

ウェーブレットとSOMを用いた エッジ特徴による類似画像検索システムの開発

久保 正明[†], ザヘル アグバリ[¶], 牧之内 顕文[¶]

[†]九州大学大学院システム情報科学府

[¶]九州大学大学院システム情報科学研究院

概要

本論文ではエッジ特徴に基づいた類似画像検索を行う。内容による検索とは画像に手動でキーワードを付加することなくシステム内で自動的に画像特徴を抽出し、それをもとに検索を行うものである。我々のシステムでは質問画像を与ると、その画像にもっとも似ている画像の集合を返す。まず、ウェーブレット変換を用い、画像を周波数分解する。分解された成分のうち高周波成分からエッジ画像を作成する。エッジ画像作成には色情報に不変な検索を行うため、輝度値のみを使用している。そのエッジ画像から高次局所自己相関関数と呼ばれる手法を用いてエッジ特徴を抽出する。この特徴には画像サイズと高次局所自己相関関数の変位に不変になるように正規化を施した。さらに画像の明るさに不変な検索を行うため距離尺度として方向余弦距離を用いている。抽出された特徴ベクトルは自己組織化マップ (SOM) によりクラスタリングされる。

キーワード: ウェーブレット変換, 高次局所自己相関関数, 自己組織化マップ (SOM), 類似画像検索, 画像データベース

Development of Edge-based Similar Image Retrieval System using Wavelet and SOM

Masaaki KUBO, Zaher AGHBARI, and Akifumi MAKINOUCI

Graduate School of Information Science and Electrical Engineering,
Department of Intelligent System, Kyushu University

Abstract

In this paper, we developed a similar image retrieval system based on Edge feature. Content-based image retrieval system can provide a method that we can search images with no needs for annotation. The system extract features automatically, and we can retrieve images by them. When a query image is selected, the system return a set of images which are most similar to the query image. At first, images are decomposed into several frequency band using Wavelet Transform. From the high frequency band, edge images are created. We use only luminance for creating edge images to retrieve images irrelevant to color. Next, edge features are extracted using higher-order autocorrelation. We normalized the method not to be influenced by image sizes and displacement of higher-order autocorrelation. Furthermore, to make our method irrelevant to difference of each image's luminance, we used direction cosine for measure of distance. The extracted features are clustered by Self-Organizing Maps (SOM).

Keywords: Wavelet Transform, Higher-order autocorrelation, Self-Organizing Maps (SOM), Similar image retrieval, Image database

1 はじめに

大量にある画像データベースから画像を検索することは近年重要なことになってきている。多くの人々がインターネットを用いてマルチメディアデータベースから画像を探したり、見たりしている。ここで画像をどのように検索するかが大きな問題である。現在広く用いられている方法の一つとして画像に対して人がテキストで画像の意味としての注釈をつけるものがある。しかし、この方法には多くの問題点がある。例えば、人それぞれ同じ画像に対して与える注釈が違っていることである。これでは探すときどのような言葉で探してよいかわからない。さらに、大きなデータベースでは注釈をつける手間がたいへんになってしまう。そこで、内容による画像検索(Content-based Image Retrieval:CBIR)が必要となる。これは、互いに意味内容が似た画像データからは互いに近い特徴量が得られるという考え方に基づいている。

内容による画像検索ではユーザが検索したい画像を指定し、大量の画像から所望の画像を検索する。検索結果は指定した画像の特徴ベクトルと、データベース内の画像の特徴ベクトルのマッチングで行われる。特徴ベクトルとは特徴量の集まりを1つのベクトルとして表現したものである。特徴ベクトルは一般的に高次元になるため検索時の計算時間も課題となる。特徴には様々なものがある。例えば、カラーヒストグラム、テキストチャ、形状等である。特徴は特徴ベクトルとして表現され、画像の内容をコンパクトに表すことができる。特徴ベクトルはB-treeやR-treeなどの空間アクセス手法によりインデックスを作成されたり、自己組織化マップ(SOM)[1]等によりクラスタリングされたりする。「画像Qに似ている画像を取り出せ」という問い合わせが与えられたとき、問い合わせ画像Qはデータベース内の画像を代表する特徴ベクトルと比較され、最もQに似ている画像の集合が結果としてユーザに返される。

本論文では Haar ウェーブレットにより画像を周波数分解し、その高周波成分を用いてエッジ画像をつくる。そのエッジ画像から高次局所自己相関関数と呼ばれる手法を用いてエッジ特徴を抽出する。大きな画像データベースでは画像は様々なサイズ、明るさ、色で蓄えられているが、これらに不変な検索を行うため従来手法[2]では行われなかった高次局所自己相関関数の正規化、及び方向余弦距離の導入を行った。画像のクラスタリングにはSOMを用いたSOMの学習は複数の特徴ベクトルからなるデータ集合を2次元上にマッピングし、特徴ベクトルとの位相関係を保った2次元の位相特徴マップを生成する。SOMでは特徴ベクトルの類似性を見だし、互いに類似した特徴ベクトル同士をマップの近傍に配置することが可能である。さらに問い合わせ画像による検索をサポートするシス

テムを構築した。このシステムは Web 上で検索可能であり、画像を指定することにより検索結果の類似画像をブラウザに表示する。特殊なソフトを必要としないため、誰でも Web 上で検索を行うことができる。

本論文は2章で関連研究、3章でシステム全体の流れ、4章でウェーブレットを用いたエッジ特徴抽出とSOMを用いたクラスタリング、5章で実験結果、そして6章でまとめを述べる。

2 関連研究

内容による画像検索システムに IBM の QBIC システム [3] がある。QBIC は画像から色のヒストグラム、テキストチャ、形状、オブジェクトの空間的関連を特徴として検索を行う。問い合わせには、例示画像を指定する、もしくは、形状をスケッチすることで検索することが可能である。しかし、このシステムは特徴抽出、検索に時間がかかる。Jacobs はウェーブレットを用いた類似画像検索システムの基礎を築いた [4]。VisualSEEK [5] は画像領域の空間関係組み合わせと圧縮領域からの特徴抽出を用いて検索を行う。特徴量は最頻出色集合とウェーブレットを用いたテキストチャである。

SIMPLICity [6] は Stanford 大学で開発されたセマンティックタイプの画像データベースである。インデックス作成には 4×4 画素のブロックに分割し分類を行う。特徴量としては色、テキストチャ、位置情報が使われる。

巨大な画像データベースでは検索スピードを向上させるために特徴ベクトルを組織化しクラスタリングすることは非常に重要なことである。Albuz は k を最大クラスタ数とする k order B-tree を利用した画像特徴量のクラスタリング手法を作成した [7]。多次元ベクトルを扱うのに便利な手法である反面、 k を事前に決めておく必要があるなど欠点もある。Oja は PicSOM [8] システムで木構造自己組織化マップ (TS-SOM) を提案している。これは SOM のマップ層を階層化することにより特徴ベクトルの絞り込を行っている。ただし、このシステムは動的な変化に対応しておらず、新しく画像がデータベースに導入された場合、多大な時間をかけて SOM のマップを再学習させなければならない。

3 システムの概要

図1はシステムの基本的流れを示す。実線は画像のインデックス作成及びクラスタリングを示し、点線は質問画像が与えられたときの流れを示す。両者はほぼ同じ経路をたどる。まず、画像はウェーブレット変換により分解される。分解後の高周波成分を用いてエッジ画像を作成する。

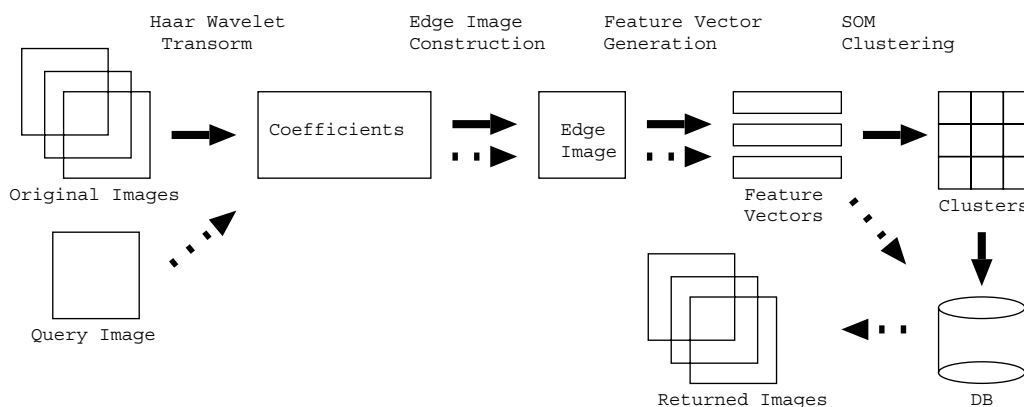


図 1: System Flow:実線は画像のインデックス作成及びクラスタリングを示し, 点線は質問画像が与えられたときの流れを示す.

エッジ画像からエッジ特徴を抽出し, 特徴ベクトルを作成する. 特徴ベクトルは正規化され, これを入力データとして, SOM の学習により各特徴ベクトルの類似性を見つけだし画像はクラスタリングされる. それぞれのクラスタはそれらを代表するノードベクトルを持つ. データベースに各代表ベクトルとそれらに最も近い特徴ベクトルを持つ画像のリスト (Best Matching List: BMIL) が格納される. 検索に対しては, 質問画像から特徴ベクトルを求め, 代表ベクトルと比較し, 最も距離の近いベクトルが示すクラスタに入っている画像をすべて返す.

4 インデックスとクラスタリング

図 1 が示すようにシステムはいくつかの処理に分割されている. これらについて詳細に述べる.

4.1 Haar ウェーブレット変換

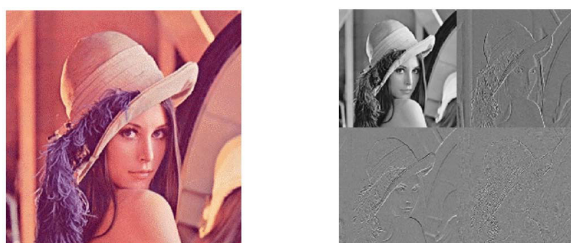


図 3: 左:原画像 右:1 回分解後の画像

まず画像の色空間である RGB 空間を YIQ 空間に変換する. YIQ 表色系は輝度情報 (Y), 色差情報 (IQ) に分けて画像情報を表す. エッジ画像作成には輝度情報である Y

値のみを利用する. Y 値のみを利用することにより

1. 色に依存しない検索
自動車など形状が似ているが色が違うものの検索
2. 特徴ベクトルの次元数の低下
RGB 空間では次元数が 3 だが Y 値のみ利用することにより次元数が 1 になる

が可能となる.

次に, 画像はウェーブレット変換をかけられる. ウェーブレット変換を画像に施すことにより, 画像を低周波成分と高周波成分に分解することができる. 低周波成分は画像の大まかな特徴, 高周波成分は詳細情報を表す. 分解を低周波成分に再帰的に施すことにより多重解像度解析 [9] が可能である. 画像をウェーブレット変換して得られるウェーブレット分解係数空間から, 色, テクスチャ, 形状情報などの様々な画像内容に対応した特徴量をシステム側で効率よく取り出すことができる. l 回分解後は 3^l 個の高周波サブ画像と 1 個の低周波サブ画像に分解される. 分解後は低周波成分 LL, 横方向差分画像 HL, 縦方向差分画像 LH, 斜め方向差分画像 HH の 4 つのサブ画像になる (図 2).

本論文では, 画像のエッジ特徴抽出のために Haar ウェーブレット変換を用いる. Haar ウェーブレット変換はデータの圧縮, 伝送, ノイズ除去, エッジ認識など様々なアプリケーションに活用されている. 画像に適用した場合高圧縮でのブロック歪みが生じるという欠点もあるが, 我々の手法にはそれほど影響を与えない. Haar ウェーブレット変換を用いるメリットは次のようなものである.

1. Haar の基底が単純なため分解が高速
Haar 基底 c と詳細係数 d は

$$c_k^{j+1} = \frac{1}{\sqrt{2}} (c_{2k-1}^j + c_{2k}^j)$$

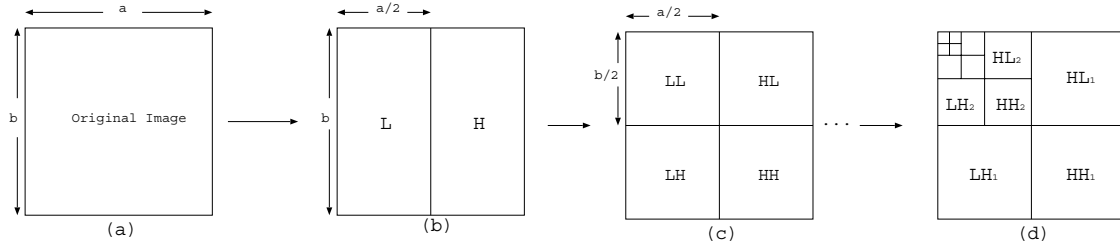


図 2: ウェーブレットでの分解:(a) 原画像 (b) 縦方向分解後の画像 (c) 横方向分解後の画像.1 回分解. 低周波成分 LL と高周波成分 HL,LH,HH に分解されている.(d) 分解を低周波成分に対して再帰的に繰り返したあとの画像

$$d_{k+\frac{1}{2}}^{j+1} = \frac{1}{\sqrt{2}} (c_{2k-1}^j - c_{2k}^j) \quad (1)$$

where $1 \leq k \leq L/2$

として計算する. ただし, c_m^l, d_m^l は分解レベル l , 左上の画素を基準とした m 番目の画素の値, L は低周波成分の高さまたは幅を表す. この式が表すように, 計算は非常に単純で計算が高速に行える.

2. 画像が縮小されるので特徴抽出時の計算が少なくなる

ウェーブレット変換は 2:1 のダウンサンプリングを行う. 図 3 に示すように, ウェーブレット変換は 1 回分解するごとに $1/2$ のサイズになる. 画像は 2 次元であるため画素数は $1/4$ である. ゆえに特徴抽出の際に計算が高速になる.

3. エッジ特徴だけでなく他の特徴抽出にも利用できる
本論文ではエッジ特徴についてのみ言及しているが, 色, テクスチャなどそれ以外の特徴抽出にも利用でき, それらの特徴は類似検索をサポートするものである.

4.2 エッジ画像作成

信号処理の見地からではウェーブレット変換はハイパスフィルタとローパスフィルタを通したものである. ウェーブレット変換後のサブ画像 HL, LH はそれぞれ横方向, 縦方向のエッジを表したものになる. この性質を用いてエッジ画像を作成する. LH サブ画像の座標 (m, n) に対応する要素を $v_{m,n}$, 同様に HL サブ画像の要素を $h_{m,n}$ とし, 対応するエッジ画像の要素を $e_{m,n}$ とするとエッジ画像の要素は以下のようにして作成できる.

$$e_{m,n} = \sqrt{(v_{m,n})^2 + (h_{m,n})^2} \quad (2)$$

where $1 \leq m \leq w \quad 1 \leq n \leq h$

この式を用いてつくったエッジ画像を図 4 に示す. ここで問題になるのがウェーブレット変換の分解度だが, 分解度

を上げると正確なエッジ画像が作成されず, 分解度を下げると計算時間がかかってしまう. そこで我々は分解度 1 のウェーブレットを用いてエッジ画像を作成した.



図 4: エッジ画像

4.3 高次局所自己相関関数を用いた特徴ベクトル作成

エッジ特徴手法には高次局所自己相関関数 [2] を用いている. この手法によるエッジ特徴抽出のメリットは,

1. 画像内にある物体の位置に関して不変な特徴抽出が行える
2. 特徴ベクトルの次元数を低くできる

ことにある. これらは類似検索を行う上で重要な性質の一つである. ベクトルの次元数は画像サイズに関わらず一定の大きさにすることができる.

まず, 局所マスクを P とする. P は P 内で定義される関数 $I(\mathbf{r}) \geq 0$ (where $\mathbf{r} \subseteq P, \text{Supp}(I) = \{\mathbf{r} | I(\mathbf{r}) \geq 0\} \subset P$) で表現される. $I(\mathbf{r})$ の P 内での平行移動は

$$T(\mathbf{a})I(\mathbf{r}) = I(\mathbf{r} + \mathbf{a}), \quad (3)$$

で表される [10]. ただしここでの変位 $\mathbf{a} \subseteq \mathbb{R}^2$ は P を越えない範囲である. 本手法では $I(\mathbf{r})$ が輝度値となる. N 次自己相関関数は参照点 \mathbf{r} 周りの N 個の変位 $\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_N$ に対して

$$R^N(\mathbf{a}_1, \dots, \mathbf{a}_N) = \sum_P I(\mathbf{r})I(\mathbf{r} + \mathbf{a}_1) \cdots I(\mathbf{r} + \mathbf{a}_N) \quad (4)$$

で与えられる [11]. もちろん, N と局所マスクの大きさによりこのベクトルの次元数は大きくなりうる. しかし, これらを小さくし, 計算効率を高めるとともに検索精度を一定の水準にとどめなければならない. そこで, 我々は N を $0 \leq N \leq 2$ とし, 局所マスクの大きさを 3×3 に限定した. この時, 平行移動を除くと 25 個までマスクパターンを減らすことができる (図 5). 次元数が 25 のベクトルであればインデックス作成にもそれほど負荷にはならず, 検索の際も効率を上げることができる. 画像の特徴の計算は, 局所パターンの対応する画素の値の積を足し合わせれば良い. 各パターンに対応する特徴を f_1, \dots, f_{25} とすると, 画像に対する高次局所自己相関特徴ベクトルを

$$\mathbf{f} = (f_1, \dots, f_{25}) \quad (5)$$

と表せる. このように計算された特徴は画像内のオブジェクトの位置に関して不変である. これは非常に有効な特徴である. 以上のように制限された 25 のベクトルにおい

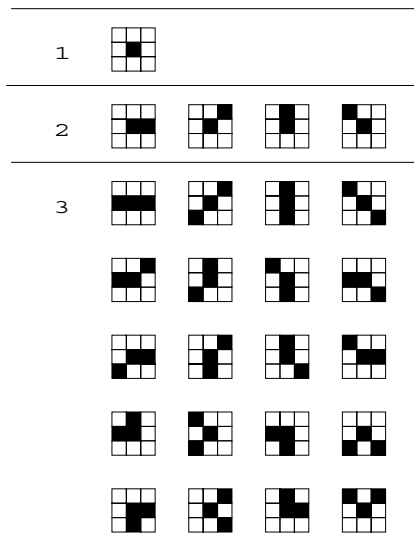


図 5: 局所マスクパターン

て, 参照点を (x, y) , その点における輝度値を $I(x, y)$ として f_i を以下のように定義する.

$$f_1 = \sum_x \sum_y I(x, y)$$

$$f_2 = \sum_x \sum_y I(x, y)I(x, y + 1)$$

⋮

$$f_5 = \sum_x \sum_y I(x, y)I(x - 1, y - 1)$$

$$f_6 = \sum_x \sum_y I(x, y)I(x - 1, y)I(x + 1, y)$$

⋮

$$f_{25} = \sum_x \sum_y I(x, y)I(x - 1, y - 1) \times I(x + 1, y - 1) \quad (6)$$

参照点 r を画像の最も左上のピクセルに適應させて計算する場合マスクパターンによっては変位が画像から出てしまうことがある. 本手法では変位が画像からでてしまう場合はその計算をしない. もし, 画像の縁のピクセルをすべて参照点としないとする, それらが無視され, 画像の正確な情報を失ってしまう. はみ出した分を平均値などで仮の計算をしようとしてもその値は正確ではない. ただし, マスクパターンにより計算するための縦横のピクセル数が異なってくるため, 後述する幅, 高さでの正規化では画像の幅, 高さではなくそれぞれのマスクパターンで計算することができる幅, 高さになる.

この特徴ベクトルは次数 N が $1 \leq N \leq 3$ の高次局所自己相関関数を用いている. しかし, この次数の違いは類似検索に悪い影響を与えることになる. また, 画像全体の明るさが明るいほど高次局所自己相関関数の値は大きくなるという性質を持つため, 画像の明るさに対して不変な検索を行うための工夫を行った. 4.4 節では特徴ベクトルの正規化, 4.5 節では画像の明るさに対して不変な手法について述べる.

4.4 特徴ベクトルの正規化

画像には様々な大きさがあり, 高次局所自己相関関数ではその大きさに特徴ベクトルの大きさが比例する. そこで, 関数を高さ h , 幅 w で正規化する. ただし, ここでの高さ h , 幅 w は前述した通り画像の高さ, 幅ではなく, マスクパターンが適用できる範囲の高さ, 幅になる. この段階でベクトルをグラフ化すると図 7 の左のようになる. このグラフは図 6 の 2 つの自動車の写真から特徴抽出を行い作成したものである.



図 6: 似ている 2 枚の写真

このグラフではベクトルの 1 番目は大きく 2 番目から

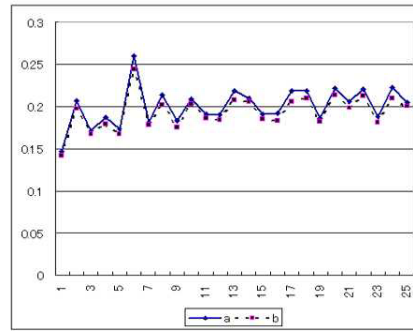
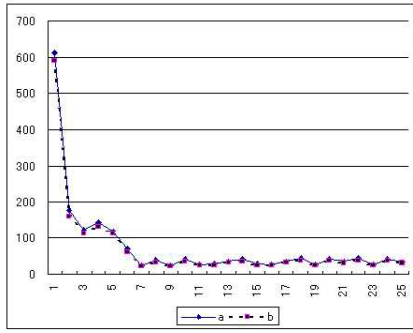


図 7: エッジ特徴ベクトル 左: N による正規化前 右: N による正規化後

徐々に小さくなっている. この原因は高次局所自己相関関数の次数 N にある. 原画像の輝度値のレベルを S としたとき, $N = 0$ では $f_1 \propto S$ であるのに対し $N = 1$ では $f_2 \cdots f_5 \propto S^2$, $N = 2$ では $f_6 \cdots f_{25} \propto S^3$ となるのである. このままでは $N = 3$ はほぼ無視されてしまうので N 乗根をとり正規化する. このようにして正規化された式は以下ようになる.

$$R^N = \frac{1}{wh} \sum_P \sqrt[N]{I(\mathbf{r})I(\mathbf{r} + \mathbf{a}_1) \cdots I(\mathbf{r} + \mathbf{a}_N)} \quad (7)$$

この式で特徴抽出したベクトルのグラフを図 7 の右に示す. 左のグラフに比べると差が小さくなっている.

もう一つの問題として画像全体の輝度値の差が上げられる. 同じ写真をとっていても光の当たり具合等により人間の目には判断できないかも知れないほどの微妙な差ではあるが画面全体に明るさの差が生じる. 図 6 は全く同じ写真のように見えるが実は明るさが少しだけ違う. それゆえ, 図 7 を見るとほぼ同じ変移とたどっているにも関わらず上下に分離していることがわかる. この補正には標準偏差を使われることがあるがさらに良い結果を出す手法を 4.5 節において述べる.

4.5 距離尺度

マルチメディアデータベースでの距離計算で最も良く利用されている距離尺度はユークリッド距離である. しかし, 我々のエッジ特徴には適していない. 画像の明るさに差がある場合, 人間が見て似ていると思われる画像がユークリッド距離で計算すると近いとされない. そこで方向余弦距離を用いた. 図 8 は 3 つの画像のエッジ特徴ベクトルを模式的に表現したものである. 人間が見て似ていると思われる画像は b と c である. 明るさが違うためベクトルの大きさは違うが方向はほぼ同じである. a と c はそれほど似ていない画像だがユークリッド距離を用いると $|a - c| \leq |b - c|$ となり似ていると判断される. 方向余弦距離ではベクトル間の角度が小さいものの方が近いとさ

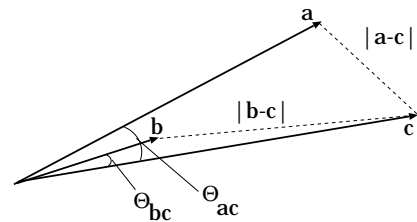


図 8: 3 つの画像の特徴ベクトルを簡略表現

れる. $\theta_{bc} \leq \theta_{ac}$ なので b と c が近いと計算されるのである. ベクトル x, y の方向余弦距離は次式で定義される.

$$dist(x, y) = \arccos \frac{x \cdot y}{|x||y|} \quad (8)$$

距離尺度として方向余弦距離を用いることが我々のエッジ特徴には非常に重要なことである.

4.6 SOM によるクラスタリング

自己組織化マップ (SOM)[1] は教師なしニューラルネットワークであり, 高次元の入力データ \mathbb{R}^n を低次元の空間に出力する 2 層型である. 第一層は 25 次元の入力層であり, 第二層は 2 次元のマップ層である. 第二層には学習前にランダムな重みベクトルが与えられ初期化される. 出力されるべき元のデータの位相関係は保持されたままとなる. つまり入力された 2 つのデータがその空間で近くにマッピングされている場合, 出力された空間でも近くにマッピングされるのである.

これまで様々なインデックスやクラスタリング手法が研究された中で SOM を用いた最大の理由は高次元になってもその検索速度がそれほど遅くならないことである. 一般的に R^* -tree[12] などのインデックスではある次元数を越えると検索効率が低下しはじめる. マルチメディアデータでは特徴ベクトルが高次元になることが多く, インデックスの選択は難しい.

クラスタリングは、以下に示す位相特徴マップ生成過程とベストマッチングリスト (BML) 生成過程からなる。

1. 位相特徴マップ生成

位相特徴マップを生成するために、学習に必要なパラメータを与え、SOM の学習を行う。位相特徴マップの生成過程は次の通りである。まず、入力として与えられた特徴ベクトルに最も近い重みベクトルを持つノードをマップ層から選ぶ。次にそのノードと近傍領域内のノードを特徴ベクトルに近づくように更新する。この処理を指定された学習回数だけ繰り返す。学習の結果、マップ層の各ノードに生成されるベクトルのことをコードブックベクトルといい、

$$CBV_i = [cv_{i1}, cv_{i2}, \dots, cv_{ij}, \dots, cv_{im}]^T \quad (9)$$

で表す。ここで、 $i(1 \leq i \leq (k \times l))$ はマップ層のノード番号、 m は入力層のベクトルの次元数、 $k \times l$ はマップ層の縦と横のノード数である。このように得られた位相特徴マップは、特徴ベクトル相互の類似関係が 2 次元上にマッピングされたものである。

2. ベストマッチングリストの生成

位相特徴マップを用いて、特徴ベクトルとコードブックベクトルとの間の距離計算を行い、最も近いノードを求め類似画像を分類する。特徴ベクトルとコードブックベクトルとの間の類似度は前述の方向余弦距離によって計算する。勝者ノード BMN_i は、

$$BMN_i = \min_j dist(FV, CBV_j) \quad (10)$$

を用いて求める。ここで、 FV はエッジ特徴ベクトルを表す。特徴ベクトルとコードブックベクトルは多対一の関係にある。以上の処理をすべての画像に対して行うことにより位相特徴マップとベストマッチングリストが生成される。つまり、ベストマッチングリストの作成とはそれぞれの入力データとすべてのコードブックベクトルの距離を計算し、一つのコードブックベクトルに最も近い入力データの集合を一つのクラスタとすることなのである。クラスタ数はコードブックベクトルと同数になる。その中には入力データをひとつも持っていないクラスタも存在するかもしれない。このようなクラスタが生まれるかどうかは最初に決定するマップのノード数、つまりコードブックベクトルの数に依存する。

4.6.1 検索

本システムでは画像からの検索をサポートしている。ユーザが質問画像 Q をを選択すると欲しい画像が検索結果としてディスプレイに表示される。画像 Q は前述の特

徴抽出が行われ、特徴ベクトルが生成される。生成された特徴ベクトルは SOM のすべてのコードブックベクトルと比較され最も距離の近いコードブックベクトルを見つける。このコードブックベクトルが代表するノードに含まれるすべての画像をユーザに返すことで検索結果とする。

SOM で分類されたマップは図 9 のようになる。表示している画像はノードの中に含まれる画像の中の一枚であり、そのノードには似ている画像がクラスタリングされている (図 10)。マップは位相特徴を保持したものとなり、類似したベクトルを近くに配置するという性質を持つ。この画像マップからもそれを見ることができる。



図 9: SOM での分類結果: 表示している画像はノードの中の一つの画像



図 10: 一つのノードに含まれる画像

5 実装と実験

5.1 実装

本システムはオブジェクトデータベースシステム Jasmine 上で動作している。Jasmine は富士通と Computer Associates 社により開発されたものであり、Web ブラウザでの検索もサポートしている。我々はこのブラウザでの検索機能を用いて画像による検索機能を作成した。プロ

グラムは C++ と ODQL(Object Database Query Language) により書かれており, このシステムは SUN Ultra5_10, 270MHz, 128MBytes のワークステーションで実装されている. Jasmine を用いることにより, Web 上での画像検索システムを実現した. 本システムは問い合わせ画像を指定することにより検索結果の画像をブラウザで見ることができる. 特殊なソフトを使う必要がないため誰にでも Web 上からの検索が可能となる.

5.2 比較実験

データベースには比較のために Stanford 大学からダウンロードした画像 [13] から非常に似た画像 5 枚ずつ 16 組, 合計 80 枚を利用した. 画像のジャンルは自然画像, 人工画像, 自動車など多岐にわたる. データベースとしては少ない枚数ではあるが比較実験を行うためにはすべての画像を人間が把握しておく必要がある. この段階での分類は人間がすべて手作業で行う. その後, 画像から特徴抽出, SOM のクラスタリングを行う. 問い合わせ画像には始めに分類した 16 組の中から 10 枚の画像をランダムに, 且つ同じ組から複数枚選ばないように決定した. SOM のマップ層には 4×4 のマップを用いた. 平均すると一つのクラスに 5 枚の画像が入っていることになる.

他の手法との比較を行うために以下の 2 つの手法に関して Precision ratio(適合率) と Recall ratio(再現率) を求めた.

1. COS

高次局所自己相関関数に式 7, 距離尺度に方向余弦距離を用いたもの. 本論文での提案手法.

2. NORM

高次局所自己相関関数に式 4, 距離尺度にユークリッド距離を用いたもの. 従来考案された手法.

Precision, Recall はそれぞれ以下のように定義される. 全データベースからの適合画像を $I_{relevant}$, 質問画像に対する検索結果を $I_{retrieved}$ として

$$Precision = \frac{I_{retrieved} \cap I_{relevant}}{I_{retrieved}}$$

$$Recall = \frac{I_{retrieved} \cap I_{relevant}}{I_{relevant}}$$

と表される. 本実験では $I_{relevant} = 5$ となる. つまり, 検索結果として出てきた画像の枚数が 10 枚, その中で適合画像が 3 枚の時, Precision は $3/10 = 0.33$, Recall は $3/5 = 0.6$ となる.

実験の結果 Precision は本手法で約 49%, 従来手法で約 48% とどちらもそれほど変わらないが Recall は本手法 88%, 従来手法は 66% で本手法は従来手法と比べて圧

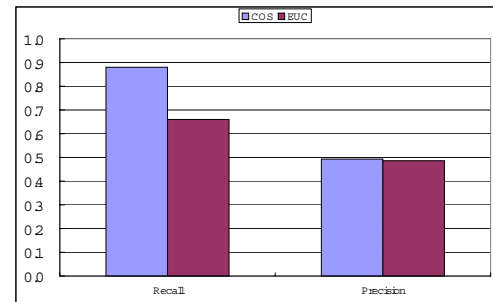


図 11: SOM での Recall と Precision の比較実験結果 (左: Recall の比較, 右: Precision の比較)

表 1: SOM での Recall と Precision の比較

	Recall	Precision
COS	0.88	0.49
NORM	0.66	0.48

倒的に高い数値を出している. これは, データベース中にある適合画像を検索結果として多く返しているということである. 検索時間は約 4.96 秒だった. また, 参考のため SOM を用いずに最近接探索による全件探索を行いそれぞれの Recall, Precision を求めた (図 12, 13). 横軸は返ってきた枚数を表し, 縦軸はそれぞれ Recall と Precision を表す. つまり, k 近傍探索 ($1 \leq k \leq 5$) を行い, その結果から Recall, Precision を求めている.

実験の結果は Recall, Precision とともに従来手法と比較して高い値を示している. これら結果からも本手法が類似画像検索に対して効果的であることを示している. 5 近傍探索の結果を SOM の結果と比較すると, Recall は SOM の方が若干高い数値を示しているが, Precision は SOM の方が低くなっている. これは SOM を用いた場合のクラスタリングでは適合画像以外の画像も多く検索結果として含まれていることに原因がある. SOM ではある代表ノードに近い画像はすべて一つのクラスとされるためそのクラスに含まれる画像の数は多い場合もあれば少ない場合もある. SOM は高次元ベクトルの高速検索が可能であるが, その反面このような欠点も持っている.

6 まとめ

本論文ではエッジ特徴による類似画像検索システムを開発した. 特徴抽出にはウェーブレット変換と高次局所自己相関関数, クラスタリングには SOM を利用した. エッジ特徴は類似画像検索にとって重要なものであり, 本論文で提案したエッジ特徴は色, オブジェクトの位置, 画像サイズ, 画像の明るさに不変なものである. 現行のシステムでは画像による検索をサポートしているがインターフェー

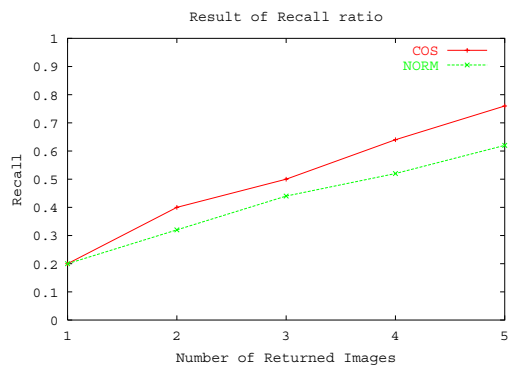


図 12: 最近接探索をしたときの Recall

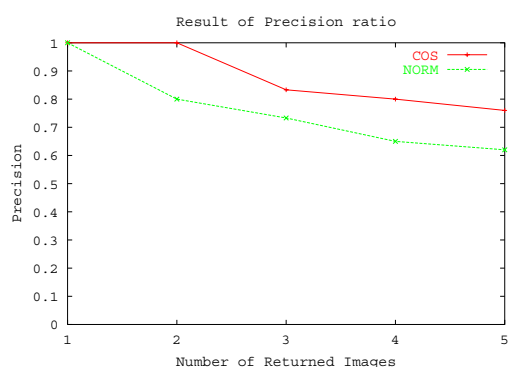


図 13: 最近接探索をしたときの Precision

ス上でスケッチした画像による検索機能を追加することも容易にできる。実験では従来手法と本提案手法とを比較した。それぞれの Precision, Recall を比較した結果、本手法は従来手法よりも高い検索精度を持つ。また、SOM を用いて検索時間の高速化を行った。SOM を用いても高い検索精度を保つことができる。

SOM を用いると、検索した結果が必ずしも距離計算の結果とは一致しないことがある。しかし、我々は完全な近傍探索が可能であり、且つ高速な近傍探索アルゴリズム「SOM-based K-NN Search」[14]を考案した。今後、本システムにこのアルゴリズムを取り入れる予定である。

参考文献

- [1] T. Kohonen, *Self-Organizing Maps Third Edition*. Springer-Verg, 2000.
- [2] 小早川 倫広, 星 守, 大森 匡, and 照井 武彦, “ウェブレット変換を用いた対話的類似画像検索と民族資料データベースへの適用,” *情報処理学会論文誌*, vol. 40, pp. 899–911, Mar. 1999.
- [3] M. Flickner, H. Sawhney, and W. Niblack, “Query by image and video content: The qbic system,” *IEEE Computer Magazine*, vol. 28, pp. 23 – 32, Sep 1995.
- [4] C. E. Jacobs, A. Finkelstein, and D. H. Salesin, “Fast multiresolution image quering,” in *Proc. SIGGRAPH95*, pp. 6–11, 1995.
- [5] J. R. Smith and S.-F. Chang, “VisualSEEK: A Fully Automated Content-Based Image Query System,” pp. 87–98, ACM Multimedia, 1996.
- [6] J. Z. Wang, “SIMPLiCity: A region-based image retrieval system for picture libraries and biomedical image databases,” ACM Multimedia Conference, 2000.
- [7] E. Albu, E. Kocalar, and A. Khokhar, “Scalable image indexing and retrieval using wavelets,” ICASSAP, 1999.
- [8] M. Koskela, J. Laaksonen, S. Laakso, and E. Oja, “The picsom retrieval system: Description and evaluations,” CIR2000, May 2000.
- [9] S. Mallat, “A Theory for Multiresolution Signal Decomposition: The Wavelet Representation,” in *IEEE Trans. PAMI*, 1989.
- [10] T. Kurita, N. Otsu, and T. Sato, “A face recognition method using higher order local autocorrelation and multivariate analysis,” in *Proc. of 11th International Conf. on PATTERN RECOGNITION*, vol. 2, pp. 213–216, 1992.
- [11] M. Kreutz, B. Volpel, and H. Janssen, “Scale-invariant image recognition based on higher order autocorrelation features,” *Pattern Recognition*, vol. 29, pp. 19–26, 1996.
- [12] N. Beckmann, H.-P. Kriegel, R. Schneider, and B. Seeger, “The R*-tree: An Efficient and Robust Access Method for Points and Rectangles,” pp. 322–331, ACM SIGMOD, 1999.
- [13] “<http://www-db.stanford.edu/~wangz/image.vary.jpg.tar>.”
- [14] Z. Aghbari, K.-S. Oh, Y. Feng, and A. Makino, “SOM-Based K-Nearest Neighbors Search In Large Image Databases,” The Sixth IFIP Working Conference on Visual Database Systems (VDB-6), Brisbane, Australia, May 2002.