点パターン検索のためのアルゴリズムの提案

松田 考史† 中村 成道‡ 山口 一章† 増田 澄男†
†神戸大学大学院工学研究科電気電子工学専攻 ‡神戸大学工学部電気電子工学科

1 はじめに

点パターンマッチングは二次元平面上の二つの点パターンとある実数値が許容誤差として与えられたとき，一方の点パターンに拡大・縮小，回転，平行移動からなる幾何変換を施し，最も多くの点が重なるような変換を求めるというものであり，指紋照合などに応用を持つ．また，その解法の高速化のため様々な研究が行われている [1]，[2]，[3].

本稿では [3] の方法に高速化の工夫を施す．実験において質問点パターンに最も良く重なるものを複数の点パターンの中から探しだす問題に対する結果を示す．

2 関連研究

2.1 相似三角形法

点パターンマッチングに相似三角形を用いた方法がある [2]．まず点パターン Q，P のそれぞれについてすべての３点の組み合わせに対し三角形を作成する．そして互いに相似な三角形の対応する辺同士はほぼ同じ幾何変換で重なるので，そのような辺の対応には同じラベルを付ける．同一のラベルを多く付けられた辺の対応から順に幾何変換を試し，マッチする点の数を調べることで，重ね合わせを行う回数を少なくしている．

2.2 確率アルゴリズム

文献 [3] では，2.1 の手法に確率アルゴリズムの手法を採り入れることで，より高速なアルゴリズムを実現している．

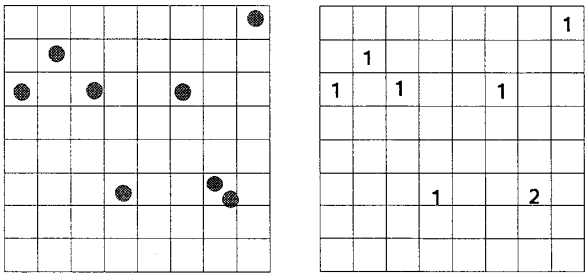
一方の点パターンをいくつかの部分集合に分割し，部分集合間をまたぐような３点については三角形を作成しない．そうすることで重ね合わせを行う回数が削減され処理が高速化できる．しかし，よりよい解を見逃す可能性もあるので見逃さない確率（成功確率）を一定以上保証するための工夫を行っている．

まず成功確率を th まで保証するとする．第一段階では点集合 P を点の数の少ない部分集合に分割する．その結果から成功確率が th 以上保証できれば処理を終了する．そうでなければ第二段階の処理として th 以上の成功確率を保証できるような分割をして再度マッチングを行う．その解が最初の解よりもよければ解を更新する．

3 提案手法

点パターンマッチングでは幾何変換をどのように求めても点の重ね合わせの処理は必要である．そこで本稿では重ね合わせを高速化する方法を提案する．

まず質問点パターン Q の点を格子状の枠で囲み，枠内にある点の数を記録する．この枠をセルと呼ぶ (図1)．重ね合わせを行う際にはセルを参照しながら行うことで高速化を図る．



(a) 格子状の枠で囲む (b) 枠内の点の数を管理

図 1: Q からセルを構成

セルの構成の手順は以下の通りである．

（１）点集合 Q の点のうち最大，最小の x，y 座標値 x_{max}，x_{min}，y_{max}，y_{min} を得る．

（２）一辺の長さが $l_{frame} = max(x_{max}-x_{min}, y_{max}-y_{min})$ で表される正方形の枠で点集合 Q を囲む．

（３）正方形の枠をある正の整数値 num_c で割って，一辺の長さ $l_{cell} = \frac{l_{frame}}{num_c}$ の小さな正方形郡（セル）を構成する．

（４）各セルの中に Q の点がいくつあるかをカウントし，その数を n_q として記録する．

次に，ある幾何変換により重なった点の数をどのように数えるかを説明する．ここで重なった点の暫定的な数を $match$，現在の最良のマッチ数を $best$ とする．

（１）$match = 0$，$P' = P$ で初期化する．

（２）$p_a \in P'$ である p_a を幾何変換した点を p'_a とする．

A proposal of point pattern retrieval algorithm
†Koji MATSUDA ‡Masayuki NAKAMURA †Kazuaki YAMAGUCHI †Sumio MASUDA
†Graduate School of Engineering, Kobe University
‡School of Engineering, Kobe University

（3）p'_a の座標 (x_a, y_a) から x 方向のインデックス i_x，y 方向のインデックス i_y を以下の式で導く．

$$i_x = \frac{x_a}{l_{cell}} \quad i_y = \frac{y_a}{l_{cell}} \tag{1}$$

（4）i_x，i_y に該当するセル内の点の数 $n_q(i_x, i_y)$ と，すでにそのセルに入った P の点の数 $n_p(i_x, i_y)$ を比較する．$n_p(i_x, i_y) < n_q(i_x, i_y)$ であれば $match$，$n_p(i_x, i_y)$ をインクリメントする．

（5）$P' = P' - \{p_a\}$ として $P' \neq \phi$ かつ $|P'| + match > best$ であれば（2）へ戻る．そうでなければ（6）へ移る．

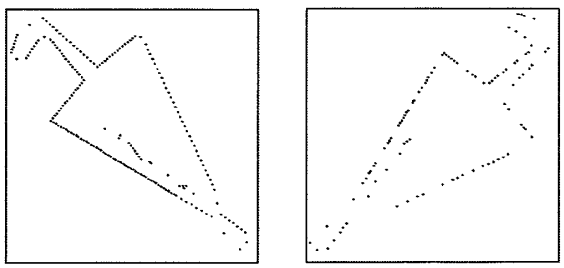
（6）$match > best$ であれば現在の幾何変換を最良の解として記録し，$best = match$ とする．そうでなければ解の更新は行わない．

すべての幾何変換に上記の（1）から（6）を行い，最も多くの点が重なる幾何変換を求める．

4 評価と考察

提案手法の評価のため計算機による比較実験を行った．プログラミング言語は Java 5.0，OS は Linux 2.6，CPU は Athlon(tm) 64 Processor 5000+を用いた．

実験では質問点パターンに最も多くの点が重なるものを複数の点パターンの中から見つける問題を扱う．20 種類の画像を元に要素数 200 の点パターン $P_1, P_2, \cdots, P_{20}$ を作成する．次に $P_1, P_2, \cdots, P_{10}$ のそれぞれについてランダムに 100 個の点を抽出し，平行移動，回転の処理を施したものを質問点パターン $Q_1, Q_2, \cdots, Q_{10}$ とする (図 2)．従来手法として文献 [3] の手法を用いる．従来手法，提案手法の双方とも 2.2 の確率アルゴリズムを採用し，成功確率 th を 0.8 とした．



P の一例　　Q の一例

図 2: 入力データの例

それぞれの質問点パターンに対して 100 回実験を行い，その元になった P の順位と計算時間を調べた．

順位はすべての試行において質問点パターンの元になった P が一位になるという結果が得られた．実行時間の比較を表 1 に示す．確率アルゴリズムには第一段階の処理と第二段階の処理があるが，実行時間は合計の時間である．表 1 から高速化されていることが分かる．

表 1: 実行時間の比較

質問点パターン	従来手法 [sec]	提案手法 [sec]
Q_1	13.43	6.21
Q_2	9.69	6.16
Q_3	11.60	5.73
Q_4	17.68	9.49
Q_5	53.19	10.12
Q_6	16.33	4.46
Q_7	24.75	7.86
Q_8	16.83	6.04
Q_9	35.31	7.20
Q_{10}	25.46	6.39

5 まとめ

本稿では質問点パターンからセルを構成し，参照しながら重ね合わせを行うことで処理を高速化した．セルによる高速化は多数のパターンから検索する場合は顕著に表れた．

しかしセルで点の位置を把握することで，正確に重ね合わせを行う場合に比べて重なる点の数が低下するという問題がある．これは，セルによる点の位置の把握が点を中心とした正方形になっていないためと思われる．したがって今後は幾何変換した点が入ったセルの周囲のセルも参照するなどして重なる点の数が同等になるような工夫が必要である．

参考文献

[1] P.B.V. Wamelen, Z. Li and S.S. Iyengar, "A fast expected time algorithm for the 2-d point pattern matching problem," Pattern Recognition vol.37, no.8, pp.1699–1711(2004).

[2] 鈴木 修人，東海林 健二，"共有辺を持つ相似三角形の最大マッチングによる点パターンマッチング," 電子情報通信学会技術研究報告，PRMU 2000–172, 2001.

[3] 篠原高広，山口一章，増田澄男 "二次元点パターンマッチング問題に対する高速な確率アルゴリズムの提案," 平成 20 年電気関係学会関西支部連合大会,G-10-10,2008.