を見ると平均検索回数は分割数を増やすことで増加していることが分かる．検索時間は，例えば Pentium4 3GHz の PC を用いた場合，データの 1 対 1 照合に掛かる時間が 0.5msec とすると， $d=10$ の場合でも平均検索回数は約 140 回なので，約 70msec となる．

3 節では拡張型マトリクス検索法について説明した．また，マトリクスの分割によるメモリ使用量低減の効果と，実験から平均検索回数への影響を述べた．次節では検索回数についての性能分析を行う．

4. 拡張型マトリクス検索法の検索性能分析

本節では検索システムの設計に利用するため，マトリクスの分割数と検索回数の関係を分析する．分割によるメモリの低減と検索回数の増加というトレードオフの関係をモデル化し，目標とする応答時間やメモリ使用量の制限から，分割数 d や対応できる登録人数 n を推定できるようにする．また検索打ち切り回数を決定するために，検索回数分布をモデル化する．

4.1 平均検索回数

検索性能を平均検索回数で考える．マトリクス検索法の検索効率を，マトリクスサイズ n の平均何%分検索すると本人が見つかるか，で定義する．マトリクス検索法において， n を 1000～10000 まで変化させた時のシミュレーション結果から検索効率 $g(n)$ を求め，図 4×印で示す．またシミュレーション結果から検索効率は，

$$g(n)=a\cdot n^{b}, a=0.608, b=-0.5548 \qquad (4.1)$$

というモデルで近似できる．図 4 に $g(n)$ のモデルを示す．

マトリクス検索法の平均検索回数 $A(n)$ は，(4.1)式から，

$$A(n)=g(n)\cdot n=a\cdot n^{b+1} \qquad (4.2)$$

と計算できる．

次に拡張型マトリクス検索法における分割数 d と平均検索回数 $B(n,d)$ の関係について述べる． (4.2)式を利用して，$B(n,d)$ は，

$$B(n,d)=A(\frac{n}{d})\cdot d-\frac{1}{d}\sum_{k=0}^{d-1}k=d^{-b}\cdot A(n)-\frac{1}{2}(d-1) \qquad (4.3)$$

と計算できる．分割マトリクス $\mathbf{X}_k$ で入力データに対応する登録データが見つかった場合，残りの分割マトリクス

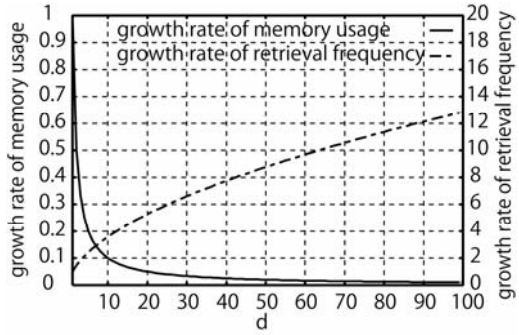

図 5 分割数とメモリ使用量・検索回数の増加率

の検索は不要である．第 2 項はその不要分の平均値を引いている．3 節図 3 に， $n=10000$ における分割数 d と平均検索回数のモデル $B(n,d)$ を破線で示している．(4.3)式を用いて分割による平均検索回数の増加率を計算すると以下のようになり，

$$\frac{B(n,d)}{B(n,1)}=d^{-b}-\frac{d-1}{2a\cdot n^{b+1}}<d^{-b} \qquad (4.4)$$

(3.6)式と(4.4)式から分割数とメモリ使用量および検索回数の増加率は図 5 の様にトレードオフの関係となる．

例えば $n=100000$ の場合，マトリクス検索法のメモリ使用量 $M(n)$ は(3.5)式から 10Gbyte で，検索回数は(4.2)式から約 100 回となる．1 対 1 照合に約 0.5msec 必要とすると，検索時間は約 50msec となる．この場合，仮に 100 分割にすると，メモリ使用量は 1/100 の 100Mbyte で，検索回数は約 13 倍の 1300 回となる．この場合でも検索時間は約 0.65sec となる．このように分割数とメモリ使用量および検索回数の関係が明らかになったので，目標応答時間やメモリの制限から，分割数や対応できる登録人数が推定できる．

ここで，平均検索回数について指紋の分類に基づく検索法との比較を行う．指紋の分類に誤りがないと仮定すると，分類検索法の平均検索回数 $C(n)$ は，

$$C(n)=(\sum_{i}^{c}P(C_i)^2)n/2$$

と計算できる． C_i は i 番目のクラスとし， $P(C_i)$ は C_i の存在する確率， c はクラスの数を表すとする．分類後は線形検索を用いるとする．指紋を Arch($P(C_1)=3.7\%$)， Tented Arch($P(C_2)=2.9\%$ Left Loop($P(C_3)=33.8\%$)， Right Loop($P(C_4)=31.7\%$)， Whorl($P(C_5)=27.9\%$)と 5

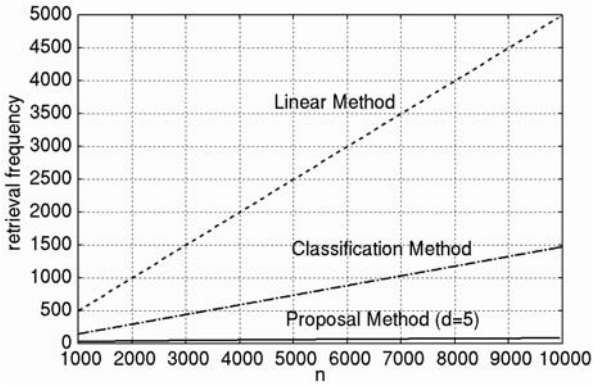

図 6 平均検索回数における他の検索法との比較

分類できるとする[6]．このときの分類検索法，分割数 $d=5$ の拡張型マトリクス検索法，および線形検索法における平均検索回数を図 6 に示す．図から，他の検索方法に比べて検索回数が少なく，短い応答時間で検索できることが分かる．従って，施設における犯罪者監視等，リアルタイム性が求められる用途に適していると考えられる．

4.2 検索回数分布

最後に検索回数分布をモデル化する．検索回数分布をモデル化することで，設定した検索打切り回数 D において，本人を検索できる確率を知ることができる．

マトリクス検索法において，マトリクスサイズ n における検索回数分布 $S_n(u)$ は対数正規分布で近似できる[7]．ある n の時，照合度マトリクスを d 分割すると，

$$S_n(u,d)=\frac{1}{\sqrt{2\pi}\sigma_n(d)}\frac{1}{u}e^{-\frac{1}{2}\left(\frac{\log(u)-\mu_n(d)}{\sigma_n(d)}\right)^2} \quad (4.5)$$

で検索回数が表せると仮定する．$\sigma_n(d)$ ・ $\mu_n(d)$ は分割数 d の場合の平均と標準偏差パラメータである．$n=10000$ としてシミュレーション実験を行い，最小二乗法で $\sigma_n(d)$ ・ $\mu_n(d)$ を求めた結果を図 7 に示し，図 8 に実験値とモデルの検索回数分布を幾つか示す．検索回数分布モデルは，図 7 で求めたパラメータモデルを用いることでシミュレーション結果を良く近似している．

このとき検索打切り回数 D まで検索した場合に，本人が見つかる確率 $P(D)$ は(4.5)式の累積，

$$P(D)=\sum_{u=1}^{D}S_n(u,d) \quad (4.6)$$

となるので，(4.6)式から打切り回数 D が適切かどうか判断することができる．

5. おわりに

本稿では，照合度マトリクスを複数に分割することで，大規模なバイオメトリクスデータベースを検索できる拡張型マトリクス検索法について述べ，マトリクスの分割によるメモリ使用量低減の効果，および検索回数への影響を分析した．その結果マトリクスを d 分割した場合，メモリ使用量は $1/d$ に低減され，検索回数は d に応じて増加し，それらがトレードオフの関係にあることを明らかにした．分割数とメモリ使用量および検索回数の関係をモデル化した結果，目標とする応答性能やメモリ使用量の制限から，分割数や対応できる登録人数を推定できるようになった．

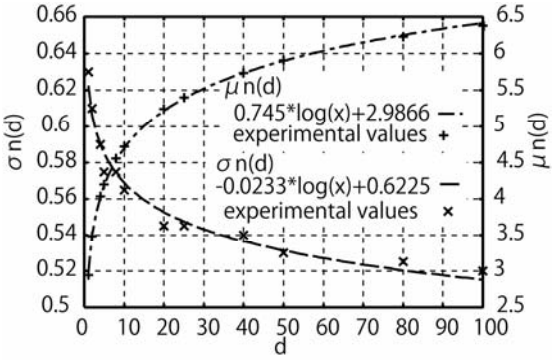

図 7 分割数と対数正規分布パラメータの関係

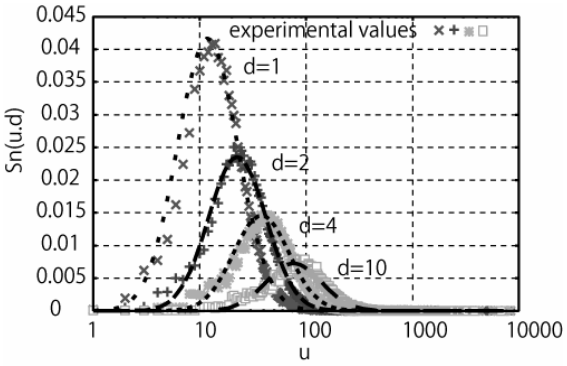

図 8 検索回数分布シミュレーション

また，検索回数分布のモデル化により適切な検索打ち切り回数を決めることができるようになった．

今後の課題としては，より効率的な検索を実現するために，照合度マトリクスの分割方法や分割したマトリクスの検索順序の制御方法等が挙げられる．

参考文献

[1] Dario Maio, and David Maltoni, “A Structural Approach to Fingerprint Classification”, Proc. of ICPR’96, pp.578-585 (1996)

[2] A.K. Jain, S. Prabhakar, and L. Hong “A Multichannel Approach to Fingerprint Classification”, IEEE Trans. Pattern Analysis and Machine Intelligence, vol.21, no.4, pp.348-359 (1999)

[3] Sen Wang, Wei Wei Zhang, and Yang Sheng Wang, “Fingerprint Classification by Directional Fields”, Proc. of the IEEE 4th International Conference on Multimodal Interface, pp.395-398 (2002)

[4] T. Maeda, M. Matsushita, and K. Sasakawa, “Identification Algorithm Using a Matching Score Matrix”, IEICE Transactions on Information and Systems, no.7, pp.819-824 (2001)

[5] R. Cappelli, A. Erol, D. Maio and D. Maltoni, “Synthetic Fingerprint-image Generation”, in proceedings 15th International Conference on Pattern Recognition (ICPR2000), Barcelona,vol.3, pp.475-478

[6] Davide Maltoni, Dario Maio, Anil K.Jain and Salil Prabhakar, “Handbook of Fingerprint Recognition”, Springer, ISBN 0-387-95431-7, pp.176

[7] 前田卓志, 松下雅仁, 笹川耕一, “バイオメトリックス検索照合システムの性能モデル”，電子情報通信学会論文誌, Vol.J87-D-I, No.6, pp.712-720 (2004)