

# パターンの疎・再帰的・階層的な表現

石川 博

名古屋市立大学 大学院システム自然科学研究科

〒467-8501 名古屋市瑞穂区瑞穂町字山の畑1

E-mail: hi@nsc.nagoya-cu.ac.jp

あらまし ビジョンシステムにおけるパターンの表現は、従来、その場限りのデータの扱い方をするプログラムの中に隠されてしまうことが多いため、一般性・柔軟性に欠け、非本質的な要素を分離しにくいという問題があった。異なるデータ構造間で自動的に変換でき、データを比較・類推できるためには、画素によるデータ配列化のような特定のデータ構造から独立に捉えられる、構造の表現を一般的に定義することが必要である。本稿では、より自動的かつ柔軟なパターンの扱いを可能にすることを目指して、統一的で一般性が高く、パターンの構造情報と実装依存部分を分離できる表現方法を提案する。

キーワード パターン表現, 知覚体制化, 疎な表現, 階層的な表現, 再帰的な表現, 図式

## Sparse, Recursive, and Hierarchical Representation of Patterns

Hiroshi Ishikawa

Department of Information and Biological Sciences, Nagoya City University

1 Yamano-hata, Mizuho, Nagoya 467-8501, Japan

E-mail: hi@nsc.nagoya-cu.ac.jp

**Abstract** Representation of structures in patterns in vision systems are not very general or flexible, or easy to separate non-essential elements, because they are usually implemented in a program that treats data in an *ad hoc* manner. To enable automatic transformation of data between different data structures, as well as comparison and analogy-making between them, it is necessary to define the notion of structure independent of specific data structures such as the pixel representation. In this paper, we propose a unified general representation of patterns that separates the essential structure from implementation-dependant data.

**Key words** Pattern representation, Perceptual organization, Sparse representation, Hierarchical representation, Recursive representation, Diagram

### 1. はじめに

画像の認識・理解をするには、画像に含まれる情報だけでは一般に不十分で、システムは画像中に見出すべき規則性などの事前知識としてのモデルを持っていなければならない。そのモデルは、物体の形や位置、ステレオ曲面、あるいはセグメンテーションなど、探すべき対象を何らかの形で機械内に表現し、それに例えば確率を与えたり、それらの間に制限をつけたりして作られる。以下では、この探すべき対象を漠然と「パターン」と呼ぶ。つまり、ビジョンや画像処理の問題は、ノイズを含むデータの中にパターンを見つけ出すことと考えることができる。例えば、特徴抽出やパターン認識はその名の通りであるし、ノイズ除去はノイズ以外の「パターン」を増やす操作と考えられる。また、パターンの認識は多数の階層が必要である。例えば、トラッキングや顔の検出は、画像中に注目する特徴を見つけ、またそれらの特徴の空間的・時間的配置などのパターン

を見つけることによってなされる。

本稿では、パターンの機械内での表現方法、特にパターンの持つ構造を捉えた表現方法について考察する。例えば、物体の形は外周 ([1], [3], [5] 他多数) や中間軸 ([6], [9] 他多数) によって表現されてきた。外周は円周と同じ構造を持ち、その表現は一周分の点の画像内での位置を指定することで与えられる。これに対して中間軸は位置情報の他に木構造を持ち、これを使ってマッチングをしたり、事前確率を変えたりすることができる。木構造は単純な部分パターンからなるより複雑なパターンの階層的表現にも使われる [8], [10]。また、ステレオやセグメンテーション、画像処理などの低レベルの問題では、しばしば画像の各画素にラベルを与えることで結果が表現される。例えば、より滑らかな曲面により高い確率を与えるなどしてモデルが作られるが、これは画素と隣接関係をグラフで表していると考えられる。ループや木構造もグラフの一種であり、これらの表現は全てグラフの頂点と辺にデータを持たせた表現と考える

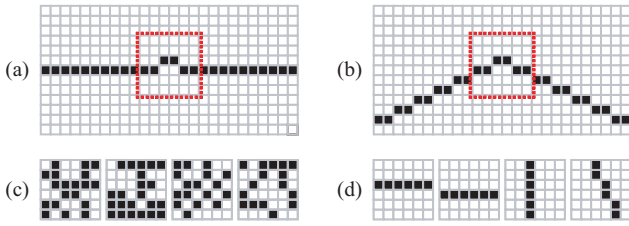


図1 2値画像のノイズ除去。(a)と(b)の中心部は同一だが、期待される結果は異なる。(c)中央部分の値の可能な組み合わせのほとんどすべてはノイズである。(d)のような特別なパターンだけを選んでノイズを除去する。

ことができる。グラフはまた文法的パターン認識[2]におけるパターンの表現にも使われる。このように、ベクトルなどの一様で単純なデータの間に構造を与える方法としては、ほとんどグラフしか使われていないようである。一般パターン理論を標榜するGrenander[4]も、表現にグラフしか使っていない。

しかし、これらの表現では表せない概念が多数存在する。例えば「直線」という概念をうまく表現することはグラフを使ってはできない。辺は2頂点間の関係を表すのみで、一連の頂点で表される点が直線上にあるという、全ての頂点間の関係を表すことがグラフではできない。もちろん、例えば連続する辺の間の角度の総和を最小化することによって、より直線に近い並び方を表現することはできる。しかし、「角度の総和を最小化する」ということをグラフまたはそれに付随するデータとして表すことはできない。このように、グラフのような一様なデータ表現で表現できないような構造は、データの扱い方としてプログラム中に埋め込まざるを得ない。

このような高次の関係を表現するには、密な表現では問題が生じる。例えば、図1において、理想のノイズ除去では、(a)中央部の2つの黒い画素はその左右と並んで一直線になることが期待される。一方、(b)においてはこれらは元のままでよい。ところが、この2つの例では点線枠内の配置は全く同じであるから、これらの状況を区別するには、かなり多くの点についての同時確率を与える必要があることがわかる。等方性を考えれば、少なくとも枠内の36画素の同時分布が必要だが、それだけで $2^{36}$ 個の確率を与える必要がある。しかし、その $2^{36}$ 通りの値の組み合わせのほとんど全ては、(c)に挙げたようなランダムな組合せであり、それらの間を区別する必要はない。必要なのは、(c)と(d)に挙げたようなパターンを区別して、後者に高い確率を与えることである。そのためには、(d)のようなパターンを特徴付ける必要がある。しかし、これを単純なデータとして持つことは困難でありまた望ましくない。

このように、最終的な表現、例えば画素の数に比例した量のデータを持つ表現を密な表現と呼ぶ。密な表現は、例えば画素のような、データ化から生じた人工物である離散性に緊密に結びついており、例えば直線を表そうとすれば、全体としてその長さに比例したデータ量が必要になる。これに対して、例えば一点とベクトルによる直線の表現が直線上の画素の数に比例したデータ量を必要としないように、少ないデータ量で最終的に大きなパターンを表現できる表現を疎な表現と呼ぶ。画像に現れる対象には、特定の解像度に抛らずに表現されるべき本来離散的でないものが多いことから、有限の表現で無限集合であるこれらの対象を表すことが出来ることは、柔軟性、一般性を得る上で不可欠である。

直線を認識するだけなら、そうするコードを書き、直線を表すデータ構造で結果を表現すればよいが、(b)の場合のように、多数存在する類似のパターンにそれぞれ対応するには困難が伴う。少しでも複雑なパターンに対処しようとするれば、そのパターンに特化したコードやデータ構造を作らなければならない。

このように、パターンの特徴づけは、密な表現で表された部分を除いて、断片的なデータと、その扱い方を表現したその場限りのプログラムの中に隠されてしまうことが多かった。それをいかに作るかが実装者の腕の見せ所でもあったのだが、これには次のような問題がある。

1. 一般性に欠ける。個別に実装されるコードは、解像度や色数などの実装のパラメータに依存したコードになりやすい。また、パターン毎に独自のデータ構造とそれを扱うコードを持つため、異なるパターン間の関係をつけるのにもまた独自コードが必要である。
2. 柔軟性に欠ける。データからの学習はパラメータのみにとどまることになり、構造そのものは実装段階で決まっていなければならないため、実装者の想像力によってシステムの能力が制限される。
3. 非本質的な要素を分離しにくい。例えば画素というものは、画像という情報の本質とは関係のない、データ化から生じた人工物である。異なるデータ構造間で自動的に変換でき、データを比較・類推できるようにするためには、画素によるデータ配列化のような特定のデータ構造にかかわらず捉えられる構造の概念を一般的に定義することが必要である。

本稿では、これらを解決するために、統一的で一般性が高く、個別のコードを書くことなくパターンの本質的な構造と実装依存部分を分離できる、パターンの表現方法を提案し、パターンをより自動的に柔軟に扱うことを可能にすることを旨とする。

提案するパターン表現は、以下の特徴を持つ。

1. 疎である。グラフやビットマップのような密な表現と違って、パターンをその本質的な構造とパラメータの組合せで表現する。
2. 階層的である。いかなるデータにも適用可能であるため、あるパターンを表現するパラメータの空間の中にパターンを表現することで、階層的な表現を可能にする。
3. 再帰的である。自由度の高い再帰的表現により、強力な表現力を持つ。

以下、次節では表現方法の定義を与え、それを第3節で例を使って説明する。第4節ではその特徴をより詳しく説明する。第5節では、表現されたパターンをビットマップなどの密な表現にレンダリングしたり、表現されたパターンを密な表現中に認識したりするための実装上の考察をする。

## 2. 図式によるパターンの表現

本節では、本稿で提案するパターンのデータ表現法の定義を与える。次節の例とあわせて理解されたい。なお、集合族とは集合の集まりのことである。

### 2.1. 定義

定義 1. 集合族 $\mathcal{S}$ の断面 $s$ とは、 $\mathcal{S}$ に含まれる集合 $S \in \mathcal{S}$ それぞれに、その部分集合 $s(S) \subset S$ を与える与え方のことである<sup>(注1)</sup>。

(注1)：この「与え方」が曖昧な表現に思われる向きには $\mathcal{S}$ から $\coprod_{S \in \mathcal{S}} \mathcal{P}(S)$ への写像 $s$ で $s(S) \in \mathcal{P}(S)$ を満たすものこと。

定義 2. 集合族  $\mathcal{S}$  の断面  $s, t$  について, 全ての  $S \in \mathcal{S}$  について  $s(S) \supset t(S)$  が成り立つとき,  $s$  は  $t$  を含むといい,  $s \supset t$  で表す.

定義 3.  $s$  を  $\mathcal{S}$  の断面とし,  $\mathcal{D} \subset \mathcal{S}$  とするとき,  $S \in \mathcal{D}$  に  $s(S)$  を与えることで定義される  $\mathcal{D}$  の断面を  $s$  の  $\mathcal{D}$  への制限と呼び,  $s|_{\mathcal{D}}$  で表す.

定義 4. 集合族  $\mathcal{S}$  に含まれる集合の冪集合の間の写像の集合  $\mathcal{M}$  があるとき, 集合族  $\mathcal{S}$  と写像の集合  $\mathcal{M}$  の組を, 図式  $(\mathcal{S}, \mathcal{M})$  と呼ぶ.

集合  $S$  の冪集合 ( $S$  の部分集合全体の集合) を  $\mathcal{P}(S)$  で表す.  $S, T \in \mathcal{S}$  のとき,  $\mathcal{M}$  に含まれる写像  $\varphi: \mathcal{P}(S) \rightarrow \mathcal{P}(T)$  について,  $S$  を  $\text{dm}(\varphi)$ ,  $T$  を  $\text{cdm}(\varphi)$  と書く. また  $\text{out}(S) = \{\varphi \in \mathcal{M} \mid \text{dm}(\varphi) = S\}$ ,  $\text{in}(S) = \{\varphi \in \mathcal{M} \mid \text{cdm}(\varphi) = S\}$  とする.

定義 5. 図式  $(\mathcal{S}, \mathcal{M})$  の断面  $s$  とは,  $\mathcal{S}$  の断面  $s$  であって,  $\text{in}(S) \neq \emptyset$  である全ての  $S \in \mathcal{S}$  について,

$$s(S) = \bigcap_{\varphi \in \text{in}(S)} \varphi(s(\text{dm}(\varphi)))$$

を満たすもののことである.

定義 6. 図式  $(\mathcal{S}, \mathcal{M})$  と  $\mathcal{D} \subset \mathcal{S}$  の断面  $s_{\mathcal{D}}$  に対して,  $\mathcal{D}$  への制限が  $s_{\mathcal{D}}$  と等しい  $(\mathcal{S}, \mathcal{M})$  の断面のうち最小のものが存在するとき, すなわち  $s|_{\mathcal{D}} = s_{\mathcal{D}}$  を満たす  $(\mathcal{S}, \mathcal{M})$  の断面  $s$  で,  $(\mathcal{S}, \mathcal{M})$  の任意の断面  $t$  について 「 $t|_{\mathcal{D}} = s_{\mathcal{D}} \implies t \supset s$ 」 が成り立つものが存在するとき, これを  $s_{\mathcal{D}}$  の決定する  $(\mathcal{S}, \mathcal{M})$  の断面という.

図式の制限が必ずしも断面を決定するとは限らない. 最も単純な反例を挙げると,  $\mathcal{S} = \{\{0\}, \{0, 1\}\}$ ,  $\mathcal{M} = \{\varphi\}$  で  $\varphi: \mathcal{P}(\{0, 1\}) \rightarrow \mathcal{P}(\{0\})$  は  $\varphi(\emptyset) = \emptyset$ ,  $S \neq \emptyset$  なら  $\varphi(S) = \{0\}$  で定義されるとき,  $\mathcal{D} = \{\{0\}\}$  の断面  $s_{\mathcal{D}}(\{0\}) = \{0\}$  は断面を決定しない. 実際,  $s(\{0\}) = s_{\mathcal{D}}(\{0\})$  として  $s(\{0, 1\})$  が  $\{0\}$  でも  $\{1\}$  でも断面である条件は満たして、この2つの断面に含まれる最小の断面は存在しない.

## 2.2. 冪集合間の写像

一般に写像  $f: X \rightarrow Y$  について, 同じ文字で冪集合間の写像  $f: \mathcal{P}(X) \rightarrow \mathcal{P}(Y)$  を表す. これは  $S \subset X$  について

$$f(S) = \{f(x) \mid x \in S\} \subset Y$$

で定義される. また, 写像  $f^{-1}: \mathcal{P}(Y) \rightarrow \mathcal{P}(X)$  は  $S \subset Y$  について

$$f^{-1}(S) = \{x \in X \mid f(x) \in S\} \subset X$$

で定義される. 本稿では  $\mathcal{M}$  に含まれる冪集合間の写像として, 写像  $f: X \rightarrow Y$  から誘導された  $f: \mathcal{P}(X) \rightarrow \mathcal{P}(Y)$  あるいは  $f^{-1}: \mathcal{P}(Y) \rightarrow \mathcal{P}(X)$  の形のもを主に使う. 他には補集合写像

$$\text{cpl}: \mathcal{P}(X) \rightarrow \mathcal{P}(X) \quad (\text{cpl}(S) = {}^cS = X \setminus S) \quad (1)$$

を使うのみである.

## 2.3. パターンの表現法

パターンを, (a) 図式と, その断面を決定するような (b) 部分的な断面を指定して表すのが, 本稿で提案するパターンの表現法である. 決定された断面が図式中の一つの集合に与える部分集合を, パターンをなすデータと解釈する.

本稿では, データは一般に何らかの集合の部分集合として表されるものとする. 例えば2値のビットマップであれば, 画素

の集合  $P$  の部分集合  $S$  と考えることができる (例えば画素  $p \in P$  が黒ならば  $p \in S$ , 白ならば  $p \notin S$  とみなす). グレースケールあるいはカラーのビットマップならば, 色の集合を  $C$  として,  $P \times C$  の部分集合  $S$  と考えられる (例えば画素  $p \in P$  の色が  $c$  ならば  $(p, c) \in S$ ). このように, 密なデータはそのまま何らかの集合の部分集合と考えられる.

集合  $A$  の部分集合  $S$  としてデータを表したとき, 図式  $\mathcal{S} = \{A\}$ ,  $\mathcal{M} = \emptyset$  の断面  $s(A) = S$  を, 入力データの図式による自明な表現と呼ぶ. つまり, 任意の密なデータは図式により表現可能である. もちろんそれだけでは意味がないが, 図式による表現は, 自明な表現により一般性が保証された上で, 次節で例を示すような, よりパターンの構造を捉えた効率のよい表現を可能にする.

## 2.4. 記号

以下, 本稿で図式  $(\mathcal{S}, \mathcal{M})$  を表すとき, 集合  $X, Y \in \mathcal{S}$  についてその冪集合  $\mathcal{P}(X)$  から  $\mathcal{P}(Y)$  への写像  $f: \mathcal{P}(X) \rightarrow \mathcal{P}(Y)$  を, 簡略化のため  $X \xrightarrow{f} Y$  のように表す.

直積集合  $X_1 \times X_2 \times \dots$  について  $\pi_i: X_1 \times X_2 \times \dots \rightarrow X_i$  は  $i$  番目の直積成分への射影を表す. 写像  $f: A \rightarrow X_1, g: A \rightarrow X_2$  があるとき, 直積写像  $f \times g: A \rightarrow X_1 \times X_2$  は  $f \times g(a) = (f(a), g(a))$  で定義される.

同じ集合や写像の複数のコピーを区別する必要があるときは,  $X, X', X'', \pi_1, \pi'_1, \pi''_1$  のようにプライムで区別する.

## 3. 例

本節では, 前節で定義した図式とその断面について, いくつか例を挙げて説明する. 例として, ユークリッド平面  $X$  の部分集合を考える. これは2値画像の抽象化と考えられる.

### 3.1. 直線

直線を表すには, 例えば次のような図式を与える.

$$\begin{array}{ccccc} X \times X' & \xleftarrow{\text{sub}^{-1}} & V & \xrightarrow{\text{mult}} & V' \times \mathbb{R} \\ \pi_1 \swarrow & & \uparrow \pi_2^{-1} & & \uparrow \pi_1^{-1} \\ X & & X' & & V' \\ & & & & \searrow \pi_2^{-1} \\ & & & & \mathbb{R} \end{array} \quad (2)$$

すなわち, 定義の形式に沿えば,  $\mathcal{S} = \{X, X', X \times X', V, V', V' \times \mathbb{R}, \mathbb{R}\}$ ,  $\mathcal{M} = \{\pi_1, \pi_2^{-1}, \text{sub}^{-1}, \text{mult}, \pi_1^{-1}, \pi_2^{-1}\}$  となる. ここで,  $\mathbb{R}$  は実数全体の集合,  $V$  はユークリッド空間  $X$  の移動のベクトル空間, また,  $\text{sub}: X \times X' \rightarrow V$  は  $(x, y) \in X \times X'$  に  $x - y$  を与える写像,  $\text{mult}: V \times \mathbb{R} \rightarrow V$  はベクトル  $v$  の実数  $c$  倍を与える写像である.

この図式の断面  $s$  で,  $\mathcal{D} = \{X', V', \mathbb{R}\} \subset \mathcal{S}$  への制限が次のようであるもの考える:

$$s(X') = \{p\}, \quad s(V') = \{v\}, \quad s(\mathbb{R}) = \mathbb{R}. \quad (3)$$

すると, 定義5を図式の矢印の方向に順に適用して,

$$s(V' \times \mathbb{R}) = \pi_1^{-1}(s(V')) \cap \pi_2^{-1}(s(\mathbb{R})) = \{v\} \times \mathbb{R} \quad (4)$$

$$s(V) = \text{mult}(s(V' \times \mathbb{R})) = \{cv \in V \mid c \in \mathbb{R}\} \quad (5)$$

$$\begin{aligned} s(X \times X') &= \text{sub}^{-1}(s(V)) \cap \pi_2^{-1}(s(X')) \\ &= \{(x, p) \in X \times X' \mid x - p = cv, c \in \mathbb{R}\} \end{aligned} \quad (6)$$

$$s(X) = \pi_1(s(X \times X')) = \{x \in X \mid x = p + cv, c \in \mathbb{R}\} \quad (7)$$

となるから,  $s(X)$  は点  $p$  を通り方向ベクトル  $v$  を持つ直線上の点全体の集合である.

このように(3)~(7)で与えられる $s$ は断面の定義5を満たし、また以上のように $s$ の各集合内に与える部分集合が一意に決まることから、断面(3)が $s$ を決定することもわかる。つまり、この直線の集合は、図式(2)と $\mathcal{D}$ の断面(3)で決定される。

### 3.2. 円 周

次に、円周を表すには、例えば次のような図式を与える。

$$\begin{array}{c}
 X \times X' \xleftarrow{\text{sub}^{-1}} V \xleftarrow{\text{len}^{-1}} \mathbb{R} \\
 \pi_1 \downarrow \quad \searrow \pi_2^{-1} \\
 X \qquad \qquad X'
 \end{array} \tag{8}$$

ここで、 $\text{len} : V \rightarrow \mathbb{R}$ はベクトルの長さを与える写像であり、他は上と同じである。この図式の断面 $s$ で、 $\mathcal{D} = \{X', \mathbb{R}\}$ への制限が

$$s(X') = \{p\}, \quad s(\mathbb{R}) = \{r\}. \tag{9}$$

であるものを考えると、定義5を順に適用して、

$$s(V) = \text{len}^{-1}(s(\mathbb{R})) = \{v \in V \mid \text{len}(v) = r\}. \tag{10}$$

$$\begin{aligned}
 s(X \times X') &= \text{sub}^{-1}(s(V)) \cap \pi_2^{-1}(s(X')) \\
 &= \{(x, p) \in X \times X' \mid \text{len}(x - p) = r\}
 \end{aligned} \tag{11}$$

$$s(X) = \pi_1(s(X \times X')) = \{x \in X \mid \text{len}(x - p) = r\} \tag{12}$$

となり、 $s(X)$ は点 $p$ を中心とする半径 $r$ の円周上の点全体の集合である。上と同様、(9)~(12)で与えられる $s$ は断面の定義を満たし、図式(8)と、 $\mathcal{D}$ の断面(9)で決定されることがわかる。

### 3.3. 部分集合

図式  $X' \xleftarrow{\text{id}} X \xleftarrow{\text{id}} X''$  で、 $s(X') = A, s(X'') = B$ の決定する断面 $s$ は $s(X) = A \cap B$ を与える。このように、断面の定義から、集合の共通部分を得るのは容易だが、和集合はどうしたらよいだろうか。補集合写像(1)を使った図式

$$X \xleftarrow[\text{cmpl}]{\text{cmpl}} X'' \xleftarrow{\text{cmpl}} X' \tag{13}$$

で、定義に従うと、

$$s(X'') = \text{cmpl}(s(X)) \cap \text{cmpl}(s(X')) = {}^c s(X) \cap {}^c s(X') \tag{14}$$

$$s(X) = \text{cmpl}(s(X'')) = {}^c ({}^c s(X) \cap {}^c s(X')) = s(X) \cup s(X') \tag{15}$$

となる。つまり、 $s(X) \supset s(X')$ である。このように、この図式は部分集合関係 $s(X) \supset s(X')$ を強制する。これを省略して  $X \xleftarrow{\text{inc}} X'$  と書くことにする。すると  $X' \xleftarrow{\text{inc}} X \xleftarrow{\text{inc}} X''$  ,で  $s(X') = A, s(X'') = B$ の決定する断面 $s$ は、 $s(X) \supset A$ と $s(X) \supset B$ を満たす最小の集合 $s(X) = A \cup B$ を与える。和集合は $A \cup B = {}^c ({}^c A \cap {}^c B)$ という関係を使っても表すことができる。

### 3.4. 等間隔の点集合

次の図式を考える：

$$\begin{array}{c}
 X \times V \xleftarrow{\pi_2^{-1}} V \\
 \uparrow \text{add} \quad \searrow \text{id} \\
 X \xleftarrow{\pi_1^{-1}} X \xleftarrow{\text{id}} X' \xleftarrow{\text{inc}} X''
 \end{array} \tag{16}$$

ここで、 $\text{id} : X' \rightarrow X$ は恒等写像、 $\text{add} : X \times V \rightarrow X$ は $X$ の平行移動である。この図式の断面 $s$ で、 $\{V, X''\}$ への制限が、 $p$ を $X'$ の点として

$$s(V) = \{v\}, \quad s(X'') = \{p\}, \tag{17}$$

となるものを考える。まず、断面の定義より

$$\begin{aligned}
 s(X \times V) &= \pi_1^{-1}(s(X)) \cap \pi_2^{-1}(s(V)) \\
 &= \{(x, v) \in X \times V \mid x \in s(X)\}
 \end{aligned} \tag{18}$$

$$s(X') = \text{add}(s(X \times V)) = \{x + v \mid x \in s(X)\} \tag{19}$$

がわかるが、 $s(X) = \text{id}(s(X')) = s(X')$ であるから、(19)より、 $s(X)$ は、ある点 $x$ を含めばその $v, -v \in V$ による平行移動 $\dots, x - 3v, x - 2v, x - v, x, x + v, x + 2v, x + 3v, \dots$ を全て含まねばならない。ところが、 $s(X) = s(X') \cap s(X'') = \{p\}$ だから、 $\dots, p - 2v, p - v, p, p + v, p + 2v, \dots \in s(X)$ となる。従って

$$s(X) = s(X') \cap \{\dots, p - 2v, p - v, p, p + v, p + 2v, \dots\} \tag{20}$$

が必要である。今、(20)の右辺で $s(X), s(X')$ を定義すると、(17)(18)と併せて断面の条件を満たすことがわかる。また、制限が(17)と等しい断面は全て条件(20)を満たすから、 $s$ がそのようなものの中で最小であることも明らかである。よって、 $s$ は制限(17)の決定する断面であり、 $s(X)$ は、 $p$ を含み、間隔 $v$ で並んだ等間隔の無限個の点の集合である。

## 4. 特 徴

本節では、図式によるパターン表現の特徴を述べる。

### 4.1. 疎な表現で無限を表す

まず、ビットマップやグラフのような密な表現と違って、図式による表現は疎な表現を可能にする。例えば直線であれば、直線上にある画素を指定する代わりに、単純な図式と、1点と方向ベクトルという、ごく少数のパラメーターを指定することで表現する。

このため、例えば直線の方向だけ変えるのは簡単である。直線の図式(2)の $V'$ 上の断面 $s(V') = \{v\}$ の $v$ を変えればよい。また、これを $\{v\}$ ではなく $\{u, v\}$ という2つのベクトルにすると、点 $p$ を通る2本の直線を表すことになるし、 $s(X')$ の方を2点にすれば平行な2本の直線を表す。また、同じ図式で表されるパターン間の比較も、対応するパラメーターを比べることにより自然に定義される。

また、図式は有限の表現で形式的に無限集合を表す。

プリミティブとなる集合と写像を固定した上で、図式は一種のグラフのようにデータとして容易に表現可能である。ユークリッド空間や実数空間などの集合や、それらの間の写像は、それぞれシステム依存で近似して実装する必要があるが、重要なのは、データとしての構造を表す部分が、それらの近似に拠らないで表現されることである。例えば直線の例なら、点とベクトルの表現方法が異なるシステム間でデータを翻訳しても、直線であるということとは変化しない。

画像に現れる対象には、直線や曲線、またそれらで囲まれた図形など、特定の解像度に拠らずに表現されるべき本来離散的でないものが多いことから、有限の表現で無限集合であるこれらの対象を表すことが出来ることは、データを扱う上での柔軟性、一般性を得る上で不可欠である。

### 4.2. モジュラー性

部分集合(3.3)のときと同様に、2つ矢のある矢印 $\xleftarrow{\text{inc}}$ は、その両側にある集合の断面に一定の関係（この場合は(13)の決める関係）を強制するものとする。例えば、(2)と(8)を

$$X \xleftarrow{\text{line}} X' \dots V'$$

$$X \xleftarrow{\text{circle}} X' \dots \mathbb{R}$$

と書くことにすれば、図式：

$$\begin{array}{c}
 X' \leftarrow \text{line} X'' \cdots V \\
 \text{inc} \nearrow \\
 X \xrightarrow{\text{inc}} X''' \leftarrow \text{circle} X'''' \cdots \mathbb{R}
 \end{array}$$

と、制限  $s(X') = \{p\}$ ,  $s(V) = \{v\}$ ,  $s(X''''') = \{q\}$ ,  $s(\mathbb{R}) = \{r\}$  は、 $s(X)$  として直線と円からなる集合を与える。

このように、図式は自然なモジュールとして理解することができ、それらを組み合わせてより複雑な図式を作ることができる。組み合わせ方も単に和集合や共通部分をとるというだけでなく、以下に述べるような複雑な組み合わせ方が可能である。

### 4.3. 階層性

図式を使ったパターンのデータ表現は、以下のような意味で階層的である。

等間隔点集合の図式(16)を  $X \leftarrow \text{repeat} X'' \cdots V$  で表す。制限  $s(V) = \{v\}$ ,  $s(X'') = A$  の決定する断面  $s$  は、 $s(X)$  として  $X$  の任意の部分集合  $A$  を  $v, -v$  で平行移動して繰り返した集合を表す。

またこれに円周の図式を繋げると：

$$\begin{array}{c}
 X' \cdots \mathbb{R} \\
 \text{circle} \downarrow \\
 X \leftarrow \text{repeat} X'' \cdots V
 \end{array}$$

制限  $s(\mathbb{R}) = \{r\}$ ,  $s(V) = \{v\}$ ,  $s(X') = \{p\}$  の決定する断面  $s$  は、 $s(X)$  として  $\{\dots, p-2v, p-v, p, p+v, p+2v, \dots\}$  を中心にする半径  $r$  の無限個の円周を与える。このように、 $X'$  は円の中心の空間、 $\mathbb{R}$  は円の半径の空間、 $V$  は円の間隔の空間というように、階層的な空間が最終的な点集合  $X$  の部分集合を決定している。

### 4.4. 再帰性

等間隔点集合の図式(16)は、一つの点を等間隔で繰り返すという再帰的な集合の定義をしている。このように、図式によって再帰的な定義をすることができる。例えば、(16)を少し変えて

$$\begin{array}{c}
 V \times M \leftarrow \pi_2^{-1} M \\
 \text{mult} \nearrow \\
 V \leftarrow \text{id} V' \xrightarrow{\text{inc}} V''
 \end{array} \quad (21)$$

とする。ここで  $M$  は  $V$  上の線形写像の集合、 $\text{mult} : V \times M \rightarrow V$  はベクトルに線形写像を適用する写像である。この図式の断面  $s$  で、制限  $s(M) = \{A\}$ ,  $s(V'') = \{v\}$  で決定されるものは、線形写像  $A$  によるが、例えば  $A$  が回転ならば円周上の点の集合になる。

図式による再帰的定義の力を端的に示すための例として、フラクタル図形であるマンデルブロ集合<sup>(注2)</sup>の表現を与える：

$$\begin{array}{c}
 \mathbb{C} \times \mathbb{C} \xrightarrow{\text{inc}} \mathbb{C} \times \mathbb{C} \xrightarrow{f} \mathbb{C} \times \text{bool} \leftarrow \pi_2^{-1} \text{bool} \\
 \pi_1 \times (\text{sq} \circ \pi_2) \downarrow \quad \pi_1 \times \text{add} \nearrow \quad \text{inc} \downarrow \quad \pi_1 \downarrow \\
 \mathbb{C} \times \mathbb{C} \quad \mathbb{C}'' \times \mathbb{C}'' \quad \mathbb{C}' \xrightarrow{\text{cml}} \mathbb{C}
 \end{array}$$

ここで  $\mathbb{C}$  は複素数全体の集合、各写像は

$$\pi_1 \times (\text{sq} \circ \pi_2) : (c, z) \mapsto (c, z^2) \in \mathbb{C} \times \mathbb{C}$$

(注2)：漸化式“ $z_0^c = 0, z_{n+1}^c = (z_n^c)^2 + c$ ”で定義される複素数列  $\{z_n^c\}$  が無限大に発散しない複素数  $c$  の集合。複素数列  $\{z_n^c\}$  が無限大に発散することと、ある  $n$  について  $|z_n^c| > 2$  であることが同値であることが知られている。

$$\pi_1 \times \text{add} : (c, z) \mapsto (c, c+z) \in \mathbb{C} \times \mathbb{C}$$

$$f : (c, z) \mapsto (c, |z| > 2) \in \mathbb{C} \times \text{bool}$$

で定義される。この図式の断面  $s$  で、 $\{\mathbb{C}'' \times \mathbb{C}'', \text{bool}\}$  への制限  $s(\mathbb{C} \times \{0\}) = \mathbb{C} \times \{0\}$ ,  $s(\text{bool}) = \{\text{true}\}$  で決定されるものは、 $s(\mathbb{C})$  としてマンデルブロ集合を与える。

## 5. 実装上の考察

本節では、図式とその断面によるパターンの表現を実装するときの考察を述べる。本節に述べる手法は、一般の冪集合間の写像には対応せず、 $\mathcal{M}$  は誘導された写像と補集合写像  $\text{cml}$  (1) のみからなることとする。

### 5.1. レンダリング

図式によって表現されたパターンをビットマップなどの密な表現にレンダリングすることを考える。

今、図式  $(\mathcal{S}, \mathcal{M})$  と集合族  $\mathcal{D} \subset \mathcal{S}$  の断面  $s_{\mathcal{D}}$  で決定される  $(\mathcal{S}, \mathcal{M})$  の断面  $s$  について、 $s(X)$  を密な表現にしたいとする。

断面の与える部分集合は一般には無限集合だが、密な表現は、有限個の点についてそれが断面の与える部分集合に含まれるかどうかを決定することで得られる。例えば  $X$  がユークリッド平面ならば、これに  $x$ - $y$  座標系を導入して座標の整数部分をとる写像を  $p : X \rightarrow \mathbb{Z}^2$  とし、図式を  $(\mathcal{S} \cup \{\mathbb{Z}^2\}, \mathcal{M} \cup \{p\})$  に拡張すると、 $s \circ s(\mathbb{Z}^2) = p(s(X))$  として一意に拡張されるので、 $\mathbb{Z}^2$  の有限範囲  $P$  を画素の集合として、その各点  $q \in P$  について  $q \in s(\mathbb{Z}^2)$  かどうかを決定すれば、2値のビットマップが得られる。写像  $p : X \rightarrow \mathbb{Z}^2$  の選択が、解像度等の決定に相当する。

### 5.2. パターン認識

与えられた図式と断面によって表現されるパターンをビットマップなどの密な表現中に見つけることは、基本的にレンダリングと同じであり、計算量はより少なくなる。密な表現を表した部分集合の各点について、与えられた図式と断面で表現される部分集合に入っているかどうかを調べればよいのである。これに対して、レンダリングでは部分集合に本当に入っていない点についても、それを確かめるための計算が必要である。もちろん、例えば線の方向のようなパラメータの変ったものは別々にレンダリングしなければならない。

パターン認識においては、階層性(4.3参照)が重要である。例えば、正方形の4頂点に小さな円のあるパターンを探すとき、このパターン全体を探すのではなく、まず小さい円を見つけて、円の中心の空間の中に正方形を見つけるということが、図式による表現の階層性から自然にできることが期待される。

### 5.3. 条件の整理

したがって、与えられた点が断面の与える部分集合に含まれるかどうかを決定することが重要である。そこで断面  $s$  について  $x \in s(S), S \in \mathcal{S}$  となる条件を整理する。 $\mathcal{M} \ni \varphi : \mathcal{P}(S) \rightarrow \mathcal{P}(T)$  は  $f : S \rightarrow T$  から誘導されるか ( $\varphi = f$  と書く)、 $f : T \rightarrow S$  から誘導されるか ( $\varphi = f^{-1}$  と書く) あるいは  $\varphi = \text{cml}$  かである。

まず  $\text{in}(S) \neq \emptyset$  のとき ( $T = \text{dm}(\varphi)$  として)

$$x \in s(S) \iff$$

$$\forall \varphi \in \text{in}(S), \begin{cases} \exists y \in s(T), f(y) = x & (\varphi = f) \\ f(x) \in s(T) & (\varphi = f^{-1}) \\ x \notin s(T) & (\varphi = \text{cml}) \end{cases} \quad (22)$$

次に  $\text{out}(S) \ni \varphi : \mathcal{P}(S) \rightarrow \mathcal{P}(T)$  とする。 $\varphi = f$  のとき、 $f$  が単

射なら  $f(x) \in s(T) \Rightarrow x \in s(S)$  がいえる。また  $\varphi = f^{-1}$  のときは、 $\exists y \in s(T), f(y) = x \Rightarrow x \in s(S)$  がわかる。最後に、 $\varphi = \text{cml}$  ならば  $x \in s(T) \Rightarrow x \notin s(S)$  である。

#### 5.4. 基本アルゴリズム

まず基本的なアルゴリズムを示す。上記の条件を使って、以下のアルゴリズムは  $x \in X$  が  $s(X)$  に入るか否かを決定する。アルゴリズムはどれも  $yes, no, unknown$  の何れかの値を返す。

```

IN-SECTION( $x, S$ )
1  if  $S \in \mathcal{O}$ 
2  then return  $x \in s(S)$  (yes or no)
3  switch
4  case  $\exists \varphi \in \text{in}(S), \text{MAP-REVERSE}(x, \varphi) = no$  :
5  return no
6  case  $\exists \varphi \in \text{out}(S), \text{MAP-FORWARD}(x, \varphi) \neq unknown$  :
7  return MAP-FORWARD( $x, \varphi$ )
8  case  $\forall \varphi \in \text{in}(S), \text{MAP-REVERSE}(x, \varphi) = yes$  :
9  return yes
10 return unknown

MAP-REVERSE( $x, \varphi$ )
1  if  $\varphi = \text{cml}$ 
2  then return not IN-SECTION( $x, \text{dm}(\varphi)$ )
3  else for each  $y \in \varphi^{-1}(x)$ 
4  do if IN-SECTION( $x, \text{dm}(\varphi)$ ) = yes
5  then return yes
6  return no

MAP-FORWARD( $x, \varphi$ )
1  if  $\varphi = \text{cml}$ 
2  then if IN-SECTION( $x, \text{cdm}(\varphi)$ ) = yes
3  then return no
4  else return unknown
5  else if  $\varphi = f^{-1}$ 
6  then if  $\exists y \in f^{-1}(x), \text{IN-SECTION}(y, \text{cdm}(\varphi)) = yes$ 
7  then return yes
8  else if IN-SECTION( $f(x), \text{cdm}(\varphi)$ ) = yes AND
9   $\nexists y \in f^{-1}(f(x)), y \neq x, \text{IN-SECTION}(y, \text{dm}(\varphi)) = yes$ 
10 then return yes
11 return unknown

```

ここで  $\varphi^{-1}(x)$  は、 $\varphi = f$  ならば  $f^{-1}(x)$  を、 $\varphi = f^{-1}$  ならば  $\{f(x)\}$  を表し、**not** は **not yes = no, not no = yes, not unknown = unknown** で定義される。

#### 5.5. 考察

上に挙げた基本アルゴリズムは図式をグラフのように探索する。そのとき IN-SECTION の **switch** における  $\varphi$  の選択、また MAP-REVERSE の3行目と MAP-FORWARD の6行目、9行目における  $y$  の選択で分岐するが、これらをどの順序で試すかが問題になる。(上では意図的にそこを曖昧にしてある。) 図式(2)や(8)のようにループのない図式ではどのような順で探索しても停止するが、閉路がある場合には、どのような順で探索するかによって無限ループに陥る可能性がある。それを避けるためには、 $\mathcal{O}$  に含まれる集合への経路を優先しながら幅優先探索をする。無限ループを避けるために探索の深さに一定の限界も設ける。

また並行して、 $\mathcal{O}$  に含まれる集合上の断面から逆向きに情報を伝播する。上記アルゴリズムと向きがちょうど逆なので、両者によって効率の上で相補的に探索することができる。例えば、MAP-REVERSE で  $\varphi = f^{-1}$  のときは  $f(x)$  のみを調べればいいが、 $\varphi = f$  のときは潜在的には無限個の  $y \in f^{-1}(x)$  を調べなければならない。これが逆向きのアルゴリズムではちょうど逆転する。

もうひとつ実装上重要なのは、集合の扱いである。まず、集合はできる限り記号的に処理するのが望ましい。上記アルゴリズムは写像の逆像などの無限集合を扱うので、集合の記号的処理が必要である。記号的に処理するとは、例えば部分集合を述

語論理として扱う場合である。これには例えば集合論的プログラム言語 SETL [7] で使われる技術が応用可能であろう。

実装にはこのように詳細な選択肢が存在するが、このような実装依存性を表現から分離できていることがわかる。

## 6. 結 び

人間の視覚系には、そのプロセスの初期段階において、物体の認識に必要な高次の知識とは独立した、また画像が生成される過程からも独立した、知覚体制化をする機構があるように思われる [11]。初めて見るものにさえ、規則性や連続性を、いくつもの階層で見つけて組織付けをする能力が人間にはある。また、そのような初期的組織付けの結果は、最終的に画像を理解し意味づけするレベルまで生き延びる、つまり、最初に見えたものを単位として、蓄えられた高次の知識で解釈したり、新しいモデルを学習したりするようでもある。そのような知覚レベルに近い「組織付けられたもの」を自然に表現し、知覚体制化を記述するための自然な言語を提供することが、本稿で提案したパターン表現方法の目標の一つである。

本稿では、新しいパターン表現の定義を与え、その特徴を述べた。これを実際に実装するのに必要になるアルゴリズムについても少し述べたが、多くは今後の課題である。特に、第5節で図式とその断面が与えられた場合の実装上のいくつかの点について考察したが、図式が与えられていないときに、密な表現で表されたデータの中に図式で効率よく表現できるパターンを見つけることができれば、非常に有用である。この、パターン発見とでも呼ぶべき、自明な表現より効率のよい表現を自動的に見つけることは今後の課題だが、それこそが本研究の元々の動機でもあり、今後の目標である。

謝辞 本研究の一部は、第15回掘情報科学振興財団研究助成、平成18年度稲盛財団研究助成、平成17年度スズキ財団科学技術研究助成、平成17年度中部電力基礎技術研究所研究助成、文部科学省科学研究費補助金萌芽研究19650065の補助を受けた。

## 文 献

- [1] I. Cohen, N. Ayache, and P. Sulger. Tracking Points on Deformable Objects Using Curvature Information. In *ECCV '92*, pp. 458–466, 1992.
- [2] K. S. Fu. *Syntactic Methods in Pattern Recognition*. Academic Press, 1974.
- [3] Y. Gdalyahu and D. Weinshall. Flexible Syntactic Matching of Curves and Its Application to Automatic Hierarchical Classification of Silhouettes. *IEEE Trans. PAMI* **21**(12):1312–1328, 1999.
- [4] U. Grenander. *Lectures in Pattern Theory*, volumes I-III. Springer-Verlag, 1976, 1978, 1981.
- [5] I. H. Jermyn and H. Ishikawa. Globally Optimal Regions and Boundaries as Minimum Ratio Cycles. *IEEE Trans. PAMI* **23**(10):1075–1088, 2001.
- [6] T.-L. Liu and D. Geiger. Approximate Tree Matching and Shape Similarity. In *ICCV'99*, pp. 456–462, 1999.
- [7] J. T. Schwartz and R. B. K. Dewar. *Programming With Sets: An Introduction to SETL*. Springer, 1986.
- [8] K. Siddiqi and B. B. Kimia. Parts of visual form: Computational aspects. *IEEE Trans. PAMI* **17**(3):239–251, 1995.
- [9] K. Siddiqi, A. Shkouflandeh, S. J. Dickinson, and S. W. Zucker. Shock Graphs and Shape Matching. *Int. J. Comp. Vision* **35**(1):13–32, 1999.
- [10] Z. Tu, X. Chen, A. Yuille, and S.-T. Zhu. Image Parsing: Unifying Segmentation, Detection, and Recognition. *Int. J. Comp. Vision* **63**(2):113–140, 2005.
- [11] A. P. Witkin and J. M. Tenenbaum. On the Role of Structure in Vision. In *Human and Machine Vision*, pp. 481–543. Academic Press, 1983.