

Title	画像内にあるオブジェクトに基づく画像検索システムの構築
Author(s)	安藤, 誠吾
Citation	
Issue Date	2001-03
Type	Thesis or Dissertation
Text version	author
URL	<a href="http://hdl.handle.net/10119/731">http://hdl.handle.net/10119/731</a>
Rights	
Description	Supervisor: 櫻井 彰人, 知識科学研究科, 修士

修 士 論 文

**画面内にあるオブジェクトに基づく  
画像検索システムの構築**

指導教官 櫻井彰人 教授

北陸先端科学技術大学院大学  
知識科学研究科知識システム基礎学専攻

950002 安藤 誠吾

審査委員： 櫻井 彰人 教授（主査）  
林 幸雄 助教授  
橋本 敬 助教授

2001年2月

Copyright \_ 2001 by Seigo Andou

# 目次

1	<a href="#">はじめに</a>	1	
1.1	<a href="#">研究の背景と目的</a>	1	
1.2	<a href="#">本論文の構成</a>	2	
2	<a href="#">画像検索システムの現状と本研究の位置付け</a>		3
2.1	<a href="#">画像検索システムの現状</a>	3	
2.2	<a href="#">本研究の位置付け</a>	6	
3	<a href="#">高次局所自己相関特徴量</a>	7	
3.1	<a href="#">高次局所自己相関関数</a>	7	
3.2	<a href="#">研究事例</a>	9	
4	<a href="#">画像検索システムの性能評価</a>		10
4.1	<a href="#">画像検索の評価方法</a>	10	
4.2	<a href="#">再現率と精度</a>	10	
4.3	<a href="#">再現率と精度の平均</a>	12	
5	<a href="#">予備実験</a>	13	
5.1	<a href="#">予備実験について</a>	13	
5.2	<a href="#">予備実験 1(形状について)</a>	14	
5.3	<a href="#">予備実験 2(面積について)</a>	20	
5.4	<a href="#">予備実験 3(幾何学図形による部分画検索)</a>	27	
6	<a href="#">高次局所自己相関特徴量に基づく部分画検索システム</a>		34
6.1	<a href="#">提案するシステム</a>	34	
7	<a href="#">むすび</a>	55	
8	<a href="#">参考文献</a>	56	
9	<a href="#">付録</a>	59	

# 目 次

<a href="#">図 2.1-A ILLUSTR &amp; VIR[1]による全体画検索の例</a>	4
<a href="#">図 2.1-B EXSIGHT[2]による部分画検索の例</a>	4
<a href="#">図 3.1-A 高次局所自己相関特徴のための局所パターン</a>	8
<a href="#">図 4.2-A 再現率-精度グラフの例</a>	12
<a href="#">図 5.2-A 予備実験 1 のシステム</a>	15
<a href="#">図 5.2-B 予備実験 1 の検索対象画像</a>	17
<a href="#">図 5.3-A 予備実験 2 のシステム</a>	22
<a href="#">図 5.3-B 変化させるキーオブジェクト</a>	24
<a href="#">図 5.3-C 予備実験 2 検索結果</a>	26
<a href="#">図 5.4-A もととなるキーオブジェクト</a>	29
<a href="#">図 5.4-B 検索対象の画像例</a>	29
<a href="#">図 5.4-C 予備実験 3 検索結果</a>	31
<a href="#">図 5.4-D レプリカ 144-400 による検索例</a>	32
<a href="#">図 6.1-A RGB を高次局所自己相関特徴に用いた部分画検索システム</a>	36
<a href="#">図 6.1-B HSB を用いた部分画検索システム</a>	39
<a href="#">図 6.1-C RGB ヒストグラムを用いた検索システム</a>	46
<a href="#">図 6.1-D 検索キーオブジェクト</a>	47
<a href="#">図 6.1-E 6009.JPG に対して適合するとした画像の例 (3/8)</a>	47
<a href="#">図 6.1-F FD1179.JPG に対して適合するとした画像の例 (3/7)</a>	47
<a href="#">図 6.1-G FD1034.JPG に対して適合するとした画像の例 (3/9)</a>	48
<a href="#">図 6.1-H その他の画像例</a>	48
<a href="#">図 6.1-I トマト[6009.JPG]の検索結果</a>	50
<a href="#">図 6.1-J バナナ[FD1179.JPG]の検索結果</a>	50
<a href="#">図 6.1-K ミカン[FD1034.JPG]の検索結果</a>	51

# 表 目 次

<a href="#">表 4.2-A 交差行列</a>	11
<a href="#">表 4.2-B 検索結果例</a>	11
<a href="#">表 5.2-A 予備実験 1 の検索結果例</a>	18
<a href="#">表 5.3-A 乱数によるオブジェクトのサイズ(1 ~ 200 までの一様乱数を用いた)</a>	24

# 1 はじめに

## 1.1 研究の背景と目的

コンピュータの性能の向上に伴い、文字だけでなく画像や音声のようなさまざまなデータがコンピュータ上で扱えるようになってきた。加えて記憶装置の大容量化とあいまって各々のデータベースが急速に大規模化してきている。

その中でも、目を通して得られる情報は人間の五感を通じて得る情報の8割を占めるとされ画像は重要な表現形態である。このため従来からさまざまな分野で画像データベースの構築、運用がなされてきた。利用例としては、地図データベース、商標画像データベース、博物館・美術館における画像データベース、医用画像データベース、リモートセンシング画像データベース、自然画像データベースなどがある。これらの多くは特定の分野での利用のために設計され、汎用性が低い。

しかし近年、デジタルカメラの普及やスキャナの低廉化により、膨大な数の画像がデジタル化され蓄積されつつある。このため、これらの多種多様な画像を簡便に管理・検索できる画像データベースが必要とされている。

データベースにおいては情報を管理もさることながら、いかにユーザの意図を反映したデータを検索できるかが重要である。テキスト検索ではその技術はほぼ確立され、検索エンジンも実用化されている。しかし画像検索に関しては人間の主観をどのように検索結果に反映させるかが難しく、いまだ確立されていない。

現在この問題に対しさまざまなアプローチがとられているが、大きく分けると、キーワードに基づく画像検索システムと画像内容に基づく検索システムの2種類に分類できる。さらに画像内容に基づく検索システムには、検索の目的に応じて、画像全体を対象とした全体が検索と、画像内の特定のオブジェクトに着目した部分画検索がある。

本研究では、これらのうち、画像内容に基づく検索システムでオブジェクトに基づく精密な検索が期待できる部分画検索システムの1手法を提案し、実験、考察を行う。

## 1.2 本論文の構成

本研究では、2章において関連研究と、本研究の位置付けを行う。3章では本研究で用いる特徴量である高次自己相関特徴量に付いて述べる。4章では高次自己相関特徴量を用いたシステムの予備実験を行う。そして、5章で本研究で提案するシステムの構成と実験を行いカラーヒストグラムによる検索と比較して有意な差が認められるか検証する。6章で課題とまとめについて述べる。

## 2 画像検索システムの現状と本研究の位置付け

### 2.1 画像検索システムの現状

画像検索システムはキーワードに基づいて検索するシステムと画像の内容に基づいて検索するシステムの2種類に分類することができる。

#### キーワードに基づく検索システム

この手法では、データベースに登録する画像に索引として索品名や作者、解説などをテキストとして付与する。画像を検索する場合には、検索者が入力したキーワードで索引をサーチして適合する索引を持つ画像を返す。このシステムの利点は検索エンジン YAHOO などのように検索者にとって問い合わせが簡単で、かつ高速に検索できることが上げられる。しかし同時に問題点として検索者側には検索対象の画像に対するある程度の知識を要求される。また知識があっても索引製作者との考え方が違くとキーワードで効率良く的確に検索することは難しい。一方、管理者側の問題点として人手による索引作成、入力、メンテナンスが大変な手間となる。国立歴史民俗博物館の歴史民族画像データベースはこの手法に基づいている。

#### 画像内容に基づく検索システム

キーワードに基づく検索システムの問題を解決する1手法として、画像内容に基づく検索システムが挙げられる。この手法では、データベースに画像の登録をする際に画像から特徴量を抽出し、索引としておく。画像を検索する場合には、画像自身をキーとして検索する。このときキー画像から抽出した特徴量と類似した特徴量を索引として持つ画像を返す。画像内容に基づく検索システムは検索する対象に応じて全体画像検索と部分画像検索とに分類できる。

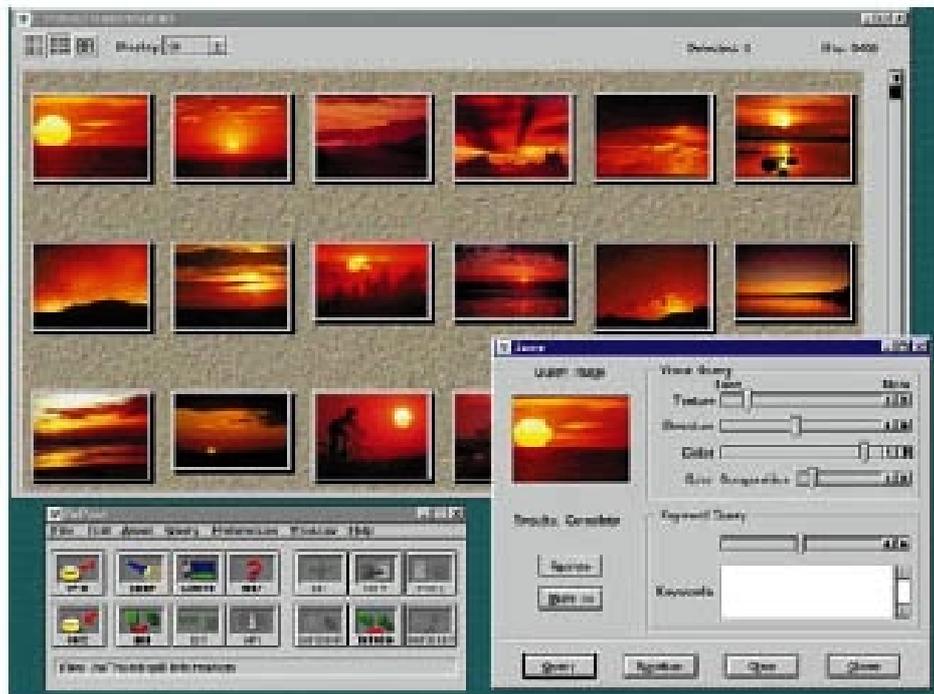


図 2.1-a Illustra & VIR[1]による全体画検索の例

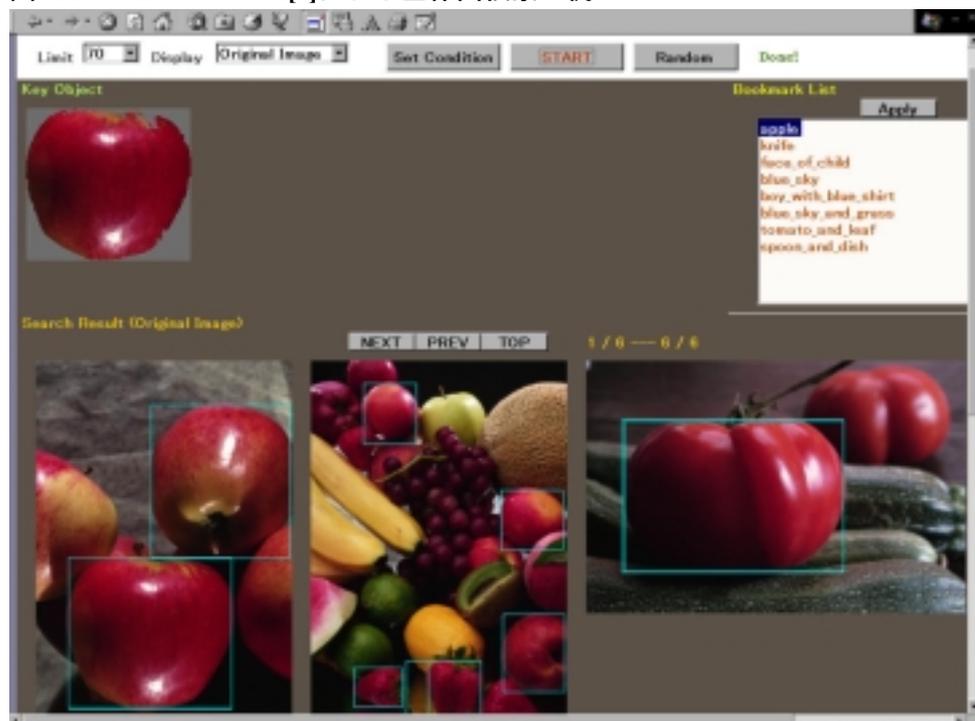


図 2.1-b Exsight[2]による部分画検索の例

Illustra & VIR の方式は全体画検索に分類される。全体画検索では、画像全体を対象に特徴量を計算して索引とする。

検索時にはキー画像の特徴量と蓄積画像の特徴量の比較が行われ、

類似度の高い方から順に出力される。画面全体に対して用いられる特徴量は色や模様などがある。海や山の風景画像、カーテンなどのパターン画像の検索に向いている。

一方、Exsight の方式は部分画検索に分類される。部分画検索ではオブジェクトを対象に特徴量を計算し索引とする。同時にもとの画像との関係を付与しておく。

検索時にはキー画像（キーオブジェクト）と蓄積画像から抽出されたオブジェクトの特徴量の比較が行われ、類似度の高い方から順に出力される。オブジェクトに対して用いられる特徴量は色、形状、位置などがある。

この方式では自動的に精度良くオブジェクトを切出せることが重要である。部分画像検索を試みる VisualSEEK[3]、VP[4]、QBIC[5]においては、画面内を主に色情報に基づいて領域分割を行いオブジェクトを切出している。Exsight ではエッジ情報を利用して色領域を分割している。

りんごや車のような特定のオブジェクトが含まれる画像の検索に向いている。

## 2.2 本研究の位置付け

本研究ではこれら従来手法の検討を検討した結果、検索者の検索意図を柔軟に反映させるために、画像検索に対し画像内容に基づいて検索するアプローチを取る。しかし従来の画像内容に基づく検索システムには以下のような問題点がある。

従来の全体画検索では、風景やパターンの検索には有効であるが、りんごが写っている絵を検索したいなどのオブジェクトに基づく精密な検索はできない。

従来のオブジェクトを単位とする部分画検索はオブジェクトをもれなく切出せれば、精度の良い検索結果が期待できる。しかし、エッジや色を用いてオブジェクトを切出す課題はそもそも難しく、また前もってあらゆる検索キーに対して有意なオブジェクトを切出しておくことができないといった根本的な問題を抱えている。オブジェクトを切出す際にどうしても画像情報の欠落が起こってしまうのである。

全体画検索では画像情報のすべてを特徴量として保持するため、画像情報の保持という点では優れている。画像全体から計算される特徴量で、画像の中に含まれるオブジェクトを区別できるような特徴量を用いれば、従来の部分画検索のモデルが抱える問題点を克服できることが予想される。

このような特徴量として、パターン認識で利用される大津らの高次局所自己相関特徴量[6]が挙げられる。高次局所自己相関特徴量は画像内の図形の形状を特徴量として反映することができる。この特徴量はまた画面内のオブジェクトの並行移動に対して不変でかつ計算が非常に簡単であるといった性質も併せ持つ。

これらを踏まえ本研究では、大津らの高次局所自己相関特徴量を全体画検索の特徴量として採用し、全体画検索のモデルを用いながらオブジェクトを検索対象とする精密な検索を試みる。

## 3 高次局所自己相関特徴量

### 3.1 高次局所自己相関関数

自己相関関数は平行移動に関して不変であることが知られている。その高次への拡張が、高次自己相関関数である。画面内の対象画像を  $f(\mathbf{r})$  とすると  $N$  次自己相関関数は、変位方向  $(\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_N)$  に対して、

$$x_N(\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_N) = \int f(\mathbf{r})f(\mathbf{r} + \mathbf{a}_1) \cdots f(\mathbf{r} + \mathbf{a}_N) d\mathbf{r}$$

(3.1)

で定義される。したがって高次自己相関関数は、次数や変位方向  $(\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_N)$  のとり方により、無数に考えられる。ここでは、高次自己相関の次数を、たかだか 2 までとする。また、変位方向を参照点の周りの局所的な  $3 \times 3$  画素の領域に限定する。この場合には、平行移動により等価な特徴を除くと、2 値画像に対して、特徴の数は全部で 25 個になる。図 3.1 に 25 次の局所パターンを示す。各特徴量の計算は、局所パターンの対応する画素の値の積を全画像に対して足し合わせればよい。たとえば、M2 の局所パターンに対応する特徴は、参照点  $\mathbf{r}$  での濃淡値とそのすぐ右隣の点での濃淡地との全画像に対する積和をとることによって計算される。こうして計算された特徴量は画像内のオブジェクトの位置に対して不変であり、オブジェクトの形状を反映している。

図 3.1-a 高次局所自己相関特徴のための局所パターン

## 3.2 研究事例

大津らは入力画像から対象の平行移動に対して不変な高次局所自己相関特徴を抽出し、認識に利用する方法を提案し、顔画像の認識に応用した[6]。高次局所自己相関特徴の計算は非常に簡単であり、汎用のワークステーションでもほぼ実時間での認識が可能である。また、長谷川ら[7]は、人間に似せたアクティブなエージェントを使ったインターフェイスシステムを試作している。このエージェントは、目でユーザを識別し、声でユーザに話しかけ、ユーザと対話することができる。ここでも高次局所自己相関特徴を用いた顔認識手法が利用されている。

## 4 画像検索システムの性能評価

### 4.1 画像検索の評価方法

画像検索のシステムの評価として行われてきた評価の方法はつぎのとおりでさまざまである。(ア) 検索精度評価をしていない、(イ) 定性的なオピニオン評価、(ウ) 2,3 の検索実地例についての詳細なデータ分析をしている、(エ) 論文の著者やシステム作成者の主観的な評価、(オ) 10 個の満足する画像を得るまでの検索結果で評価、(カ) 再現率・適合率を算出して評価、などである。

最近では木本[8]のようにテキストの類似度検索の場合と同様に再現率-適合率グラフが用いられるようになった。本研究でもこれを用いることでより客観的、数値的な評価を目指すことにする。

### 4.2 再現率と精度

画像検索システムに再現率-適合率での評価では、あらかじめ個々の検索質問画像に対して適合する画像を用意しておき、かく検索質問画像に対してシステムがどのくらい適合画像を検索できるかによってシステムを評価する。このようにあらかじめ用意した評価用のデータはテスト・コレクションと呼ばれている。

テストコレクションが用意されたとき、各検索質問画像に対して表 4.1 の交差行列を作ることができる。

	検索された画像	検索されなかった画像
適合画像	w	x
非適合画像	y	z

表 4.2-a 交差行列

表 4.1 が与えられると再現率 (recall)  $R$  と精度 (precision)  $P$  は式 (4.1) と式 (4.2) のようにそれぞれ定義できる。

$$\text{再現率} : R = \frac{w}{w+x}$$

(4.1)

$$\text{精度} : P = \frac{w}{w+y}$$

(4.2)

再現率は検索にどれだけ「漏れ」がないかを表している。一方精度は検索結果にどれだけ「ゴミ」がないかを表している。したがって再現率と精度は大きい値ほど性能がよいことになる。

例としてある検索質問画像に対してシステムが表 4.2 のような検索結果を出力したとしよう。

順位	画像番号	適合性	再現率	精度
1	24	○	0.2	1
2	43	○	0.4	1
3	11	×	0.4	0.67
4	35	○	0.6	0.75
5	89	○	0.8	0.8
6	33	×	0.8	0.67
7	14	×	0.8	0.57
8	96	×	0.8	0.5
9	39	○	1	0.56
10	22	×	1	0.5

表 4.2-b 検索結果例

このシステム中には検索質問画像に対して適合する画像が全部で 5 つ含まれていると仮定し、第 1 位から各順位までの画像をすべて検索結果として採用したとすると、再現率、精度は表 4.2 の第 4 列、5 列のように計算できる。

再現率を横軸に、精度を縦軸にとって、検索画像数を変化させて（再現率、精度）の点をプロットすると、再現率-精度グラフを書くことができる。図 4.1 は表 4.1 から書いた再現率精度グラフである。高い再現率でできるだけ高い精度を得ること、つまり曲線ができるだけ水平に近くなるようにすることがシステムの目標となる。

図 4.2-a 再現率-精度グラフの例

### 4.3 再現率と精度の平均

図 4.1 はシステムの性能を評価するための手がかりとなるがこれだけでは不便な面もある。2つのシステム A、B を比較する場合に、ある検索質問に対してはシステム A の性能がよいが、別の検索質問では B のほうが性能がよいこともあり得る。したがって、複数の検索質問について検索結果が得られたときに、総合的な性能を評価するためには何らかの方法で検索質問集合に対する平均的な値を計算する必要がある。本研究においては検索質問ごとに再現率、精度を計算し、それらの値をすべての検索質問にわたって平均するマクロ平均を採用する。再現率、精度のマクロ平均はそれぞれ式(4.3)式(4.4)で計算できる。

は質問要求の総数、添え字は検索質問を表している。

$$\bar{R} = \frac{1}{Q} \sum_{i=1}^Q \frac{w_i}{w_i + x_i} \quad (4.3)$$

$$\bar{P} = \frac{1}{Q} \sum_{i=1}^Q \frac{w_i}{w_i + y_i} \quad (4.4)$$

## 5 予備実験

### 5.1 予備実験について

高次自己相関に基づく特徴は、課題に依存しない一般的で基本的な特徴となっていて、画像中のオブジェクトの形状が反映されている。大津らは、幾何学図形およびタイプ英数字の認識に高次局所自

自己相関特徴量と判別分析を組み合わせオブジェクトの認識、個数のカウントに有効であることを示した。画像検索に適応した場合、任意のオブジェクトを対象とするためこのような方法は使えない。したがって25次の高次局所自己相関特徴量をそのままオブジェクトの形状を表す特徴量として扱い、オブジェクト検索に用いる特徴量として有意な結果が出るか実験を行う。

予備実験として特徴量が

- 1) オブジェクトの形状を認識して検索できるか
  - 2) オブジェクトの面積の変化に対応して検索ができるか
  - 3) さまざまな形状や面積のオブジェクトがある画像から任意のオブジェクトの検索ができるか
- について実験を行った。

## 5.2 予備実験 1(形状について)

### 実験設定

25 次の高次局所自己相関特徴量画がオブジェクトの形状を認識して検索できるかを確認する。また同じ形状で同じ面積のオブジェクトが複数ある場合にはベクトルの向きは変わらず大きさのみが2倍3倍になることが予想されるので内積をとることによって検索できるかを確認する。検索対象とする画像は図 5.2 のとおりである。

システムの構成は図 4.1 に示すとおりで、登録時には、図 4.2 に示す画像に対し高次局所自己相関特徴量をそれぞれ計算して特徴量をファイル名と共に配列に保持する。

図 4.2 の画像のうちオブジェクトが1個の画像をキー画像とする。ここでは処理を簡単にするため、あらかじめキーを蓄積画像に含めて登録しておく。これらは2値画像である。

検索時には、あらかじめ登録されたキーを指定して、素の特徴量と登録されている25個の特徴量をそれぞれ25次のベクトルとみなし、ベクトルの内積をとり値を降順でソートした。これらをJava言語によって実装した。

図 5.2-a 予備実験 1 のシステム

1つのオブジェクトサイズ  
[H:50.8mm, W:53.45mm]

1つのオブジェクトサイズ  
[H:50.8mm, W:50.8mm]

1つのオブジェクトサイズ  
[H:50.8mm, W:58.74mm]

1つのオブジェクトサイズ  
[H:50.8mm, W:50.8mm]

1つのオブジェクトサイズ  
[H:50.8mm, W:25.4mm]

**図 5.2-b 予備実験 1 の検索対象画像**  
図形は Microsoft Word のオートシェイプ機能を使って作成した

## 実験結果と考察

	検索キー：maru.gif		検索キー：sankaku.gif	
1	1.0000000000000000 0	maru3.gif	1.0000000000000000	sankaku.gif
2	1.0000000000000000 0	maru.gif	0.9999999992412240	sankaku2.g if
3	1.0000000000000000 0	maru2.gif	0.9999999991454170	sankaku3.g if
4	0.999999304638338 0	sikaku3.gif	0.9999956794057470	sikaku.gif
5	0.999999304638338 0	sikaku2.gif	0.9999956794057470	sikaku2.gif
6	0.999999304638338 0	sikaku.gif	0.9999956794057470	sikaku3.gif
7	0.999995504678653 0	sankaku3.g if	0.9999954105250400	maru2.gif
8	0.999995449730151 0	sankaku2.g if	0.9999954105250400	maru.gif
9	0.999995410525040 0	sankaku.gif	0.9999954105250390	maru3.gif
10	0.999964612626523 0	hosi.gif	0.9999765935921850	hosi.gif
11	0.999964612626523 0	hosi2.gif	0.9999765935921850	hosi2.gif
12	0.999964176041424 0	hosi3.gif	0.9999762834099130	hosi3.gif
13	0.999953266029611 0	tuki.gif	0.9999674888347440	tuki.gif
14	0.999952608781963 0	tuki3.gif	0.9999669627697010	tuki3.gif
15	0.999952390787082 0	tuki2.gif	0.9999668262234950	tuki2.gif

表 5.2-a 予備実験 1 の検索結果例

図 5.2 の画像のうちオブジェクトが 1 個の画像をキー画像として、高次局所自己相関特徴量と内積を用いた予備実験 1 の検索システムから検索結果例として表 3.1 を得た。他の検索結果は付録 1 とした。

結果から同種のオブジェクトの形状についてうまく検索できていることがわかる。また同種のオブジェクトを複数含む画像についても検索できていることがわかった。他の検索結果も同様であった。ベクトルの大きさに依存しない内積の性質は類似度計算として妥当であろうと考える。

## 5.3 予備実験 2(面積について)

### 実験設定

同種のオブジェクトが2個、3個と含まれた場合、高次局所自己相関特徴量のそれぞれの値は2倍3倍となる性質がある。この性質は特徴量をベクトルとして捉えたときにその成分が2倍、3倍にするが、ベクトルの向きを変えない。同種のオブジェクトの面積が大きく変わったときには、ベクトルの向きが変わることが予想され、検索できなくなる可能性がある。

同種のオブジェクトの面積の変化に対応して検索できるよう、検索時にオリジナルの検索キーの縮小、拡大画像(レプリカ)をつくり、オリジナルに加え、これも検索キーとして検索することでオブジェクトの面積の変化に対応できる有意な結果が出るか実験を行う。検索キーは図5.4であり、検索対象とする画像は図5.4をもとに表5.2のように拡大縮小した画像である。

オリジナルの検索キーとレプリカの検索キーの組み合わせを以下のように設定し、それぞれのシステムで各オブジェクトの検索を行う。そして各々の検索結果の上位20のところを平均再現率と平均精度を計算し、システムとして有効なレプリカのサイズの組み合わせを調べた。組み合わせはつぎのとおりである。

- ・オリジナルの検索キーと4%の面積のレプリカの検索キー
- ・オリジナルの検索キーと16%の面積のレプリカの検索キー
- ・オリジナルの検索キーと36%の面積のレプリカの検索キー
- ・オリジナルの検索キーと64%の面積のレプリカの検索キー
- ・オリジナルの検索キーと144%の面積のレプリカの検索キー
- ・オリジナルの検索キーと196%の面積のレプリカの検索キー
- ・オリジナルの検索キーと256%の面積のレプリカの検索キー
- ・オリジナルの検索キーと324%の面積のレプリカの検索キー
- ・オリジナルの検索キーと400%の面積のレプリカの検索キー
- ・オリジナルの検索キーと4%~64%の面積のレプリカの検索キー
- ・オリジナルの検索キーと4%~400%の面積のレプリカの検索キー
- ・オリジナルの検索キーと144%~400%の面積のレプリカの検索キー

**図 5.3-a 予備実験 2 のシステム**

システムの構成は図 5.3 に示すとおりである。検索対象とする画像は、キーオブジェクトを 1% ~ 200%までランダムに変化させた画像とする。

登録時には図 5.4 のオブジェクトの面積を変化させた画像に対し高次局所自己相関特徴量をそれぞれ計算して特徴量をファイル名と共に配列に保持する。

検索時にはオリジナルのキー画像を特徴抽出して類似度計算する。同時にオリジナルキー画像の拡大、縮小したもの（レプリカ）を数枚生成して、それぞれ特徴抽出して類似度計算を行う。拡大縮小の機能は Java の `AreaAveragingScaleFilter` クラスを使って行った。

最終的な検索結果は各々のキー画像から得られたな遺跡の値の大きいものから順に検索結果として採用し、降順でソートするシステムである。

図 5.3-b 変化させるキーオブジェクト

hosi.gif	maru.gif	sankaku.gif	sikaku.gif	tuki.gif
112	31	59	3	52
139	40	95	82	20
60	54	97	192	104
157	187	167	6	21
8	9	1	8	161
165	5	156	93	181
48	40	96	174	154
10	75	103	100	125
64	31	113	109	102
20	74	93	124	117
55	81	147	178	185
185	128	67	11	66
36	198	154	177	14
63	88	32	33	24
169	22	72	85	46
108	92	48	31	89
139	200	135	164	54
80	109	174	75	194
100	50	116	31	123
31	150	152	52	10

表 5.3-a 乱数によるオブジェクトのサイズ(1~200までの一様乱数を用いた)  
 hosi.gif 112 とは元のオブジェクトの縦と横をそれぞれ 1.12 倍した画像を意味する。

面積は図 5.4 のオブジェクトの 0.01% ~ 400%の値を取ることになる。

## 実験結果と考察

図 5.3-c 予備実験 2 検索結果

レプリカ 4、レプリカ 16 と重なり合っ て見にくい が、オリジナルはこのグラフの一番左下に位置している。

このグラフから面積の変化に対しレプリカを使った検索がある程度有効であることがわかる。またレプリカ 4-400 が一番よい成績を示しているようにオリジナルキーに対して拡大したレプリカと縮小したレプリカを複数併せ持つものがよい結果を示すと考えられる。

ただし、ここでは実験を行わなかったが、オリジナルを拡大するレプリカに対しては、その拡大率があまり大きいともとの特徴を反映しない別のオブジェクトを検索することが予想される。一般の検索において、レプリカをどこまで拡大すればよいかわからないので、たかだか 4 倍の面積までとした。

## 5.4 予備実験 3(幾何学図形による部分画検索)

### 実験設定

予備実験 1、および 2 において高次局所自己相関特徴量を用いてオブジェクトの形状と面積に対応した検索について調べ、ある程度有効であることがわかった。

部分画像検索では 1 つの画像中にさまざまな形状と面積を持ったオブジェクトが含まれている。一般的な画像の検索の実験をする前にオブジェクトがはっきりしている幾何学図形による部分画像検索に対し、有意な結果が出るか実験を行う。

予備実験 2 に基づき、オブジェクトの面積の変化に対応するためレプリカを用いる。

システムの構成は予備実験 2 と同じく図 5.3 に示すとおりである。

検索キーは図 5.6 のとおりでまた検索対象とする画像はそれをもとに一様乱数で作られた図 5.7 のとおりとする。

予備実験 2 と同じようにオリジナルの検索キーとレプリカの検索キーの組み合わせを以下のように設定し、それぞれのシステムで各オブジェクトの検索を行う。そして予備実験 3 では各々の検索結果の上位 10 のところで平均再現率と平均精度を計算し、システムとして有効なレプリカのサイズの組み合わせを調べた。組み合わせはつぎのとおりである。

- ・オリジナルの検索キーと 4% の面積のレプリカの検索キー
- ・オリジナルの検索キーと 16% の面積のレプリカの検索キー
- ・オリジナルの検索キーと 36% の面積のレプリカの検索キー
- ・オリジナルの検索キーと 64% の面積のレプリカの検索キー
- ・オリジナルの検索キーと 144% の面積のレプリカの検索キー
- ・オリジナルの検索キーと 196% の面積のレプリカの検索キー
- ・オリジナルの検索キーと 256% の面積のレプリカの検索キー
- ・オリジナルの検索キーと 324% の面積のレプリカの検索キー
- ・オリジナルの検索キーと 400% の面積のレプリカの検索キー
- ・オリジナルの検索キーと 4% ~ 64% の面積のレプリカの検索キー
- ・オリジナルの検索キーと 4% ~ 400% の面積のレプリカの検索キー
- ・オリジナルの検索キーと 144% ~ 400% の面積のレプリカの検索キー

-

図 5.4-a もととなるキーオブジェクト

図 5.4-b 検索対象の画像例

検索対象となる画像は中に入るオブジェクトの個数、オブジェクトの形状、面積などすべて乱数によって生成する。オブジェクトの個数は1~5個、形状は図 5.6 のいずれか、また面積は図 5.6 のオブジェクトの0.01%~400%の値を取る。

こうしてできた検索対象画像は全部で50個である。このテストデータは付録2とする。

## 実験結果と考察

### 図 5.4-c 予備実験 3 検索結果

このグラフでの平均再現率と平均精度は図 5.6 のそれぞれのオブジェクトで部分画検索した結果の上位 10 について計算している。

オリジナルの値は (0.338175、0.47) である。これに対しオリジナルキーとオリジナルの 4%、16%、36%、64% の面積を検索キーとするレプリカ 4-64 の平均再現率-平均精度での値はよくない。しかしながら、レプリカ 4-400 の値はやはりよい値が出ている。検索した結果の上位 10 の画像はオリジナルキーより面積が大きいものが含まれているのだろうか。レプリカ 144-400 の検索結果例をつぎに示す。

### 図 5.4-d レプリカ 144-400 による検索例

左から順に星、丸、ダイア、三角をレプリカ 144-400 で検索して 1、2 番目にきたものである。これらはうまく検索できていて、検索対象のオブジェクトの面積はいずれもキーオブジェクトよりも大きい。

これらの画像では他のオブジェクトが相対的に小さい分ノイズが少なく、このシステムで検索されやすいことが予想される。このような構図の画像においては、しばしばキーオブジェクトよりサイズが大きいことが予想されるので、一般的な部分画検索でもオブジェクトより大きなサイズのレプリカを用いることは重要であると考えられる。

# 6 高次局所自己相関特徴量に基づく部分画検索システム

## 6.1 提案するシステム

予備実験を通して、高次局所自己相関特徴量が幾何学図形の部分画像検索に対して適用できることを示した。ここでは写真画像への適用を試みて、一般的な画像に対しても有意な部分画像検索が行えるかを実験する。また同時に RGB ヒストグラムでの検索も行い、比較する。

これまで 2 値の幾何学図形を取り扱ってきたため高次局所自己相関特徴量画像内のオブジェクトの形状を捉えることができた。写真画像においては色を扱うため、本研究では、画像中のオブジェクトの形を捉えるために RGB 色尺度と、色相 (Hue)、輝度 (Saturation)、明度 (Brightness) によって色を表す HSB 色尺度を用いる。

色尺度を用いたシステムとして、

- 1) RGB を用いて高次局所自己相関特徴量を計算し部分画検索を行うシステム
- 2) HSB を用いて高次局所自己相関特徴量を計算し部分画検索を行うシステム
- 3) RGB ヒストグラムで検索を行うシステム

の 3 つのシステムで写真画像を対象として部分画検索を行い、有意な結果が出るか実験を行う。

## RGBを用いて高次局所自己相関特徴量を計算し部分画検索を行うシステム

### 図 6.1-a RGBを高次局所自己相関特徴に用いた部分画検索システム

基本的に予備実験2のシステムと変わらない。画像の登録や検索の高次局所自己相関特徴量の計算に前述の方法でRGBを用いる。RGBを用いるのは単に取り扱いやすいからである。

類似度計算でキーオブジェクトと対象画像それぞれRGBについて25次の高次局所自己相関特徴量がもとまるので、75次で内積をとって検索結果としたものと、キーオブジェクトのRGBの総量から重みを求めRGBそれぞれの内積の結果に重みをつけて足し合わせ検索結果としたものもちいて実験した。

拡大縮小にはJavaのReplicateScaleFilterクラスを用いて、オリジナルの面積の20%、40%、60%、80%、(100%)、120%、140%、...400%のレプリカを作ってオリジナルの100%を含めて20個の検索キーで検索した。

プログラム上では特徴量を  $25 \times 3$  の配列として確保しているので  
キー画像のベクトルを  $KeyFeature[1] \sim KeyFeature[75]$ 、  
また対象画像のベクトルは  $Feature[1] \sim Feature[75]$  である。  
1~25 までが R 成分、26~50 までが G 成分、51~75 までが B 成分と  
なる。

ここで高次局所相関特徴量の 0 次にあたる  $KeyFeature[1]$ 、  
 $KeyFeature[26]$ 、 $KeyFeature[51]$ 、 $Feature[1]$ 、 $Feature[26]$ 、 $Feature[51]$   
は R、G、B の総量の足し合せなので重みは以下のように計算できる。

$$R \text{ の重み} : \alpha = \frac{KeyFeature[1]}{KeyFeature[1] + KeyFeature[26] + KeyFeature[51]}$$

$$G \text{ の重み} : \beta = \frac{KeyFeature[26]}{KeyFeature[1] + KeyFeature[26] + KeyFeature[51]}$$

$$B \text{ の重み} : \gamma = \frac{KeyFeature[51]}{KeyFeature[1] + KeyFeature[26] + KeyFeature[51]}$$

とし、RGB25 次ごとにとった内積にそれぞれの重みをかけて計算し  
た。

RGB を用いた部分画検索システムでは以上より  
オリジナルの検索キーだけで検索したもの

- ・ 「RGB 自己相関」と表記する

オリジナルとその面積を変えたレプリカで検索したもの

- ・ 「RGB 自己相関+複製」と表記する

について調べた。

## HSBを用いて高次局所自己相関特徴量を計算し部分画検索を行うシステム

### 図 6.1-b HSB を用いた部分画検索システム

これも基本的に予備実験 2 のシステムと変わらない。HSB をもちいるのは色相によってオブジェクトを見たときオブジェクトが固有の色を持っていると仮定すると、形状がわかりやすくなることが予想されたからである。色相を前述のように 6 つで分けて取り扱うことにした。

類似度計算でキーオブジェクトと対象画像それぞれの色相について 25 次の高次局所自己相関特徴量がもとまるので、150 次で内積をとって検索結果としたものと、キーオブジェクトのそれぞれの色相の総量から重みを求めそれぞれの色相の内積の結果に重みをつけて足し合わせ検索結果としたものを持ちいて実験した。

拡大縮小には Java の `ReplicateScaleFilter` クラスを用いて、オリジナルの面積の 20% , 40% , 60% , 80% , ( 100% ) , 120% , 140% , ...400% のレプリカを作ってオリジナルの 100% を含めて 20 個の検索キーで検索した。

プログラム上では特徴量を  $25 \times 6$  の配列として確保しているの

キー画像のベクトルを KeyFeature [ 1 ] ~ KeyFeature [ 150 ]

また対象画像のベクトルは Feature[1] ~ Feature[150]である。

1 ~ 25 までが R 成分、26 ~ 50 までが G 成分、51 ~ 75 までが B 成分、  
76 ~ 100 までが C 成分、101 ~ 125 までが M 成分、126 ~ 150 までが Y  
成分となる。

ここで高次局所相関特徴量の 0 次にあたる KeyFeature[1]、  
KeyFeature[26]、KeyFeature[51]、...、KeyFeature[126]は R、G、B、C、  
M、Y の総量の足し合せなので重みは以下のように計算できる。

R の重み：

$$\alpha = \frac{KeyFeature[1]}{KeyFeature[1] + KeyFeature[26] + KeyFeature[51] + \dots + KeyFeature[126]}$$

G の重み：

$$\beta = \frac{KeyFeature[26]}{KeyFeature[1] + KeyFeature[26] + KeyFeature[51] + \dots + KeyFeature[126]}$$

B の重み：

$$\gamma = \frac{KeyFeature[51]}{KeyFeature[1] + KeyFeature[26] + KeyFeature[51] + \dots + KeyFeature[126]}$$

...

C の重み：

$$\zeta = \frac{KeyFeature[126]}{KeyFeature[1] + KeyFeature[26] + KeyFeature[51] + \dots + KeyFeature[126]}$$

とし、RGBCMY ごとにとった内積にそれぞれの重みをかけて計算した。

HSB を用いた部分画検索システムでは色相(H)を取り扱う上で後述する 2 種類の特徴抽出を用いて実験した。

それぞれ特徴抽出方法 1、特徴抽出方法 2 とすると、

特徴抽出方法 1 をつかってオリジナルの検索キーだけで検索したものの

- ・ 「HSB 自己相関 1」と表記する

オリジナルとその面積を変えたレプリカで検索したものの

- ・ 「RGB 自己相関 1+複製」と表記する

特徴抽出方法 2 をつかってオリジナルの検索キーだけで検索したもの

- ・ 「HSB 自己相関 2」と表記する

オリジナルとその面積を変えたレプリカで検索したもの

- ・ 「RGB 自己相関 2+複製」と表記する

について調べた。

R G B色尺度をそのままちいて高次局所自己相関特徴量を計算すると白色R:255,G:255,B:255が多い画像について値が大きい領域をオブジェクトとして計算するため、オブジェクトをうまく検索できないことが予想される。そこでオブジェクトがはっきりとした色を持つと仮定して、彩度が高くまた明度も高い部分のみ色相Hの値を大きくするような手法を取った。高次局所自己相関特徴量の局所パターンのM2の計算は

対象画像  $f(\mathbf{r})$ 、変位  $\mathbf{a} = (1,0)$  とすると

$$\sum f(\mathbf{r}) \cdot f(\mathbf{r} + \mathbf{a})$$

であった。

HSBカラーモデル

H S B色尺度の場合、彩度が高く明度も高い部分のみ値を大きくすることにした。

彩度  $S$ 、明度  $B$ 、

対象画像

$$\text{彩度}(S) \quad R(\mathbf{r}, h), Y(\mathbf{r}, h), G(\mathbf{r}, h), C(\mathbf{r}, h), B(\mathbf{r}, h), M(\mathbf{r}, h)$$

とする

$$\begin{aligned} & \sum s \cdot b \cdot R(\mathbf{r}, h) \cdot s \cdot b \cdot R(\mathbf{r} + \mathbf{a}, h) \\ & \sum s \cdot b \cdot Y(\mathbf{r}, h) \cdot s \cdot b \cdot Y(\mathbf{r} + \mathbf{a}, h) \\ & \sum s \cdot b \cdot M(\mathbf{r}, h) \cdot s \cdot b \cdot M(\mathbf{r} + \mathbf{a}, h) \end{aligned} \quad \dots$$

特徴抽出方法1では、H S B尺度の場合の対象画像の関数は色相にあるかないか0, 1の2値をとる関数を用いる

一方、特徴抽出方法2では、試験的に滑らかな値をとるようにしてみた。

明度(b)

色相(h)

h は Red : 0 度、Yellow : 60 度、Green : 120 度、Cyan : 180 度、Blue : 240 度、Magenta : 300 度をとる。そこで、特徴抽出方法 1 では 6 個のブロックに区切り、

$$\begin{aligned} 330 < h & 30 & R(\mathbf{r}, h) &= 1 \\ 30 < h & 90 & Y(\mathbf{r}, h) &= 1 \\ 90 < h & 150 & G(\mathbf{r}, h) &= 1 \\ 150 < h & 210 & C(\mathbf{r}, h) &= 1 \\ 210 < h & 270 & B(\mathbf{r}, h) &= 1 \\ 270 < h & 330 & M(\mathbf{r}, h) &= 1 \end{aligned}$$

となる関数を用いた。ここで、 $\mathbf{r}$  は画像内の参照点である。

一方、特徴抽出方法 2 では h の値を 6 倍して連続値をとるようにした

$$\begin{aligned} 0 < h & & R(\mathbf{r}, h) &= \cosh + 1 \\ & < h & 3 & G(\mathbf{r}, h) = \cosh + 1 \\ 3 & < h & 5 & B(\mathbf{r}, h) = \cosh + 1 \\ 5 & < h & 6 & R(\mathbf{r}, h) = \cosh + 1 \\ 0 < h & 2 & Y(\mathbf{r}, h) &= \sin\left(h - \frac{\pi}{2}\right) + 1 \\ 2 & < h & 4 & C(\mathbf{r}, h) = \sin\left(h - \frac{\pi}{2}\right) + 1 \\ 4 & < h & 6 & M(\mathbf{r}, h) = \sin\left(h - \frac{\pi}{2}\right) + 1 \end{aligned}$$

このように求めた値を高次局所自己相関特徴の計算に用いた。

## RGB ヒストグラムで検索を行うシステム

### 図 6.1-c RGB ヒストグラムを用いた検索システム

RGB カラーヒストグラムでは RGB 色尺度で表現された画像全体に関して、RGB の各色ごとに画像内の画素値を積算することで各 RGB 値を横軸とするヒストグラムを構成できる。実際には RGB の 256 階調そのものを用いると人間にとって近い色であっても計算上、別の色相として判断されるため、次元縮退が行われる。ここでは RGB それぞれについて 16 次元に縮退させ、計 48 次元で特徴量とし、内積をとって検索結果とする。

前述した 2 つの部分画検索との比較に用いる。

## 実験データ

図 6.1-d 検索キーオブジェクト

図 6.1-e 6009.jpg に対して適合するとした画像の例 (3/8)

図 6.1-f FD1179.jp g に対して適合するとした画像の例 (3/7)

図 6.1-g FD1034.jpg に対して適合するとした画像の例 (3/9)

図 6.1-h その他の画像例

\*実験データは 具満タン写真素材 具満タン 6 写真素材 2 から 224 枚とした。これを付録 3 とする。

## 実験結果と考察

図 6.1-i トマト[6009.jpg]の検索結果

図 6.1-j バナナ[FD1179.jpg]の検索結果

**図 6.1-k ミカン[FD1034.jpg]の検索結果**

赤い色のトマトや黄色のバナナでは、RGB ヒストグラムや RGB の値を使った高次局所自己相関特徴による検索よりは、HSB の値をつかった高次局所自己相関特徴による検索のほうが精度がよい状態で検索できる。

その中間職のミカンではあまりよい精度で検索できていない。

検索キーのレプリカ(複製)を用いた検索は必ずしもよい結果を出さないことがわかる。

RGB の値を使った高次局所自己相関特徴による検索では、白色 (R:255,G:255,B:255)は大きな値をとるので形状が捉えられなくなっていることが考えられる。

幾何学図形の部分画検索では部分画検索においてレプリカを使うほうが成績がよい傾向があったので、HSB 自己相関 2+複製は形状をうまく捉えている可能性があると考えられる。

最後に HSB 自己相関 2+複製による検索例を示す。トマトと、ミカンを上位 20 位まで検索結果として採用したものである。





## 7 むすび

本研究では従来の部分画像検索の問題点であるオブジェクトの切出しを行うことなく部分画の検索を行うシステムとして、全体画像の検索をモデルにオブジェクトの切出しによる画像内情報の欠落を防ぎながら検索を行うことを試みた。

この実現のために、画像全体に高次局所自己相関特徴量をもちいて部分画像検索を行うことを試みた。予備実験では、検索キーオブジェクトの面積を変えた複製画像を用意することによって、他のオブジェクトより相対的に大きなオブジェクトは検索されやすいことがわかり、幾何学図形の部分画検索に有意な結果が出るようになった。

一般の部分画像検索では、色尺度を用いることによってオブジェクトの形状を見分けることを試みた。今回の結果においては、一般的な全体画検索の手法の一つである、カラーヒストグラムと同様か、若干よい成績を示すことがわかった。

色尺度の改良もさることながら、キーオブジェクトから高次局所自己相関特徴量を計算して得られる 25 次の特徴のうち最もキーの特徴をよく表すものだけを何らかの方法で(例えば複数の検索キーからの統計的特徴を用いる等)抽出して、類似度の計算に反映させることが高次局所自己相関特徴量を部分画像検索に用いる上で本質的な改良となると考えられる。

## 8 参考文献

- [1] A.Gupta, R.Jain: Visual Information Retrieval. Communications of the ACM, Vol40, No.5,pp.71-79(1997)
- [2] 串間和彦, 赤間浩樹, 紺谷精一, 木本晴夫, 山室雅司, オブジェクトに基づく高速画像検索システム: Exsight, 情報処理学会論文誌, Vol.40, No.2, pp.732-741, 1999
- [3] J.R.Smith, S.F.Chang: VisualSEEK: A Fully Automated Content-Based Image Query System, Proc. ACM International Conference on Multimedia, pp.87-93(1996)
- [4] 金原史和, 佐藤真一, 浜田喬, プリミティブ分解による多様な検索条件を扱うカラー画像検索, 情報処理学会論文誌, Vol.37, No.11, pp1989-2000, (1996)
- [5] M.Flicker et al: Query by Image and Video Content: The QBIC System, IEEE Computer, Vol.28, No.9, pp.23-32(1995)
- [6] N.Otsu and T.Kurita,: A new scheme for practical flexible and intelligent vision systems, Proc. IAPR Workshop on Computer Vision, Tokyo, pp.431-435, (1988)
- [7] O.Hasegawa, K.Itou, T.Kurita, S.Hayamizu, K.Tanaka, K.Yamamoto, and N.Otsu: Active agent oriented multimodal interface system, Proc.of IJCAI'95, pp.82-87, (1995)
- [8] 赤間浩樹, 紺谷精一, 三井一能, 串間和彦, 画像内オブジェクトの自動抽出を使った画像検索システム Exsight-写真(PhotoDisk)への適用-, 電子情報通信学会技術報告書, Vol.DE97, No.31, pp.25-30, (1997)
- [9] 小早川倫広, 星守, 大森匡, 照井武彦, ウェー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, 情報処理通信学会論文誌, Vol.40, No.3, pp.899-911(1999)
- [10] 木本晴夫, 感性語による画像検索とその精度評価, 情報処理学会論文誌, Vol.40, No.3, pp.866-898(1999)

- [11] 串間和彦, 赤間浩樹, 紺谷精一, 山室雅司, 色や形状に基づく画像内容検索技術, 情報処理学会論文誌, Vol.40, No.SIG3, pp.171-184(1999)
- [12] 小早川倫広, 星守, 画像内容に基づいた画像検索システム, bit, Vol.31, No.10, pp.23-34(1999)
- [13] 大津展之, 島田俊之, 森俊二, N次自己相関マスクによる図形の特徴抽出, 信学技報, PRL78-31, (1978)

## 9 付録

### 予備実験 1 の検索結果

検索キー : key:maru.gif W:50.8mm]	検索キー : key:sankaku.gif W:58.74mm]
1.0000000000000000maru3.gif	1.0000000000000000sankaku.gif
1.0000000000000000maru.gif	0.999999992412240sankaku2.gif
1.0000000000000000maru2.gif	0.999999991454170sankaku3.gif
0.999993046383380sikaku3.gif	0.999956794057470sikaku.gif
0.999993046383380sikaku2.gif	0.999956794057470sikaku2.gif
0.999993046383380sikaku.gif	0.999956794057470sikaku3.gif
0.999955046786530sankaku3.gif	0.999954105250400maru2.gif
0.999954497301510sankaku2.gif	0.999954105250400maru.gif
0.999954105250400sankaku.gif	0.999954105250390maru3.gif
0.9999646126265230hosi.gif	0.9999765935921850hosi.gif
0.9999646126265230hosi2.gif	0.9999765935921850hosi2.gif
0.9999641760414240hosi3.gif	0.9999762834099130hosi3.gif
0.9999532660296110tuki.gif	0.9999674888347440tuki.gif
0.9999526087819630tuki3.gif	0.9999669627697010tuki3.gif
0.9999523907870820tuki2.gif	0.9999668262234950tuki2.gif

検索キー : key:sikaku.gif W:50.8mm]	検索キー : key:hosi.gif W:53.45mm]
0.9999999999999990sikaku3.gif	1.0000000000000000hosi2.gif
0.9999999999999990sikaku.gif	1.0000000000000000hosi.gif
0.9999999999999990sikaku2.gif	0.999999968154570hosi3.gif
0.999993046383380maru.gif	0.9999915536582780tuki.gif
0.999993046383380maru3.gif	0.9999914789620980tuki2.gif
0.999993046383380maru2.gif	0.9999914412471900tuki3.gif
0.999957614430340sankaku3.gif	0.9999765935921850sankaku.gif
0.999957081547400sankaku2.gif	0.9999764941910140sankaku2.gif
0.999956794057470sankaku.gif	0.9999764256312500sankaku3.gif
0.9999669707939400hosi.gif	0.9999669707939400sikaku3.gif
0.9999669707939400hosi2.gif	0.9999669707939400sikaku2.gif
0.9999665785082530hosi3.gif	0.9999669707939400sikaku.gif
0.9999556852773130tuki.gif	0.9999646126265230maru2.gif
0.9999550242845320tuki3.gif	0.9999646126265230maru.gif

0.9999548241308900	tuki2.gif	0.9999646126265230	maru3.gif
検索キー : key:tuki.gif	W:25.4mm]		
1.0000000000000000	tuki.gif		
0.9999999961528250	tuki3.gif		
0.9999999947953150	tuki2.gif		
0.9999917871119630	hosi3.gif		
0.9999915536582780	hosi2.gif		
0.9999915536582780	hosi.gif		
0.9999674888347440	sankaku.gif		
0.9999672988902080	sankaku3.gif		
0.9999672887380190	sankaku2.gif		
0.9999556852773130	sikaku3.gif		
0.9999556852773130	sikaku2.gif		
0.9999556852773130	sikaku.gif		
0.9999532660296110	maru2.gif		
0.9999532660296110	maru.gif		
0.9999532660296110	maru3.gif		

### 予備実験3 テストデータ

pict[0] dia(143) plus(72) heart(77)plus(169)  
pict[1] plus(178) heart(175) sikaku(146)  
pict[2] heart(107) heart(139) dia(17) tuki(183)  
pict[3] inazuma(94)  
pict[4] heart(90)dia(153) heart(148) tuki(84)  
pict[5] tuki(84)  
pict[6] sikaku(79)  
pict[7] plus(150) dia(41) maru(102)  
pict[8] maru(1)  
pict[9] hosi(176) dia(102)  
pict[10] tuki(192) tuki(117) inazuma(193) heart(113)  
hosi(103)  
pict[11] plus(26) plus(36) hosi(10)  
pict[12] inazuma(84)  
pict[13] sikaku(116) hosi(86) sankaku(21)  
pict[14] plus(169)  
pict[15] taiyou(96) inazuma(27) taiyou(90) maru(82)  
pict[16] taiyou(124) maru(177) sikaku(162) heart(111)  
pict[17] sankaku(48) hosi(75) plus(156) taiyou(79) hosi(180)  
  
pict[18] sankaku(60) inazuma(49) dia(120)  
pict[19] maru(52) hosi(132) heart(101) tuki(108)  
inazuma(98)  
pict[20] taiyou(187)  
pict[21] hosi(122)  
pict[22] sikaku(55) sikaku(146) taiyou(159) taiyou(23)  
pict[23] dia(118) inazuma(30) taiyou(77) taiyou(16)  
sankaku(161)  
pict[24] sikaku(129) sankaku(13) maru(131) dia(160) taiyou(46)  
  
pict[25] heart(187) sankaku(115) plus(130) inazuma(65)  
pict[26] hosi(165) plus(9) plus(31) taiyou(7)plus(19)  
pict[27] tuki(87) sikaku(158)  
pict[28] tuki(75) sankaku(137) dia(32) hosi(143)  
pict[29] sikaku(14) sankaku(24) heart(62) taiyou(1) hosi(80)  
pict[30] heart(78) dia(165) hosi(53)

pict[31] hosi(139) maru(19) sikaku(42) sikaku(43)  
 dia(16)

pict[32] sankaku(72) maru(26)

pict[33] maru(171) hosi(15) heart(53)

pict[34] taiyou(180)

pict[35] sikaku(63) sikaku(186) sikaku(4) sankaku(37)

pict[36] heart(93)sankaku(166) sikaku(69)

pict[37] sankaku(62) sikaku(132) inazuma(168) tuki(29) sikaku(27)

pict[38] maru(134) dia(192) dia(65) hosi(148)

pict[39] heart(28)

pict[40] maru(26) sankaku(52) dia(90) dia(33) hosi(109)

pict[41] sikaku(10) sikaku(23) tuki(104) plus(90)

pict[42] maru(43)

pict[43] tuki(96) dia(22) maru(47) heart(193) maru(155)

pict[44] hosi(186) tuki(68) sankaku(174)

pict[45] sankaku(135) maru(157) inazuma(131)

pict[46] plus(17) taiyou(156) sikaku(15)

pict[47] maru(182) sankaku(71) plus(61) taiyou(193) tuki(151)

pict[48] maru(13) taiyou(192) sikaku(64) maru(64)  
 taiyou(37)

pict[49] dia(71) sikaku(169)



6001



6002



6003



6004



6005



6006



6007



6008



6009



6010



6011



6012



6013



6014



6015



FD1001



FD1002



FD1003



FD1004



FD1005



FD1006



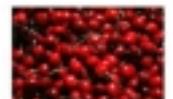
FD1007



FD1008



FD1009



FD1010



FD1011



FD1012



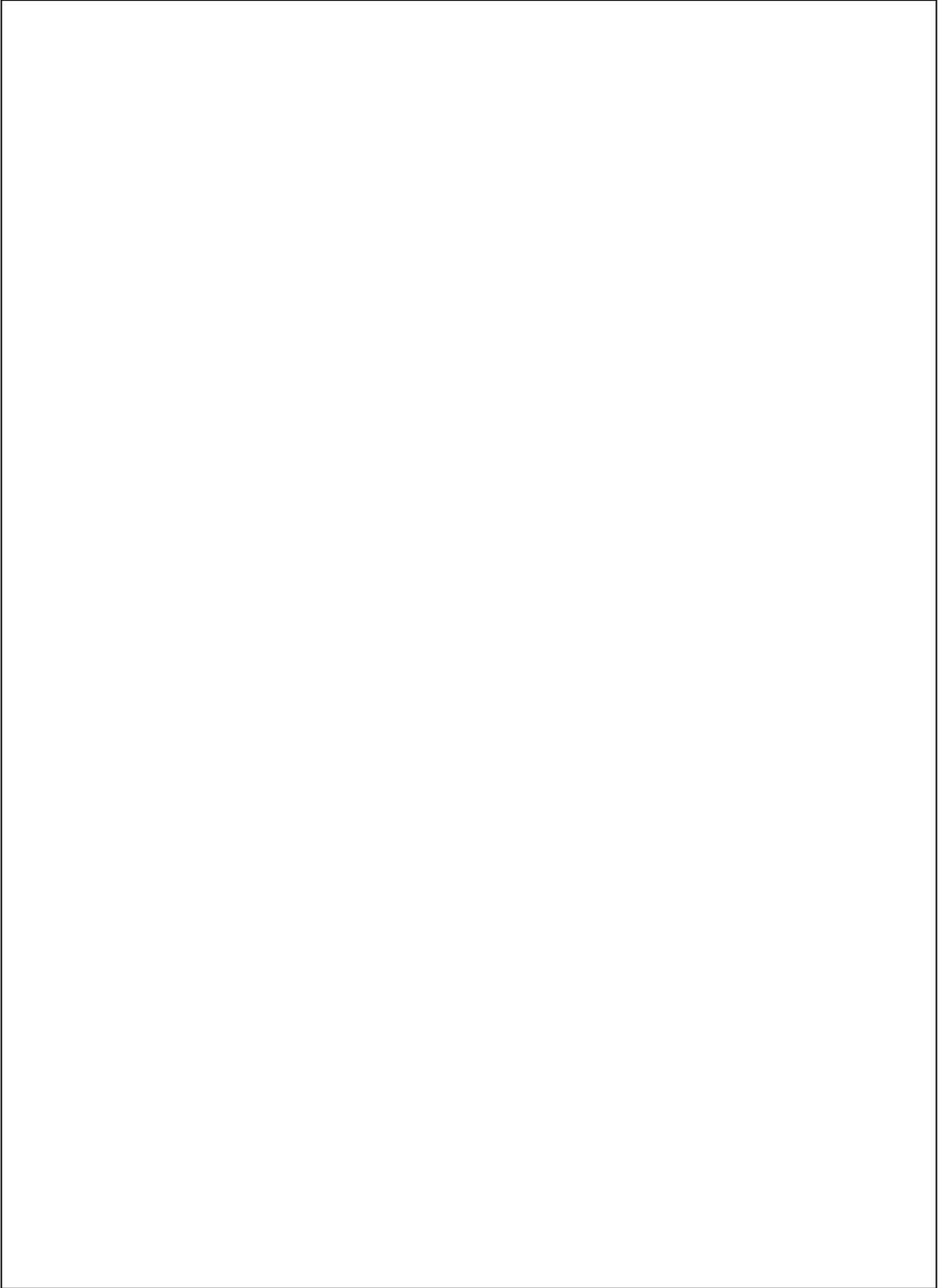
FD1013

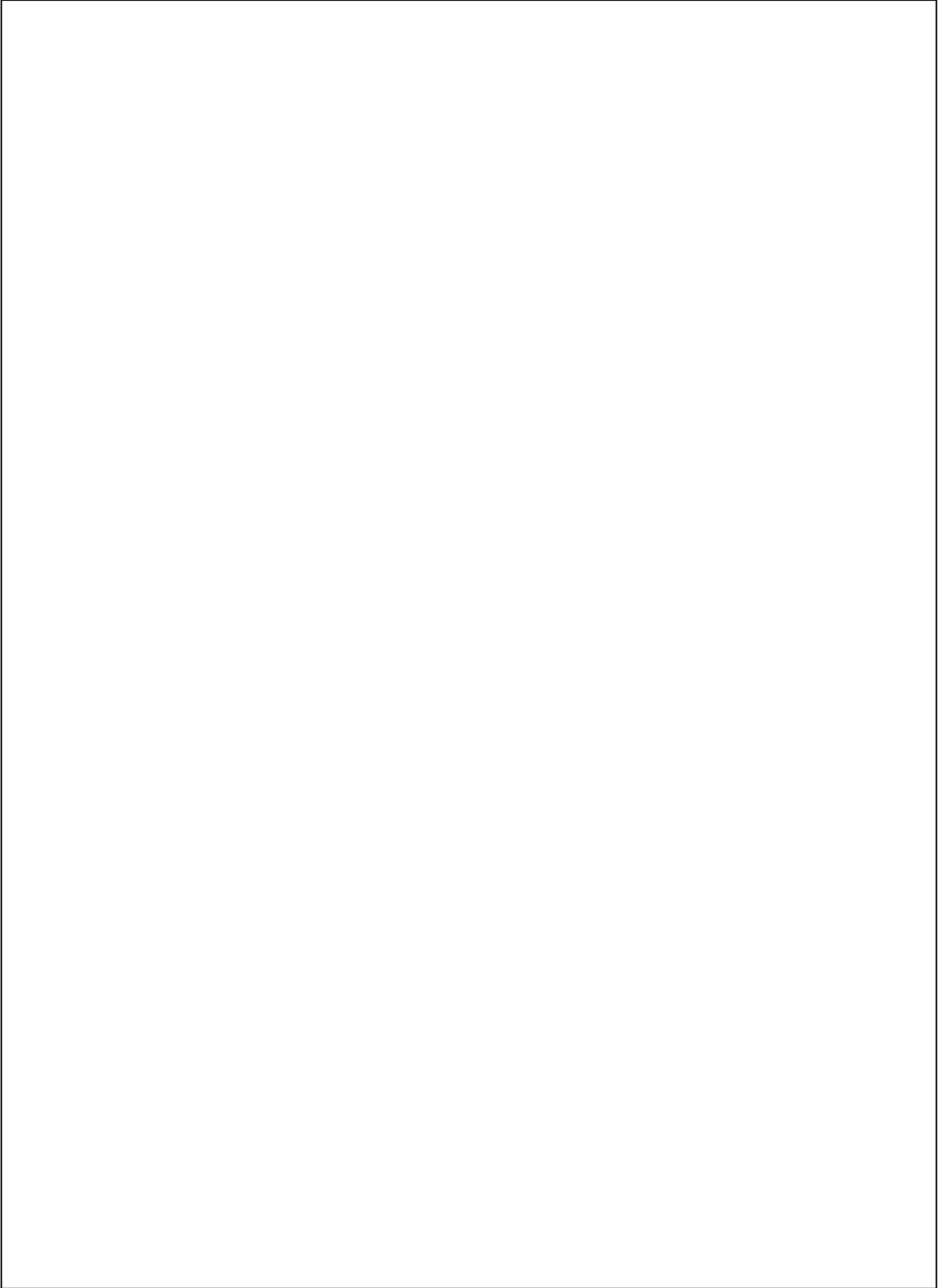


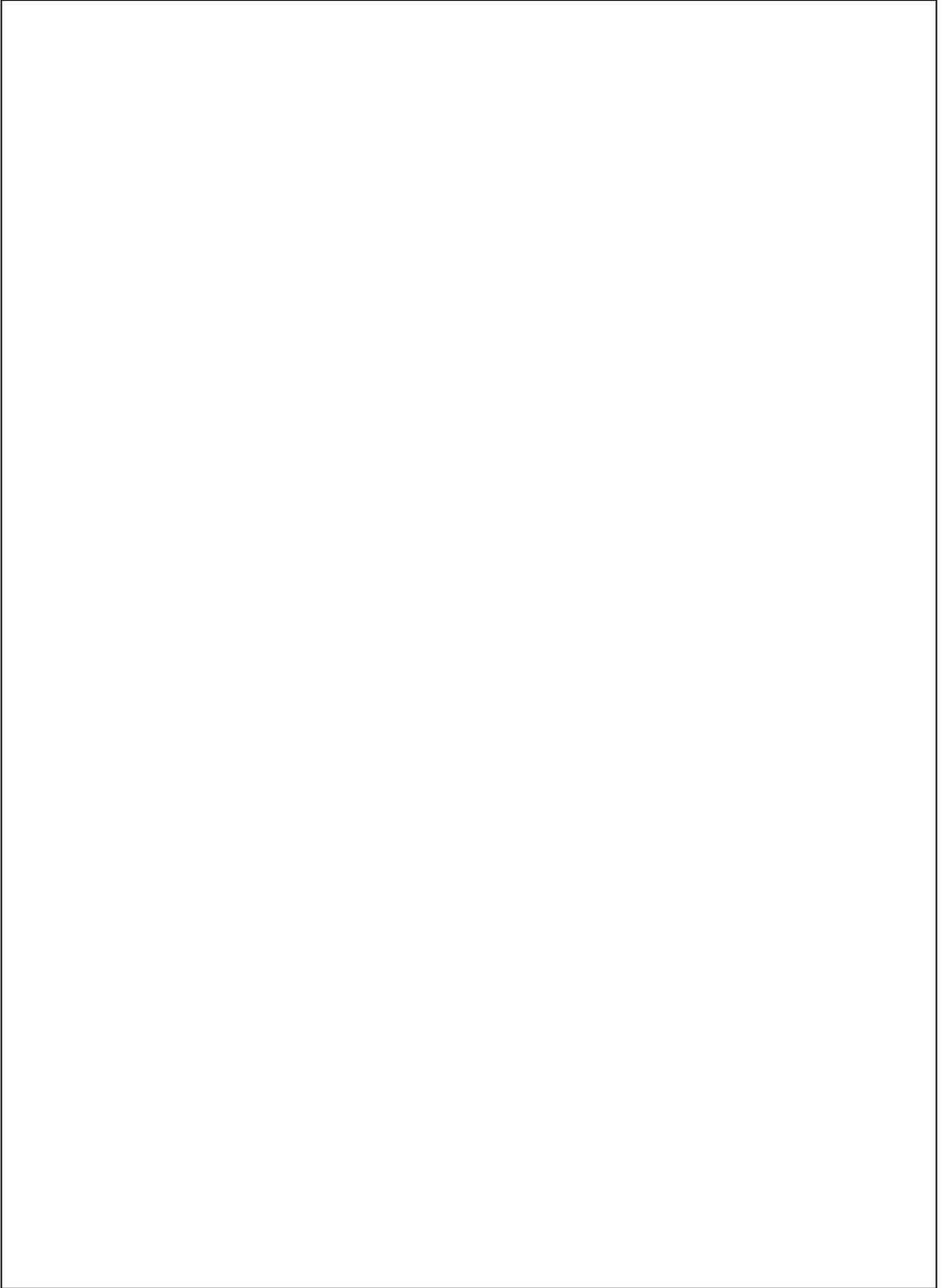
FD1014



FD1015









FD1106



FD1107



FD1108



FD1109



FD1110



FD1111



FD1112



FD1113



FD1114



FD1115



FD1116



FD1117



FD1118



FD1119



FD1120



FD1121



FD1122



FD1123



FD1124



FD1125



FD1126



FD1127



FD1128



FD1129



FD1130



FD1131



FD1132



FD1133



FD1134



FD1135



FD1136



FD1137



FD1138



FD1139



FD1140



FD1141



FD1142



FD1143



FD1144



FD1145



FD1146



FD1147



FD1148



FD1149



FD1150



FD1151



FD1152



FD1153



FD1154



FD1155



FD1156



FD1157



FD1158



FD1159



FD1160



FD1161



FD1162



FD1163



FD1164



FD1165



FD1166



FD1167



FD1168



FD1169



FD1170



FD1171



FD1172



FD1173



FD1174



FD1175



FD1176



FD1177



FD1178



FD1179



FD1180



FD1181



FD1182



FD1183



FD1184



FD1185



FD1186



FD1187



FD1188



FD1189



FD1190



FD1191



FD1192



FD1193



FD1194



FD1195



FD1196



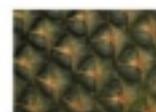
FD1197



FD1198



FD1199



FD1200



FD1201



FD1202



FD1203



FD1204



FD1205



FD1206



FD1207



FD1208



FD1209

