

## 大規模画像 DB からの類似画像検索による改変画像検出

馬越 健治<sup>†</sup> 糟谷 勇児<sup>††</sup> 山名 早人<sup>†††,††††</sup>

<sup>†</sup> 早稲田大学理工学部 〒169-8555 東京都新宿区大久保 3-4-1

<sup>††</sup> 早稲田大学大学院理工学研究科 〒169-8555 東京都新宿区大久保 3-4-1

<sup>†††</sup> 早稲田大学理工学術院 〒169-8555 東京都新宿区大久保 3-4-1

<sup>††††</sup> 国立情報学研究所 〒101-8430 東京都千代田区一ツ橋 2-1-2

E-mail: <sup>†</sup>, <sup>††</sup> {umakoshi,kasuya-u}@yama.info.waseda.ac.jp, <sup>†††</sup> yamana@waseda.jp

**あらまし** 本稿では、元画像を改変した画像を、大量の画像が登録された元画像 DB に対して類似画像検索を行うことで検出する手法を提案する。提案手法では、画像全体の特微量を用いた第一段階の検索で元画像の候補画像を絞り込み、部分特微量を用いた第二段階の検索で元画像を特定する。この二段階検索によって元画像 DB の大規模化に対応する。約 48 万枚の元画像を用意して実験を行い、DB が大規模な場合にも、提案手法が多様な改変画像を平均約 7 秒で 81.4% 検出できることを確認した。

**キーワード** 大量画像, 著作権保護, 類似画像検索, アクティブ探索

## Modified Images Detection using Content Based Image Retrieval from large image-database

Kenji UMAKOSHI<sup>†</sup> Yuji KASUYA<sup>††</sup> and Hayato YAMANA<sup>†,†††</sup>

<sup>†</sup>Science and Engineering, Waseda University 3-4-1 Okubo, Shinjuku-ku, Tokyo 169-8555 Japan

<sup>††</sup>Graduate School of Science and Engineering, Waseda University 3-4-1 Okubo, Shinjuku-ku, Tokyo 169-8555 Japan

<sup>†††</sup>National Institute of Informatics 2-1-2 Hitotsubashi, Chiyoda-ku, Tokyo 101-8430 Japan

E-mail: <sup>†</sup>, <sup>††</sup> {umakoshi,kasuya-u}@yama.info.waseda.ac.jp, <sup>†††</sup> yamana@waseda.jp

**Abstract** In this research, we propose the method for detecting the modified images by content based image retrieval (CBIR) on large original image-database. Proposed method consists of two stages. In the first stage of retrieving, the candidate images of original images are extracted from original image-database using the feature of an entire image. In the second stage of retrieving, original images are specified using the feature of a partial image. With those two stages of retrieval, proposed method is capable of dealing with large-scale of original image-database. We experimented on 480,000 original images, and confirmed that proposed method were able to detect 81.4% modified images in 7 seconds.

**Keyword** MassImages, Copyright Protection, CBIR, Active Search

### 1. はじめに

近年、コンピュータの高性能化やストレージの低価格化、ネットワークの高速化、デジタルカメラの普及などによって、個人が扱えるデジタル画像の量が爆発的に増加している。また、市販あるいはフリーのソフトウェアによって、コンピュータに詳しくない人でも容易に画像を加工できるようになっている。このような状況の下で、著作権を持つ画像が無断で改変され、公開されてしまう事態が増加している。特に最近では、画像掲示板や個人のブログなどに簡単に画像を掲載できるようになっているため、以前よりデジタル画像の公開が簡単になっている。著作権を持つ元画像及び改変画像の数が少ない場合には、人手で改変画像を検出することも可能であるが、元画像、改変画像共に常に大量に生み出されている現在において、人手で改変画像を検出することは困難である。そこで、改変画像を自動で検出できるシステムが必要となる。

改変画像を検出するシステムとして、電子透かしを用いる方法が広く研究されている[1]。電子透かしは、画像中に人間の目にはわからない程度の色の变化を施すことで、著作権情報を埋め込む技術である。しかし、

1. 透かしの画像公開前につける必要があるため、過

去にアップロードされた画像には対応できない。

2. アルゴリズムが非公開の透かし方式は、アルゴリズムが流出すると透かしを消す攻撃に弱くなる。

などの欠点がある。電子透かし以外の方法として、文献[2-4]のシステムでは、類似画像検索技術を用いて改変画像の検出を試みている。類似画像検索とは、画像の内容情報から特微量を取り出し、画像間の特微量の類似度を計算することで、見た目に似た画像を検索する技術である。類似画像検索を利用した改変画像検出では、画像の公開前に、画像に対して特別な操作をする必要がない。よって、本論文においても、これらの文献と同様に、元画像のデータベース(元画像 DB)を手元に持っているとして、類似画像検索を用いた改変画像の検出を目指す。

類似画像検索を用いた改変画像の検出においては、部分画像に注目する手法が有効であると考えられる[4]。これは、改変によって画像全体としての情報は変化してしまうが、部分的には不変である情報が存在するためである。部分画像を用いた検索のためには、一枚の画像を領域分割し、それぞれの領域ごとにマッチングを取る必要がある。しかし、この手法では、一枚の元画像に対して複数の部分画像を元画像 DB に登録しなければ

ばならず、元画像 DB が大きくなる。その結果、検索時間が長くなる。このため、従来の研究では、多くても約 10 万枚の画像を対象に検索を行っており [5]、Web のように大量の画像を対象として検索を行う場合、実用時間で改変画像の検出を実現することは困難になる。

以上のような問題に対し、本研究では、元画像を分割せずに元画像 DB に登録し、以下に述べる二段階検索を用いて、元画像 DB に大量画像が登録されている場合の、実用時間での改変画像検出を実現する。なお、提案するシステムでは、改変されていると疑われる画像を入力として用いる。システムは、入力画像が改変画像であるなら、処理結果として改変前の元画像を出力する。

二段階検索の第一段階では、画像全体の特徴量を使い、SR-Tree[6]を用いたインデクシングを利用して、大量画像から元画像候補の絞込みを行う。言い換えれば、大量の元画像 DB から、粗い検索で高速に元画像候補を取得するということである。第二段階では、高速なテンプレートマッチング手法であるアクティブ探索[7]を用いて、元画像候補から元画像の特定を行う。これは、少数の元画像候補を詳細に検索するということである。その際、システムへの入力画像を分割して、アクティブ探索に対するクエリとすることで、加筆や切り取りなどの改変に頑健な探索を行うことができる。

提案手法の評価にあたっては、実際に元画像 DB の大規模化に適応可能であることを確かめるため、Web 上から約 48 万枚の画像を収集して、元画像 DB を構築し、実験を行う。前述したように、48 万枚もの画像からの類似画像検索の実験は過去に行われていない。これまでにない規模の大量画像から類似画像検索を行うという点においても、本論文は大きな意味を持つといえる。

本システムを用いることで、複数の改変及び、それらを組み合わせた改変を施された画像を検出することができる。実用上の利用形態としては、大量の画像情報を持つ企業や写真家が、Web 上の無断改変画像を検出するために利用する形が考えられる。クローラーに Web を巡回させ、画像を自動で次々に入力していき、改変と思しき画像を発見したら、ユーザに警告を出して確認してもらい、といった方法である。他にもサービスとしては様々なものが考えられるが、詳細な議論についてはまたの機会に譲る。

本稿の構成は次のとおりである。2 節では、関連する研究及び要素技術について述べる。3 節では本研究で扱う改変の定義について述べる。4 節では類似画像検索とアクティブ探索を用いた改変画像検索システムを提案する。5 節にシステムの実験とその結果を示す。最後に 6 節で全体をまとめる。

## 2. 関連研究

本節では、提案手法に関連する研究について述べる。2.1 では、従来の類似画像検索を用いた改変画像検出手法について、概要と問題点を述べる。2.2 では、提案手法で使用した、多次元データを高速に検索できるインデックス手法である SR-Tree[6]について述べる。2.3 では、提案手法で使用した、高速なパターンマッチング手法であるアクティブ探索[7]について述べる。

### 2.1. 類似画像検索を用いた改変画像検出

ここでは、従来研究されてきた、類似画像検索を用いた改変画像検出手法[2-4]について述べる。これらのシステムでは、ユーザが、改変が施されたと思われる画像

を入力し、システムは、入力画像の改変元の画像が元画像 DB 中に存在するかどうかを検索する。元画像 DB 中に改変元と思われる画像があれば、その画像を結果として出力し、改変元と思われる画像がなければ、入力画像は元画像 DB 中の画像を改変した画像ではないとして、処理結果として画像を出力しない。

これらのシステムでは、電子透かしと異なり、公開前に画像に情報を埋め込む必要がなく、どんな画像でも入力画像として利用することができる。画像の特徴量としては様々なものが考えられ[8]、特徴量を適切に選択することで、検出の精度を上げることができる。

#### 2.1.1. 局所領域に注目した手法[2][3]

文献[2][3]のシステムでは、画像の特徴的な点に注目した類似画像検索を用いて、改変画像の検出を行っている。[2]ではまず、Harris detector を用いて画像中の特徴点を探す。次に、それぞれの特徴点の周囲にガウシアンフィルターをかけ、結果の値を特徴量として用いている。実験では、1 枚の画像に対して、200 個ほどの特徴点を抽出し、約 1 万枚の画像から元画像 DB を構成している。そのため、対象画像約 1 万枚に対して、元画像 DB 中の特徴点の数は約 260 万個になっている。また、[3]では、スケール変化や回転に頑健な特徴点を利用しており、画像一枚あたりの特徴点の数は、平均して約 1100 個にもなっている。

[2]における実験結果では、明るさ変更、切り取り、ぼかしなどの改変を高い精度で検出できている。しかし、元画像 DB 中に含まれる特徴点の数が画像枚数と比べて多く、また、画像同士の類似度を求める処理に時間がかかる(入力画像の特徴点の数だけ k-近傍探索を行わなければいけない)ために、近似を利用した高速化を用いても、画像を入力してから出力を得るまでに、15 秒あまりの応答時間がかかっている(CPU:Pentium 4, 2.4GHz,RAM: 1GB)。更に、[3]のシステムでは、より高い検出率、精度が得られているが、画像一枚あたりの特徴点の数が多いために、200 枚の元画像 DB からの検索に 1 秒近くの時間がかかっている(CPU: Pentium 4, 3GHz, RAM: 1.00GB)。よって、これらのシステムを、このまま大量の画像に対して適用することは困難である。

#### 2.1.2. オブジェクト画像に注目した手法[4]

画像をオブジェクト領域に分割し類似画像検索を行うシステムとして、ExSight[9]がある。画像の分割は、一意に行うことが困難な問題であるため、ExSight では、画像内の領域の重なりを許容し、冗長性を持たせた分割を行う。そして、元画像 DB には、分割によって得られた、全てのオブジェクト画像を登録する。[9]の実験では、この画像分割方法を用いて元画像 DB を構築した結果、1000 枚の画像から、10 万近くのオブジェクト画像が得られ、元画像 DB に登録された。更に[4]では、ExSight を利用して、実際の画像流通市場における著作権侵害画像の検出実験を行っている。実験の結果、改変画像を一定の検出率、適合率で検出できているが、元画像 DB に含まれる画像が約 500 点と少ないので、大量画像に適用できるかどうかは未知である。ただし、実際の市場における実験を行ったことで、想定していなかった改変が出現したり、特徴量の重み付けに対する重要な知見が得られたりしているため、[4]の実験が行われた意義は大きい。

## 2.2. SR-Tree [6]

次に本論文で要素技術として用いている SR-Tree について説明する。

本論文では大量の画像を扱うことを主要な目的としている。そのため、大量の元画像 DB 中から、高速に検索を行うためのインデクシングが不可欠である。高速な検索のためのインデックス方法は数多く研究されているが[6][10-12]、本研究においては、近似を用いずに高い速度性能を発揮する SR-Tree[6]を用いた。

SR-Tree は、[10]の SS-Tree で用いられている包囲球と、[11]の R\*-Tree で用いられている包囲長方形を併用するインデックス手法である。包囲球と包囲長方形の共通部分によって領域を区分し、tree を構築する。対象データは高次元の点データである。

SS-Tree で利用している包囲球のみの区分では、領域の半径が大きくなったときに体積が必要以上に大きくなってしまい、領域の重なりが大きくなる可能性が高くなる。領域の重なりは検索時間の増大につながるため、望ましくない。また、R\*-Tree で利用している包囲長方形による区分では、次元が高くなるにつれて、一辺の長さと同角線の長さの差が大きくなるため、距離の近い点だけを集めることが難しくなる。これらの SS-Tree と R\*-Tree の相補的な特徴を、二つを組み合わせることで解決した手法が SR-Tree である。[6]では、実際の高次元データを用いた実験によって、SR-Tree が、SS-Tree や R\*-Tree よりも、検索にかかる CPU 時間やディスクアクセスを減らせることを示している。

なお、SR-Tree は作者によってライブラリが公開されており[13]、今回の実験でも[13]を使用している。

## 2.3. アクティブ探索[7]

アクティブ探索は、本論文で提案する二段階検索の第二段階目で用いる要素技術である。第二段階では、SR-Tree による検索結果の画像に対してテンプレートマッチングを行う。テンプレートマッチングは、クエリ画像のサイズを変えながら、探索対象画像を網羅的に走査する、計算量の大きな処理である。そのため、単純に利用すると探索時間が長くなってしまふ。この問題を解決するために、アクティブ探索を用いている。アクティブ探索は、類似度計算の枝刈りを行い、無駄な計算を省くことで、テンプレートマッチングを高速化する手法である。枝刈りは、あるピクセルにおけるクエリ画像と探索対象画像の類似度を元に、その周囲の領域における類似度の上限値を求めることで行う。以下、アクティブ探索の詳細を述べる。

アクティブ探索では、正規化された色ヒストグラムを特徴量として用いる。クエリ画像の正規化色ヒストグラムを  $M$ 、探索対象画像中の、クエリ画像と同じ大きさのある領域における正規化色ヒストグラムを  $H$  とする。二つの領域の類似度は、ヒストグラムインタセクション  $S_{HM}$  で表され、

$$S_{HM} = \sum_{i=1}^I \min(H_i, M_i) \quad (1)$$

で得られる。 $I$  はヒストグラムのビン数である。3チャンネルで表現される画像において、色空間の各軸を  $Q$  分割した場合には、 $I = Q^3$  となる。

ここで、探索対象画像中の任意の領域  $A, B$  に対して、

クエリ画像  $M$  との類似度  $S_{AM}, S_{BM}$  を求めることを考える。 $A \cap B$  を  $A$  と  $B$  の共通部分、 $A - B$  を  $A$  の領域から  $B$  の領域を除いた部分、 $|A|$  を  $A$  の画素数としたとき、 $|A| \leq |B|$  ならば、次の関係が成り立つ。

$$S_{AM} \leq \frac{\min(S_{BM} |B|, |B \cap A|) + |A - B|}{|A|} \quad (2)$$

これは、 $S_{BM}$  が計算されている場合には、その値を利用して  $S_{AM}$  の上限値を求められるということを表している。つまり、ある領域  $B$  での類似度が計算されれば、 $B$  における類似度を利用して、 $B$  周辺の領域  $A$  における類似度の上限値が計算できるということである。 $A$  の上限値が、既に探索を終えた領域における類似度の最大値より小さければ、 $A$  の類似度計算は省略できる。文献[7]では、この上限値を利用した類似度計算の枝刈りによって、総当り法と同じ精度を保ちつつ、計算時間を 10~1000 倍短縮できることを示している。

## 3. 改変画像

画像の改変には様々な手法が考えられ、程度も軽くいじる程度のものから、元画像と同じだとは思えない程に変えてしまう場合まである。本節では、提案システムにおいて検出の対象とする改変画像を定義する。

以下に、考えられる画像の改変を記す。これらのうち、提案システムでは、1~9 の改変を検出対象とする。10. の改変については本研究では対象としない。10. を検出の対象としない理由としては、

- ・改変の前後で変化が小さい特徴量の抽出が難しい
- ・その他には 1~9 以外の全ての改変が含まれるため、種類が多い

などが挙げられる。今後、10. の改変も検出できるよう、特徴量の選択、改変の区分などを見直す必要がある。

### ・改変の種類

1. 回転、反転
2. 拡大縮小
3. ぼかし
4. ノイズ
5. 尖鋭化
6. 加筆
7. トリミング(切り取り)
8. 明るさ変更
9. 色変更
10. その他(歪み、モザイク等)

1~9 の改変を施した画像は、改変の前後で変化の小さい特徴量を利用することで検出できる。以下に、1~9 の改変を施す前後で変化が小さい特徴量を示す。

### ・色ヒストグラムの変化が小さい改変

改変 1~7 は、改変前後で色ヒストグラムの変化が小さいため、色ヒストグラムを用いることで、改変画像を検出できる。1,2 においては、改変後の各ピクセルの色を、周囲のピクセルの色を利用して求める。そのため、改変の前後で、完全には色ヒストグラムが一致しないが、ほぼ変化がない。3,4,5 についても、各ピクセルの色が大きく変化しないため、色ヒストグラムの変化は小さい。また、6,7 の前後で色情報の変化があるのは、加筆、切り取りをされた部分のみである。それ以外の部分については、色情報は不変であるので、加筆、切り取り部分が画像全体に比べて小さければ、画像全体としての色ヒストグラムの変化は小さい。

・エッジ方向情報の変化が小さい改変

改変 2,6~9 は,改変の前後でエッジ方向情報の変化が小さいため,エッジ方向情報を用いることで,改変画像を検出できる.2,8,9 においては,エッジ方向情報には変化を加えないため,改変の前後で,エッジ方向情報は不変である.6,7 においては,色情報の場合と同様に,改変を加えた部分のみ,エッジ方向情報が変化する.それ以外の部分についてはエッジ方向情報が不変であるので,加筆,切り取り部分が画像全体に比べて小さければ,画像全体としてのエッジ方向情報の変化は小さい.

以上より,本システムでは 1~9 の改変を検出対象とする.それぞれの改変を検出するのに有効であると考えられる特徴量の組み合わせを表 1 に示す.なお,どちらの特徴量を用いても検出できる改変は,両方を組み合わせた特徴量を用いることで,より頑健な検出を実現できる.10.に分類されるその他の改変については,今回のシステムでは対象とせず,今後の課題とする.

表 1. 改変の種類と有効な特徴量の組み合わせ

改変	色ヒストグラム	エッジ方向情報
1.回転,反転	○	×
2.拡大縮小	○	○
3.ぼかし	○	×
4.ノイズ	○	×
5.尖鋭化	○	×
6.加筆	○	○
7.トリミング	○	○
8.明るさ変更	×	○
9.色変更	×	○

4. 提案手法

4.1. 提案手法概要

提案手法では,画像を入力として,大量の元画像 DB から類似画像検索を行うことで,入力画像が改変画像であるかどうかを判断する.システムによる類似画像検索の結果得られた画像が,入力画像の元画像であれば,入力画像は改変画像であり,改変画像の検出に成功したことになる.逆に,入力画像の元画像以外の画像が検索された場合は誤検出となる.また,検索結果が 0 件だった場合,つまり入力画像がリジェクトされた場合には,入力画像は元画像 DB 中の画像を改変した画像ではないということになる.

大量の画像を扱うために,第一段階では SR-Tree によるインデックスを利用し,画像全体の特徴量を用いた粗い検索を高速に行う.この時点で,元画像候補となる画像群を複数取得する.第二段階では,それらの元画像候補に対して,アクティブ探索を利用して,分割した入力画像をクエリとした細かい検索を行う.この二段階検索によって,元画像 DB が大きな場合にも,実用時間で,入力画像が元画像 DB 中の画像かどうか,またはその改変画像かどうかを判断することができる.概要を図 1 に示す.

なお,本システムでは,画像の特徴量として,正規化色ヒストグラム及び正規化エッジ方向スペクトラムを用いる.特徴量としてこの二つを用いたのは,

1. 抽出処理が自動で高速に行える
2. 類似画像検索において広く用いられている
3. 画像の多少の変化に頑健である

という理由からである.具体的には,Lab の各軸を 16 次

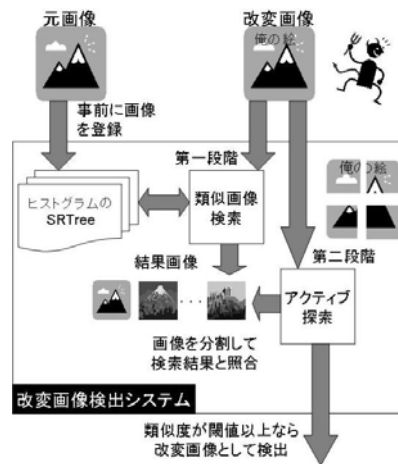


図 1. 提案システム概要

元に量子化したものと,エッジの方向を 18 次元に量子化したものを用いた.

ただし,3 節で述べたように,検出対象の改変には,色ヒストグラムを用いることで検出できる改変,エッジ方向スペクトラムを用いることで検出できる改変,両方を組み合わせた特徴量を用いることで検出できる改変がある.つまり,色ヒストグラム,エッジ方向スペクトラムのどちらか片方,もしくは組み合わせ特徴量のうち,どれか一つのみを用いた検索では,全ての改変を検出することはできない.よって,本システムにおいては,(a)色ヒストグラムを用いた検索,(b)エッジ方向スペクトラムを用いた検索,(c)組み合わせ特徴量を用いた検索の 3 種類の検索を,それぞれ独立に行う.これにより,より多くの種類の改変を検出できるようになると考えられる.各検索における特徴量の次元は,(a)の検索が  $16+16+16=48$  次元,(b)の検索が 18 次元,(c)の検索が  $48+18=66$  次元である.

なお,2.3.ではアクティブ探索の特徴量として正規化色ヒストグラムを用いると述べたが,特徴量として正規化エッジ方向スペクトラムや組み合わせ特徴量を用いても,色ヒストグラムを用いた場合と同様にアクティブ探索を行うことができる.

4.2. SR-Tree による元画像候補の絞込み

提案システムの第一段階では,画像全体の特徴量を利用した元画像候補の絞込みを行う.第一段階におけるクエリは,入力画像そのものである.絞り込み検索の対象は,元画像 DB に含まれる全画像であり,対象画像の数は膨大である.よって,高速な検索を実行できる SR-Tree によるインデックスを利用する.

ここで,第一段階の検索によって元画像候補を何枚取得するかという問題があるが,これに対しては以下の 2 つの方法が考えられる.

1. あらかじめ取得する候補画像枚数を決めておく.
2. ある閾値以上の類似度を持つ画像全てを候補画像として取得する.

本研究では,簡単のために 1 の方法を用いた.しかし,2 の方法において,適切な閾値の決定法を利用することで,入力画像毎に適応枚数の元画像候補を取得することができる.元画像候補を必要十分な枚数だけ取得できれば,SR-Tree による検索及び第二段階のアクティブ探索において,無駄な検索をしないで済むようになるため,検出精度の向上,検索時間の短縮が見込まれる.

### 4.3. アクティブ探索による元画像決定

第二段階では、探索対象画像を、第一段階検索によって得られた元画像候補としたアクティブ探索を行う。元画像候補のうち、この第二段階検索における入力画像との類似度が一定値を超えた元画像候補を、入力画像の元画像であると判断する。

3章で述べたように、加筆、切り取りの改変においては、部分的な特徴量のみが変化する。これらの改変を検出するためには、クエリ画像、元画像を分割し、それぞれの分割画像同士で検索を行う手法が有効である。[4]では画像をオブジェクト単位に分割しているが、本手法ではグリッド分割を行う。これは、オブジェクト単位で分割を行うと、唐草模様のような、オブジェクトが写っていない画像に対しては、分割がうまくいかないためである。また、グリッド分割は、オブジェクト分割に比べて、高速、簡単に実行できるという利点もある。具体的には、入力画像を縦横 1/2 の長さになるように、左上、右上、左下、右下、中央の五つの部分画像に分割した(図 2)。

本手法では、元画像候補を探索対象画像としたアクティブ探索のクエリ画像として、入力画像を分割し、得られた分割画像及び入力画像そのものを用いる。アクティブ探索を行うことで、分割画像中にオブジェクトの一部が含まれている場合でも、元画像中の分割オブジェクトを検出することができる(図 3)。



図 2. グリッド分割の例(画像は著者撮影)

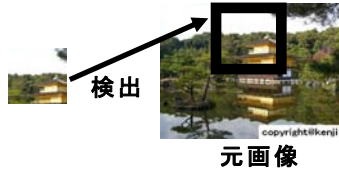


図 3. 分割オブジェクト検出の例

アクティブ探索は、元々テンプレートマッチングの手法であるため、その出力は「探索対象画像中、クエリ画像はどこにどの大きさで存在するか」という情報である。しかし、本手法においては、正確なクエリ画像の位置を知る必要はなく、「対象画像中に、クエリ画像に似た部分があるかどうか、また、その部分の類似度はいくらか」という情報のみで十分である。よって、本手法においては、厳密なアクティブ探索ではなく、高速化のための枝刈り手法のみを利用することにする。

つまり、各分割画像及び入力画像そのものをクエリとしたアクティブ探索の出力値は、クエリ画像と、探索対象画像中の最も類似している領域との類似度となる。これを  $S_{ij}$  ( $i=1..I, j=1..J$ ) とする。 $S_{ij}$  は、分割した入力画像群中  $i$  番目の画像をクエリとし、 $j$  番目の元画像候補を探索対象画像としてアクティブ探索を行った際の出力値である。ただし、 $I$  は(画像の分割数+1)、 $J$  は第一段階目の検索によって得られた元画像候補の枚数である。

更に、 $j$  番目の元画像候補と入力画像そのものとの類似度を  $S_j$  とし、

$$S_j = \sum_{i=1}^I w_j f(S_{ij}) \quad \left( f(x) = \begin{cases} 0(x < a) \\ x(x \geq a) \end{cases} \right) \quad (3)$$

で与える。 $a$  はパラメータとして与える閾値である。 $w_j$  は入力画像の分割位置(図 2 の中央や左上など)による重みである。式(3)は、元画像候補  $j$  と入力画像との類似度  $S_j$  を、 $j$  と入力画像の部分画像  $i$  との類似度  $S_{ij}$  の重み付きの総和とする、ということを表す。ただし、 $j$  と  $i$  との類似度が閾値  $a$  を超えない場合は、 $j$  は部分画像として  $i$  を含まないとして、全体の類似度  $S_j$  に加算しない。 $S_j$  が閾値  $a_s$  を超えれば、入力画像は、元画像候補  $j$  の改変画像であるとする。 $a_s$  も、 $a$  同様パラメータとして与える。すべての元画像候補に対して  $S_j$  を求めた後、 $a_s$  を超える  $S_j$  を持つ元画像候補が一枚もない場合は、入力画像は元画像 DB 中の画像の改変画像ではないと判断する。更に、 $S_{ij}$  が、類似度の最大値である 1.0 に近い閾値  $a+(>a)$  を超えた場合には、元画像候補  $j$  中に、分割した入力画像  $i$  と完全一致する部分があると考えられる。この場合には、 $S_j$  が  $a_s$  を超えなくても、 $j$  は  $i$  の元画像であるとする。

なお、実装においては、高速化のために、 $S_j$  が  $a_s$  を越えた時点、または、越えることがないと判断した時点で探索を打ち切っている。つまり、各元画像候補において、1 から  $I$  までの全ての  $i$  について  $S_{ij}$  を求めるとは限らず、必要な  $i$  についてのみ、 $S_{ij}$  を計算する。

また、テンプレートマッチングにおいて、探索時間短縮のために類似度を計算するピクセルを減らすには、粗い探索を行う手法が有効である。事前実験を行い、 $x$  軸、 $y$  軸、それぞれ 10 ピクセルに 1 ピクセルのみ、類似度または上限値を計算する粗いアクティブ探索を行ったところ、探索精度に大幅な劣化は見られなかった。よって、本手法においても、高速化のために、10 ピクセルに 1 ピクセルのみ計算を行う粗い探索を行う。同様に、類似度計算対象のピクセルを減らす目的のため、画像サイズの正規化を行った。具体的には、探索の前に、探索対象画像のサイズを、縦横どちらか長いほうのサイズが最大 500 ピクセルとなるように縮小した。

### 5. 実験

本節では、提案手法が、大量の画像から実用時間で改変画像を検出できることを確認するために行った 2 つの実験(実験 1,2 とする)について述べる。実験 1,2 の概要はそれぞれ以下の通りである。

- ・ **実験 1** : 画像を入力してから改変画像かどうかを判断するまでの時間の測定
- ・ **実験 2** : 改変画像の検出率および誤検出率の測定

実験 1 及び 2 では Web 上から収集した約 48 万枚の jpeg 画像から特徴量を収集し、元画像 DB とした。画像の URL リストとして、e-Society「基盤ソフトウェアの総合開発」プロジェクト内のサブプロジェクトである、「インターネット上の知識集約を可能にするプラットフォーム構築技術」[14]によって収集された全世界の

150 億 URL のうち,日本国内の URL の一部を用いた. 実験環境は CPU: Pentium 4 3.2GHz, RAM: 2GB である. 以下,5.1.では実験 1 及び 2 で用いた改変画像について,5.2.では実験 1 について,5.3.では実験 2 について述べる.5.4.では実験結果の考察を行う.

なお,本実験においては,第一段階の結果得られる元画像候補の枚数を,4.1.で述べた(a)の検索,(b)の検索,(c)の検索のそれぞれについて,入力画像から最も近い 3 件とした.つまり,第一段階の検索の結果として,合計 9 件の元画像候補を得る.そして,得られた 9 件の元画像候補に対して,(a),(b),(c)それぞれの特徴量を用いた第二段階の検索を行うことになる.

また,パラメータとして,エッジ方向スペクトラムを用いた検索における  $a=0.9822$ ,色ヒストグラムおよび両方を組み合わせた特徴量を用いた検索における  $a=0.948$ ,全検索において共通で  $a_S=5$ ,  $a+=0.987$  を手動で設定した.類似度の重み  $w_j$  は,図 2 の中央部分の分割画像について 2.0,分割しない入力画像全体について 1.4,その他の分割画像については 1.0 とした.

### 5.1. 実験用に用意した改変画像

本実験では,画像の改変に,フリーの画像加工ツールである ImageMagick[15]を用いた.ImageMagick では多種多様な改変が可能であるが,本実験では文献[3]を参考にすると共に,Web 上で行われることが多い文字挿入などの改変も含めた 39 種類の改変を施した画像を改変画像として用いた.以下に詳細を示す.改変を施す前の元画像として,元画像 DB から無作為に 100 枚を選んだ.実験用画像は,元画像 100 枚と,元画像に改変を施した画像が 39 種類についてそれぞれ 100 枚,元画像 DB 外の画像 100 枚の,合計 4100 枚である.図 4 に改変画像の例を示す.元画像 DB 外の画像については,元画像 DB に含まない画像を,URL リストから無作為に 100 枚用意した.

1. 縮小:縦横の長さを(a)90%, (b)80%, (c)70%, (d)60%, (e)50%, (f)30%, (g)10%に縮小
2. 拡大:縦横の長さを 200%に拡大
3. 加筆:右下に,山名研究室のロゴ(図 5)を,元画像の縦横の長さの(a)1/6, (b)1/4 の長さで貼り付け
4. 文字挿入:横の長さの 1/10 のフォントで,画像の中央に 45° 回転させた「sample」の黒文字を挿入
5. トリミング:外周を,縦横の長さが(a)95%(面積 90%), (b)90%(同 81%), (c)80%(同 64%), (d)70%(同 50%)になるよう削減
6. 明るさ変更:明るさを(a)10%増加, (b)20%増加, (c)10%減少, (d)20%減少
7. 彩度変更:彩度を(a)10%増加, (b)20%増加, (c)10%減少, (d)20%減少, (e)30%減少
8. 回転:(a)90°, (b)180°, (c)270° 回転
9. 色調変更:(a)赤, (b)緑, (c)青をそれぞれ 10%強調
10. コントラスト:(a)強める, (b)弱める
11. 枠付け:縦横それぞれ 5%の太さの枠を付ける
12. 縮小+明るさ変更:1(e)と 6(a)
13. 縮小+加筆:1(e)と 3(b)
14. 縮小+トリミング:1(e)と 5(a)
15. 明るさ変更+トリミング:6(a)と 5(a)
16. トリミング+加筆:5(a)と 3(b)
17. 明るさ変更+加筆:6(a)と 3(b)

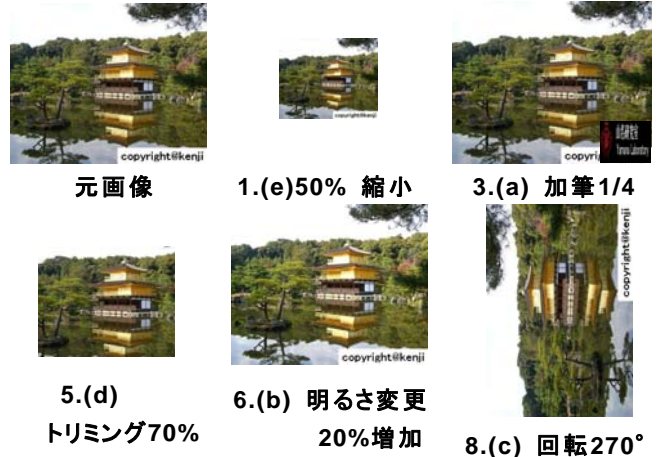


図 4. 改変画像の例



図 5. 山名研究室のロゴ

表 2. 実験 1 の結果

	第一段階(s)	第二段階(s)	合計(s)
元画像	1.06	3.54	4.6
改変画像	1.15	4.99	6.14
DB 外画像	1.41	4.36	5.77

### 5.2. 実験 1: 検出時間の測定

実験 1 では,システムに画像を入力してから改変画像かどうかを判断するまでの時間について検証を行う. テスト用の入力画像として,5.1.で述べた画像 4100 枚を用いた.この 4100 枚の画像を,48 万枚の元画像 DB を持つシステムに入力し,第一段階の SR-Tree による類似画像検索の時間及び,第二段階のアクティブ探索による改変画像検出の時間を,元画像,改変画像,元画像 DB 外画像それぞれについて測定した.表 2 に入力画像一枚辺りの平均測定結果をまとめる.なお,SR-Tree によるインデックスのサイズは,三つの特徴量のインデックスの合計で 905MB であった.また,実験はインデックスがすべてメモリに乗っている状態で行った.

表 2 より,第一段階,第二段階ともに,元画像を入力した場合に,最も高速に検索結果が得られることがわかる.この理由は以下のとおりであると考えられる.

まず,第一段階の検索において,元画像を入力した場合,入力画像と同一の画像が元画像 DB 内に存在する.そのため,検索結果を得るまでにたどる枝が少なく,検索時間が短くなる.改変画像を入力した場合には,DB 内に,入力画像と似てはいるが同じではない画像しか存在しないため,たどる枝が少し多くなる.さらに,DB 外画像を入力した場合には,DB 内に入力画像と似た画像が存在する可能性が低くなるため,たどる枝の数はより多くなり,最も長い検索時間がかかる.

次に,第二段階において,元画像を入力した場合には,元画像候補中に入力画像と同一の画像が入っていることが見込まれる.そのため,4.3 で述べたアクティブ探索の打ち切りが早々に行われ,短時間で結果を得ることができる.また,改変画像を入力した場合よりも,DB 外画像を入力した場合のほうが,短い検索時

間が得られている。これは、改変画像は入力画像と同一ではないが、ある程度は似ているために、アクティブ探索を早々に打ち切ることが難しいためである。逆に DB 外画像を入力とした場合には、元画像候補に、入力画像と似ている画像が存在する可能性が少ないため、早々に打ち切りが行われる。

また、元画像を入力した場合には、 $S_{ij}$  が  $a+$  を超えた場合に、最短で一回のアクティブ探索のみを行えば元画像であると判断できる。それに対し、DB 外画像を入力とした場合では、 $S_j$  が  $a_5$  を超えないと判断するまでに複数回の探索を行う必要があるため、元画像を入力とした場合よりは長い探索時間がかかっている。

しかし、いずれの場合においても、48 万枚程度の元画像 DB からの検索は、7 秒以内で実行できている。文献 [2] では、本実験より多少性能の劣るマシンを用いているが、1 万枚程度の元画像 DB を利用した改変画像の検出に約 15 秒かかっている。提案システムでは、[2] と比べて高速な検索を実現した。

### 5.3. 実験 2: 改変画像の検出率と誤検出率の測定

実験 2 では、48 万枚の元画像 DB を持つシステムに、5.1 で述べた 4100 枚の画像を入力し、検出率、第一段階の検索結果である元画像候補に元画像が入っている割合(第一段階含有率)、元画像候補に元画像が入っているとき、第二段階の検索で元画像を検出できた割合(第二段階検出率)、誤検出率をそれぞれ測定した。ここで検出率とは、検索結果に入力画像の元画像が入っていた割合である。検索結果は得られたが、結果画像中に入力画像の元画像が含まれていない場合は誤検出となる。また、元画像 DB 外の画像を入力して、検索結果が得られてしまった場合も誤検出となる。誤検出率は、これらの誤検出が起こる割合である。

表 3 に元画像、改変画像、元画像 DB に含まれていない画像それぞれについて、測定結果を示す。

表 3 より、39 種類の改変の平均で、検出率 81.4% が得られたことがわかる。よって、システムが多様な改変にある程度対応可能であることが確かめられた。しかし、いくつかの改変においては、検出率が極端に低い。表 4 に、それぞれの理由と解決法をまとめる。

一方、DB 外の画像についても、24% が改変画像であるとして検出されてしまっている。この誤検出は、今後減らしていく必要がある。誤検出は、白黒画像のように色の種類が少ない画像(白黒様画像と呼ぶことにする)が入力画像である場合に多く起きている。これは、白黒様画像の Lab 色ヒストグラムにおいては、ab のチャンネルで他の白黒様画像の色ヒストグラムとの違いがあまりないためである。そこで、入力画像が白黒様画像の場合には、エッジ方向スペクトラムのみを用いた検索を行えば、誤検出率を減らすことができる。白黒様画像であるかどうかは、色ヒストグラムにおける各ビンの値の分散を計算することで判断できる。白黒様画像では、Lab 中の ab チャンネルにおいて、値が 0 になるビンが多くなり、分散が大きくなるためである。

### 5.4. 考察

提案システムにおいては、48 万枚の DB からの検出を 7 秒以内という短時間で実現しているため、従来研究に対して優位性を持つ。

検出性能については、提案システムでは改変画像全体に対して平均で約 80% の検出率が得られている。そ

れに対して文献 [3] では、同様の改変に対してほぼ 100% の検出率を達成している。単純な検出率の比較は、以下の表 5 の理由によって大きな意味を持たない

表 3. 実験 2 の結果(単位は全て%)

		検出率	第一段階含有率	第二段階検出率	誤検出率	
元画像		100	100	100	0	
改変画像	1. 縮小	a.90	94	97	96.9	2
		b.80	91	94	96.8	5
		c.70	90	94	95.7	4
		d.60	88	92	95.7	6
		e.50	84	91	92.3	8
		f.30	74	86	86.0	9
		g.10	29	60	48.3	14
	2. 拡大		94	97	96.9	3
	3. 加筆	a.1/6	99	99	100	1
		b.1/4	92	95	96.8	5
	4. 文字		99	100	99.0	0
	5. トリミング	a.95	97	98	99.0	2
		b.90	95	98	96.9	3
		c.80	79	83	95.2	7
		d.70	50	53	94.3	16
	6. 明るさ	a.10	92	98	93.9	1
		b.20	66	87	75.9	12
c.-10		91	94	96.8	3	
d.-20		71	86	82.6	3	
7. 彩度	a.10	96	100	96.0	0	
	b.20	90	97	92.8	2	
	c.-10	97	100	97.0	1	
	d.-20	93	98	94.9	0	
	e.-30	90	94	95.7	0	
8. 回転	a.90	98	100	98.0	0	
	b.180	100	100	100	0	
	c.270	100	100	100	0	
9. 色調	a.赤	90	92	97.8	2	
	b.緑	87	92	94.6	2	
	c.青	93	93	100	0	
10. コントラスト	a. 強め	90	94	95.7	2	
	b. 弱め	94	98	95.9	0	
11. 枠付け		16	50	32.0	13	
12. 1(e)+6(a)		43	73	58.9	16	
13. 1(e)+3(b)		74	89	83.1	8	
14. 1(e)+5(a)		77	84	91.7	11	
15. 6(a)+5(a)		67	85	78.8	8	
16. 5(a)+3(b)		76	81	93.8	9	
17. 6(a)+3(b)		28	58	48.3	14	
改変画像平均		81.4	89.2	91.2	4.9	
DB外画像		0	0	0	24	

表 4. 検出率の低い改変とその解決法

改変	理由	解決法
1.(g) 縮小 10%	縮小の度合いが強いため、ピクセル数が少なくなり、色やエッジ情報が失われる。	DBにマルチスケールの特徴量を登録
5.(d) トリミング 70%	画像全体の情報が大幅に欠損するため、第一段階の検索で取得できない。	部分画像をDBに登録
6.(b) 明るさ 20%増加	強い明るさ変更によりエッジが消失する。	Labの内、明るさを表すLチャンネルを無視
11. 枠付け	5.(d)同様の理由で第一段階の検索により取得できない。また、第二段階の検索で分割された入力画像中、4枚に枠が入ってしまい検出できない。	グリッド分割以外の分割を使用
12. 縮小+ 明るさ	組み合わせにより、色、エッジ、両方を組み合わせ特徴量のすべてが変化。	Labの内、明るさを表すLチャンネルを無視
17. 明るさ+ 加筆	6.(b)の効果に加えて、四角い画像の加筆により、縦横のエッジが強く検出される。また、加筆により色も変化。	Labの内、明るさを表すLチャンネルを無視

表 5 検出率が低い理由

1.	[3]では12000枚が登録された、本研究とは異なるDBからの検索を行っている。
2.	DBが大きくなればなるほど、元画像以外にも入力画像に似た画像が存在する可能性が高くなるため、検出率は低くなる。

とはいえ、この事情を差し引いても、提案システムの検出率のほうが優れているとは言いがたい。今後、異なる特徴量や画像分割手法の利用、第一段階の検索の結果として得る元画像候補枚数などを検討することで、さらに検出率を向上していく必要がある。

今回用いた特徴量以外の特徴量としては、[9]に述べられているように、オブジェクト分割時の形状や位置情報などがある。テクスチャ情報としては、粗さや明瞭さが利用できる。Lab以外の異なる表色系を利用して実験をすることも有効である。これらの特徴量を利用することで、今回の実験に用いた改変画像以外にも、より多くの、より厳しい改変を検出できるようにすることを目指していく。

また現在は、 $a$  や  $a+$ 、 $a_s$  などの閾値を、パラメータとして手動で設定している。さらに、テストセットごとに適切な閾値が異なる。一般的な画像データに対して利用できるようにするためには、最も適切な閾値を発見するためにより多くの実験を行うか、もしくは自動で適切な閾値を設定する手法を利用する必要がある。

## 6. おわりに

本稿では、改変画像検出のための、大量画像からの類似画像検索手法について述べた。大量の画像を扱うために、SR-Tree とアクティブ探索を用いた二段階検索を提案し、48 万枚以上の元画像 DB を利用した実験において、39 種類の改変画像の平均で検出率 81.4%の結果を 7 秒以内に得られることを確認した。しかし、誤検出率が 24%となってしまったことや、検出率が大きく低

下してしまう改変があるなど、改善しなければならない点も見つかった。これらの点に対しては、  
① Lab やエッジ方向以外の特徴量を利用  
② 第二段階においてグリッド分割以外の分割を利用などの方法を用いて、検出率及び誤検出率の改善を図っていく。また、DB に登録する画像数をさらに増やし (~1000 万枚レベル)、その際に生じる問題を明らかにしていくことで、より大規模な画像 DB を扱えるようになることを目指していく。

## 謝辞

本論文を執筆するに当たり、画像の URL リストを提供していただいた同研究室、田代君に感謝します。

## 文 献

- [1]画像電子学会編, “電子透かし技術—デジタルコンテンツのセキュリティー,” 東京電機大学出版局, 220pp (2004).
- [2]S. Berrani, L. Amsaleg, and P. Gros, “Robust content-based image searches for copyright protection,” In Proc. of ACM Workshop on Multimedia Database, pp.70-77 (2003).
- [3]Y. Ke, R. Sukthankar and L. Huston, “Efficient near-duplicate detection and sub-image retrieval,” In ACM Conf. on MultiMedia, pp.869-876(2004).
- [4]吉田忠城, 赤間浩樹, 山室雅司, 串間和彦, 榎谷浩章, 辻敏弘, 原田健次, “類似画像検索方式の改変画像検出への適用—デジタルコンテンツ保護・流通システムへの適用と評価,” 情処学論, Vol.42, No.1, pp.171-181 (2001).
- [5]H. Mueller, S. Marchand-Maillet, and T. Pun, “The truth about corel – evaluation in image retrieval,” In Proc. of The Challenge of Image and Video Retrieval (CIVR), pp.38-49 (2002).
- [6]片山紀生, 佐藤真一, “SR-Tree:高次元点データに対する最近傍検索のためのインデックス構造の提案,” 信学論, Vol.J80-D- I , No.8, pp.703-717 (1997).
- [7]村瀬洋, V. V. Vinod, “局所色情報を用いた高速物体探索—アクティブ探索法—,” 信学論, Vol. J81-D- II , No.9, pp.2035-2042 (1998).
- [8]串間和彦, 赤間浩樹, 紺谷精一, 山室雅司, “色や形状等の表層的特徴に基づく画像内容検索技術,” 情処学論, Vol.40, No.3, pp.171-184 (1999).
- [9]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,” 情処学論, Vol40, No.2, pp732-741 (1999).
- [10]D. A. White and R. Jain, “Similarity Indexing with the SS-tree,” Proc. of the 12<sup>th</sup> Int. Conf. on Data Engineering, pp.516-523 (1996).
- [11]N. Beckmann, H.-P. Kriegel, R. Schneider, and B. Seeger, “The R\*-tree: an Efficient and Robust Access Method for Points and Rectangles,” Proc. of ACM SIGMOD, pp.322-331 (1990).
- [12]北研二, 獅々掘正幹, “1次元自己組織化マップを用いた高次元データの高速近傍検索,” 自然言語処理, Vol.10, No.5, pp.41-54 (2003).
- [13]SR-Tree, “<http://research.nii.ac.jp/~katayama/homepage/research/srtree/Japanese.html>”
- [14]村岡洋一, 山名早人, “e-Society 「基盤ソフトウェアの総合開発」プロジェクト「インターネット上の知識集約を可能にするプラットフォーム構築技術」,” “<http://www.yama.info.waseda.ac.jp/~yamana/e-society/>”.
- [15]ImageMagick, “<http://www.imagemagick.org/>”.