

データ圧縮を利用したオンライン・スケッチ認識手法 OSR

近藤 邦 広[†] 加藤 直 樹[†] 渡 辺 俊 典[†]

発展が期待される柔軟なヒューマンインタフェースの基礎技術開発を目的としてオンラインスケッチ認識問題を取り上げ、新たな方式を1つ提案した。新たなアイデアは、スケッチ筆跡のサンプルデータ列を一群のデータ圧縮器で別々に圧縮し、各圧縮率を並べて定義したベクトルによって入力スケッチの特徴を表現する点である。さらに、LZ77-LZB アルゴリズムを基礎とした圧縮器を利用することで、圧縮器構成の際の事前トレーニングの手間を省くことなく特徴抽出機能を実現できるようにした。特徴ベクトルの検索の際に使用する距離関数を工夫し、スケッチの全体および局所双方の特徴を考慮できる能力を高めた。提案方式の詳細を示すとともに実験によってその有効性を示した。

OSR: Online Sketch Recognition by Data Compression Technique

KUNIHIRO KONDOU,[†] NAOKI KATO[†] and TOSHINORI WATANABE[†]

Aiming the development of flexible user-interface that will be a basic technology for future HI (Human-Interface), online sketch recognition problem is considered and a new solution is given. Sketch stroke stream is filtered by a set of data compressors, giving a feature vector composed of compression ratios. Modified LZ77-LZB is used to avoid the need of compressor pre-training, thereby realizing rapid construction of feature extractors. A new scheme for vector selection that incorporates both local and global features of a sketch is also proposed. Prototype system is detailed and experiment results illustrating the usefulness of this approach are given.

1. ま え が き

計算機の進歩とともにその入力インタフェースは、初期のフロントパネル、バンチカードなどから徐々に進歩し、最近のパーソナルコンピュータではキーボード、マウス、ライトペン、タッチパネルなどが常識となってきた。今後は音声入力やジェスチャー入力などへの発展が予想される。

計算能力や記憶能力の乏しい初期の計算機ではこのように高度な機能の実現は困難であり、入力情報に曖昧さが混入することを避けるために、利用者には強度に制約された入力手段しか提供できなかった。しかしながら、半導体技術進歩に支えられた計算能力や記憶能力の向上が、上記のような多様で便利なインタフェースを実現可能とした。計算機の社会への浸透とともに、柔軟なインタフェースへの要求はますます高まると予想される。

このような背景から、本研究では、現状のインタフェース技術では実現できていない自由手書きスケッチ

認識問題を取り上げ、柔軟なインタフェース実現に向けた新たなアイデアの提案とプロトタイプシステムの実現を試みた。

本研究に関連した問題に文字認識問題がある。周知のように文字認識技術は大きく分けてスキャナーで取り込んだ静止文字データを認識するオフライン方式と、筆跡を実時間で取り込んで認識するオンライン方式とに大別される。いずれもデータ入力、前処理、特徴抽出、特徴量による辞書検索、検索結果の出力機能などから構成される。認識対象も、文字レベルから単語あるいは文章レベルに、定型文字から自由手書き文字やサイン文字などに拡大されてきている^{1),2)}。この内の自由手書き文字やサインのオンライン認識問題は自由手書きスケッチ認識問題と類似した側面を持っているが、後者には、文字のように有限分類できないとか、意味と表現との関係が多様な曖昧さを持っているなどの著しい特徴があり、文字認識向けの特徴抽出機構や認識機構をそのまま利用することはできない。

自由手書きスケッチ認識技術の応用としては、計算機へのコマンド指示、サインによる個人識別、手書き入力図面の自動清書ソフトウェアなどがある。時系列物理信号の意味認識機構であるととらえれば、個人の

[†] 電気通信大学大学院情報システム学研究科
Graduate School of Information System, University of
Electro-Communications

癖を計算機利用時に矯正する必要のない柔軟なジェスチャ対話システムにも応用できる。マルチメディア分野で重要な画像情報の記憶や検索においても、スケッチ画を直接入力して所望の画像を検索する問題に関心が寄せられるようになってきている^{3)~5)}。この場合、事前に特徴抽出して記憶済みの画像データを、事後入力スケッチで検索するが、現状は比較的簡単な自由入力スケッチを受理できるのみであり、入力の自由度を向上させる際に本研究で扱う問題に遭遇する。

本研究では、文字に比べて非定型な手書きスケッチ認識問題を解決する際に、基本的技術課題となるスケッチの特徴表現問題を、データ圧縮技術の利用という新たなアイデアによって解決する。その際、文字認識と異なり、認識機構を利用者自身が作り込む場合が多いことを想定し、数少ないスケッチ事例で認識機構自体を定義できるように実現方式を工夫する。本アイデアに基づくプロトタイプシステムの認識性能は非常に良好であった。以下、2章では問題の定義、技術課題とその解決策、3章でシステム概要、4章で実験結果を報告する。

2. 自由手書きスケッチ認識システム

2.1 問題の定義

(1) オンライン自由手書きスケッチ認識問題

たとえば「木」を表すスケッチ S_{tree} とその対応概念を表す文字列 $C_1 = "Tree"$ とのペア (S_{tree}, C_1) 、同様に「家」を表す (S_{house}, C_2) など、認識させたいと希望する対象について、合計 m 個のペアを利用者が事前に登録しておく。なお、上例では「木」「家」などの概念ごとにペア1個を登録するとしたが、一般には1個の概念に対して複数個のペアを記憶させて構わない。登録を終えた後、新たに事後入力としてスケッチ S^* を与え、それに対応する概念 C^* を解答させる。できるだけ少ない登録データで多様な S^* に対し正しい C^* を解答することが望ましい。

(2) 入力方法

スケッチはタブレット上にマウスや電子ペンで描画する。その筆跡を座標点の時系列サンプリングデータとしてオンラインで計算機に取り込む。スケッチに対応する概念はキーボードからテキスト形式で入力する。

(3) 出力方法

事後入力スケッチ S^* の対応概念 C^* をテキスト形式でディスプレイ装置に表示する。

2.2 技術的課題

スケッチ S_1, \dots, S_m の特徴を抽出し、上記のペアデータの形式で事前登録しておく、事後入力 S^* によ

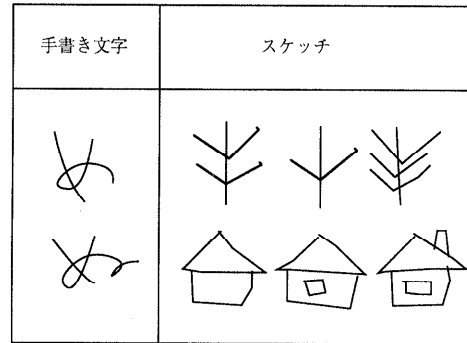


図1 手書き文字とスケッチとの差異

Fig.1 Handwritten characters vs. sketches.

る類似検索を行えば認識問題を解決できる。以下に、その際の課題を文字認識とスケッチ認識の場合を比較しつつ説明する。

(1) 対象概念の多様性

文字の場合にはもともと有限個の記号だけが許されていて多様性は少ないがスケッチでは何を描いてもよいため無制限である(図1)。このため、文字認識では個々の文字の特徴と文字との関係を認識辞書として事前作成しておき、事後入力文字の特徴と認識辞書項目とをマッチングさせる方法が有効である。スケッチの中でも、アイコンのように表現が限られた領域では同様の手法が利用できるが⁶⁾、一般のスケッチでは無限の対象を効果的に特徴表現できる新たな手法が必要である。

(2) 表現と概念との対応の曖昧さ

たとえば「め、ぬ」の文字は互いに似てはいるが、対応する概念は異なる。対応概念が異なることを示すために構造的な差異が局部的に設けられており、これに着目すれば認識精度を向上できる。スケッチでは枝が何本あっても「木」であり、「家」は煙突や窓があってもなくても「家」と判断されるなど、異なる構造が同一の意味を持つ。このように位相構造すら一定でないことと上述した表現の無限の多様性から、文字認識で有効な生成文法規則^{7)~9)}も問題解決に利用できない場合が多い。

2.3 課題の解決策

(1) 無限の多様性への対処

無限個の対象を有限個の手がかりで表現するには有限個の基底で張られたベクトル空間を利用するのが有効である。このベクトル空間は一般に特徴空間と呼ばれ、文字認識や画像認識で古くから利用されてきている⁷⁾。本研究でもこの枠組みを使用する。ただし、2.2節の課題を解決できるような基底を工夫しなければならない

い. この枠組みのもとでは, スケッチ S_1, S_2, \dots, S_m はそれぞれ F_1, F_2, \dots, F_m なる特徴ベクトルで表現され, 事後入力スケッチ S^* も同様に特徴ベクトル F^* で表現される. 認識は F^* と距離的に近い F_i を探索することで実現される.

スケッチ認識では, たとえば風景図, 人物図, 設計図といった分野別の特徴があると期待できる. よって, 利用者サイドで分野に固有のベクトル空間基底を与えて自前の特徴空間を簡単に定義できれば, スケッチの多様性への対応性が向上する. 詳細は後述するが, 数点の簡単な分野依存スケッチを空間の基底として直接利用できる方式を導入する.

(2) 表現と概念との対応の曖昧さへの対処

図1の「木」のスケッチでは枝の本数などの位相構造が異なっている。「家」では煙突や窓があつたりなかつたりしている. しかしながら, 「木」では幹を表す縦方向や枝を表す斜め方向あるいはV字型の筆跡が特徴的である. 「家」では屋根を示す閉三角形や壁を示す四角形の筆跡が見られる. このことから, スケッチに含まれる縦, 横, 斜めなどの単純成分やそれらから形成される連成分の出現様相をスケッチの特徴として利用できる可能性に気づく. ただし, スケッチの多様性から, 単純成分や連成分を事前に一般的な辞書として定義し利用することはできない. また, 「木」の枝の個数や, 「家」の煙突の有無の例に見られるように成分の出現様相も多様に変化するため, ある成分があるかないかには比較的敏感であるが, 存在すればその個数には比較的鈍感であるような特徴抽出メカニズムが望ましい.

音声認識分野で開発され, 最近では手書き文字認識への応用の盛んな隠れマルコフモデル^{10),11)}では文字を筆跡の基本成分が確率的につながったものと見なし, マルコフ確率過程で近似表現する. 文字の崩れなどは確率的ノイズとしてモデルに容易に取り込み, 基本成分のみでなくその連をも筆跡成分と見なすことでモデルの精度を上げることもできる. 可能性のある文字を仮説とし, 観測された筆跡を最大の確率でもたらすような仮説を採択することで認識を行う. このモデルを利用できるのは, 対象が有限でありノイズも含めてマルコフモデルで記述可能, かつ状態遷移確率を知るために相当量の教示例が存在する場合であり, スケッチ認識ではこれは難しい.

そこで, データ圧縮技術を利用する新たな特徴表現手法の導入を試みる.

(3) データ圧縮を利用したスケッチの特徴表現法

データ圧縮技術は通信文の短縮を目的として開

発されたものであり, 周知のシャノンの符号化手法に従って「多頻度に発生する文の成分に短い符号長を与える」ことで通信文の全体長を圧縮する^{12),13)}. これまで様々な圧縮方式が開発され, それらをプログラム化したものが実用に供されている. 本論文ではこの種のプログラムをデータ圧縮器と呼ぶことにする. 文の成分の頻度分析のためにデータ圧縮器は通常, 文成分の辞書を保持している. 辞書はあらかじめ例文を与えて作成しておく場合と, 入力文を圧縮する過程でオンライン的に作成する場合とがある.

さて, ある成分 P_1 を含む文 EMP_1 と, 別の成分 P_2 を含む文 EMP_2 とを別々に与え, 2種類の圧縮器 A_1, A_2 が構築できたとする. A_1 と A_2 に同時にある任意の文 M を入力して圧縮させると次のような結果が生じる.

(場合1) M が P_1 を P_2 より多頻度を含む場合

$$f(A_1, M) < f(A_2, M)$$

(場合2) M が P_2 を P_1 より多頻度を含む場合

$$f(A_1, M) > f(A_2, M)$$

ただし, $i = 1, 2$ について,

$$f(A_i, M) = |M_i|/|M| \\ = A_i \text{ による } M \text{ の圧縮率,}$$

$$M_i = A_i \text{ による } M \text{ の圧縮後の文,} \\ |M_i|, |M| = M_i, M \text{ の長さ.}$$

これを利用すると任意の文 M の特徴を次の n 次元ベクトル F で表現できる.

$$F = (f(A_1, M), \dots, f(A_n, M)),$$

ここで n は事前に用意した別々の特徴的な文に適応するように構築した圧縮器の個数である.

以上の方式において, M を, スケッチ筆跡成分を符号化した時系列データ, EMP_i ($i = 1, \dots, n$) を, 圧縮器形成のための教示用スケッチの時系列データ, A_i ($i = 1, \dots, n$) を教示用スケッチに適応させたデータ圧縮器群とすることで, 無限に多様なスケッチをベクトル空間で表現する方法を得る. $n = 3$ の場合の状況を図2に示す. A_i などの圧縮器が空間の基底の役割を果たすことが分かる.

圧縮アルゴリズムのうち, LZ77¹²⁾は圧縮処理中に入力データを直接辞書として利用し, 事前に辞書を作成する必要がない. LZ78系¹³⁾では圧縮処理速度向上のために事前作成する. 後者を利用すると, 特徴抽出処理に先だって圧縮器に辞書をインストールするための余分なステップ(圧縮器の事前トレーニング)が必要となる. この難点を避けるために本研究ではLZ77系, 特にLZ77-LZB¹⁵⁾アルゴリズムを採用する. 以下図3でその動作を説明し, 続いて本研究への適用に

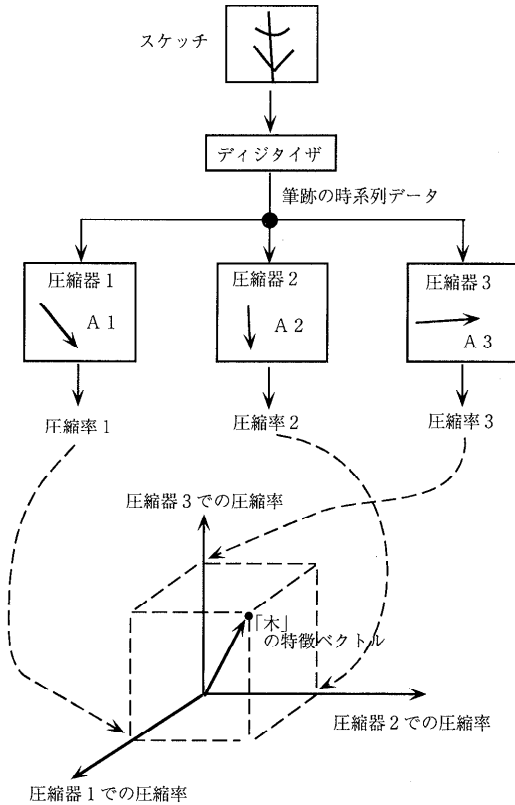


図2 データ圧縮率ベクトルによるスケッチ表現

Fig.2 Sketch representation by data compression ratio vector.

際して加えた変更点を示す。以下の%部は図3の説明部である。

LZ77-LZB：データ圧縮アルゴリズム；

INPUT：入力記号列 Istr, スライド窓最大長 MWL, 最短一致長 ML；

%Istr = ababbababaa,

%MWL = 8, ML = 1

OUTPUT：圧縮済み出力記号列 Ostr；

%Ostr = <0,"a"><0,"b">

<1, 0, 2><1, 1, 3>

<1, 0, 3><1, 0, 1>

<1, 2, 1>

STEP1：Istrの左端にスライド窓の左端 SW1, および右端 SW2をあわせる；

%第1行目では SW1 = SW2 = 0

STEP2：窓内に文字がなければ STEP4へ

文字があればそれと窓の右側の文字列とを最長一致させる

%第3行目では abで一致長 = 2

一致長 ≥ MLなら成功, 否なら不成功, 右側に文

| 入力記号列とスライド窓 | 一致位置 | 一致長 | 出力符号語 |
|--------------------------------------|------|-----|---------|
| Istr ababbababaa | --- | 0 | <0,"a"> |
| スライド窓 ababbababaa 0 | --- | 0 | <0,"b"> |
| SW1 SW2 ababbababaa 01 | 0 | 2 | <1,0,2> |
| ababbababaa 0123 | 1 | 3 | <1,1,3> |
| ababbababaa 0123456 | 0 | 3 | <1,0,3> |
| MWL = 8 ababbababaa 01234567 | 0 | 1 | <1,0,1> |
| ababbababaa 01234567 | 2 | 1 | <1,2,1> |

"a"などは不一致時に出力されるスライドウィンドウ直後の文字を示す。

図3 LZ77-LZB データ圧縮アルゴリズムの動作

Fig.3 Behaviour of LZ77-LZB data compression algorithm.

字がなければ終了 (%第7行目)；

STEP3：出力 Ostr 生成。

成功のときは <1, 一致位置, 一致長> を出力,

不成功のときは <0,"窓の右の文字"> を出力；

%第1行目は不成功で <0,"a"> を出力

STEP4：一致部分を取り込むまで SW2 を右移動。

不一致の場合は1だけ右移動。窓長 ≤ MWL を守るよう, SW1も右移動；

%第6行目は第5行目に対し SW2は3文字,

%第SW1は2文字右移動

STEP5：STEP2へ；

本アルゴリズムにより, 図3の11文字の入力文は7個の符号列に圧縮され, 圧縮率 = 7/11となる。以上の説明から分かるように LZ77-LZB では入力文内のスライド窓で覆われた部分が辞書として使用され, 窓の右側の未処理部分が圧縮される。

この特徴に着目し, 先の教示用スケッチ EMP_i (i = 1, ..., n) を入力記号列の左側に配置し (この部分を以下では基底データと呼ぶ), その右側に入力記号列 M を配置したものを Istr とし, スライド窓の初期条件を SW1 = 0, SW2 = |EMP_i|, MWL = ∞ とし,

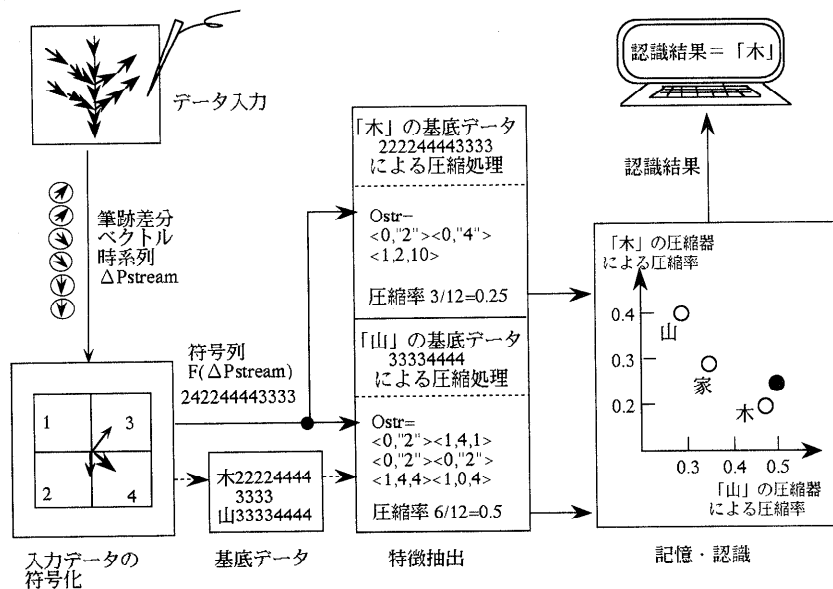


図4 スケッチ認識システムのブロック図

Fig. 4 Block diagram of the sketch recognition system.

STEP2の「窓内の文字列」を「EMP_i上の文字列」と変更することで、EMP_iを用いたMの圧縮が可能となる（データ配列の詳細は3.3節で述べる）。

なお、STEP2で使用する最短一致長MLの値はマッチング性能を左右し、特徴抽出能力に影響を与えるが、その適正値を理論的に決定することが困難なため通常は経験的な適正値が用いられている。本研究でも実験的に適正値を求めて使用した。

3. スケッチ認識システムのプロトタイプینگ

スケッチ認識システムは図4に示すように下記の諸機能で構成される。

- (1) データ入力,
- (2) 入力データの符号化,
- (3) 特徴抽出,
- (4) 記憶・認識.

以下、これらの内容を図4を利用しつつ説明する。

3.1 データ入力

図4の左上部に示すように、スケッチは電子ペンで入力する。図では「木」のスケッチが入力されている。定められた周期ごとにペンの位置座標 P_t を計算機に取り込む。時刻 t での計測位置を $P_t = (X_t, Y_t)$ とするとき、相続く2つの位置の差分ベクトル、

$$\Delta P_t = (X_t - X_{t-1}, Y_t - Y_{t-1})$$

の列 ($t = 1, 2, \dots$) を筆跡差分時系列データと呼ぶ。1つのスケッチを描き終えたら、そのことをシステムに知らせ、対応する差分時系列データ $\Delta Pstream$ (図

4左部の楕円で囲んだ短い矢印の連鎖)を作成して以後の処理に渡す。

3.2 入力データの符号化

図4左下部に示したように、平面区画割りパターンを用いて $\Delta Pstream$ を、あらかじめ定めた有限個の記号の時系列に符号化する。結果を $F(\Delta Pstream)$ と略記する(図では242244443333)。

区画割りパターンの例を図5に示す。図中(1), (2), (4), (5)は差分ベクトル ΔP_t の方向のみを符号化する。(1), (2)は粗く、(4), (5)は詳細である。(3)は差分ベクトルの長さにも敏感で、スケッチ描画スピードが符号に反映される。本研究では形が似たスケッチは描画スピードによらず同符号に符号化する方針とし、(3)のタイプは使用しない。(1), (4)に対し(2), (5)は人がスケッチを描く場合に多用する縦、横、斜め(45度)を安定に符号化することに留意したパターンである。本研究では最終的に(5)が最良であると判明した。

3.3 特徴抽出

図4の中心部に示すように、 $F(\Delta Pstream)$ に2.3節(3)のデータ圧縮アルゴリズムを作用させる。データ圧縮器を動作可能とするためには基底データが必要であり、図の例ではシステム利用者があらかじめ「木」「山」のスケッチを各1件ずつ与え、それぞれを上記3.2節の機構で符号化して得た基底データ222244443333および33334444を圧縮器に与えている。これらの基底データのもとで $F(\Delta Pstream)$ は圧縮され、結果的

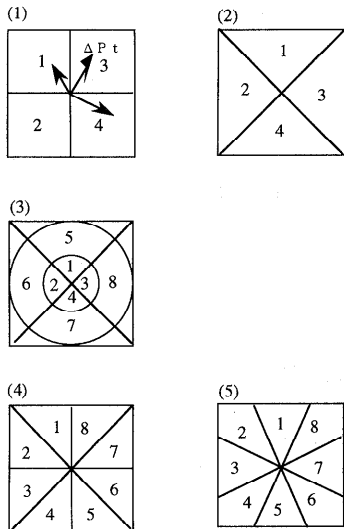


図5 筆跡差分ベクトル符号化機構
Fig.5 Coding scheme for ΔP_t .

に特徴ベクトル $(0.5, 0.25)$ が求まっている。

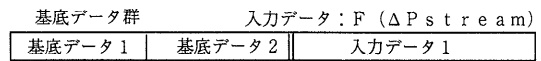
圧縮器での特徴抽出のためのデータ配列状況を図6により詳しく説明する。図6(1)に示すように、入力データ $F(\Delta Pstream)$ の前に基底として使用するデータ群を付加した1次元配列が使用される。図の例は基底データを2個、入力データを1個含む。入力データ1の基底データ1による圧縮率を計算するには、図6(2)に示すように、基底データ2の部分にマスクをかけてスライドウィンドウの一部を無視させつつ、2.3節(3)の圧縮アルゴリズムを適用する。その結果、圧縮率 $f(1,1)$ を得る。同様に基底データ2による処理では基底データ1の部分にマスクをかけて $f(2,1)$ を得る。これらを集めて、

$$F_1 = (f(1,1), f(2,1))$$

を入力データ1の特徴量とする。本方式では基底データ群を1次元配列の左端部にセットするのみで即座に特徴抽出が可能となり、圧縮器に圧縮用辞書をもたせるための事前トレーニングの手間が省ける。

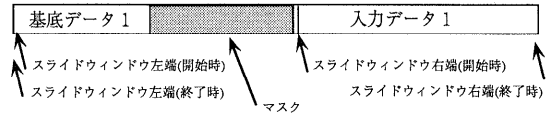
なお、2.3節(3)で実験的に適正值を定めると述べた最短一致長の値は $ML = 3$ とした。実験では「線分」「木」「山」などの簡単なスケッチを2セット用意し、片方のセットを基底データとして他のセットを圧縮させた。その結果、 $ML = 2$ ではともに斜線で表された「木」の枝と「山」の陵線とがマッチングして「木」の基底が「山」をよく圧縮するという傾向が増大し、 $ML = 5$ ではデータに混入する手ぶれノイズの影響を受けて「木」の基底による「木」のデータの圧縮率自体が低下する傾向が見られた。 $ML = 3, 4$ では

(1) 基底データと入力データの1次元配列

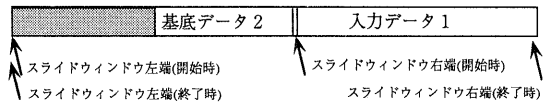


(2) 圧縮による特徴抽出

基底データ1による圧縮: 圧縮率 $= f(1,1)$



基底データ2による圧縮: 圧縮率 $= f(2,1)$



$$F_1 = \text{入力データ1の特徴ベクトル} \\ = \text{圧縮率ベクトル} \\ = (f(1,1), f(2,1))$$

図6 特徴抽出のための各種データの配列

Fig.6 Data arrangement for feature extraction.

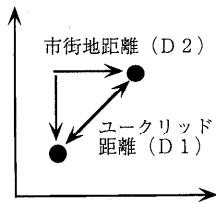
これらの問題が少なかったため、最終的に $ML = 3$ を使用した。

3.4 記憶・認識

事前登録しておきたいスケッチ $S_i (i = 1, \dots, m)$ それぞれに対応する $F(\Delta Pstream)$ があらかじめ上記3.3節の方法で処理され、得られた特徴ベクトル F_i をスケッチに対応する概念、たとえば「木」を示すテキストコード C_i とともに (F_i, C_i) のペア形式で記憶してあるものとする。図4では右下部の2次元特徴空間内の3個の○が「山」「家」「木」をこの方法で記憶させたことを示している。この事前準備の後、事後入力スケッチ S^* を入力すると対応する $F(\Delta Pstream)$ を特徴抽出して、図中の●に相当する検索用ベクトル F^* が得られる。 F^* と事前に記憶させたベクトル $F_i (i = 1, \dots, m)$ との距離を計算し、 F^* から最短距離にある F_{min} を探索し、 F_{min} とペア記憶させてある概念 C_{min} を S^* の認識結果とする。図4では「木」と認識されている。

距離の定義としてユークリッド距離を採用し、平均的には良好な特性を得たが、スケッチの大局的類似性の検出に失敗し、人の直感と合致しない結果を招くことがあった。この問題に対処するため、以下に示すように、特徴の異なる複数の距離関数を統合的に利用する「統合的類似度」を導入した。図7(1)に示す3種の距離関数をまず用意した。D1はユークリッド距離、D2は市街地距離、D3はエディット距離である。D1、D2が各圧縮器によるスケッチの圧縮率を平等に評価するのにに対し、D3は以下に示すように圧縮器のスケッ

(1) 使用する距離



エディット距離 (D3)

F1 = (0.8, 0.4, 0.5)

F2 = (0.7, 0.5, 0.3)

→

ssfx(F1) = (2, 3, 1)

ssfx(F2) = (3, 2, 1)

→

D3 = ed((2,3,1), (3,2,1))

= 2 × 0.1

= 0.2

(2) 統合的類似度

| 距離 | ユークリッド D1 | 市街地 D2 | エディット D3 |
|---------------|--------------|-----------|-------------|
| 候補1 | ↓ 10 | ↓ 10 | ⬆ 10 |
| 候補2 | ⬆ 9.8 | ∧ 9.6 | ↓ 9.8 |
| 候補1, 2 間距離 | 0.2 | 0.4 | 0.2 |

↓ 統合的類似度 = 29.8

⬆ 統合的類似度 = 19.8

図7 類似ベクトルの検索方式

Fig. 7 Nearest vector selection mechanism.

チへの相性の良さ(圧縮率の良さ)の順位に着目してスケッチの特徴をロバストに表現する。

すなわち,

$$D3 = ed(ssfx(F^*), ssfx(F_i))$$

ここで, $ssfx(F)$ はベクトル F の生成に関与した圧縮器の番号を圧縮能力順に並べたものであり図7の例では,

$$ssfx((0.8, 0.4, 0.5)) = (2, 3, 1)$$

などとなる。 $ed(X, Y)$ はベクトル X と Y の非一致要素数(すなわち圧縮率順位に関して見解の異なる圧縮器の個数) × 調整定数である。調整定数は $D3$ 値のオーダーを, 定義のまったく異なる $D1$ や $D2$ のオーダーと揃えるためのものであり,

$$\text{調整定数} = \mu(D1 + D2) / \mu(D3)$$

と定義した。ここで $\mu(X)$ は X の平均値であり, 「木」「山」「家」などのスケッチの特徴ベクトルを使用した実験の結果から 0.1 を使用することとした。以上より, 図7では,

$$ed((2, 3, 1), (3, 2, 1)) = 2 \times 0.1 = 0.2$$

が得られている。

これらの距離を用いて「統合的類似度」を次のように定義した。すなわち, 検索用ベクトル F^* が与えられたとき, 3種の距離 $D1, D2, D3$ を用いて F^* に近接する上位 P 件の候補ベクトルを各々選択し, 多くの距離で高順位となった候補を上位から P 件選出す

る。 $P = 2$ とした場合の具体例を図7(2)に示す。検索ベクトル F^* (図には示していない) に対し, たとえば $D1$ では「木」「家」の順となっている。多くの距離で高順位となったものを優先するために, 各距離 $D1, D2, D3$ ごとに, 第1候補に最高点 Max を与え, より劣った第 k 候補には第1候補からの乖離状態に応じた罰金を差し引いた評点を次式で与える。

$$Max - (\text{第1, 第}k\text{候補間の } D_i \text{ で計った距離}).$$

図には $Max = 10$ とした場合が示されている。

最後にこの評点を各候補ごとに集計したものを各候補の「統合的類似度」とし, その上位 P 件を F^* の類似物と決定する。図では「木」が 29.8 で第1候補, 「家」が 19.8 で第2候補と決定されている。

4. スケッチ認識実験

4.1 実験の目的と方針

提案手法の有用性は以下の要因に左右される。

- (1) 符号化器の良否
- (2) 特徴抽出に使用するデータ圧縮器の種類
- (3) 事前登録スケッチの個数
- (4) 事後入力スケッチの自由さ
- (5) 特徴ベクトル検索時に利用する距離関数

このうち(2)では, なるべく簡単かつ少数の圧縮器で多様なスケッチを認識できれば好都合である。(3)では同一概念のスケッチについてはできるだけ少数事例の事前登録で多様な事後入力に対応できることが望ましい。(4)では事後入力スケッチを事前登録スケッチよりもできるだけ簡単なもので済ませたい。(5)では新たに工夫した統合的類似度とユークリッド距離などの通常の距離との効果の違いを把握したい。

これらの要因の影響を調べるために3種類の実験を行った。なお, 事前登録スケッチと, 事後入力スケッチとを与えるのが同一人物の場合と, 別人の場合とが考えられるが, 本実験では前者の場合に限定した。これは, 本研究がデータ圧縮を利用したスケッチの特徴表現という新原理の検討段階にあり, 応用段階で議論すべき後者の課題とは検討を切り分けるべきであると考えたためである。

4.2 実験内容と結果

(1) 実験1

前節の要因(1)~(4)の認識能力への影響を調べるために各要因を変動させてみる。図8の中央列に示す5種類の簡単な, すなわちスケッチが線分, 円弧などの単純要素で構成され, その表す概念も単一であるようなスケッチを実験対象とする。要因(1)については図5の(2), (4), (5)を使用し, 要因(2)の基底デー

| 基底データ : 下記8個 | 事前登録データ | | 事後入力データ | | |
|-----------------|---------|----|---------|-----|-----|
| | スケッチ | 概念 | 原形 | 変形1 | 変形2 |
| ↑ ↓ | | 木1 | | | |
| ↖ ↗ | | 山 | | | |
| ↙ ↘ | | 家 | | | |
| ← → | | 木2 | | | |
| ↖ ↗ | | 川 | | | |

図8 実験1のデータ諸元

Fig. 8 Data items used in experiment 1.

| 正解 | システムの応答 (正解回数/実験回数) | | | | | |
|----|---------------------|-----------|----------|-----------|-----------|-----------|
| | 事前登録数=20個 | 事前登録数=10個 | 事前登録数=5個 | 符号化器図5(2) | 符号化器図5(4) | 符号化器図5(5) |
| 木1 | 49/50 | 48/50 | 49/50 | 34/50 | 38/50 | 49/50 |
| 山 | 50/50 | 48/50 | 49/50 | 35/50 | 43/50 | 50/50 |
| 家 | 49/50 | 46/50 | 46/50 | 35/50 | 45/50 | 49/50 |
| 木2 | 50/50 | 50/50 | 50/50 | 32/50 | 50/50 | 50/50 |
| 川 | 50/50 | 50/50 | 50/50 | 32/50 | 49/50 | 50/50 |

図9 実験1の結果

Fig. 9 Results of experiment 1.

タについては対象とするスケッチが簡単なため、図8左端の縦、横、斜め線を書き順を変えて各2個、合計8個使用する。要因(3)については各スケッチごとに20件、10件、5件を対応概念とともに事前登録し、認識率の差を調べる。要因(4)については図8の右列に示した原形、変形1、変形2などをランダムに選び合計50個を事後入力して認識率を調べる。要因(5)についてはユークリッド距離を用いるにとどめる。

実験1の結果を図9の左半分に示す。要因(2)については単純な線分スケッチ8個の基底データで高精度の認識が可能であることが示された。要因(3)については事前登録スケッチを5個程度に削減しても認識率に大差は見られない。要因(4)については、大きく形状が変化する事後入力のもとでも高精度の認識が可能であることが示された。以上の結果は、要因(1)の符号化器として図5(5)を使用した場合のものであるが、これを変更した場合が図9の右半分に示されている。この実験では事前登録データは20件とした。図5(2)の符号化器は粗すぎてスケッチの特徴を十分に表現できず性能が悪い。図5(4)、(5)の区画数(=符号語の数)はともに8個だが(5)が最も良い。その理由は(4)がスケッチに多用される縦、横、斜め線の多少の変動に対して別の符号語を割り当ててしまうという意味で不安定なのに対し(5)はそれを防止できたためと考えられる。

(2) 実験2

多種類の単純スケッチを要素として含んでいる複雑なスケッチに対する認識性能を調べる。要因(1)については実験1で良好と判明した図5(5)の符号化器を用いる。要因(2)については複雑なスケッチの要素を取り出して作成した図10左端列の3案を比較検討す

| 基底データ | 事前登録データ | | 事後入力データ | |
|---|---------|----|---------|-----|
| | スケッチ | 概念 | 変形1 | 変形2 |
| 基底群1 : 8次元 基底群2 : 5次元 基底群3 : 13次元 | | 町 | | |
| | | 村 | | |
| | | 山 | | |
| | | 森 | | |
| | | 滝 | | |

図10 実験2のデータ諸元

Fig. 10 Data items used in experiment 2.

る。要因(3)については図10中央列のスケッチを各1件のみ事前登録する。要因(4)については図右端列の詳細度の異なる2種類の事後入力を与える。実験回数は各スケッチで30回とする。要因(5)についてはユークリッド距離と統合的類似度との認識率の差を比較する。

結果を図11に示す。要因(2)については基底群1, 2, 3の順に認識性能が高まる傾向が見られる。特に「森」の例でこれが顕著である。この結果は、認識対象スケッチ内の低レベル構造である単純線分に敏感なものから、「木」「山」などの中間レベルの構造に敏感なものまで、基底成分の複雑さレベルを多様化することが認識率向上に有効であることを示している。要因

| 正解 | システムの応答 (正解回数/実験回数) | | | |
|----|---------------------|--------------|---------------|---------------|
| | 統合的類似度 | | | ユークリッド |
| | 基底群 1 8次元 | 基底群 2 5次元 | 基底群 3 13次元 | |
| | | | | 基底群 3 13次元 |
| 町 | 27/30 | 28/30 | 30/30 | --- |
| | 21/30 | 22/30 | 29/30 | 20/30 |
| | 21/30 | 23/30 | 30/30 | 30/30 |
| 村 | 28/30 | 28/30 | 30/30 | --- |
| | 12/30 | 24/30 | 28/30 | 25/30 |
| | 30/30 | 30/30 | 30/30 | 30/30 |
| 山 | 30/30 | 30/30 | 30/30 | --- |
| | 29/30 | 28/30 | 30/30 | 28/30 |
| | 30/30 | 30/30 | 30/30 | 30/30 |
| 森 | 24/30 | 26/30 | 30/30 | --- |
| | 2/30 | 4/30 | 27/30 | 25/30 |
| | 3/30 | 11/30 | 27/30 | 30/30 |
| 滝 | 30/30 | 26/30 | 30/30 | --- |
| | 29/30 | 30/30 | 30/30 | 17/30 |
| | 30/30 | 30/30 | 30/30 | 30/30 |

上段は検索時に変形1を入力した場合
 中段は検索時に変形2を入力した場合
 下断は検索結果の第2候補までを正解と認めた場合

図 11 実験 2 の結果

Fig. 11 Results of experiment 2.

(4)については、変形1の事後入力に事前入力に近いので認識率が良好であるが、省略の多い変形2では一般に認識率は低下する。基底群1, 2ではこの傾向は高いが、基底群3はその特徴抽出能力によって性能劣化を防止している。「森」の例で基底群1, 2は非常に成績が悪い。森のスケッチ画の特徴は木が多いことであるが、変形2の問合せ入力はこの特徴を表現していないためと考えられる。基底群3を利用し、検索の際に第2候補までに正解を含めばよいとすると変形2の事後入力の場合でもほぼ100%の認識率となる(「森」は90%)。要因(5)については図11の右端にユークリッド距離を用いて基底群3のもとで変形2の事後入力を与えたときの認識率を示す。第2候補までを正解と見なすとすべて100%正解を与えているが、第1候補で正解を引き出す能力は総合的類似度に比べて相当悪い。よって、類似性判定尺度としては総合的類似度の方が優れているといえる。

(3) 実験3

実験2では、要因(2)に関して事前登録スケッチに含まれる要素を基底データとして採用し、高精度の認識能力を実現できたが、ある対象向けに用意した基底を他の対象に流用できると、対象分野ごとに基底データを用意する必要もなくなりさらに好都合である。本実験ではこの可能性を調べる。図12に概要と結果を示す。

本実験では要因(1)については図5(5)を、要因(2)については図12の左列に示す実験2の基底データ群3を使用した。また要因(3)については図12の中央列のものを各1件事前登録し、要因(4)につい

| 基底データ 基底群3 : 13次元 | 事前登録データ | | 事後 入力 | 応答 | |
|-------------------------|---------|---------------|----------|-------|-------|
| | | 概念 | | 上位1件 | 上位2件 |
| | | ドラえもん | | 24/30 | 27/30 |
| | | モナリザ | | 27/30 | 28/30 |
| | | クレヨン しんちゃん | | 25/30 | 30/30 |
| | | トトロ | | 22/30 | 24/30 |
| | | ドラミちゃん | | 20/30 | 30/30 |
| | | ムーミン | | 22/30 | 30/30 |

図 12 実験 3 のデータ諸元

Fig. 12 Data items used in experiment 3.

ては図の右列の事後入力を30件与える。要因(5)については統合的類似度を利用する。第1候補のみを正解とする場合と第2候補までを正解とする場合の認識率を調べる。

図12の右列に結果を示す。上位1件を正解とすると70~80%の認識率である。「ドラえもん」と「ドラミちゃん」、「トトロ」と「ムーミン」が似ているためお互いの認識率を下げているが、上位2件まで正解とすると80~100%となる。認識対象には関係のない「木」「山」などで構成される基底群がそれらを直接部分要素に含まないスケッチの特徴抽出にも有効であることが示されている。

5. ま と め

5.1 結 論

(1) ヒューマンインタフェースの高度化には人と計算機との柔軟な対話が重要である。これに有効な技術を開発する目的で、筆跡時系列データが得られるオンライン・スケッチ認識問題を取り上げ、以下を特徴とする解決策を提示した。

- それぞれが固有のスケッチを効率的に圧縮するデータ圧縮器を複数集め、あるスケッチをそれらで圧縮させたときの圧縮率ベクトルによって、多様で有限分類できないスケッチの特徴を表現する。
- 圧縮器としてLZ77-LZBを利用することで圧縮器の事前トレーニングを不要化し、特徴ベクトル空間構築の手間を削減する。
- 特徴ベクトル空間内の近接点を探索する際に、スケッチの全体的および局所の特徴を考慮できる統

合的類似度を用いる。

(2) プロトタイプシステムを開発し、単一利用者環境での実験によって下記を確認した。

- 単純なスケッチを対象とする場合、利用者がわずかの基底と数個のスケッチの事前登録を行うのみで、多様な事後入力スケッチをほぼ100%認識できる。
- 複雑なスケッチの場合、そのスケッチに含まれる低レベルから中間レベルの図形に対する基底を利用者が用意すると、事前登録スケッチと相当乖離した事後入力スケッチに対しても高精度の認識が可能である。
- ある分野のスケッチに含まれる要素を抽出して作成した基底を、別分野のスケッチの認識に有効に流用することが可能である。

5.2 今後の課題

複数利用者の場合への対応、文脈の利用など認識性能を向上させるための工夫、マルチメディア問題への適用などが今後の課題である。

謝辞 データ圧縮技術に関して情報をいただいた電気通信大学情報システム学研究所、伊藤秀一、星守両教授に感謝いたします。また、論文の完成過程で有益なコメントをいただいた査読者の方々に感謝いたします。

参考文献

- 1) Impedovo, S., Ottaviano, L. and Occhincgro, S.: Optical Character Recognition - A Survey, *International Journal of Pattern Recognition and Artificial Intelligence*, Vol.5, No.1, 2, pp.1-24 (1991).
- 2) Mori, S., Suen, C.Y. and Yamamoto, K.: Historical review of OCR Research and Development, *Proc. IEEE*, Vol.80, No.7, pp.1029-1058 (1992).
- 3) 盛田 栄, 長嶋祐二, 長嶋秀世: 商標図形の記述方法に関する検討, 信学技報, PRU87-111, pp.15-22 (1988).
- 4) 加藤俊一, 下垣弘行, 藤村是明: 画像対話型商標・意匠データベース TRADEMARK, 信学論, Vol.J72-DII, No.4, pp.535-544 (1989).
- 5) 岡野栄司, 木下宏揚, 酒井善則: 手書き画像とのマッチングによる画像検索, 信学技報, IE93-96, pp.53-60 (1994).
- 6) Chang, S.K., Ichikawa, T. and Ligomenides, P.A.: *Visual Language*, Plenum Press (1986).
- 7) Young, T.Y. and Fu, K.S.: *Handbook of Pattern Recognition and Image Processing*, Academic Press (1986).
- 8) Freund, R.: Syntactic Analysis of Handwritten Characters by Quasi-Regular Programmed Array Grammers, *Advances in Structural and Syntactic Pattern Recognition*, Bunke, H. (Ed.), pp.310-319, WSP (1992).
- 9) Rosenfeld, A.: *Picture Languages: Formal Models for Picture Recognition*, Academic Press, New York (1979).
- 10) Rabiner, L.R.: A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition, *Proc. IEEE*, Vol.77, No.2, pp.257-286 (1989).
- 11) Bercu, S. and Lorette, G.: On-line Handwritten Word Recognition: An Approach based on Hidden Markov Models, *Proc. 3rd International Workshop on Frontiers in Handwriting Recognition (IWFHR-3)*, pp.385-390 (1993).
- 12) Ziv, J. and Lempel, A.: A universal algorithm for sequential data compression, *IEEE. Trans. Info. Theory*, IT-233, pp.337-343 (1977).
- 13) Ziv, J. and Lempel, A.: Compression of individual sequences via variable-rate coding, *IEEE. Trans. Info. Theory*, IT-245, pp.530-536 (1978).
- 14) Welch, T.A.: A technique for high-performance data compression, *IEEE Computer*, 176, pp.8-19 (1984).
- 15) Timothy, C.B., Cleary, J.G. and Witten, I.H.: *Text Compression*, Prentice Hall (1990).
- 16) Ziv, J.: On Classification with Empirically Observed Statistics and Universal Data Compression, *IEEE. Trans. Info. Theory*, IT-34, 2, pp.278-286 (1988).

(平成8年4月10日受付)

(平成9年10月1日採録)

近藤 邦広

1969年生。1991年電気通信大学電子工学科卒業。1993年同大学大学院情報システム学研究科修了。同年、NTTデータ通信(株)入社。金融情報システムの設計開発に従事。



加藤 直樹

1970年生。1994年電気通信大学電子工学科卒業。1996年同大学大学院情報システム学研究科修了。同年、日本オラクル(株)入社。情報システム設計開発に従事。





渡辺 俊典（正会員）

1947年生。1971年東京大学工学部航空学科卒業。工学博士。日立製作所を経て、1992年より電気通信大学大学院情報システム学研究科勤務。情報システムモデリング、適応情報システムなどの研究教育に従事。ACM, IEEE, ALP, 日本ソフトウェア科学会各会員。
