

## グローバルな特徴を利用した画像検索

1D-5

望月貴裕 伊藤崇之

NHK 放送技術研究所

### 1. はじめに

各映像・画像メディアで、電子化された画像情報が大量に蓄積されつつある。そうした過去の映像資産を積極的に再利用する必要性が高まるにつれ、画像検索分野が注目され様々な検索手法が提案されている。通常の画像検索は、キーワード検索、オブジェクト検索などのように、被写体の内容（「～が映っているか」等）を検索のための手がかりとする手法が主流である。しかし、扱うべき画像が多様化するに従い、「全体的な印象が似ている」あるいは「構図が似ている」等の人間の感性に適合した画像検索手法も有効となると考えられる。また、将来的には画像検索が家庭の映像情報機器で行われる可能性もあり、その場合には人間の感性に合致した手法が有効となる。

本稿では、輝度変化の画像全体でのフラクタルを利用した複雑度、およびエッジ量の画像全体でのフラクタルを利用した複雑度という、複数のグローバルな画像特徴量の組み合わせによる画像検索手法を提案する。

### 2. 画像特徴量

#### 2.1. 従来の技術

近年、画像の被写体の内容ではなく画像全体の類似性に基づいて画像検索を行う技術が研究されている。フラクタル次元を特徴量として用いる手法[文献 1][文献 2]、図形の概略形状、周波数特徴、局所相関の各特徴をベクトルとして用いる手法[文献 3]などが提案されている。

フラクタル次元は画像の相似性を表す特徴量で、複雑な画像ほど次元は大きく、単純な画像では次元が小さくなるため、画像の複雑さという人間の感性に類似した特徴を表現する特徴量として注目されている。一般に、天然の物体表面や自然の風景などではフラクタル次元は求められるが、ビルなどの街の風景やデザイン画など人工的な物については次元は求められない。[文献 1][文献 2]では、このような性質を持つフラクタル次元を特徴量として用いているが、フラクタル次元そのものは情報が少ないため、検索対象画像の種類を限定しなくてはならない。すなわち、デザイン画のみ、テキスト画像のみ、あるいは自然画のみなど、特定の種類の画像だけを対象とせざるを得ない。[文献 3]は、フラクタル次元とは異なる複数の特徴量を用いているものであるが、

デザイン画にしぼった上で必要な画像特徴を求めている。このように、これまでの画像全体の特徴量を利用した画像検索手法の多くは、対象となる画像の種類をある程度限定する必要があった。

#### 2.2. 本研究で用いる特徴量

本研究では、人間が画像の初見から受ける印象（構図、複雑度など）をよりよく表すと思われる特徴量として、輝度変化のフラクタル性およびエッジ量のフラクタル性を組み合わせたものを用いる。ただし、従来の、フラクタル次元を特徴量とする手法ではなく、そのフラクタル次元を求める際に抽出される数値列全体を特徴量することで、データベースの画像の種類に依存せずに構図、複雑度等が類似していると思われる画像を検索することを可能とした。

#### 2.3. 輝度変化のフラクタル性(特徴量 1)

輝度変化のフラクタルに基づく特徴量は自然界に存在するテクスチャに対して良好な分類性能があることが知られている。そのフラクタル次元の計算方法としては、ミンコフスキーブランケット[文献 1]が有名であるが、掛村らは画像を分割した各格子の四角の輝度値による「仮想体積」なるものを定義し、格子一辺の長さと同体積の一格子あたりの平均値を log-log プロットしてその傾きを計算し高速にフラクタル次元を求めている[文献 2]。

本手法では、この掛村らの手法を計算法として適用した。ただし、フラクタル次元を求めそれを特徴量とするのではなく、log-log プロットした際の縦軸数値列全体（グラフの形そのものと考えられる）を特徴量とすることで、フラクタル性の少ない画像にも適用することを可能としている。詳細を図 1 に示す。

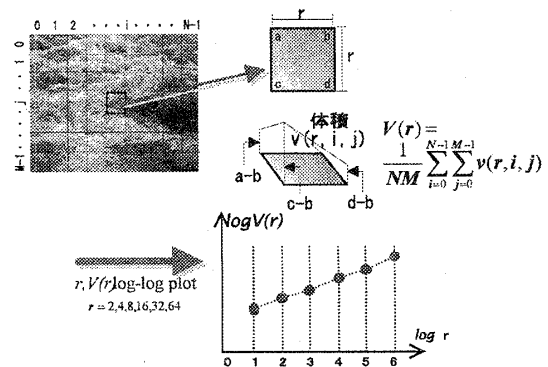


図 1

ここで、図 2 のように、複雑度が類似した物体が画像上全く異なる場所に存在するような 2 枚の画像を比較す

ることを考える。この2枚の画像からそれぞれ、上記の手法で数値列を抽出した場合、複雑度が同じであるため2つの数値列の類似度は高くなると思われる。しかし、この2枚の画像を「類似していない」としたい場合、すなわち構図・位置を重視して類似度を考えたい場合は、画像全体からの数値列のみでは不十分であると考えられる。そこで、本手法では、画像全体から求めた数値列だけでなく、画像を12×12分割したそれぞれの領域からの数値列もあわせて特徴量とする(1+12×12=145個の数値列)ことにした。ここで、12という値は実験を繰り返した上で決定した値である。以降、上述の特徴量を次のように記述することにする。

- ・画像全体から求めた数値列  
→ 「特徴ベクトル 1-1」
- ・12×12 分割した各領域から求めた数値列を接続した数値列  
→ 「特徴ベクトル 1-2」

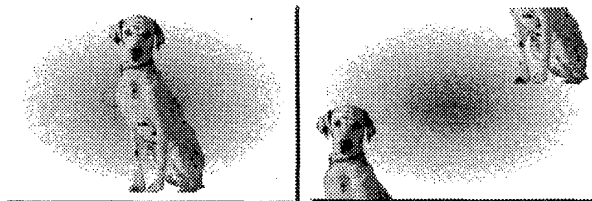


図 2

#### 2.4. エッジ量の複雑性(特徴量 2)

輪郭(エッジ量)についても、多くの自然物にはフラクタル性があることが知られている。例えばアマゾン川に対し、粗視化の度合い(格子の一边)rと川全体を被覆する格子の数 N(r)を log-log プロットすると直線になり、その傾きからフラクタル次元が求まる[文献 4]。本手法でもエッジ量の複雑性の計算法として、上記の格子被覆手法を適応した。さらに本手法ではより多くの特徴情報を抽出するために次の2点の拡張を行った。

1. 2値化・細線化したエッジ画像よりもそれらの処理を行う前のエッジ画像のほうがより多くの特徴情報を含んでいると考えられるため、本手法で扱うエッジ画像は「各画素値に、エッジ抽出計算をしたそのままの値が入っている」ものとする。
  2. 各格子が、エッジ領域を「被覆するもの」か「被覆しないもの」か判別するための含有エッジ量の閾値  $E_{th}$ をその格子の面積(=r<sup>2</sup>)に比例したものとする。
- 以上2点を踏まえ、N(r)を次のように定める。

式 1

$$N(r) = \text{エッジ量の合計が } E_{th}(=R \times Op \times r^2) \text{ 以上である格子の数}$$

ここで、R は格子内の面積比率をあらわす比例定数(0 ≤ R ≤ 1)である。また、エッジ計算時のエッジ抽出オペレータに依存して画素値は大きく変化するため、 $E_{th}$ もエッジ抽出オペレータによって変化させる必要がある。Opはそのための定数である。

このエッジのフラクタル性についても log-log プロットした際の縦軸数値列全体(グラフの形そのものと考えられる)を特徴量とすることで、フラクタル性の少ない画像にも適用可能とする。詳細を図 3に示す。

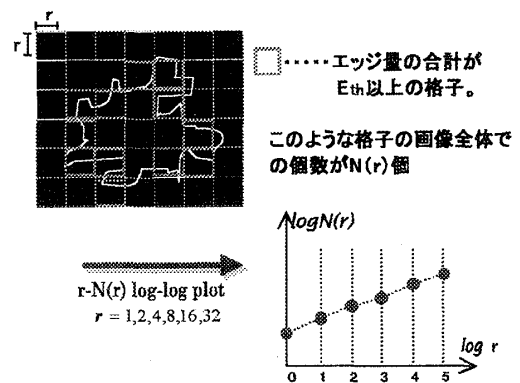


図 3

さらに、輝度変化のフラクタル性の際と同様の理由により、画像全体から求めたグラフだけでなく、画像を12×12分割したそれぞれの領域からの数値列もあわせて特徴量とすることにした。以降、上述の特徴量を次のように記述することにする。

- ・画像全体から求めた数値列  
→ 「特徴ベクトル 2-1」
- ・12×12 分割した各領域から求めた数値列を接続した数値列  
→ 「特徴ベクトル 2-2」

#### 2.5. 特徴量の抽出

2.3節、2.4節で定義した各特徴量ベクトルの抽出アルゴリズムの詳細は次の通りである。

●特徴ベクトル 1-1 の抽出(図 1参照)

1. k=0 とする。
2. 入力画像を一边 r(=2<sup>k+1</sup>)画素の格子で分割する。
3. その各格子について仮想体積を計算し、その一格子あたりの平均を V<sub>1</sub>(k)とする。
4. k=k+1とし2へ戻る。kがある値 K<sub>Th1</sub>へ達したならば5へ。

5. 数値列  $a_{11}=(\log V_1(0), \log V_1(1), \dots, \log V_1(K_{Th1}-1))$  を特徴ベクトル 1-1 とする。

●特徴ベクトル 1-2 の抽出

入力画像を縦 12 個、横 12 個に分割

1. その 144 個の各領域  $(i,j)$  について、その領域を 1 枚の画像とみなし「特徴ベクトル 1-1 の抽出」と同様の処理を行なう。その処理で計算された要素数  $K_{Th2}$  の数値列を  $a_1(i,j)=(\log V_2(i,j,0), \log V_2(i,j,1), \dots, \log V_2(i,j, K_{Th2}-1))$  とする。

2. 全ての領域  $(i,j)$  に対する 2 の処理によって計算された要素数  $144 \times K_{Th2}$  の数値列

$$a_{12}=(a_1(0,0), a_1(0,1), \dots, a_1(i,j), \dots, a_1(11, 11))$$

を特徴ベクトル 1-2 とする。

●特徴ベクトル 2-1 の抽出(図 3 参照)

1. エッジ画像を抽出する。
2.  $k=0$  とする。
3. 入力画像を一辺  $r(=2^k)$  画素の格子で分割する。
4. 全格子の中で、エッジ量の合計が  $E_{th}$  (式 4 参照) 以上含む格子の数を  $N_2(k)$  とする。
5.  $k=k+1$  とし 2 へ戻る。  $k$  がある値  $K_{Th1}$  へ達したならば 5 へ。
6. 数値列  $a_{21}=(\log V_1(0), \log V_1(1), \dots, \log V_1(K_{Th1}-1))$  を特徴ベクトル 1-1 とする。

●特徴ベクトル 2-2 の抽出

1. エッジ画像を縦 12 個、横 12 個に分割
2. その 144 個の各領域  $(i,j)$  について、その領域を 1 枚の画像とみなし「特徴量ベクトル 2-1 の抽出」と同様の処理を行なう。その処理で計算された要素数  $K_{Th2}$  の数値列を  $a_2(i,j)=(\log N_2(i,j,0), \log N_2(i,j,1), \dots, \log N_2(i,j, K_{Th2}-1))$  とする。
3. 全ての領域  $(i,j)$  に対する 2 の処理によって計算された要素数の  $144 \times K_{Th2}$  の数値列

$$a_{22}=(a_2(0,0), a_2(0,1), \dots, a_2(i,j), \dots, a_2(11,11))$$

を特徴ベクトル 2-2 とする。

3. 類似度

2章で定義した特徴ベクトルについて、検索したい画像(入力画像)とデータベース中の画像  $i$  の類似度を計算する。

3.1. 各類似度の計算

入力画像の各特徴ベクトルと、データベースの画像の特徴ベクトルとの類似度をすべてのデータベース画像について計算する。以降、便宜上、入力画像の特徴ベクトル 1-1, 1-2, 2-1, 2-2 をそれぞれ  $a'_{11}, a'_{12}, a'_{21}, a'_{22}$  と記述し、データベース中の画像  $i$  ( $i=0, 1, \dots, N-1$ , ここで、 $N$  はデータベースの全画像の数とする)の特徴ベクトル 1-1, 1-2, 2-1, 2-2 を

それぞれ  $a_{11}[i], a_{12}[i], a_{21}[i], a_{22}[i]$  記述する。さらに、

- ・特徴ベクトル 1-1 に関する類似度:  $S_{11}[i]$
- ・特徴ベクトル 1-2 に関する類似度:  $S_{12}[i]$
- ・特徴ベクトル 2-1 に関する類似度:  $S_{21}[i]$
- ・特徴ベクトル 2-2 に関する類似度:  $S_{22}[i]$

と表すことにする。ここでまず、2 つの長さ  $n$  のベクトル  $a=(a_0, a_1, \dots, a_{n-1}), b=(b_0, b_1, \dots, b_{n-1})$  の類似度を求める関数  $\text{Sim}(a, b, n)$  を次のように定義する。

式 2

$$\text{Sim}(a, b, n) = \frac{\sum_{k=0}^{n-1} \min(a_k, b_k)}{\sum_{k=0}^{n-1} \max(a_k, b_k)}$$

この関数  $\text{Sim}$  をもちいて各類似度  $S_{kl}[i]$

( $k=1, 2, l=1, 2, 0 \leq S_{kl}[i] \leq 1$ ) (式 4 参照) を次のように計算する。

式 3

$$S_{11}[i] = \text{Sim}(a'_{11}, a_{11}[i], K_{Th1})$$

$$S_{12}[i] = \text{Sim}(a'_{12}, a_{12}[i], K_{Th2})$$

$$S_{21}[i] = \text{Sim}(a'_{21}, a_{21}[i], K_{Th1})$$

$$S_{22}[i] = \text{Sim}(a'_{22}, a_{22}[i], K_{Th2})$$

式 3 による各類似度の計算を、全ての画像  $i$  ( $i = 0, 1, \dots, N-1$ ) について行う。

3.2. 特徴ベクトルの使い分け

2.3 節でも触れたが、図 2 のように複雑度が類似した物体が画像上全く異なる場所に存在するような 2 枚の画像を比較する場合、

ア) 構図・位置関係よりも、画像の全体的な複雑性の類似を重視するならば、この 2 枚の画像は「類似度が高い」と考えられる。

イ) 構図・位置関係を重視して類似を考えるならばこの 2 枚の画像は「類似度が低い」と考えられる。

上記のア) のような場合は画像全体から抽出した特徴量ベクトル 1-1 および 2-1 を重視して類似度を計算するべきであり、イ) のような場合は、画像をより細かく分割した格子単位の比較を行うことで構図および位置情報を保持して類似度を計算することが可能な特徴量ベクトル 1-2 および 2-2 に重きを置いたほうがよいと考えられる。また、例えば、一般的に輝度変化についてのフラクタル性が強いと考えられている「山・海」などを含んだ画像の検索を行いたいときは、特徴ベクトル 1-1, 1-2 を用いて類似度を計算したほうがよいであろうし、輪郭線のフラクタ

ル性が強いとされる「(はっきり輪郭の出た)地図・海岸線」などの画像について検索するのならば、特徴ベクトル 2-1,2-2 に重きを置くべきであろう。このように「何に重きをおいて検索をするか」というユーザの意図に従って類似度計算に用いる特徴ベクトルを使い分けることが検索精度の向上のために非常に有効と思われる。

そこで、まず検索したい画像(入力画像)とデータベース中のある画像  $i$  との類似度  $S[i]$  を次のように定義し、

**式 4**

$$S[i] = W_G(W_I S_{11}[i] + W_E S_{21}[i]) + W_S(W_I S_{21}[i] + W_E S_{22}[i])$$

各重み  $W_G, W_S, W_I, W_E$  の値の設定を、ユーザが次の9種類から選択できるようにした。

検索名	$W_G$	$W_S$	$W_I$	$W_E$
Intensity Global	1	0	1	0
Edge Global	1	0	0	1
Global	1	0	1	1
IntensityStructural	0	1	1	0
Edge Structural	0	1	0	1
Structural	0	1	1	1
Intensity Normal	1	1	1	0
Edge Normal	1	1	0	1
Normal	1	1	1	1

表 1 各検索における重み付け

表 1の各検索の意味は次のように考えることができる。

<Intensity Global Search>

輝度変化に重点を置き、かつ構図・位置に関しては緩い条件で検索したい場合。

例えば、「どの部分でもいいからとにかく大部分山が映っている画像」を検索する場合など。

<Edge Global Search>

エッジに重点を置きかつ構図・位置に関しては緩い条件で検索したい場合。

例えば「地形を問わず(輪郭のはっきりした)地図」を検索する場合など。

<Global Search>

構図・位置に関して緩い条件で検索したいが、輝度変化・エッジどちらに重点を置くかは判断できない場合。

<Intensity Structural Search>

輝度変化に重点を置き、かつ構図・位置に関して厳しく検索したい場合。

例えば「海の風景で、かつ、水平線が画面の中央にある」画像を検索したい場合など。

<Edge Structural Search>

エッジに重点を置き、かつ構図・位置に関して厳しく検索したい場合。

例えば「画面の真中にぼっかり浮かぶ島の地図」を検索したい場合など。

<Structural Search>

構図・位置に関して厳しく検索したいが、輝度変化・エッジどちらに重点を置くかは判断できない場合。

<Intensity Normal Search>

「とりあえず」、輝度変化に重点を置いて検索したい場合。

<Edge Normal Search>

「とりあえず」、エッジに重点を置いて検索したい場合。

<Normal Search>

「とにかくとりあえず」検索したい場合。

このようにしてユーザがどのような検索を行うかによって選択・決定した各重み  $W_G, W_S, W_I, W_E$  の値および、式 3による各類似度  $S_{kl}[i](k=1,2, l=1,2)$  の値を用いて、式 4に従い、検索したい画像(入力画像)とデータベース中のある画像  $i$  との類似度  $S[i]$  が計算される。

この類似度  $S[i]$  の計算をすべての画像 ( $i=0,1,\dots,N-1$ ) について行った後、 $S[i]$  の値の大きい、つまり類似度が高い順にデータへ順位付けをする。この順位が検索結果である。

4. 検索実験例

4.1. 実験に用いたデータベース

データベース画像は、イラスト、絵画、テクスチャ、実画像など 3157 枚、大きさは全て正規化(240x162)してある。図 4に画像例を示す。

4.2. 実験例

本実験では、まず式 1のパラメータを  $R=2/3, Op=180$  に設定した。この  $Op$  の値は、エッジ抽出オペレータとして雑音に強いとされる Sobel を適用したことに依存している。また、特徴量抽出時(2.5節参照)に使用するパラメータ  $K_{Th1}, K_{Th2}$  は、画像の大きさ(240x162)を考慮してそれぞれ、 $K_{Th1}=6, K_{Th2}=3$  に設定した。つまりこれにより、各特徴ベクトル抽出時の格子一辺の最大値がそれぞれ、1-1抽出時: $2^6=64$ 、1-2抽出時: $2^3=8$ 、2-1抽出時: $2^6=32$ 、2-2抽出時: $2^3=4$ 、となる。図 5~図 8に実験結果(入力画像と、その入力画像と類似度の高い上位 20 枚)を数例示す。実験例 1,2 は「画像全体を(テクスチャが)埋めている」、実験例 3 では、「画像中央に(大きな楕円が)ある」といったように、これらの

実験については構図・位置的特徴を重要視した検索、すなわち表 1の「Structural Search」を適用し検索を行った。また、実験例 4 については、「木」や「細かい葉の集まり」に見られる輝度のフラクタル性が画像内のどこに存在してもよいという考え方で「Global Intensity Search」を適用し検索を行った。主観的な評価になってしまうが、各実験例とも（とくに実験例 1~3については）入力画像と構図・複雑度などが類似していると思われる画像がおおむね上位に来ているといえるであろう。

5. まとめ

画像全体およびある程度分割した画像の各領域から、フラクタル次元（単一のスカラー）ではなく数値列として抽出される、輝度変化およびエッジ量のフラクタルを利用した複雑度といった複数の特徴量を、ユーザの検索意図によって選択された重み分配に従い組み合わせた類似度を尺度とすることで、データベース画像の種類に制限を設けることなく、構図や複雑度等人間の「似ている」感覚に近いと思われる検索を行う手法を提案し、特にテクスチャ画像や、背景がプレーンな画像（構図的、位置的要素が特徴として重要視されると思われる画像）に対して良好な検索が可能であることを示した。

本手法の応用として、類似度を用いた動画画像カット検出、大量の TV データからの構図をキーとした検索、例えば、野球の試合の画像を集めたデータベースから「バッターのバストショット」あるいは「バッテリーのルーズショット」を検索する等のシーン検索への利用などが考えられる。

今後の課題としては、まず、構図・位置を重視した検索(Structural)に比べやや精度が低かった構図・位置を重視しない検索(Global)の精度向上があげられる。また、現在実験に用いているデータベースは、絵画風の画像が大半を占めているので、もっと実画像を増やす必要がある。また、現在の検索精度評価の尺度が、検索結果を眺めての主観のみになっているので、検索精度を定量化し客観的な評価を行う必要がある。

参考文献

文献 1 林,田中,“混合フラクタルモデルに基づく画像特徴量とその画像検索応用における性能評価”,情処会論文誌 Vol.38 No.10 (1997)  
 文献 2 掛村,東,入江,“仮想体積を用いたテクスチャ変数とその応用”,信学会論文誌 Vol.J80-D-II No.9.pp2411-2420 (1997.9)  
 文献 3 大津他,“パターン認識 理論と応用”,朝倉書店  
 文献 4 高安秀樹 “フラクタル”,朝倉書店



図 4 画像例

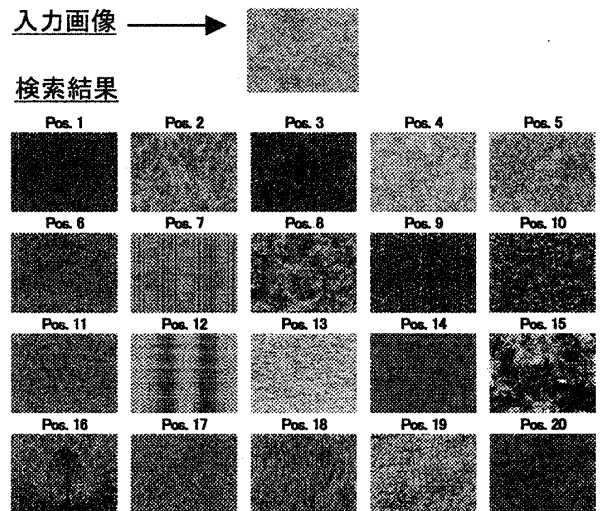


図 5 実験例 1 (Structural Search)

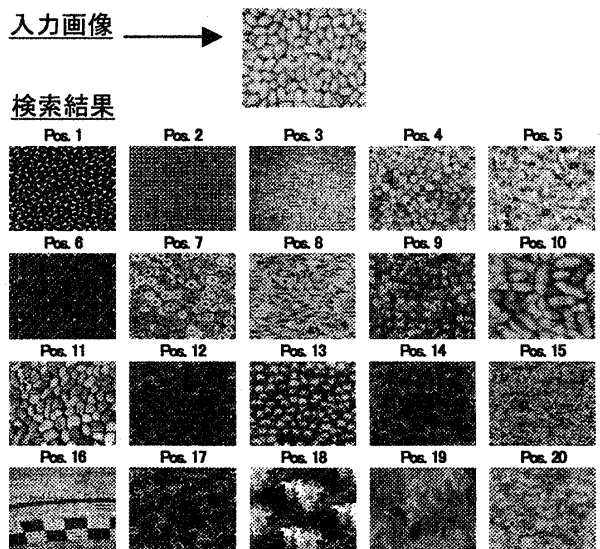


図 6 実験例 2 (Structural Search)

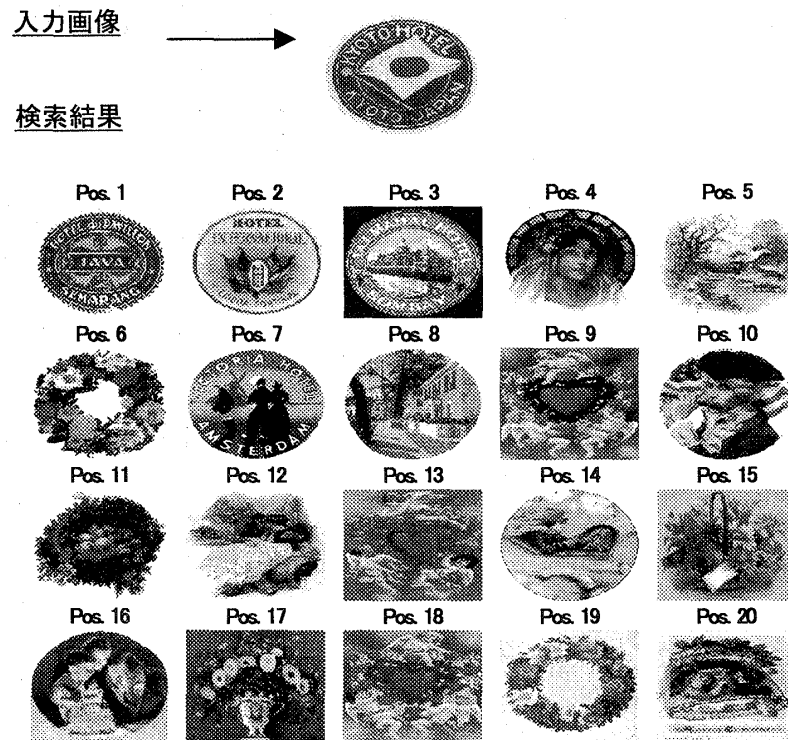


図 7 実験例 3 (Structural Search)

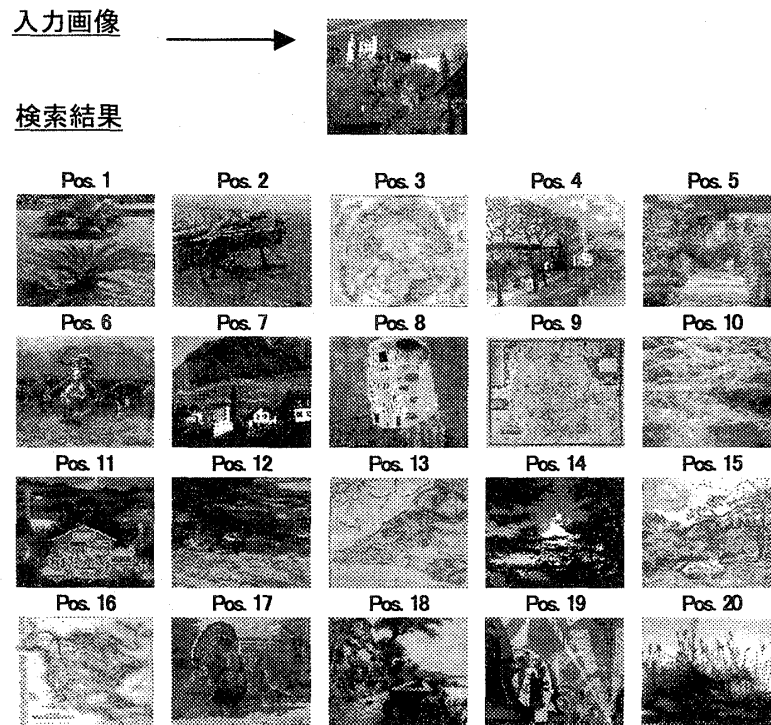


図 8 実験例 4 (Global Intensity Search)