

マッチンググラフを用いた飾り文字の認識

女川 俊一[†] 大町 真一郎[†] 阿曾 弘具[†]

[†] 東北大学大学院工学研究科
〒 980-8579 仙台市青葉区荒巻字青葉 05
電話番号:022-217-7088, E-mail:mega@aso.ecei.tohoku.ac.jp

我々が普段目にする字体は一樣なものではない。特にカタログ等において企業名・商品名の表示に用いられる飾り文字は、既成のフォントの形状とは異なり、多様である。このような飾り文字に対して、画像にぼかしを与えて得られた濃淡画像から幾何的な特徴を求めて大局構造を表す画像を抽出する手法が提案されている。本論文では、飾り文字の大局構造からストローク・特異点を特徴として抽出し、未知入力の特徴と辞書の特徴の対応関係をマッチンググラフを用いて調べることにより認識する手法を提案する。さらに、様々な飾り文字に対して認識実験を行い、提案手法の有効性を確認する。

文字認識, マッチンググラフ, 飾り文字, 構造解析

Decorated Character Recognition Using Matching Graphs

Shunichi MEGAWA[†], Shin'ichiro OMACHI[†], and Hiroto ASO[†]

[†]Graduate School of Engineering, Tohoku University
Aoba 05, Aramaki, Aoba-ku, Sendai-shi, 980-8579 Japan
TEL:022-217-7088, E-mail:mega@aso.ecei.tohoku.ac.jp

Many kinds of fonts are used in our life. Especially, the fonts which are used for trade name or enterprise name in catalogs are often decorated. A method which extracts global structures from decorated characters has been proposed. In this paper, a new method to recognize decorated characters is proposed. Based on matching graphs, we use exhaustive search of optimal matching between strokes and singular points extracted from input pattern and those extracted from standard patterns. The effectiveness of the new character recognition method is shown by experiments with decorated character images.

character recognition, matching graphs, decorated character, structural analysis

1 まえがき

文字認識システムは多様な文書に対応できることが重要である。我々が普段目にする字体は一樣なものではなく、多様なフォントが存在する。特に、カタログ等の企業名・商品名に用いられる飾り文字は文字本来の目的である“情報伝達”に加え、デザイン性を高めることにより消費者の注意を引くといった特別な目的を持つため、既成のフォントにとらわれない多様な形状を有する。

既存のフォントに対する文字認識手法は、大きくパターン整合法と構造解析法の2種類に分類できる。

パターン整合法は、パターン同士の重なり具合で評価し認識を行なう手法である。入力された未知の文字パターンとあらかじめ作成しておいた標準パターン（以後、辞書と呼ぶ）との重ね合わせを行ない、最も近いものを認識結果とする。パターンの特徴量を用いた整合法として、文字画像から方向線分を抽出して辞書との近さをもとに認識を行う手法 [1] は日本語手書き文字認識において高い認識率を得ている。

構造解析法は、入力文字画像から文字の線分などの構造情報を抽出し、線分の位置関係や接続関係などに着目して識別を行う手法である。構造解析法として、文字パターンの輪郭線を折線近似し文字自体を多角形近似することで線分を抽出し、線分間の対応付けを行う手法 [2] や、文字画像からストロークを抽出しグラフ表現に変換して形状を解析する手法 [3] 等が提案されている。前者は漢字のようなほぼ直線からなる文字に適した手法であり、後者は曲線のストロークも柔軟に扱うことのできるモデルになっている。

これらの手法ではいずれも印刷文字や手書き文字を対象とするため、特徴抽出は文字画像の黒画素の連結性に着目している。しかし、飾り文字の場合、黒画素の連結性に着目しただけではその文字の形状を表す特徴を抽出できるとは限らない。このような飾り文字に対し、複数のパラメータ（スケール）で画像にぼかしを与え尾根や谷等の幾何的な特徴 [4] を抽出し大局構造を求める手法 [5] が提案された。ここで得られる大局構造は同一字種ではほぼ同じトポロジとなる。このことは、構造解析法による飾り文字認識の可能性を示唆している。

本論文では、まず、より大きな変形に対応できるようなマッチンググラフの作成法を提案し、その手法を組み込んだ認識手法を提案する。提案手法により飾り文字の認識実験を行い、提案手法の有効性を示す。

2 提案手法

提案する飾り文字認識手法の概要を述べる。

認識対象となる飾り文字の例を図1に示す。これらの飾り文字に対して、まず文献 [5] で提案された手法で文字

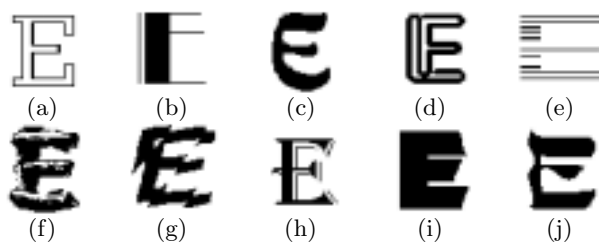


図 1: 飾り文字の例

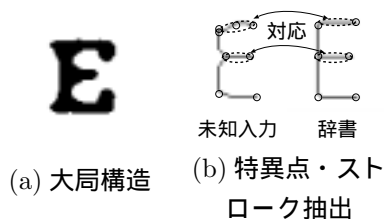


図 2: 大局構造・特徴抽出

の大局構造を抽出する。本論文ではこれを前処理と呼ぶ。図1(a)の画像から構造を抽出した結果を図2(a)に示す。

次に各飾り文字をパターン整合法で大分類を行う。このため、 64×64 に正規化し、次元圧縮と文字画像の変形の吸収のために、隣接する4画素を1つの小領域とし、 32×32 個の小領域を重複しないように設け、各小領域内の黒画素数をその小領域の値とする1024次元の特徴ベクトルを作成する。識別関数としてユークリッド距離を使用し、上位K個の候補を選出する。

最後に、提案するマッチンググラフを用いた手法で細分類を行う。未知入力から得られた画像(図2(a))から線分と特異点を抽出し、マッチンググラフにより、辞書の線分と特異点との対応付けを行う(図2(b))。対応付けをするために未知入力の線分を変形する必要があるが、その変形と整合の度合いがコストとなる。コストの最小のものを候補とする。マッチンググラフの詳細は第4章で説明する。

3 マッチンググラフ作成の準備

本章では、マッチンググラフを作成するための準備について説明する。準備として、適応的正規化、特徴抽出、ブリッジの導入を行う。

3.1 適応的正規化

前処理で得られた大局構造を表す画像は歪みが大きい。そのため、そのまま認識するのは難しい。そこで、提案手法では適応的正規化 [6] を導入する。この正規化は未知入力と標準文字パターンの画像を細線化し、未知入力に対し Global Affine Transformation を行い、次に Local Affine Transformation を行うことで未知入力を標準文字パター



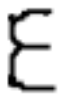
(a) 大局構造



(b) 細線化



(c) 標準文字パターン“E”



(d) “E” に適応的正規化



(e) 標準文字パターン“B”



(f) “B” に適応的正規化

図 3: 適応的正規化

ンに近づくように変形し、歪みの除去を試みる。

図 3 に正規化した画像の例を示す。図 3(a) は前処理により得られた飾り文字の大局構造を表す画像であり、(b) は (a) を細線化したものである。(c)、(e) はそれぞれ文字“E”、“B”の標準文字パターンであり、(d)、(f) はそれぞれ(c)、(e) に適応するように(b)を正規化した後、スムージングの目的で黒画素の膨張処理を施したものである。

3.2 特徴抽出

適応的正規化後の画像を再細線化し、ストローク、特異点を抽出する。特異点としては、

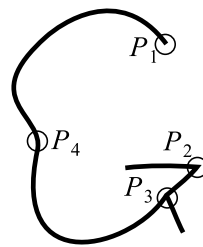
- ストロークの端点 (図 4(a) 点 P_1)
- 鋭角の頂点 (図 4(a) 点 P_2)
- ストロークの交点 (図 4(a) 点 P_3)
- 凹形の頂点 (図 4(a) 点 P_4)

を抽出する。以下では、2 つの特異点を結ぶ線分や曲線を改めてストロークと呼ぶ。具体例を図 4(b) に示す。図 4(b) は図 3(d) の画像を細線化したものである。図 4(b) における f_1, f_2, \dots, f_7 は抽出されたストロークを、 P_1, P_2, \dots, P_8 は特異点を示す。

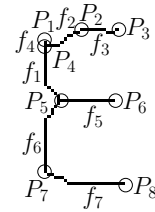
3.3 ブリッジ

ブリッジとはストロークの欠如に対応するために導入する仮のストロークである。

図 5(a) の実線は“A”のある飾り文字より、(b) の実線は“H”のある飾り文字より抽出された大局構造であるが、一部のストロークが欠けてしまい、ストロークの対応が取れず誤認識の原因となる。これに対処するためブリッジ

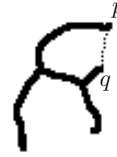


(a) 特異点

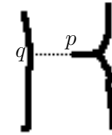


(b) 特異点・ストロークの例

図 4: 特異点・ストロークの抽出



(a)



(b)

図 5: ブリッジ

を導入する (図 5, 点線)。ブリッジは次の条件を満たす 2 点 p, q が存在する場合に p, q 間に導入する。

- 点 p はストロークの端点であり、点 q は黒画素上の点である
- $d(p, z) < d(p, q)$ かつ $d(q, z) < d(p, q)$ を満たす黒画素上の点 z が存在しない

但し $d(p, q)$ は 2 点 p, q 間のユークリッド距離である。

4 マッチンググラフ

本章では、Rocha らの提案した形状分析モデル [3] を基に、飾り文字認識に対応できるマッチンググラフの作成法を提案する。

まず、未知入力の複数の連続するストロークの組をパスと呼ぶ。標準文字パターンに対して、その構造を表すグラフを辞書として用意しておく。

マッチンググラフとは、未知入力のパスの集合を辞書の 1 つの文字の枝集合 E に、パスの端点の集合を節点集合 V に対応させたグラフである。マッチンググラフを作成することで、未知入力と辞書との、パス-ストローク間の対応関係、対応するパス-ストローク間のコスト、特異点の対応関係などを表すことができる。

Rocha らのモデルではマッチンググラフを作成していく際にその時点でのコスト最小のもののみを考慮する。この手法では、比較的変形の小さい文字には対応できるが、飾り文字のような変形の大きい文字に対しては局所解に陥り易い。そこで、提案する手法では複数の候補を残し、最適なグラフ (コスト最小のグラフ) を作成する。

4.1 マッチンググラフの作成

マッチンググラフを作成する手法を説明する。

マッチンググラフを作成するには、最初に初期部分マッチンググラフ (以下単に初期部分グラフ) を作成する。

次に作成した初期部分グラフを拡張したマッチンググラフを作成する。これを再帰的に繰り返すことにより、文字全体を表すマッチンググラフを作成する。

最後に作成された全グラフのコストを比較し、コスト最小のグラフを解とする。

以上を実行するために、4.1.1 節で部分マッチンググラフ (以下単に部分グラフ) を作成する手法、4.1.2 節で部分グラフのパスを延長する手法、4.1.3 節で部分グラフを拡張する手法を説明する。4.1.4 節において、これらの手法を再帰的に適用することにより、候補となる全てのマッチンググラフを作成し、最適なグラフを見つける手法について述べる。

4.1.1 部分グラフの作成

ある特異点の周りのストロークの対応関係を表す部分グラフを作成する手法を説明する。

対応済みの未知入力ストローク f 、辞書のストローク g 、未知入力の特異点 P 、辞書の特異点 b が与えられた場合、その特異点の周囲のストロークの対応関係を示す部分グラフ N_1, N_2, \dots, N_N を作成する。

図 6 に未知入力、辞書の組 (図 6(a)) に対し、対応済みのストローク $f-g$ 、特異点 $P-b$ が与えられた場合の例を示す。

最初に未知入力の特異点 P に隣接するストロークを f から時計周りに探索し、 f_1, f_2, \dots, f_m を見つける。同様に辞書の特異点 b に隣接するストロークを g から時計周りに探索し、 g_1, g_2, \dots, g_n を見つける。次に、この順番を保存するような未知入力と辞書のストロークとの対応付けを全て求める。そして、それぞれの対応付けに関して部分グラフ $N = (\{P-b\}, \{f-g, f_{i_1}-g_{j_1}, \dots, f_{i_{n'}}-g_{j_{n'}}\})$ (n' は m, n の小さい方の値) を作成する。図 6 の例では、 $\{f-g, f_1-g_2, f'-g_3\}$ 、 $\{f-g, f_1-g_1, f'-g_3\}$ 、 $\{f-g, f_1-g_1, f'-g_2\}$ の 3 通りの対応付けを求めることができ、 N_1, N_2, N_3 (図 6(b),(c),(d)) を作成する。

この際、最適な部分グラフを作成する妨げとなるストロークが存在する可能性がある。そこで、あるストロークが次の条件を満たす場合、そのストロークを削除し、そのストロークの両端の特異点を同一の特異点とみなして、同様の対応付けを行った部分グラフも作成する。

- 両端が共に 3 本以上のストロークの交点
- 接しているストロークの中で最短

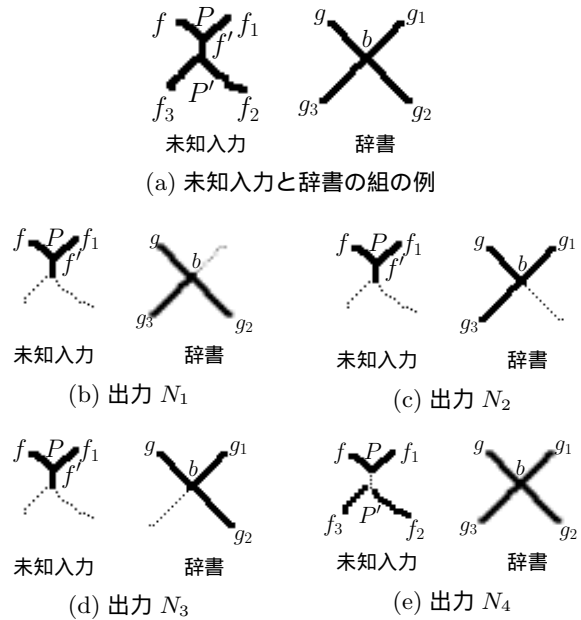


図 6: 部分グラフの作成

図 6 の例では f' は f, f_1, f_2, f_3 より短く、かつ、その両端の点 P, P' がそれぞれ 3 本のストロークの交点となっているので、 f' を削除し、点 P, P' を同一の点とみなして、同様の対応付けを行い、部分グラフ N_4 を作成する。

4.1.2 パスの延長

前節で作成された部分グラフは未知入力と辞書のストロークが 1 本対 1 本で対応 (一対一対応) している。そこで、未知入力と辞書のストロークが複数本対 1 本で対応 (多対一対応) する可能性を考慮して、多対一対応の部分グラフを作成するためにパスを延長する。

一対一対応の部分グラフ N が与えられた場合、多対一対応のグラフ N'_1, N'_2, \dots, N'_N を作成する。

図 7 に未知入力 (図 7(a))、辞書 (図 7(b)) の部分グラフ N_1 (図 7(c), (d)) が与えられた場合の例を示す。

与えられた部分グラフが $N = (\{P-b\}, \{f_{i_0}-g_0, \dots, f_{i_n}-g_n\})$ の場合、まず、辞書のストローク g_0, \dots, g_n に対応するパスとして、 $\alpha_{00} = \{f_{i_0}\}, \dots, \alpha_{n0} = \{f_{i_n}\}$ を作成する。図 7(c)(d) の N_1 の例では、辞書のストローク g_1, g_2, g_3 に対応する未知入力のパスとして、 $\alpha_{00} = \{f_1\}, \alpha_{10} = \{f_6\}, \alpha_{20} = \{f_5\}$ を作成する。

次にこれらのパスをそれぞれ延長する。 α_{j0} を延長する際、まず、 α_{j0} の終点を P_{j0} とおく。 P_{j0} は未知入力のストローク f_{i_j} に隣接し、 P の反対側の特異点である。このパスの終点に隣接するストローク $f'_1, \dots, f'_{n''}$ を探索する。これらのストロークをパス α_{j0} にそれぞれ挿入し、パス $\alpha_{j1} = \{f_{i_j}, f'_1\}, \dots, \alpha_{jn''} = \{f_{i_j}, f'_{n''}\}$ を作成する。このパスの延長を長さが辞書のストローク g_j の 1.5 倍以上になるか、終点に隣接するストロークがなくなるまで再帰的に

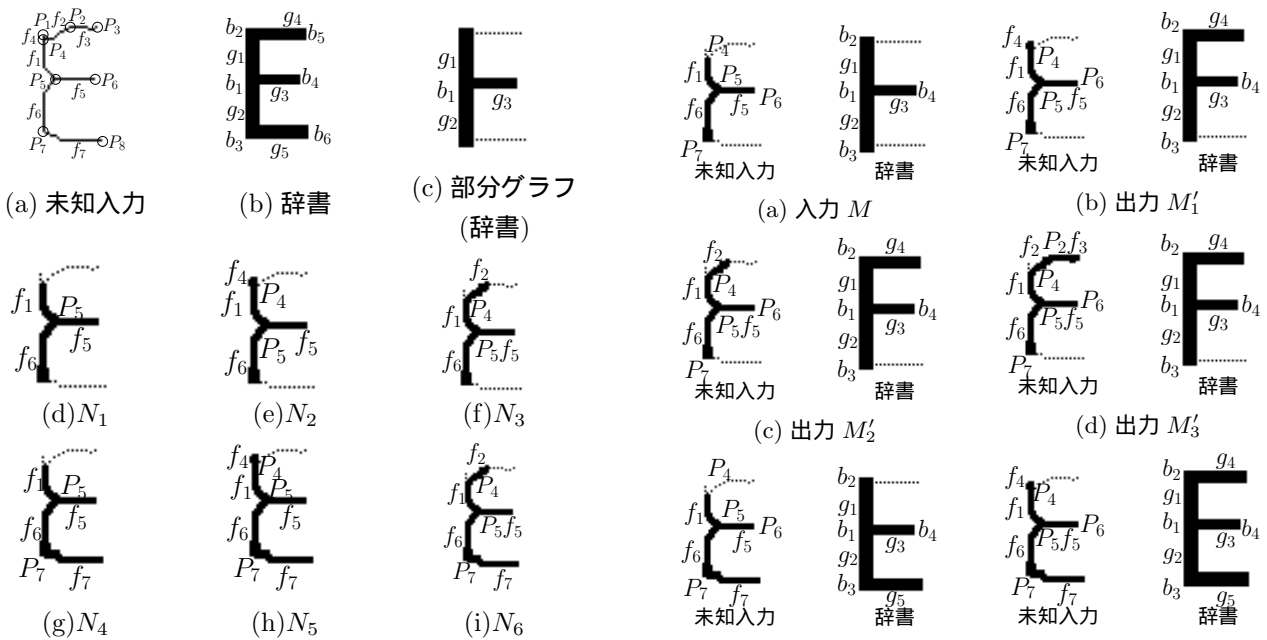


図 7: パスの延長

繰り返す. 図 7(c)(d) の N_1 の例では, パス $\alpha_{00}, \alpha_{10}, \alpha_{20}$ を延長したパスとして, $\alpha_{01} = \{f_1, f_4\}, \alpha_{02} = \{f_1, f_2\}, \alpha_{11} = \{f_6, f_7\}$ を作成する.

最後に作成されたパスの全組合せ (初期組合せは除く) に対して部分グラフを作成する. 図 7(d) の N_1 の例においての全組合せは, $\{\alpha_{01}-\alpha_{10}-\alpha_{20}\}, \{\alpha_{02}-\alpha_{10}-\alpha_{20}\}, \{\alpha_{00}-\alpha_{11}-\alpha_{20}\}, \{\alpha_{01}-\alpha_{11}-\alpha_{20}\}, \{\alpha_{02}-\alpha_{11}-\alpha_{20}\}$ である, それぞれについて部分グラフ N_2, \dots, N_6 (図 7(e), ..., (i)) を作成する.

4.1.3 部分グラフの拡張

部分グラフを拡張する手法について説明する.

部分グラフ M が与えられた場合, 部分グラフ M に含まれる各パスの終点で部分グラフを作成する. それらを M に結合することで部分グラフ M'_1, M'_2, \dots, M'_N を作成する. 図 8 は図 8(a) の部分グラフ $M = (\{P_5-b_1\}, \{\{f_1\}-g_1, \{f_6\}-g_2, \{f_5\}-g_3\})$ が与えられた場合の例である.

部分グラフ $M = (V, \{\alpha_0-g_0, \dots, \alpha_n-g_n\})$ が与えられた場合, それぞれのパス α_i の終点 P_i (対応する辞書の特異点 b_i) で, P_i が未調査の場合, 4.1.1 節にて説明した部分グラフを作成する手法を用いることにより部分グラフ $N_{i1}, N_{i2}, \dots, N_{in'}$ を作成する. 更に, 得られた $N_{i1}, N_{i2}, \dots, N_{in'}$ に対し 4.1.2 節で説明したパスを延長する手法により, 多対一対応の部分グラフ $N'_{i1}, N'_{i2}, \dots, N'_{in'}$ を作成する. 適切な部分グラフを作成することができないパスも存在することを考慮して, $N'_{i0} = (\{P_i-b_i\}, \phi)$ も作成しておく (ϕ は空集合). これを部分グラフ M に含まれる全てのパス $\alpha_0, \dots, \alpha_n$ について実行する. 図 8(a) の部

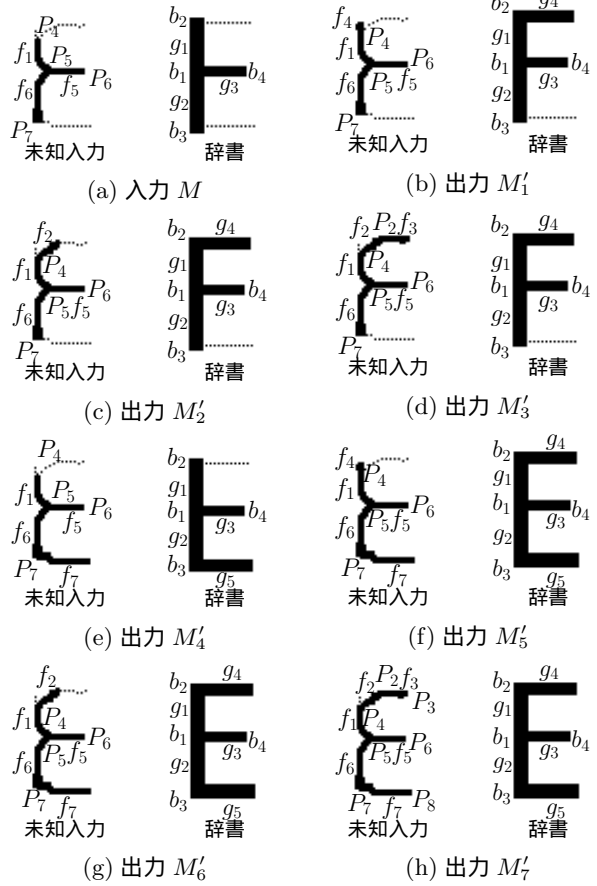


図 8: 部分グラフの拡張

分グラフ M の例では, $\alpha_0 = \{f_1\}$ の終点 P_4 で部分グラフ $N'_{00} = (\{P_4-b_2\}, \phi)$, $N'_{01} = (\{P_4-b_2\}, \{f_1-g_1, f_4-g_4\})$, $N'_{02} = (\{P_4-b_2\}, \{f_1-g_1, f_2-g_4\})$, $N'_{03} = (\{P_4-b_2\}, \{f_1-g_1, \{f_2, f_3\}-g_4\})$ を得る. $\alpha_1 = \{f_6\}$ の終点 P_7 では部分グラフ $N'_{10} = (\{P_7-b_3\}, \phi)$, $N'_{11} = (\{P_7-b_3\}, \{f_6-g_2, f_7-g_5\})$ を得, $\alpha_2 = \{f_5\}$ の終点 P_6 では部分グラフ $N'_{20} = (\{P_6-b_4\}, \phi)$ を得る.

最後に作成された部分グラフの全組合せ (初期組合せは除く) に対して部分グラフ M と結合する. 図 8 の例においての全組合せは $\{N'_{01}-N'_{10}-N'_{20}\}, \{N'_{02}-N'_{10}-N'_{20}\}, \{N'_{03}-N'_{10}-N'_{20}\}, \{N'_{00}-N'_{11}-N'_{20}\}, \{N'_{01}-N'_{11}-N'_{20}\}, \{N'_{02}-N'_{11}-N'_{20}\}, \{N'_{03}-N'_{11}-N'_{20}\}$ である. それぞれについて M と結合し部分グラフ M'_1, \dots, M'_7 (図 8(b), ..., (h)) を作成する.

4.1.4 最適なマッチンググラフの探索

前節までで説明した部分グラフの作成, パスの延長, 部分グラフの拡張の手法を用いて, 未知入力のグラフ及び辞書のグラフから候補となる全てのマッチンググラフを作成して, コスト最小のマッチンググラフを選出する手法について述べる.

最初に, 辞書の任意のストローク g (終点 b) と未知入力

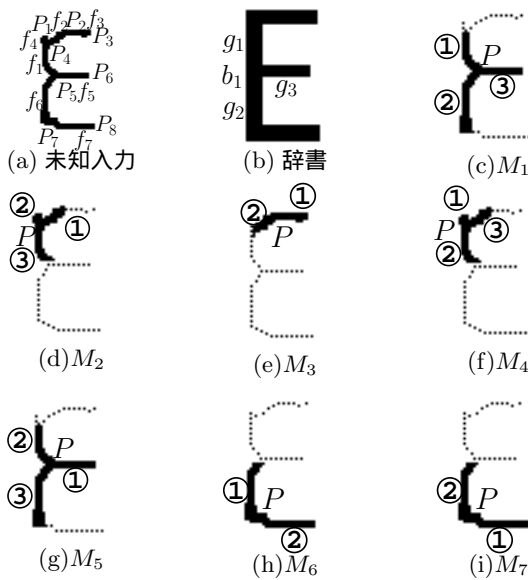


図 9: 初期部分グラフ (一部)

の全てのストローク f_i (終点 P_i) に対して, 4.1.1 節で説明した部分グラフを作成する手法を実行する. 作成された部分グラフに対し 4.1.2 節で説明した部分グラフのパスを延長する手法を実行し, 初期部分グラフ M_1, M_2, \dots, M_{N_1} を作成する. 具体例を図 9 に示す. 図 9(a) は未知入力, (b) は辞書であり, ストローク g_1 (終点 b_1) に対して実行した結果である. 図 9(c), \dots, (i) は初期部分マッチンググラフ M_1, M_2, \dots, M_7 である. 図 9(c), \dots, (i) 各図の丸数字は (b) の辞書と対応するストロークの添字の番号を示している.

次に, M_1, M_2, \dots, M_{N_1} を 4.1.3 節で説明した部分グラフを拡張する手法により拡張し, 部分グラフ $M_{N_1+1}, M_{N_1+2}, \dots, M_{N_2}$ を得る. 更に部分グラフ $M_{N_1+1}, M_{N_1+2}, \dots, M_{N_2}$ に対しても, 新しいグラフが作成されなくなるまで拡張を繰り返す.

最後に全てのマッチンググラフ M_1, M_2, \dots, M_N の中でコスト最小のグラフを解とする. 図 9(a) が未知入力として与えられた場合, 図 8(h) の M_7' が最適な部分グラフとして選出された.

コストの計算方法については次節で述べる.

4.2 コスト計算

マッチンググラフのコストは認識の基準になるように, 次のように定義する.

$$\begin{aligned} cost = & w_1 \sum_i trans(\alpha_i, g_i) + w_2 \sum_i match(\alpha_i, g_i) \\ & + w_3 \sum_i length(b_i) + w_4 \sum_i length(f'_i) \\ & + w_5 \sum_i length(g'_i) + w_6 D \end{aligned}$$

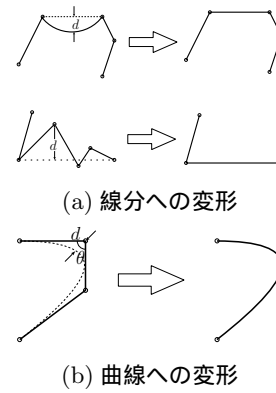


図 10: 変形コスト

上式の第 1 項の $trans(\alpha_i, g_i)$ は未知入力のパス α_i を辞書のストローク g_i に合わせて変形する際の変形コスト, 第 2 項の $match(\alpha_i, g_i)$ は未知入力のパス α_i と辞書のストローク g_i との整合コストを与える. 第 3 項は使用したブリッジ b_i の長さ, 第 4 項は使用されなかった未知入力のストローク f'_i の長さ, 第 5 項は使用されなかった辞書のストローク g'_i の長さである. 第 6 項 D は 3.1 節で説明した適応的正規化にかかるコストである. また, $w_1 \sim w_6$ は重み定数で実験により定める.

変形コスト, 整合コスト, 及び適応的正規化にかかるコストについて詳しく説明する.

4.2.1 変形コスト

未知入力のパスの対応の良さを知るために, 対応する辞書のストロークに合うようにパスを変形し, 変形の度合を示すコストを定める.

図 10(a) は対応する辞書のストロークが線分である未知入力の曲線を持つパス, 折線パスにおける変形を示す. その変形コスト $trans(\alpha_i, g_i)$ はパスの始点と終点を結んだ線分とパスの各点との距離の最大値 d とする. 図 10(b) は対応する辞書のストロークが曲線の場合のパスの変形を示す. その変形コストはパス上の特異点をはさむストロークが成す角を θ とおいて, $d = \cos \frac{\theta}{2}$ と定める. 特異点が複数存在する場合はそのような d の総和を変形コストとする.

4.2.2 整合コスト

整合コストは, 辞書のストロークと未知入力のパスとの一対一対応に対して整合の度合を定めるものである.

未知入力のパスの長さを l_α , 辞書のストロークの長さを l_g , 未知入力と辞書の始点のストロークの水平方向に対する角度をそれぞれ θ_α, θ_g , 未知入力と辞書の終点のストロークの水平方向に対する角度をそれぞれ ϕ_α, ϕ_g とし, 未知入力のパスの始点と終点を結ぶ線分の長さを l_s , その

始点における線分の水平方向に対する角度を ψ_α として、以下のように整合コストを定める。

- 未知入力のパスと辞書のストロークが共に線分の場合

$$l_s^2 + l_g^2 - 2l_s l_g \cos(\psi_\alpha - \theta_g))^{\frac{1}{2}}$$

- 未知入力のパスと辞書のストロークが共に曲線の場合
 $\frac{1}{2}\varepsilon_{arc}(|\theta_\alpha - \theta_g|) + \frac{1}{2}\varepsilon_{arc}(|\phi_\alpha - \phi_g|) + \varepsilon_{st}(|l_\alpha - l_g|)$
ただし、

$$\varepsilon_{arc}(\theta) = \begin{cases} 0 & \text{if } \theta < \theta_0 \\ \theta - 1 & \text{if } \theta \geq \theta_0 \end{cases}$$

$$\varepsilon_{st}(l) = \begin{cases} 0 & \text{if } l < l_g/4 \\ l - l_g/4 & \text{if } l \geq l_g/4 \end{cases}$$

θ_0 は実験で定める。

- パスが曲線、辞書が線分の場合
曲線をその始点・終点を結ぶ線分に変形し、その弦と辞書の線分とのコストとする。変形コストと線分どうしの整合コストの和となる。
- パスが線分、辞書が曲線の場合
曲線に歪みが生じて線分になることは無いという仮定をし ∞ とする。

4.2.3 適応的正規化のコスト

適応的正規化によって、入力文字が異なる字種の標準文字パターンにも近づいてしまうため、入力文字がどの程度変形されたか考慮する必要がある。そこで D を適応的正規化によって各黒画素が移動した距離の総和で定義する。

4.3 マッチンググラフ拡張の限定

4.1 節で説明した手法により最適なマッチンググラフを見つけることができる。しかし、この手法では可能な部分グラフを全て作成してしまうため、明らかに不要な部分グラフも作成してしまう。そこで、本節では 4.1.3 節で説明した部分グラフの拡張を行う際にその拡張を限定する手法を提案する。

余分な部分グラフの拡張を限定するために部分グラフ M のコスト下限を次の式で定義する。

$$\begin{aligned} \text{limit} = & w_1 \sum_i \text{trans}(\alpha_i, g_i) + w_2 \sum_i \text{match}(\alpha_i, g_i) \\ & + w_3 \sum_i \text{length}(b_i) + w_4 \sum_i \text{length}(f_i'') \\ & + w_5 \sum_i \text{length}(g_i'') + w_6 D \end{aligned}$$

上式第 1 項～第 3 項及び第 6 項、 $w_1 \sim w_6$ は 4.2 節におけるコスト計算の式の定義と等しい。第 4 項は部分グラフ M を拡張して作成されるどのグラフでも決して使用されない未知入力のストローク f_i'' の長さ、第 5 項は決して使用されない辞書のストローク g_i'' の長さである。

コスト下限はその部分グラフ M を拡張して作成される部分グラフのコストの最小値以下の値をとる。従って、そのグラフのコスト下限が全部グラフのコストの最小値より大きい場合、その時点でのコストの最小値よりコストの小さなグラフを作成するのは不可能なので、部分グラフの拡張を行わない。

上式第 4 項、第 5 項の計算をするために必要な、拡張後決して使用されないストロークの条件を以下説明する。

部分グラフ M に含まれるパス $\alpha_i = \{f_1, f_2, \dots, f_n\}$ 、それに対応する辞書のストローク g_i について考える。また、パス α_i の始点を P_{si} 、終点を P_{ei} とし、それぞれに辞書の特異点 b_{si}, b_{ei} が対応しているとする。この際、決して使用されないストロークはグラフ M に含まれておらず、

- (1) f_j, f_{j+1} に挟まれる特異点に隣接している未知入力のストローク
- (2) P_{si} と隣接している未知入力のストローク
- (3) b_{si} と隣接している辞書のストローク
- (4) b_{ei} に隣接していて、部分グラフ M に含まれない辞書のストロークが存在しない場合、 P_{ei} に隣接している未知入力のストローク
- (5) P_{ei} に隣接していて部分グラフ M に含まれない未知入力のストロークが存在しない場合、 b_{ei} に隣接している辞書のストローク

及び、それに連結しているストロークである。

上記の条件の理由を説明する。(1)については、そのストロークを使用すると未知入力と辞書とでトポロジの異なるマッチンググラフを作成してしまう。(2)、(3)については、 P_{si}, b_{si} はすでに部分グラフを作成するための調査済みであり拡張後調査されることはない。(4)、(5)については、 P_{ei}, b_{ei} の周りに関する部分グラフを作成する際、一方のストロークが存在しないので部分グラフとして $N = (\{a_{ei-b_{ei}}, \phi\})$ しか作成されない。

図 11(a) のグラフ M_1 の例では未知入力のストローク f_4 が、パス $\{f_1, f_2\}$ に注目すると上記 (1) より、拡張後も決して使用されない。また、辞書のストローク g_5 が、パス $\{f_6, f_7\}$ の終点 P_8 及び辞書の特異点 b_3 に注目すると上記 (5) より、使用されない。図 11(b) のグラフ M_2 の例では未知入力のストローク f_4 が、パス $\{f_2\}$ の始点 P_4 に注目すると上記 (2) より、使用されない。また、未知入力のストローク f_3 が、パス $\{f_1, f_2\}$ の終点 P_2 及び辞書の特異点 b_5 に注目すると上記 (4) より、使用されない。

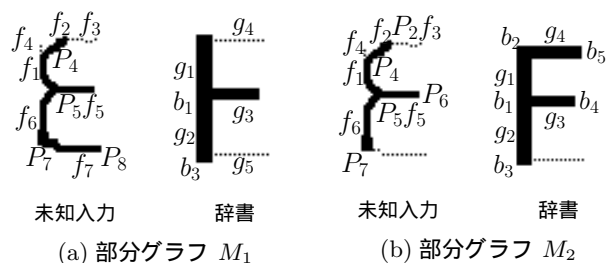


図 11: 部分グラフのコスト下限

5 実験

本手法の有効性を検証する為に、飾り文字を認識対象とする性能評価実験を行う。テストサンプルには、アルファベット大文字 26 文字・各 10 種の飾り文字を用いた (図 1 に一部を示した)。標準文字パターンとしてはアルファベット大文字 26 文字・各 10 種のゴシック体を用いた。また、コスト計算の重み定数を $w_1 = w_2 = w_3 = 1, w_4 = w_5 = 2, w_6 = 1$ とした。

本認識手法において、大分類による絞り込み数を $K = 5$ とした場合、認識率 68.8% の結果を得た。パターン整合法のみで飾り文字を認識した場合の認識率は 57.7% であり、本認識手法により認識率を向上させることができ、有効性が示された。しかし、パターン整合法のための第 5 位累積認識率は 87.7% であり、さらなる向上の余地が残されている。

また、大分類を行わず、最初からマッチンググラフのみを用いて認識した場合の認識率は 63.4% であった。モデルが柔軟に様々な文字画像と整合されるため、概形の異なる文字とも対応してしまい、認識率が低下したものと考えられる。これは、パターン整合法と構造解析法との融合が有効であることを示す証拠でもある。

ここで、提案手法で正しく認識されたマッチンググラフの例を示す。図 12(c) の上段は飾り文字 “E” に対して標準文字パターン “E” を対応させた例。下段は標準文字パターン “B” を対応させた例である。図 12(c) 上段の細破線は使用しなかったストローク、下段の太破線はブリッジを表す。まず上段では、グラフの形状は “E” に近い形状であり、辞書のストロークは全て使用されている。未知入力側に使用されていないストロークはあるもののその長さは短くマッチングコストは小さくなっている。一方、下段はマッチンググラフが文字 “B” と同じトポロジを持つものの、使用されるブリッジの長さが長くマッチングコストが大きくなっている。この飾り文字は “E” と認識された。

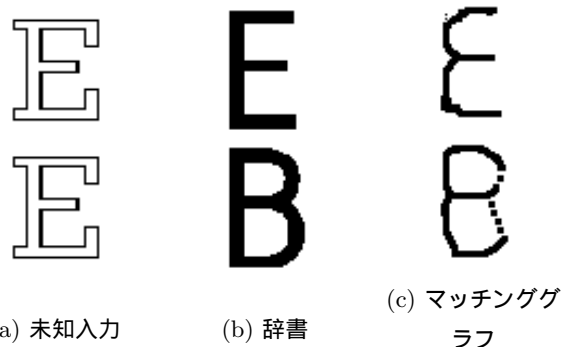


図 12: マッチンググラフの例

6 むすび

本論文では、マッチンググラフを用いた飾り文字の認識手法を提案した。パターン整合法により大分類を行い、適応的正規化を導入し、複数の候補を残しながら分岐限定法により最適なマッチンググラフを作成することで細分類を行った。飾り文字を用いた認識実験を行い、その結果提案手法により 68.8% の認識率が得られ、パターン整合法単独、マッチンググラフ単独の認識実験より高精度であることを示した。

しかし、重みの決め方、同一のトポロジを持つ文字の識別に関してはまだ改良の余地があり、今後の課題である。

参考文献

- [1] 鈴木雅人, 大町真一郎, 加藤寧, 阿曾弘具, 根元義章, “混合マハラノビス関数による高精度な類似文字識別手法,” 信学論 (D-II), vol.J80-D-II, no.10, 1997.
- [2] 山本和彦, “弛緩整合法による手書き教育漢字認識,” 信学論 (D), vol.J65-D, no.9, pp.1167–1174, 1982.
- [3] J. Rocha and T. Pavlidis, “A shape analysis model with application to a character recognition system,” IEEE Trans. PAMI, vol.16, no.4, pp.393–404, 1994.
- [4] R. M. Haralick, L. T. Watson, and T. J. Laffey, “The topographic primal sketch,” International Journal of Robotics Research, vol.2, no.1, pp.50–72, 1983.
- [5] 大町真一郎, 井上政樹, 阿曾弘具, “マルチスケール画像を用いた飾り文字の構造抽出,” 信学技報, PRMU98-30, 1998.
- [6] T. Wakahara and K. Odaka, “Adaptive Normalization of Handwritten Characters Using Global/Local Affine Transformation,” IEEE Trans. PAMI, vol.20, no.12, pp.1332–1341, 1994.