グリッド・コンピューティングによる画像からの顔検索システムに関する検討

通山　拓馬　　加藤　誠巳
（上智大学理工学部）

1.まえがき

近年、バイオメトリクス認証・ロボットビジョン・エンターテイメント等、顔認識技術の需要が高まっている。それにあわせて、高精度で高速に行える顔認識技術も発達してきている。

本稿では、GA（遺伝的アルゴリズム）を用いたテンプレートマッチングにより顔の位置検出および特徴抽出を行い、分散顔データベースから顔を検索する手法を提案し、高速かつスケーラブルな顔検索システムの構築に関し検討を行った結果について述べる。

2.システムの目的

現在、実用化されている顔認識システムは、バイオメトリクス認証に焦点を絞ったものが多く、企業の入退室管理などに応用されている。そのため、たかだか千人程度での利用しか想定していない。

本システムでは、何万人・何億人というオーダーでの顔認識・顔検索に対応しうるスケーラブルな顔検索システムを目指し、構築を行った。

3.システムの概要

3.1　システム構成

本システムでは、顔画像を与えることにより自動的に顔の特徴を抽出し、その特徴を元に顔の特徴データが格納されているＤＢと比較して高速に検索を行い、個人を特定する。

はじめに、与えられた顔画像を 1 台のＰＣ（マスター）で処理し、顔の特徴を抽出する。抽出する特徴については後述する。

次に、その特徴データを各地に配置された顔の特徴を所持したＰＣ（ワーカー）に伝達し、並列に顔検索を行う。

最後に、最もマッチング率が高かったものを結果としてマスターに提出する。マスターは返された結果の中で最も優秀なものを最終的な結果とする。

システムの構成を図 1 に示す。

3.2　顔認識の流れ

顔認識を行う流れを図 2 に示す。はじめに、GA によるテンプレートマッチングを用いて全体画像の中から目と思われる領域を検出する。その後、その近辺を中心に目、鼻、口、顔の輪郭等の特徴量を抽出する。GA によるテンプレートマッチングについては後述する。

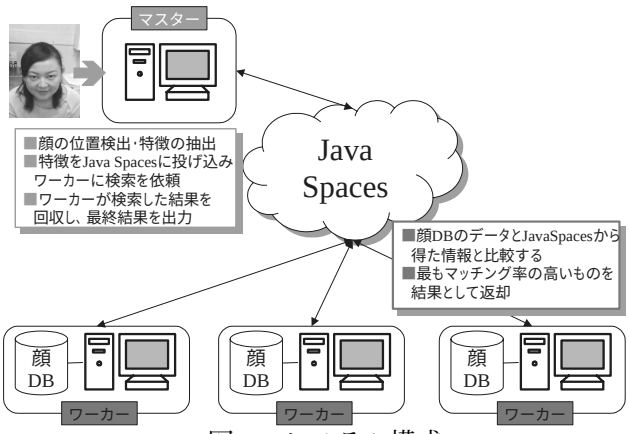


図 1　システム構成

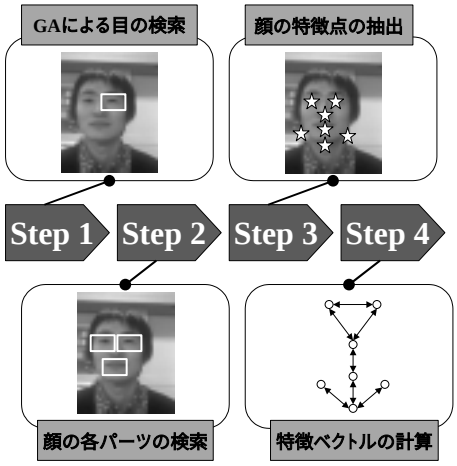


図 2　顔認識の流れ

3.3　GA によるテンプレートマッチング

本システムでは、GA（遺伝的アルゴリズム）を用いて顔位置検出を行う[1]。GA を用いたテンプレートマッチングの概要を図 3 に示す。

これにより全体画像の中から目の位置を検出し、顔の特徴抽出の足掛かりとする。

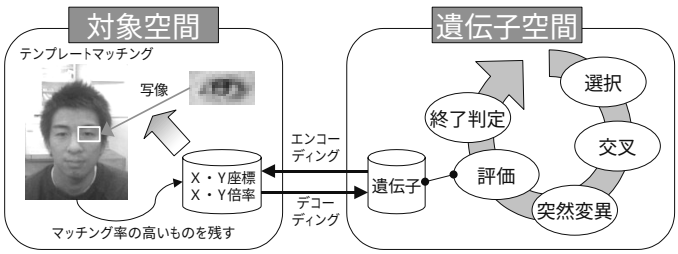


図 3　GA を用いたテンプレートマッチング

A Face Searching System Based on Grid Computing
Takuma TOORIYAMA, Masami KATO
Sophia University

3.4　抽出する顔の特徴点

本システムにおいて、個人を特定するのに用いる顔の特徴量を図 4 に示す。

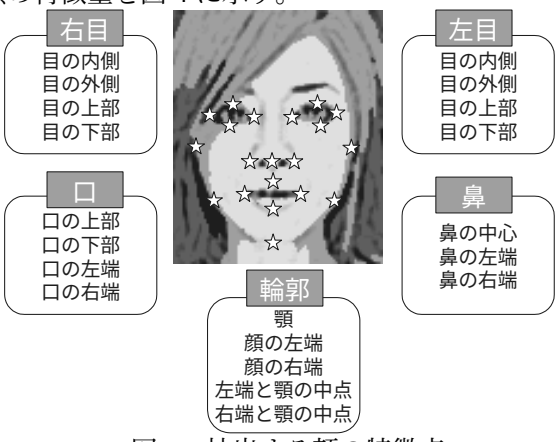


図 4　抽出する顔の特徴点

上図に示す 20 箇所のベクトル（x 成分、y 成分）を用いて、40 次元のベクトル量の比較を行う。なおマッチング率は以下に示す相互相関によって求める。

$$R = \frac{\int_a^b f(x)g(x)dx}{\sqrt{\int_a^b f^2(x)dx}\sqrt{\int_a^b g^2(x)dx}}$$

3.5　JavaSpaces による分散並列処理

1 台の PC で一定時間内に検索できる顔には限界がある。既に実用化されているシステム[3]でも、25 万（フェイス/秒）程度である。仮に、1000 万人の顔と照合を行うとすると 40 秒もかかる計算になる。

そこで、本システムでは JavaSpaces による分散並列処理[2]を提案する。複数台の PC で並列処理を行うことにより高速化を図る。これにより今後、比較する顔の数が増えても PC を増やすことにより対応が可能になる。JavaSpaces による分散並列処理の様子を図 5 に示す。

4.実行例

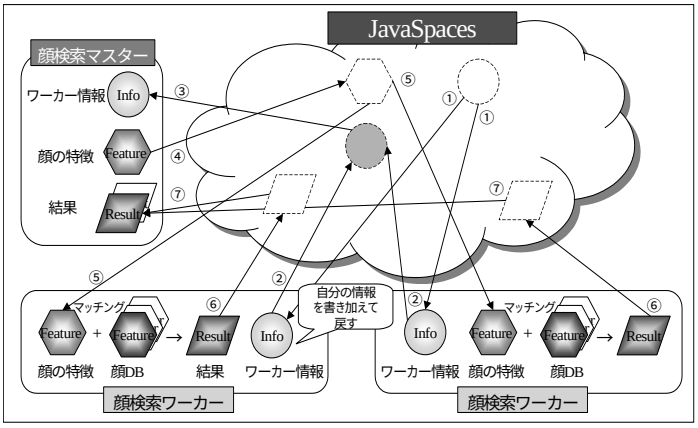
乱数により作成したテストデータを用いてパフォーマンスを測定した。表 1 に 1 台あたりの処理能力の測定結果、図 6 に複数台での分散並列処理のイメージを示す。これより、ワーカーを増やせば、処理時間を一定のまま比較する顔の数を増やせることが分かる。

5.むすび

本稿では、大人数（何万人・何億人）の顔の検索を想定したスケーラブルな顔検索システムについて提案した。

今後は、特徴点の数・マッチング法の妥当性についての検討、ならびに顔特徴抽出の精度の向上・検索の効率化に関する検討を行っていく予定である。

最後に、有益な御討論を戴いた本学 e-LAB/マルチメディア・ラボの諸氏に謝意を表する。



①ワーカー情報を取得
②自分の情報を書き加えて JavaSpaces に戻す
③マスターがワーカー情報を取得する
④検索したい顔の特徴を JavaSpaces に書き込む
⑤ワーカーは、JavaSpaces を監視しており、顔の特徴が入ってくると取得する
⑥持っている顔 DB とマッチングを行い、最もマッチング率の高いものを返す
⑦結果を回収する
⑧検索要求オブジェクトを削除する

図 5　JavaSpaces による分散並列処理

表 1　1 台あたりのパフォーマンス比較

CPU	平均 (FACE/sec)	最大 (FACE/sec)
450MHz	900000	1100000
600MHz	1100000	1300000
1.7GHz	2000000	2300000

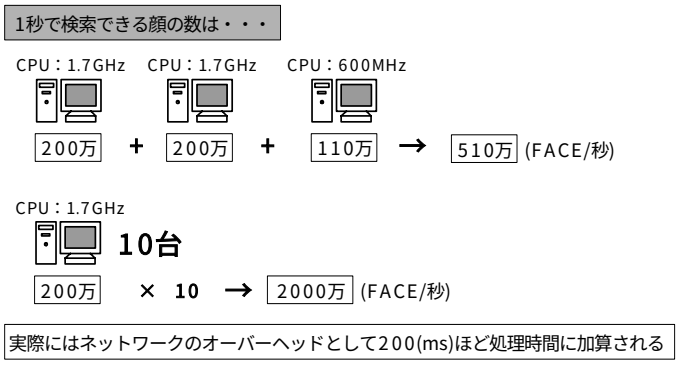


図 6　分散並列処理のイメージ

参考文献

[1]安居院　猛：“ C 言語による画像処理入門, ” 昭晃堂, (2000-11-20).

[2]中山　茂：“ Java 分散オブジェクト入門, ” 技報堂出版, (2000-03-25).

[3]http://ascii24.com/news/inside/2002/08/08/637794-000.html