

Yahoo セミナー資料

コンテンツベースの類似画像検索技術

2007年7月18日

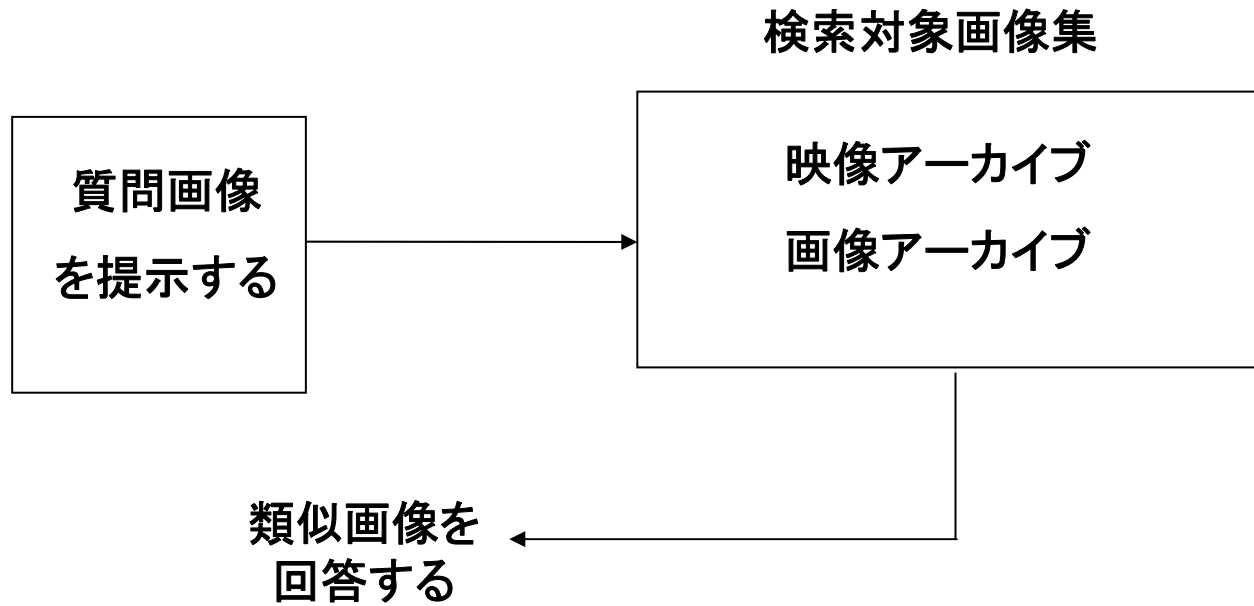
(株)ライブグラフィ研究所

石井光雄

mishi@livegraphy.com

[URL://www.livegraphy.com](http://www.livegraphy.com)

画像を提示して類似画像を検索する問題



画像の特徴による検索問題の現状分析

1. 質問画像を提示しただけでは類似検索の問題設定が曖昧である
 - 画像のどこが似ているのか、どのように似ているのか。注目する特徴点の指示が不明確
2. 画像の特徴(コンテンツ)表現が曖昧である
 - 色、形、サイズ、等の項目も抽出方法も自由度が大きい
 - コンテンツの多次元ベクトル化への対応付けも任意性大
3. 多次元ベクトルによる類似度評価方法の曖昧さ
 - 異なるベクトルであってもベクトル間のユークリッド距離は同じ場合がある。方向余弦のスカラー値も同一値になる場合がある
4. 多次元ベクトルはアドレスとしてメモリを直接アクセスできない
 - 多次元ベクトルはコンテンツの格納番地を表してはいないのでインダイレクトアクセスになる
5. リアルタイムでコンテンツIndexを生成付与する方式ではない
 - 検索用のコンテンツIndexは撮影時に日付と同様にヘッダー情報として記録されるべきものだがそのようなものは未だ見当たらない

多次元ベクトル間の類似度比較の問題点

JPEG,MPEGのDCT特徴ベクトル間の距離

検索画像Q とターゲット画像T 間の距離を定義する

$$D = \left(\sum w_{ij} (Q[i,j] - T[i,j])^2 \right)^{1/2}$$

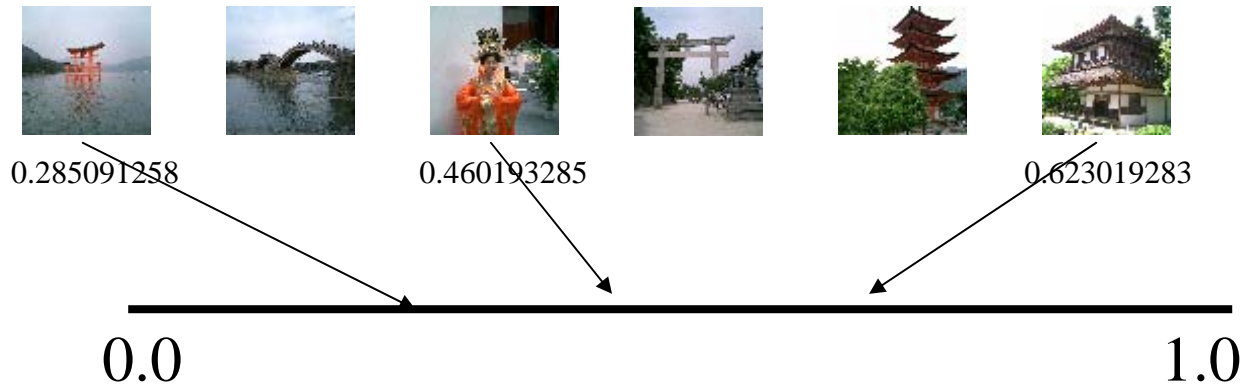
w_{ij} : ベクトル i,j 要素の重み

検索結果は最小値(D)の或いは $D < d$ (閾値)の画像を出力する

距離Dによる検索の問題点

- ① 多次元ベクトル間の距離を単純なスカラーにしているため縮退する
- ② データベース内の全ての画像と距離計算するので対象DBのサイズに比例して計算量が増える。
- ③ DCT係数の支配的なものに絞るとか、係数を量子化して計算量を抑えるなど工夫を凝らすと比較する点で全件検索の本質は変わらない。

画像の特徴量を実数値に変換してIndexとする

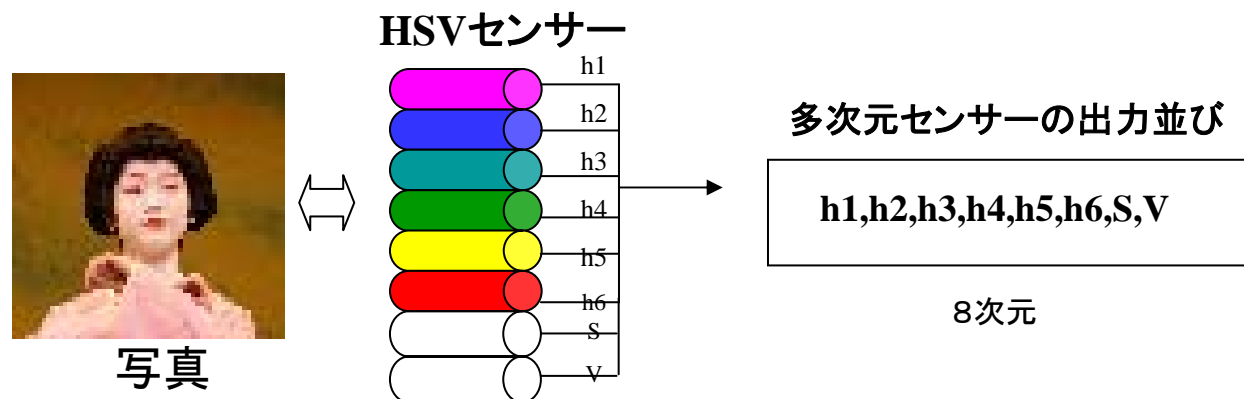
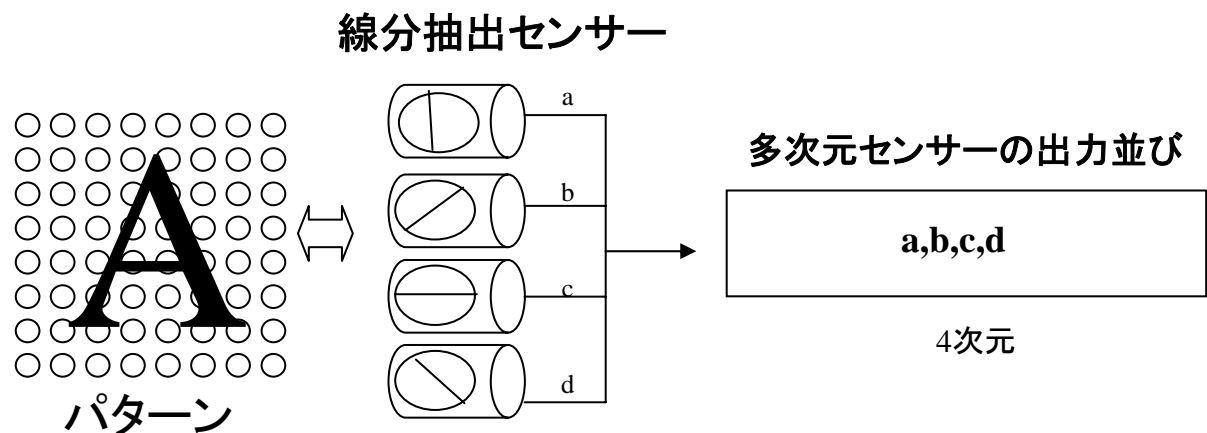


1. 多次元特徴ベクトルデータを1個の実数値に変換する
2. 類似画像は実数値も近い値になるように変換する
3. 実数の桁数を増やせば画像の詳細特徴を表せる
4. 検索を高速に行える実数インデックスデータ構造にする

これによって

実数直線上の位置が画像のコンテンツアドレスと見做せるので、
従来のように全ての画像間での距離計算をする必要はなく
類似検索は実数直線上の位置アドレスをアクセスするだけとなる

特徴抽出センサーによる多次元ベクトルの例



多次元データの一次元マッピング方法

1. 2次元平面上的の点を1次元直線上の点に対応させる事ができる

- $(0.a_1a_2a_3a_4a_5\dots; 0.b_1b_2b_3b_4b_5\dots)$

- $0.a_1b_1a_2b_2a_3b_3a_4b_4a_5b_5\dots$

交互に並べるカントールの方法 「我見るも我信せず」1877年6月29日

2. N次元空間の点も1次元実数直線上に対応させることができる

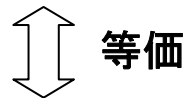
3. 画像は多次元空間内の点として表現できる

カオスの離散力学系による固有値変換

1. カオスを生成する力学系
2. 任意のビット列を生成する初期値がある

多次元ベクトルデータを順々に並べてビット列にする

10010101000011110101010100010101001011100001111...



等価

$X_0 = 0.r_1r_2r_3\cdots r_n\cdots$

X_0 を初期値として同じビット列を再生できる。ビット列の固有値と呼ぶ

3. 力学系は単峰関数なら何でも良い

ロジスティック関数

$$X_{n+1} = F(X_n) : F(x) = 4x(1-x)$$

あるいは三角関数など何でもよい。

任意ビット列を生成する固有値の求め方

カオス生成関数の性質

$$X_{n+1} = f(X_n) \quad : \quad f(x) = 4x(1-x)$$

初期値 X_0 を与えるとカオス数列が得られる

$$X_0, X_1, X_2, \dots, X_n$$

2値化するとカオスビット列が得られる

$$b_0, b_1, b_2, \dots, b_n$$

カオス逆プロセスから固有値が得られる

任意のビット列 b_0, b_1, \dots, b_n を生成できる初期値 X_0 がある。それはビット列固有の値である。

X_0 の求め方: $X_{n-1} = f^{-1}(X_n; b_n)$

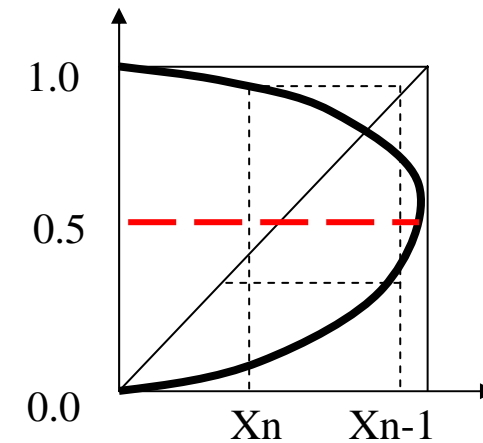
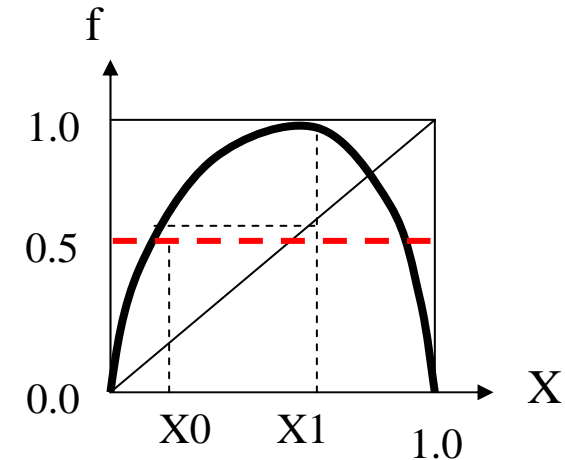
X_n からカオス逆プロセスを開始する。

$b_n, b_{n-1}, \dots, b_2, b_1, b_0$ と順に N ビット畳み込めんだ結果が X_0 である。

X_0 を初期値としてカオス力学系から元のビット列

$b_0, b_1, b_2, \dots, b_n$ を再生できる。

右図の点線は $b_n=1, b_{n-1}=0$ の例



初期値 X_n の違いによる固有値 X_0 の収束状況

$b_0b_1b_2\cdots\cdots\cdots b_{31} \longrightarrow X_0 (0.r_1r_2r_3\cdots\cdots)$

0011 0101 1111 0100 0101 0111 0010 1001 \longrightarrow 0.15099490832

畳み込んだビット長 n	$n=4$ $b_{31}\sim b_{28}$	$n=8$ $b_{31}\sim b_{24}$	$n=12$ $b_{31}\sim b_{20}$	$n=16$ $b_{31}\sim b_{16}$	$n=20$ $b_{31}\sim b_{12}$	$n=24$ $b_{31}\sim b_8$	$n=8$ $b_{31}\sim b_4$	$n=32$ $b_{31}\sim b_0$
初期値 $X_n=0.1$	0.x	0.19x	0.36x	0.3976x	0.47514x	0.6546965x	0.41591853x	0.150994908
初期値 $X_n=0.6$	0.x	0.19x	0.36x	0.3976x	0.47514x	0.6546965x	0.41591853x	0.150994908
初期値 $X_n=0.9$	0.x	0.19x	0.36x	0.3976x	0.47514x	0.6546965x	0.41591853x	0.150994908

表の数値 x はその桁位置で収束値が異なる
初期値 X_n によらず ビット長32個で 9桁程度の精度まで収束する

ビット列の変化位置による固有値の違い

$b_0b_1b_2\cdots b_{31} \longrightarrow X_0(0.r_1r_2r_3\cdots)$

0011 0101 1111 0100 0101 0111 0010 1001 \longrightarrow 0.15099490832

	最近 ←							昔
変化させたビット位置	b4	b8	b12	b16	b20	b24	b28	b31
影響が現われた固有値の桁位置	2	2	3	6	7	8	8	10

昔の変化ほど固有値への影響は小さい

固有値の精度による再生ビット数の制御

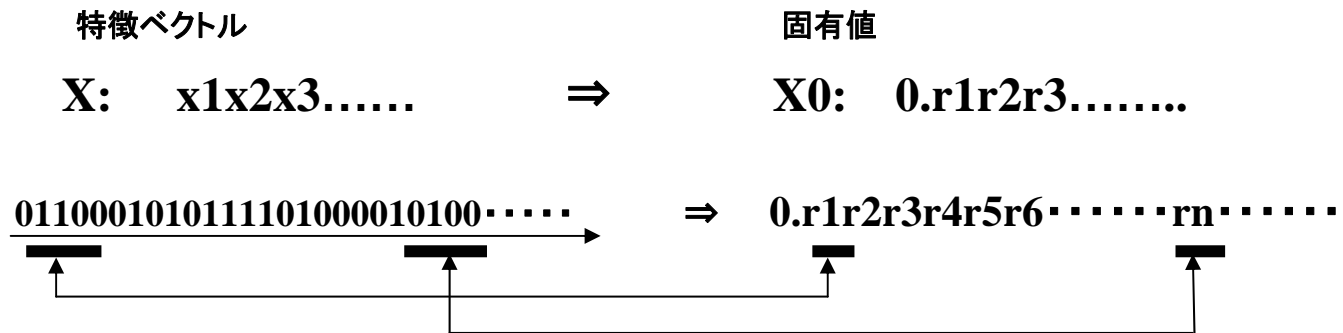
b0b1b2.....b31 ← X0 (0.r1r2r3.....)

0011 0101 1111 0100 0101 0111 0010 1001 ← 0.15099490832

桁数以下を打ち切って力学系を動作させたとき
正しく再生したビット数

固有値精度	2 桁 0.15	4 桁 0.1509	6 桁 0.150994	8 桁 0.15099490	10 桁 0.1509949083
正しい再生ビット数	b0~b7	b0~b11	b0~ b21	b0~b27	b0~b31

特徴ベクトルの並び順と固有値の関係

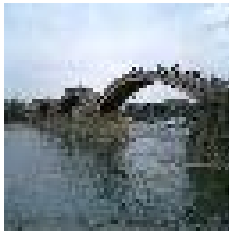


特徴ベクトルの並び順がそのまま固有値の桁位置に反映されるので注目する特徴ベクトルを優先順に並べればよい

例

- ・低周波成分から高周波成分の順
- ・粗い画像から精細画像への特徴ベクトルの並び

二次元DCTによる特徴ベクトル



-22	-2	14	-1	-6	1	-1	0
10	-2	-4	1	2	0	-1	0
0	1	-6	0	2	0	0	0
-2	3	1	0	0	0	0	0
-2	0	0	0	0	0	0	0
1	1	0	0	0	0	0	0
0	0	-1	0	0	0	0	0
0	0	0	0	0	0	0	0

DCT(8x8)変換出力
Y成分



係数出力の並び

$c(1,1), c(1,2), \dots, c(8,8)$

64次元データ

類似画像は固有値も近い値にする方法

2次元DCT係数をジグザグスキャンに取り出す。

大まかな特徴から詳細な特徴を示すデータ順になる

画像の大まかな変化は固有値の上位桁の変化に現れ、細かな変化は下位桁に現れる

1	2	6	7	15	16	28	29
3	5	8	14				
4	9	13					
10	12						
11							
21							
22							
							64

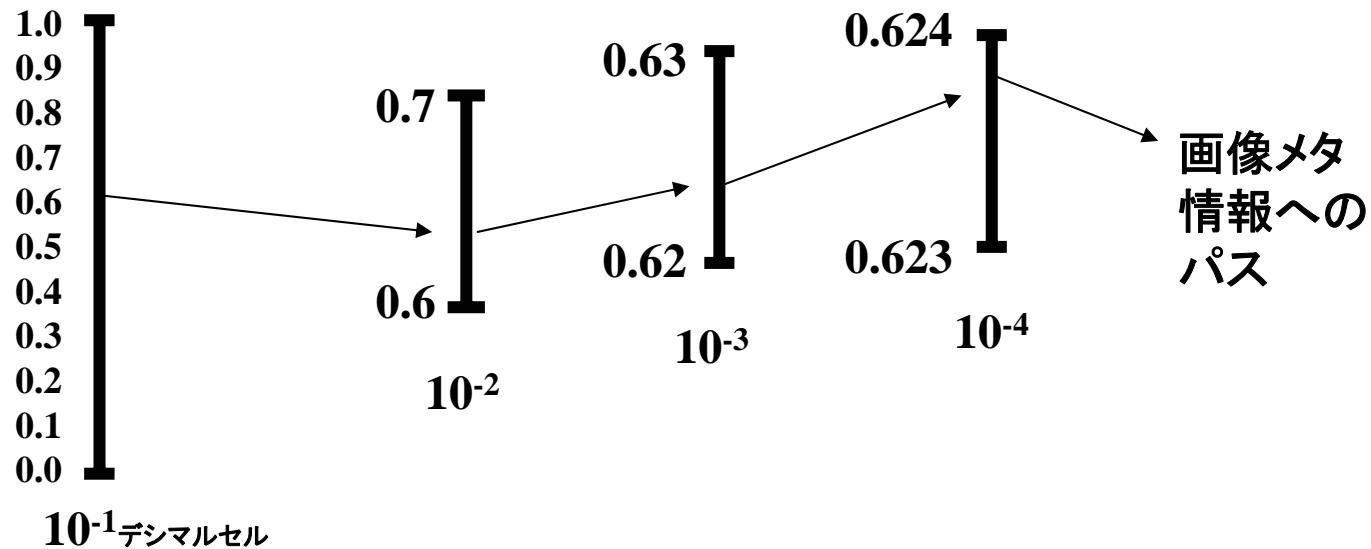
JPEG、MPEG符号化方式でのDCT係数のジグザグスキャン方式

高速アクセスのための実数直線メモリの実装構造

1. 直線上の位置をアドレスと見做してダイレクトアクセス可能とする
2. アドレスはコンテンツそのものだから無限のアドレス空間を生成できるようにする
3. 類似検索は
実数アドレスの上位桁からアクセスして絞ってゆき、
任意桁の隣のアドレスもダイレクトアクセスできるようにする
4. 実数アドレスへのRead/Writeアクセス内容は
そのコンテンツのメタ情報が格納されているファイルへのパス

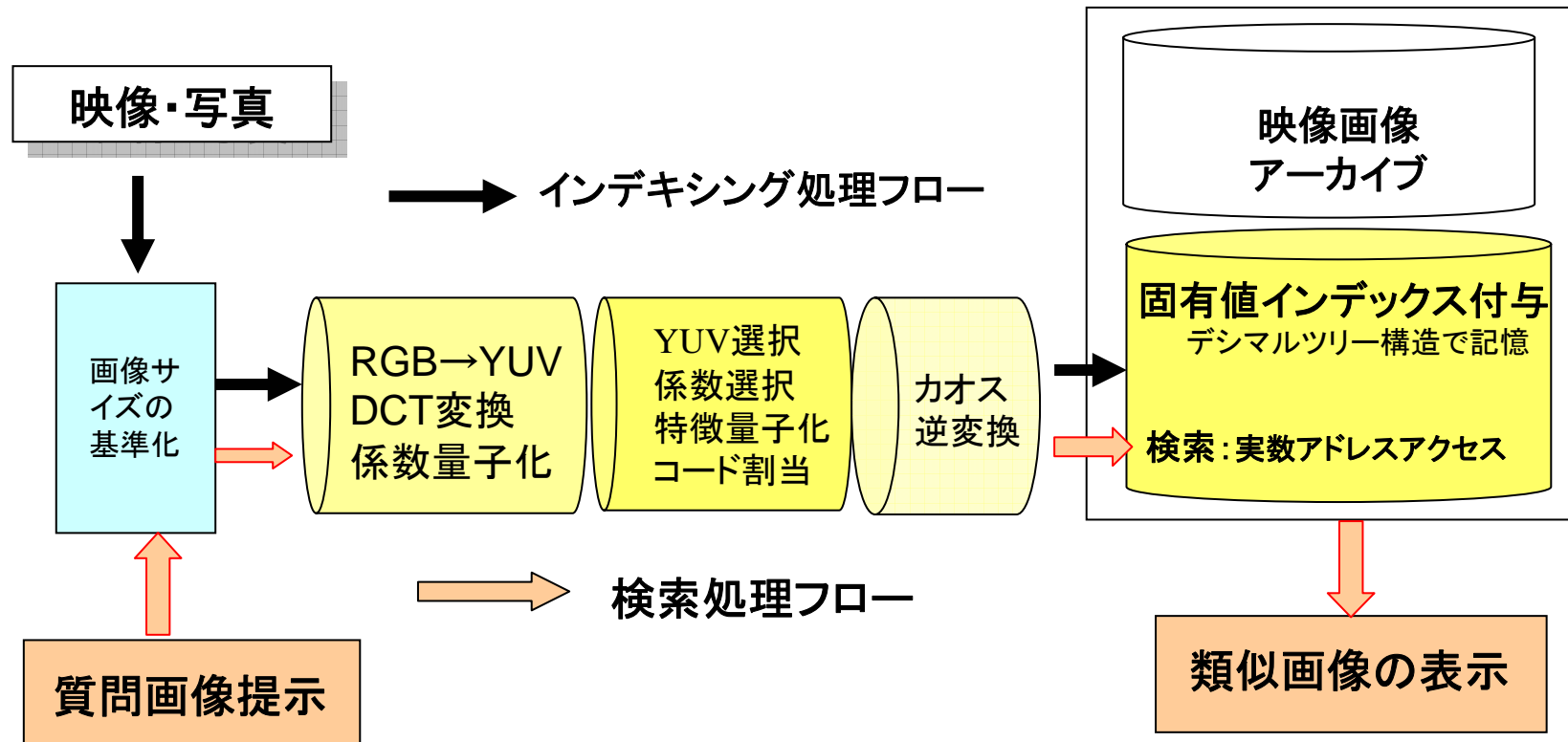
実数直線メモリのデシマルツリー構造

固有値 0.6238の記憶例



- ・デシマルセルは実数値を記憶するとき生成することで無限のアドレスを生成可能とする
- ・ツリー構造にすることで固有値をインデックスとしてアクセスできる

DCT特徴による類似画像検索 試作システム



質問画像に類似した画像検索結果

The screenshot displays a software window titled "画像検索システム" (Image Search System). At the top, the search target is "Index UV TV-PH 32273.index". Below this, several index files are listed, with "Index UV TV-PH 32273.index" highlighted. The search conditions section, labeled "検索条件 (1)", shows four query images: [1]0000521.bmp (Sensor: Det UV), [2]0000733.bmp (Sensor: Det UV), [3]0000891.bmp (Sensor: Det UV), and [4]0000863.bmp (Sensor: Det UV). A "GO 検索実行" button is present. The results section, titled "(該当した画像: 3枚)", shows a list of folders on the left and a grid of three image thumbnails on the right. The thumbnails are labeled "該当した画像の一覧". At the bottom, there are controls for metadata (FilePath), thumbnail size, and a maximum display count of 100.

検索対象 : Index UV TV-PH 32273.index

IndexDB

Index Y TV-PH 32273.index | Index UV TV-PH 32273.index | Index UV Y TV-PH 32273.index | Index YUV TV-PH 32273.index | Index

検索条件 (1)

KeyPicture

[1]0000521.bmp
センサ:Det
UV

[2]0000733.bmp
センサ:Det
UV

[3]0000891.bmp
センサ:Det
UV

[4]0000863.bmp
センサ:Det
UV

GO 検索実行 (該当した画像: 3枚)

Root

- 2
- 3
- 4
- 5
- 6
- 7
- 0
- 1
- 2
- 3
- 6
- 6
- 7
- 0
- 8
- 3
- 4
- 7
- 1
- 0
- 9
- 3
- 7
- 1
- 2
- 8
- 9
- 8
- 0

0.7678710937

該当した画像の一覧

Image検索プログラム

メタデータ:
FilePath:

サムネイルのサイズ:
最大表示件数: 100

ETL8 漢字の検索結果

The screenshot displays the ETL8 image search system interface. At the top, the search target is set to 'Index Y4hen.index'. Below this, there are four database selection buttons: 'Index UVYTV-PH 32273.index', 'Index YTV-PH32273.index', 'Index UVUYYTV-PH96816.index', and 'Index Y4hen.index' (highlighted in orange). A '+ データベースの追加' button is on the right. The search conditions section shows four criteria: '六', '連', '論' (highlighted in orange), and 'ゆ', each with a sensor ID and 'センサ:Dct ETL Y4hen'. A '+ キー画像の追加' button and a '条件の再設定' button are on the right. The search results section shows '検索結果 (該当した画像: 2枚)'. On the left is a tree view of the file system. On the right, two images of the character '論' are displayed. Below the images, the text 'Image検索プログラム' is visible. At the bottom, there are fields for 'メタデータ: FilePath:' and 'サムネイルのサイズ: 最大表示件数 50'.

一次元実数値処理の特長

	多次元ベクトル表現	一次元実数値表現
検索の決定方法	ベクトル間の距離計算・縮退あり	直線上の位置検索・縮退なし
注目する特徴についての処理	特徴の重み付け	特徴を並べる順序
インデキシング機能	無い	デシマルツリー構造で記憶する有
検索範囲	全件アクセス	ダイレクトアクセス
類似検索方法	距離の許容値で制御	直線上の位置指定精度で制御
コンテンツ記憶形態	ビット列をそのまま記憶	ダイナミック動作で再生
汎化性能	多次元ベクトルの汎化性能を活かせる	画像の同定探索に適している

固有値インデックスの応用

1. AND検索

- Y,UV, UVYなど個別のデシマルツリーのアクセス結果をAND操作する

2. キーワード検索

- 質問画像が無いときキーワード入力で得た複数画像の中から一枚を質問画像とする。

3. リアルタイムインデキシング

- 撮像時にJPEG, MPEG2符号化と同時に固有値Indexをつけておく

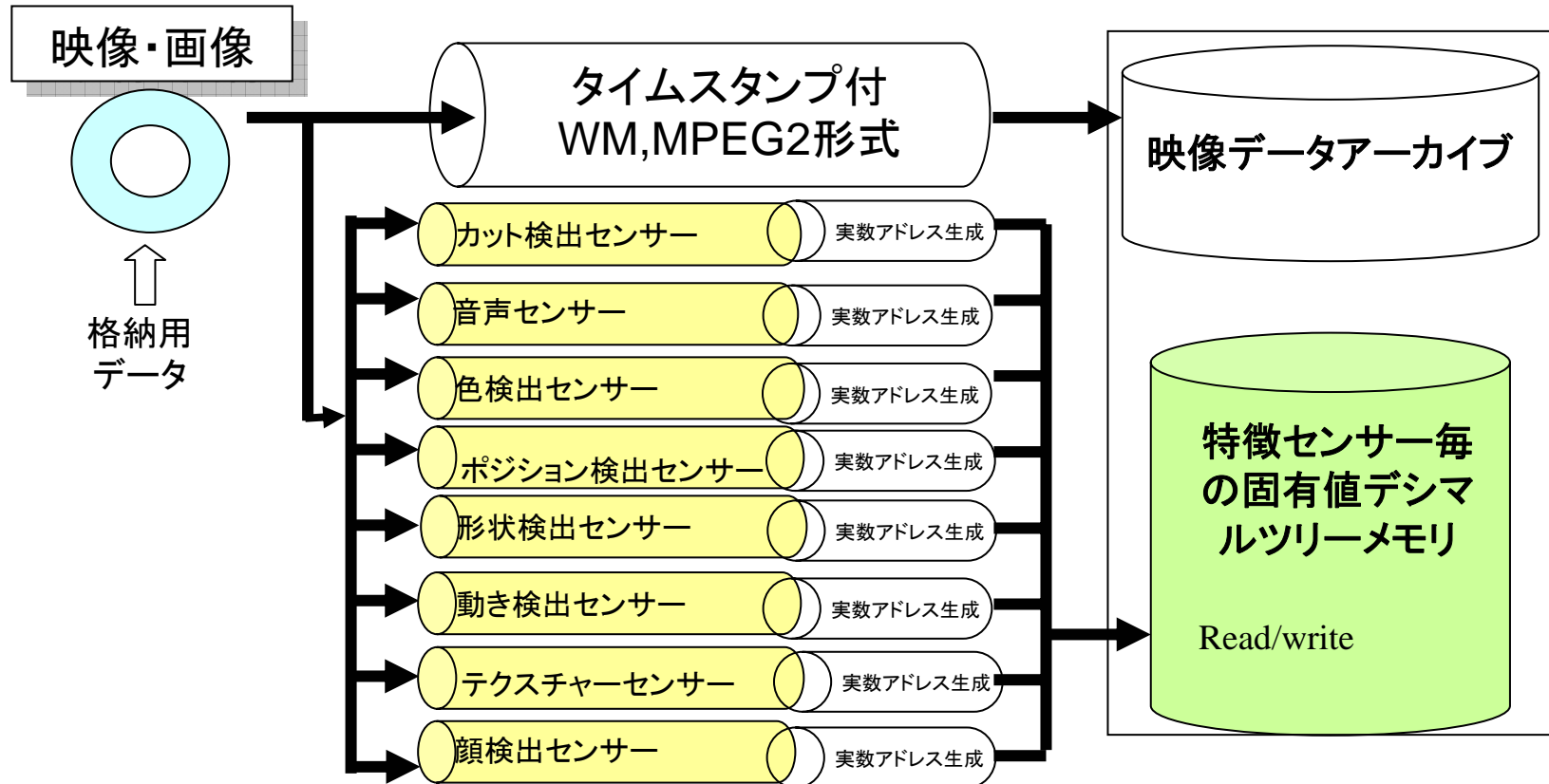
4. カラー画像を白黒画像で検索する

- Y成分の特徴ベクトルで固有値を記憶しておけば簡単である

5. マイニングへのツールとして

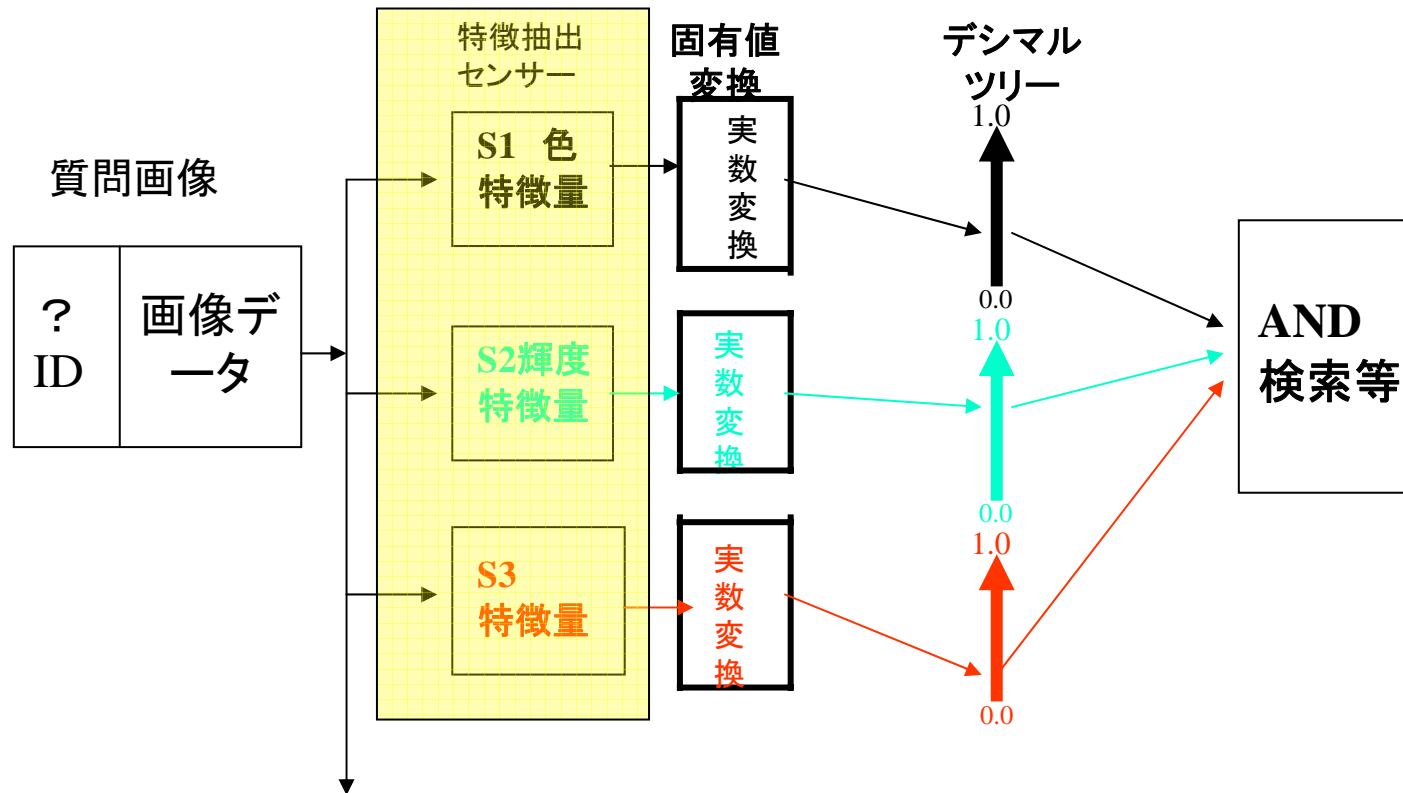
- 多次元データを任意の次元に落としてみる事ができる

特徴ベクトル毎の固有値Index作成



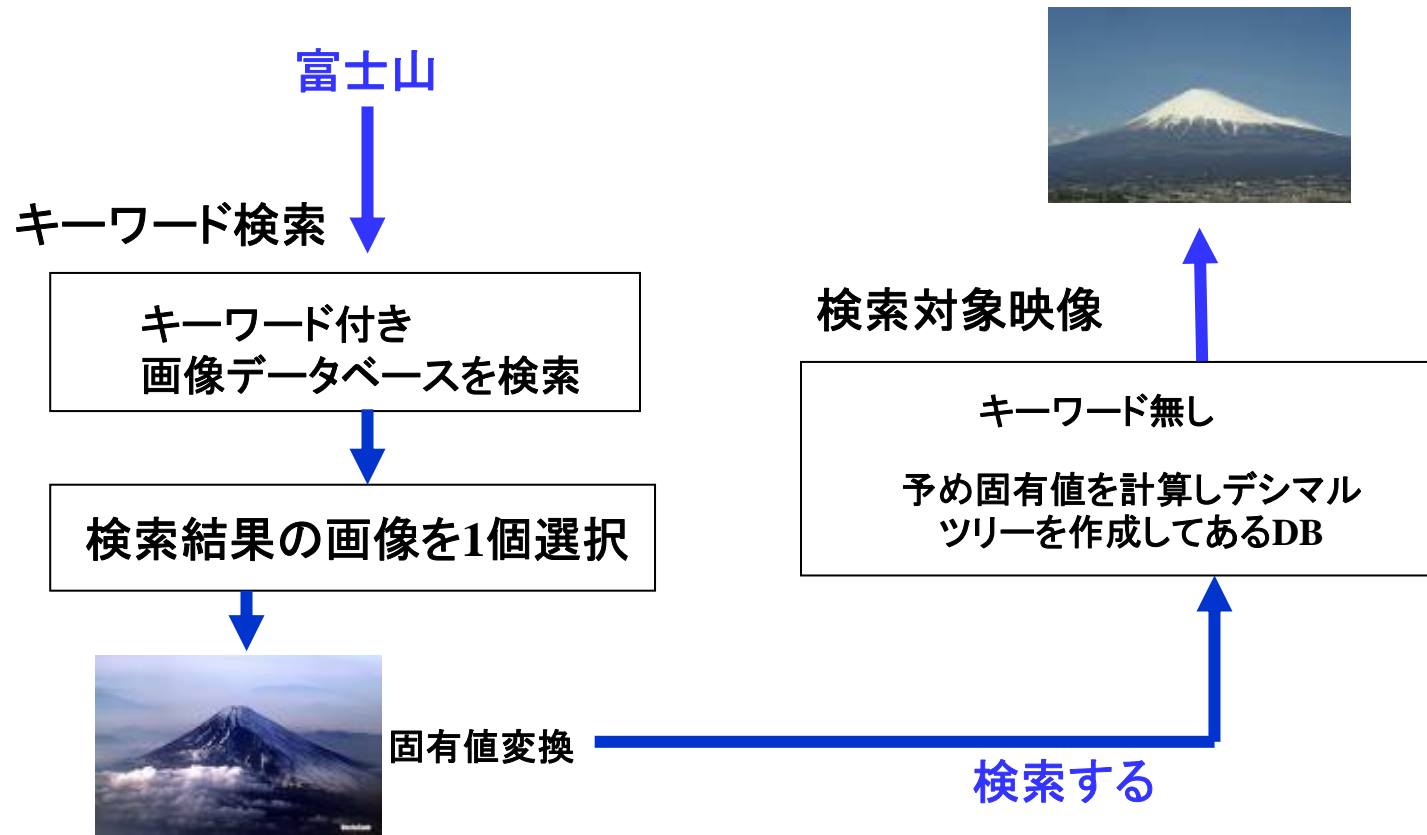
複数の特徴を使った検索

特徴抽出センサー毎にデシマルツリーを構築し記憶しておく。
ツリー間のAND検索で絞り込むことができる



キーワード検索との連携

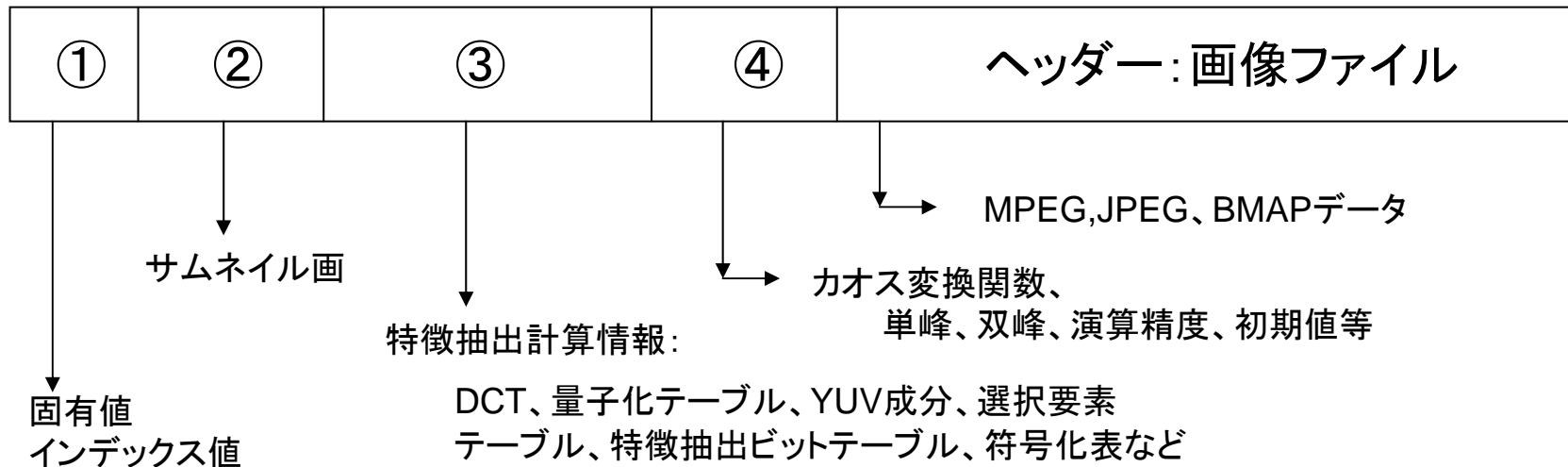
キーワードが無い映像データをキーワードで検索する



画像撮影時に固有値をヘッダーに書く

1. 画像ファイルのヘッダ情報に固有値を記入する

- 固有値計算方法を標準化しておけば①のみ追加すれば良い
- デジカメ、ビデオカメラで撮影したとき圧縮処理と同時に固有値インデクスも計算してリアルタイムで付記しておく



画像検索性能向上に向けて

1. 画像の認識と検索の違いは
 - 本質は同じ
2. 特徴ベクトルの性能が良ければ少ない次元で検索できる
 - 検索空間が小さくてよい
 - 認識では辞書数とも呼ばれるが、汎化能力が高いアルゴリズムを追求するのは記憶する辞書数が少なくて済むから。
 - 検索においては良い特徴ベクトルを求めることに相当するが、画像認識に良い特徴ベクトルの追求はパターン認識よりはるかに亡羊としている
3. 辞書数を少なくする努力のトレードオフとして、膨大な辞書数でも高速でマッチング、検索できるなら結果は同じことになる
 - パターンの辞書を少なくすることは現実的に限界があり、さらに画像の場合は検索対象データ数は無限とも言え、汎用的な良い特徴ベクトルの追求は現実的でない
 - 検索対象データの中から同じ画像を探す問題では特徴ベクトル空間の次元数削減と同等以上に、無限のアドレス空間を高速でアクセスできるメモリ構造の追及が重要となる
 - 幾らでも記憶でき、しかもコンテンツアドレスでのダイレクトアクセスが可能なメモリが必要

本技術に関する知的財産権

1. 発明の名称 : データ記憶装置、検索装置 等
No1 出願番号 : 特願2005-111301号
No2 出願番号 : 特願2005-356985号
No3 出願番号 : 特願2007-044743号
2. 出願人 : 国立大学法人広島大学
3. 発明者 : 石井光雄
No3は富士通と共願

お問い合わせ先

国立大学法人広島大学

産学連携センター 知財部門

知的財産マネージャー 植田 栄治

Email ueda05@hiroshima-u.ac.jp

関連分野

1. パターン認識
2. 画像の圧縮符号化
3. カオス非線形力学
4. 無限空間のアドレスアクセス
5. 神経細胞系の記憶

参考文献

1. 「カオスとフラクタル 非線形の不思議」 山口昌哉 講談社 ブルーバックス
昭和61年6月20日 発行
2. 「無限に魅入られた天才数学者たち」 アミール・D・アクセル 青木薫 訳
早川書房 2002年2月20日

関連文献

1. 「THE MPEG-7 Color LAYOUT DESCRIPTOR:A COMPACT IMAGE FEATURE DESCRIPTION FOR HIGH-SPEED IMAGE/VIDEO SEGMENT RETRIEVAL」
Eiji KATSUTANI Akio YAMADA, Multimedia Resarch Laboratories, NEC Corporation
2. 「Fast Multiresolution Image Quering」
Charles E.Jacobs Adam Finkelstien David H. Salesin Department of Computer Science and Engineering
University of Washington

END