モバイル端末によるWebページ閲覧のための画像分類

前川　卓也† 　原　　隆浩† 　西尾章治郎†

† 大阪大学大学院情報科学研究科マルチメディア工学専攻
E-mail: †takuya.maekawa@gmail.com, ††{hara,nishio}@ist.osaka-u.ac.jp

あらまし　携帯電話などの貧弱な端末で通常のWebページの閲覧を実現するために，ページ内の必要ないと考えられる画像を削除したりして端末の小さな画面に表示可能なWebページを作成する研究や，Webページから重要な画像を抽出してユーザに提示する研究などが数多く行われている．しかし，既存研究において用いられている画像の検出方法は単純であるため，多くの検出誤りが発生する．一般に，Webページに含まれる画像は，それぞれが役割をもっており，例えば，サイトのメニューとして用いられる画像，箇条書きの行頭のアイテムとして用いられる画像などさまざまである．これらの画像の役割を正しく把握することで，Web画像に対して適切な処理を行うことができる．そこで本研究では，その役割ごとに，11のWeb画像のカテゴリを定義する．そして，定義したカテゴリへの画像の自動分類を実現するために，40のサイトから収集した3901の画像を手動でカテゴライズし，その結果から分類のための画像特性を抽出した．抽出した画像特性から画像の自動分類手法を提案し，実際のWebページに含まれる画像を用いて評価を行ったところ，最高で83.1%の分類精度を達成した．

キーワード　モバイルコンピューティング，Webブラウジング，Web画像，画像分類

Web Image Classification for Web Browsing Using Mobile Devices

Takuya MAEKAWA†, Takahiro HARA†, and Shojiro NISHIO†

† Dept. of Multimedia Engineering, Grad. School of Information Science and Technology, Osaka Univ.
E-mail: †takuya.maekawa@gmail.com, ††{hara,nishio}@ist.osaka-u.ac.jp

Abstract　It is difficult for mobile devices such as cellular phones equipped with a small screen and a poor input interface to browse Web pages designed for large displays such as desktop PCs. Many researches and products try to solve this problem and various solutions were proposed. In general, Web page includes images with various kinds of rolls, e.g., site menu, a line head in itemization, and the title of the page. However, most conventional works for mobile Web browsing haven't given much attention to the rolls of Web images. In this paper, we define eleven Web image categories according to their rolls for a proper Web image handling. Then, we manually categorize collected 3,901 Web images from forty real Web sites, and extract image features of each category according to the classification result. By making use of the extracted features, we propose an automatic Web image classification method. Furthermore, we evaluate the proposed automatic classification method using real Web pages, where we show that up to 83.1% accuracy is achieved.

Key words　Mobile computing, Web browsing, Web image, Image classification

1. まえがき

近年，モバイル端末の小さな画面によるWeb閲覧の制約を改善するために，Webページをモバイル端末の画面サイズに合うよう再構成することを目的とした研究や製品開発が数多く行われている [2] [7] [10]．これらにおいては，ページのレイアウトなどに用いられるような画像を削除したり，大きな画像を縮小したりする処理が一般的であるが，画像の検出方法は単純であるため多くの検出誤りが発生する．例えば，サイト内のメニューのための画像が削除されたり縮小されたりする誤りなどは非常に深刻である．この問題を解決するために，画像の役割を把握することが有効であり，これにより画像の処理方法を適切に決定することができる．例えばサイト内メニューの画像という役割をもつ画像が，ALT文字列をもつときはその文字列と置き換え，もたないときはそのまま表示するという処理が考えられる．

また，モバイル端末の貧弱な入力インタフェースによる制約 (ほとんどの携帯電話は4方向の入力キーと数字ボタンしか持たない) を改善するために，ユーザによる最低限の入力動作でWebページ内の情報を自動で提供する研究もこれまでに行わ

れている [3]．このような研究では，Web ページ内のメインコンテンツとなるコンテンツ画像やコンテンツ画像に付随するテキストを抽出し，それらを順番に提示する．その際，画像の抽出手間を省くために，構造が機知な特定のページのみを対象とすることが多いが，ページ内の画像の役割を検知することにより，任意のページにおいてコンテンツ画像を抽出することができる．また，例えば，箇条書き文章の行頭の画像を抽出することにより，文章を順番に提示する応用例なども考えられる．なお，箇条書きされている文章はページの中で重要な情報であることが多い．

以上のように，モバイル端末による Web ページ閲覧において，Web ページ内の画像の役割を適切に把握することは重要である．しかし，多くのモバイル端末による Web ページ閲覧の研究において Web 画像は丁重に扱われていない．例えば画像は，その高さ，幅，アスペクト比といった単純な属性のみからその役割や重要性が決定されている．そこで，本研究では，Web 画像の正確な取り扱いを可能とするため，また，画像の役割を利用した新しいアプリケーションの構築を支援するため，11 の Web 画像のカテゴリを定義する．そして，様々な種類の 40 の Web サイトから収集した 3901 の画像を，それらの 11 のカテゴリに手動で分類する．手動分類の結果から，ページ内の画像を効果的に分類するための 37 の画像特性を定義し，それらの特性から作成した決定木を用いて画像の自動分類を実現する [9]．

ここで画像から抽出可能な特性には，HTML ソースの解析から抽出できるもの，Web サーバに通信で問い合わせて取得できるもの，ページを実際にレンダリングした際に DOM Tree のレイアウト情報から取得できるもの，画像処理から取得できるものがある．モバイル環境では，端末の性能や通信環境から取得できる特性はさまざまである．また，端末を取り巻く環境によっても取得できる特性は異なる．例えば，プロキシサーバを利用できる環境なら，全ての特性を取得することも不可能ではない．さらに，アプリケーションによっては 11 ものカテゴリへの分類を求めないものや，特定のカテゴリへの分類に対する精度のみを要求するものもあり，全ての特性を利用する必要がないこともある．そこで本研究では，利用できる特性ごとに 11 のカテゴリへの分類を評価した．そして，11 のカテゴリに対して最高で 83.1%の分類精度を達成した．

以降，2. では定義した 11 のカテゴリについて説明し，3. で本研究の応用例について説明したあと，4. で関連研究について述べる．5. で画像の収集方法，手動による分類結果，および，自動分類に利用する画像特性について述べる．6. で自動分類の評価を行い，7. で考察を行う．最後に，8. で結論を述べる．

2. 画像のカテゴリ

本研究の詳細を述べる前に，定義した 11 のカテゴリについて説明する．また，収集した画像群に見られたカテゴリごとの特徴についても簡単に説明する．ここで，以降の理解を容易にするために，主に文字列のみを含む画像が属する 4 つのカテゴリをまとめて文字列画像と呼び，小さい画像が属する 2 つのカテゴリをまとめて小画像と呼ぶ．以下の説明は，本研究で収集した 40 の Web サイトに含まれる画像を例として用いる．

- 文字列画像: 以下の四つのカテゴリに分類される．
  - **MENU**: サイト内のメニュー画像を含む．図 1(a) における “HOME”，“SCHEDULE” などの画像や，図 1(b) における “THE COLLECTION”，“EVENTS & PROGRAMS” がこれに当たる．ページの上部や下部に横並びで配置されることが多い．ページの左部に縦並びで配置されることも多い．収集した 3901 の画像を調査したところ，MENU に属する画像のうち，67.6%の画像において 3 つ以上横に並ぶ同じ高さの画像があった．11.5%の画像において 3 つ以上縦に並ぶ同じ幅の画像があった．また，ページの上部や下部に単独で存在するホームへのリンク画像なども MENU に属する．アスペクト比が小さいことが多く，平均で 0.320 であった．
  - **SECTION**: ページ内のセクションの表題画像を含む．図 1(c) における，“U.S.” の画像がこれに当たる．この画像の下に文字列が配置されていることが多い．実際，92.8%の画像の下に文字列が配置されていた．アスペクト比が小さいことが多く，平均で 0.142 であった．
  - **DECORATION**: 装飾文字列の画像を含む．図 1(d), (e) の画像がこれに当たる．HTML 言語では表現できないような文字列を利用するときに用いる．ハイパーリンクをもたない．
  - **BUTTON**: ハイパーリンクをもつ画像を含む．図 1(f), (g) の画像がこれに当たる．文字列が周辺に配置されていることが多く，周辺の文字列に関連するページへのハイパーリンクとなっている．BUTTON に属する画像のうち，16.1%が上に文字列をもち，8.0%が下に文字列をもち，36.8%が左に文字列をもち，13.8%が右に文字列をもち，25.3%が文字列を周辺にもたなかった．アスペクト比が小さいことが多く，平均で 0.266 であった．
- 小画像: 以下の二つのカテゴリに分類される．
  - **ITEM**: 箇条書きにおける行頭のアイテムの画像を含む．図 2(a) の画像がこれに当たる．同じ幅の画像が縦並びで配置されることが多く，74.6%が 3 つ以上縦に並ぶ同じ幅の画像をもっていた．また，文字列が右に配置されていることが多く，

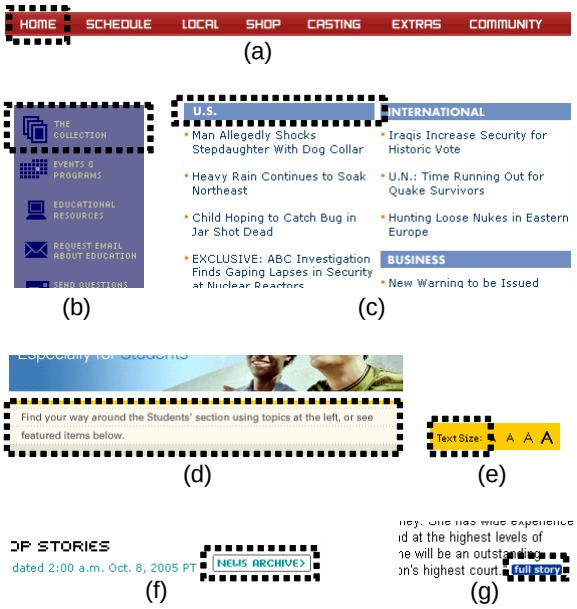

図 1　文字列画像

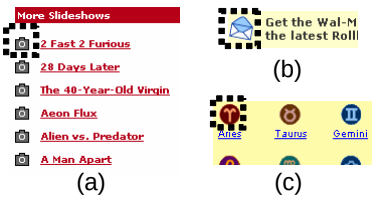

図 2　小　画　像

図 3　TITLE，MAP，AD，CONTENT，LAYOUTER

99.4%がそうであった．それ以外は，画像が右に配置されていた．周辺に配置されている文字列の平均は 31.7 文字であった．文字列を含む画像でも，箇条書きのアイテムとして用いられているときはこの分類に属する．アスペクト比が 1 に近いことが多く，その平均は 1.052 であった．

– **ICON**: あるものを表現する画像を含む．図 2(b)，(c) の画像がこれに当たる．文字列が右や左に配置されていることが多く，58.3%が右に文字列をもち，22.0%が左に文字列をもっていた．アスペクト比が 1 に近いことが多く，その平均は 0.942 であった．

- **TITLE**: ページのタイトル画像を含む．図 3(a) の画像がこれに当たる．ページの上部に配置され，サイトのインデックスページや，そのページ自身へのハイパーリンクであることが多い．アスペクト比が小さいことが多く，その平均は 0.279 であった．
- **MAP**: マップメニューの画像を含む．図 3(b) の画像がこれに当たる．サイトのメニューとして用いられることが多い．
- **AD**: バナー広告の画像を含む．図 3(c) の画像がこれに当たる．外部サイトへのハイパーリンクであることが多く，25.5%がそうであった．また，アスペクト比が小さいことが多く，その平均は 0.459 であった．少量の文字列が下に配置されていることがある．14.0%が下に文字列をもち，78.7%が周辺に文字列をもたなかった．下に配置する文字列の平均は 45.2 文字であった．
- **CONTENT**: Web ページ内のメインのコンテンツに関連付けられている画像を含む．図 3(d) の画像がこれに当たる．アスペクト比が 1 に近いことが多く，その平均は 0.951 であった．文字列が右や下に配置されていることが多く，35.1%の画像が右に文字列をもち，51.7%の画像が下に文字列をもっていた．周辺文字列の平均は 99.7 文字であった．JPEG 形式であることが多く，55.4%がそうであった．一方，CONTENT 以外に属する画像では 6.6%が JPEG 形式であった．
- **LAYOUTER**: ページのデザインやレイアウトを調整するための画像を含む．空白や単一色の画像であることが多く，ページ内の画像や文字列の表示位置を調整するために用いられることが多い．1 つの画像がページ内で複数回利用されることが多く，平均で 10.7 回用いられていた．図 3(e) のように，他の重要な画像のレイアウトを調整するために用いられることもある．

3. 応　用　例

前述のように，画像の自動分類はモバイル端末による Web 閲覧において重要である．以下では，そのさまざまな応用例について述べる．

3.1　Web ページ解析

モバイルの小さな画面による制約を改善するために，Web ページを解析してユーザによる Web ページ閲覧を補助する研究は多くある．例えば Chen ら [4] は，Web ページを小さな画面でも十分に表示できるコンポーネントにセグメント化する手法を提案している．ユーザは，任意の 1 つのコンポーネントを端末に表示させることで，大きなページを小さなコンポーネントごとに閲覧できる．また我々の研究グループでは，コンポーネントを複数のモバイル端末に分配し，近くにいるモバイルユーザ同士でページを協調して閲覧するための研究をおこなっている [8]．Web ページのセグメント化において重要なポイントは，ページ内に連続して存在する文字列や画像をどこで区切るかである．コンポーネント抽出に関する研究の多くは，タグ解析や，レンダリングしたページを画像解析することで，コンポーネントを区切るセパレータの位置を検出している．

ここで，画像の特性を併用することで，より正確にセパレータを検出できるものと考えられる．例えば，メニュー画像が連続してあるときは，それらのメニュー群を 1 つのコンポーネントとするようにセパレータを設定する．また，セクション画像があるときは，その画像の上部をセパレータとして設定する．

いくつかの研究では縦，横幅，または，アスペクト比が非常に小さい画像をセパレータとしてみなしているが，それらの簡単な条件だけでセパレータを把握するのは，実際には困難である．例えば，画像の縦と横がともに 10 ピクセル以下の画像をセパレータと定義したとき，われわれが収集した画像の中では，32.0%の誤りがあった．

3.2　Web ページの再構成

1. で簡単に説明したように，モバイル端末のための Web ページ再構成に関する研究や製品開発は数多く行われている．例えば，Anderson ら [1] は，ユーザの Web 閲覧履歴を用いて，Web ページから不必要とみなされるコンテンツを削減し，小さな画面の端末でも閲覧可能なページを作成している．このようなコンテンツの削減において，画像の役割は特に重要である．例えば，図 2(a) のように，箇条書き文章があるとき，アイテムごとの閲覧履歴や嗜好情報から，不必要なアイテムを決定できる．これは，画像が ITEM に属することが検出できなければ行えない．文章のみを解析したとしても，行頭の画像は取り除かれない．またメニュー画像群を認識し，あまり利用されないメニューを削除したり，セクションのタイトル画像のみを残して，セクションの中身を削除できる．メニューを削除するときは，他のリンクと重みを変えて削除することもできる．

Opera for Mobile [10] などでは，1. で述べたように Web ページを端末の画面の横幅に合うように再構成するために，画像の縮小や削除などが行われる．1. では，必要な画像が削除されたり，文字列を含む画像が縮小されたりする問題について述べたが，ここでは画像の削除によりページの可読性が大幅に減少する問題について述べる．例えば，テーブルタグを用いて実現されている箇条書きにおいて，テーブルのレイアウトを無視して，テーブル内の文字列を改行なしで表示する場合を考える．テーブルによるレイアウトを無視することは，限られた画面しかもたないモバイル Web ブラウジングにおいて有効である．しかし，箇条書きの行頭画像は，小さい画像であるため削除されやすい．これにより，アイテム間の境目がなくなるため，ユーザ

はアイテムを区別して認識できない．しかし，画像の役割を検出することで，ITEM 画像を削除しないようにしたり，代替文字列を代わりに挿入したりできる．

4. 関 連 研 究

以上に述べたように，画像の役割を把握することにより，Web ページ解析の精度向上や，さまざまな新しいアプリケーションの構築に役立つにもかかわらず，Web 画像を詳細に分類する研究はこれまでにほとんど行われていない．Peak [11] は，Web 画像の 5 つのカテゴリを定義している．そして，Sony と BBC の Web サイト内の画像から，ページ内のコンテンツ画像のみを抽出している．具体的には，“広告画像”，“コンテンツ画像”，ボタンやセクションタイトルなどを含む “装飾画像”，工事中の情報などを表す画像を含む “情報画像”，“ロゴ画像”，ホームへ戻るボタンなどを含む “ナビゲーション画像” のカテゴリを定義している．画像を分類するために，主に画像処理により取得した特性を用いており，HTML ソースなどから取得できる特性はほとんど用いていない．さらに画像の ALT 文字列を解析しているため，特定の言語で記述されたページのみにしか適用できない．

Hu ら [6] は，BBC や CNN などのニュースサイト内の画像に対して 7 つのカテゴリを定義している．具体的には，ニュースの文章に関連する画像を含む “記事画像”，リンク先のニュースの内容を表す画像を含む “プレビュー画像”，“広告画像”，コラムやテレビの番組を表す画像を含む “ホスト画像”，コラムや文章の表題画像を含む “表題画像”，および，“アイコンロゴ画像” のカテゴリを定義している．そして，ニュースサイトにおいて重要である，“記事画像”，“プレビュー画像”，“ホスト画像” を抽出している．この研究でも，HTML ソースから得られた簡単な特性と画像処理および簡単な文章解析を用いたカテゴライズを行っている．また，対象をニュースサイトとしているが，本研究ではポータルサイトやショッピングサイトなどのさまざまなページを対象とする．

本研究ではこれらの研究と異なり，定義した全ての画像を分類している．また，モバイル環境下を考慮し，利用できる画像特性ごとに画像分類の精度を評価している．

5. データセットの収集と検証

5.1 画像の収集と分類

まず，ポータルサイト，オンラインショッピングサイト，ニュースサイト，製造会社のサイトなど，表 1 に示すランダムに選んだ 40 の Web サイト (20 の米国サイトと 20 の日本サイト) において，インデックスページおよびその他の 2 つのページから画像を収集した．2 つのページはインデックスページから互いにリンクしており，そのサイト内で重要と思われるものを選んだ．そして，40 のサイトから収集した計 3901 の画像を，2. において定義した 11 のカテゴリに手動で分類した．その結果を表 2 に示す．ここで，1 つのページに同じ画像が複数回使われているときは，最初に現れる画像のみを対象とした．

3901 もの画像の中には，定義した 11 のカテゴリに容易に分類できない画像も存在する．以下に，そのような特殊な画像の分類例を示す．

- 2 つの役割をもつ画像は，より重要と思われる役割を選択した．アイコンの絵の右にセクションのタイトル文字列の絵が配置されている画像がこの例である．この場合は，SECTION として分類する．
- 複数の部分画像を組み合わせて 1 つの絵を構成するような画像群が稀にある．このような画像群は，画像ごとにそれぞれがもつ役割を決定して分類した．
- 文字の含まれないボタンが稀に存在する．絵のみが記されたリンクなどが，その例である．このような画像は小画像として，ICON に分類した (文字列画像ではない)．

表 1　40 の Web サイト

msn.com	google.com	walmart.com
ebooks.com	buy.com	nationalgeographic.com
weather.com	nbci.com	metmuseum.org
abc.go.com	flowgo.com	en.wikipedia.org
ed.gov	lycos.com	whitehouse.gov
intel.com	hp.com	sonymusic.com
wired.com	nytimes.com	yahoo.co.jp
goo.ne.jp	amazon.co.jp	ntt.com
nhk.or.jp	honda.co.jp	reuters.co.jp
asahi.com	sankei.co.jp	casio.co.jp
nifty.com	nikkei.co.jp	jp.fujitsu.com
infoseek.co.jp	nnn24.com	watch.impress.co.jp
livedoor.com	rakuten.co.jp	ufjbank.co.jp
chunichi.co.jp		

表 2　収集画像の分類結果

MENU	686	TITLE	141
SECTION	469	MAP	53
DECORATION	69	AD	329
BUTTON	87	CONTENT	951
ITEM	311	LAYOUTER	541
ICON	264		

5.2 画像の特徴

2. では各カテゴリに属する画像の特徴について説明した．以下では 2. であげたような特徴のみでは分類困難な画像を，画像処理により分類する方法について述べる．

5.2.1 画像の複雑さ

ページの上部に存在する MENU や SECTION 画像は，TITLE 画像とアスペクト比が似ている．しかし画像の内容に注目すると，TITLE のほうがロゴマークやデコレーションを施した文字列を含むことが多く，複雑な構造をしている．そこで，画像の色数と同色領域の数に注目する．ここで同色領域とは 4 近傍の関係にある同じ色のピクセル同士をグループ化したものである．MENU 画像の色数の平均は 42.32 で，領域数の平均は 22.79 であった．SECTION 画像は 40.53 および 41.83，TITLE 画像は 89.58 および 109.55 であった．このように色数と領域数に大きな隔たりが発見された．実際に t 検定 ($p < 0.001$) により MENU，TITLE 間および SECTION，TITLE 間に有意差が認められた．ただし，色数の平均は GIF 画像のもののみの平均である．

ここで，SECTION に属する画像の 92.8%がその下に文字列をもっていた．また，BUTTON に属する画像の 16.1%がその下に文字列をもっていた．大きめのボタンとセクションは形状も類似している．しかし，BUTTON に属する画像の同色領域数の平均は 26.90 となっており，SECTION と大きな差が見られた．

5.2.2 近くにある画像との類似度

ページの左端に存在する縦並びの MENU 画像群と SECTION 画像群は，画面の左端に縦並びに存在する AD 画像との分類が困難である．ここで，MENU や SECTION に属する画像は，近くに使用色のやデザインが類似する画像をもつことが多い．そこで，画像 P_1 と P_2 の類似度を以下の式で定義する．

$$S(P_1,P_2)=\frac{\sum_{x=0}^{min(W_1,W_2)-1}\sum_{y=0}^{min(H_1,H_2)-1}D(P_1(x,y),P_2(x,y))}{min(W_1,W_2)\cdot min(H_1,H_2)\cdot 255\sqrt{3}}.$$

ただし，W_1，H_1 は画像 P_1 の幅と高さを表し，W_2 と H_2 は画像 P_2 の幅と高さをそれぞれ表す．$D(p_1,p_2)$ は，ピクセル p_1 と p_2 の RGB 空間におけるユークリッド距離である．画像の周辺画像の中で最小の類似度を分類に利用する．ここで周辺の画像とは，HTML ソースにおいてその画像の img タグに対してタグのインデックスが 30 以内のものである．インデックスは，HTML ソースにおいて，そのタグが出てくる順番である．また，類似度は高さか幅が同じ画像間でのみ計算する．類似画像が見つかった MENU，SECTION，AD に属する画像群の最小類似度の平均はそれぞれ，0.139，0.136，0.373 となっており，AD は大きな値となった．実際に t 検定 ($p < 0.001$) により MENU，AD 間および SECTION，AD 間に有意差が認められた．

5.2.3 BUTTON 画像の特徴

大きめ BUTTON 画像は，SECTION だけでなく AD 画像とも形状が類似する．また，小さめの BUTTON 画像は ICON 画像と形状が類似する．これらの画像と BUTTON 画像を区別するために，BUTTON 画像に多く見られる特徴を利用する．BUTTON 画像は，図 1(f)，(g) のように，角が丸まった矩形であることが多く，実際に BUTTON に属する画像の 37.9%がその形状をもっていた．一方，SECTION，AD，ICON に属する画像はそれぞれ 1.9%，6.1%，7.2%であった．

5.2.4 画像に含まれる文字列

画像内の文字列を検出することで，文字列画像を区別することができる．例えば，LAYOUTER 画像はさまざまな形で至るところに存在するため，SECTION や DECORATION，TITLE 画像と混同しやすい．また，AD 画像は複数の位置に文字列をもつことが多い．例えば，図 3(c) では，複数の領域に異なるフォントの文字列が描かれている．一方で，MENU や SECTION は 1 つの領域に文字列が描かれていることが多い．例えば，図 1(a) では，1 つの領域に同じフォントの文字列をもつ．

画像内の文字列を認識するアルゴリズムは多くあるが，今回は文字列の位置を認識するアルゴリズムとして，文献 [12] を利用した．このアルゴリズムを実装したところ，多くの文字列ではない部分が文字列として誤認識された．図 4(a) は，アルゴリズムをある画像に適用した際の出力であるが，人の顔が誤って文字列と認識されている．そこで，文字列の候補となる隣接関係にあるピクセルのグループを，穴の数，アスペクト比，密度を考慮してフィルタリング [14] した後，横に隣接する大きさの近いグループ同士をグループ化した．これにより，同じ行の文字列領域が取得できる．さらに，領域のアスペクト比が大きいものをフィルタすることにより最終的な結果を得る（図 4(b)）．

実装した認識アルゴリズムを今回収集した全ての画像に適用し，どの程度文字列を認識できるか評価したところ，79.70%の画像において，ほとんどの文字列が抽出されており，ノイズもほとんどなかった．10.18%の画像において，半分程度文字列が抽出されているか，ある程度のノイズを含んでいた．10.13%の画像において，文字列がほとんど抽出できていなかった．

以上のアルゴリズムを用いたところ，LAYOUTER, SECTION, DECORATION, TITLE 画像の文字列領域占有率の平均はそれぞれ 0.40%, 37.89%, 55.19%, 44.85%であった．また，AD, MENU, SECTION 画像の文字列領域数の平均はそれぞれ 2.75, 1.04, 1.19 であった．実際に t 検定 ($p < 0.001$) により

(a) (b)

図 4 文字列の抽出

AD，MENU 間および AD，SECTION 間の文字列領域数に有意差が認められた．

5.3 自動分類に用いる画像の特徴

2. や前節の議論に基づいて，画像の自動分類を実現するため，抽出する画像の特徴を以下に示す．特性 F1〜F20 は HTML ソースの解析により取得できる値である．なお，幅や高さなどの値がソースの IMG タグに明示されていないときは欠損値となる．F21，F22 は画像をダウンロードする際や，Web サーバに HEAD メソッドを用いてリクエストする際に取得できる値である．F23〜F30 は，Web ページを受信してレンダリングする際に，DOM ツリーなどから取得できる値である．本研究では MSHTML DOM ツリーを用いてこの値を取得した．F31〜F37 は，画像処理を用いて取得できる値である．ただしアニメーション GIF の画像処理は，一枚目の静止画に対して適用している．

- **F1-面積**
- **F2-幅**
- **F3-高さ**
- **F4-アスペクト比**
- **F5-マップを利用しているか否か**{TRUE, FALSE}: MAP に属する画像が TRUE となる．
- **F6-ハイパーリンクか否か**{TRUE, FALSE}: SPACER に属する画像の多くや，DECORATE に属する画像は FALSE となる．
- **F7-アウトリンクか否か**{TRUE, FALSE}: 他のドメインへのリンクか否かを示す．
- **F8-ループバックリンクか否か**{TRUE, FALSE}: サイトのインデックスページや，その画像が含まれるページ自身へのハイパーリンクか否かを示す．
- **F9-ALT の有無**{TRUE, FALSE}: 文字列画像に属する画像などの多くは TRUE となる．MENU, SECTION, DECORATION, BUTTON に属する画像のうち，それぞれ 85.4%, 74.0%, 66.7%, 63.2%の画像が TRUE であった．また，TITLE, AD, ICON, ITEM, CONTENT, LAYOUT に属する画像のうち，それぞれ 78.0%, 58.1%, 45.4%, 40.1%, 40.1%, 18.9%の画像が TRUE であった．BUTTON が予想外に低い値となった．
- **F10-ALT の文字数**: CONTENT に属する画像はこの値が大きく，ALT をもつものの平均は 26.8 文字であった．MENU, SECTION, DECORATION, BUTTON, ICON, ITEM, TITLE, AD の平均はそれぞれ 8.5, 11.3, 19.6, 9.18, 3.8, 9.9, 19.0, 19.7 文字であった．また，MENU, SECTION, DECORATION, BUTTON の ALT 文字列はほとんどがそれらの画像に含まれている文字列であった．
- **F11-周辺文字列の文字数**: 画像の周辺に配置されている文字列の文字数を示す．MENU に属する画像はこの値が小さく，その平均は 2.7 文字であった．一方，全ての画像における平均は 69.8 文字であった．
- **F12-JPEG か否か**{TRUE, FALSE}: CONTENT 以外に属する画像はほとんどが FALSE となる．
- **F13-HTML ソースにおけるインデックス**: ソースにおいてその画像の IMG タグが現れるインデックスを示す．TITLE はこの値が小さいことが多く，その平均は 48.4 であった．一

方，全ての画像における平均は 424.7 であった．

- **F14-同じページ内で画像の現れる回数**: MENU, SECTION, DECORATION, TITLE, MAP, AD に属する画像はほとんどが 1 だった．ITEM と LAYOUTER に属する画像はこの値が大きいことが多く，その平均は 10.65, 6.77 であった．全ての画像における平均は 2.97 であった．
- **F15-同じページ内の面積が同じ画像の数**: CONTENT, ICON, ITEM に属する画像はこの値が大きいことが多く，その平均はそれぞれ 7.5, 4.3, 4.0, であった．一方，MENU, SECTION 画像は平均で 2.0, 3.0 とそれほど大きな値とならなかった．
- **F16-同じページ内の幅が同じ画像の数**: CONTENT, AD, ICON, ITEM, SECTION に属する画像はこの値が大きいことが多く，その平均はそれぞれ 8.1, 3.5, 4.3, 4.5, 4.4 であった．AD 画像は，縦揃えでページの端に並べられていることが多いため，この値が大きくなった．
- **F17-同じページ内の高さが同じ画像の数**: CONTENT, MENU, SECTION, ICON, ITEM に属する画像はこの値が大きいことが多く，その平均は, 8.1, 8.5, 4.8, 4.4, 4.8 である．
- **F18-同じページ内の近くに位置する面積が同じ画像の数**: 近くに位置する画像とは，インデックスが前後 100 以内のものである．BUTTON, TITLE, MAP, LAYOUTER に属する画像はこの値が小さいことが多く，その平均は, 0.64, 0.04, 0.04, 0.27 であった．全ての画像における平均は 2.20 であった．
- **F19-同じページ内の近くに位置する幅が同じ画像の数**:BUTTON, TITLE, LAYOUTER に属する画像はこの値が小さいことが多く，その平均は，0.78, 0.41, 0.85 であった．全ての画像における平均は 2.67 であった．
- **F20-同じページ内の近くに位置する高さが同じ画像の数**:MENU に属する画像はこの値が大きいことが多く，その平均は 7.76 であった．BUTTON, TITLE, MAP に属する画像はこの値が小さいことが多く，その平均は，1.22, 0.94, 0.55 であった．全ての画像における平均は 3.69 であった．
- **F21-バイトサイズ**
- **F22-面積あたりのバイトサイズ**: CONTENT や AD 画像などの複雑な画像はこの値が大きいことが多く，その平均は 0.83, 0.71byte/pix^2 であった．また，小さい画像はこの値が大きいことが多い．ICON, ITEM, LAYOUTER 画像は，その平均がそれぞれ 1.2, 1.0, 8.9 であった．これは，画像ファイルのヘッダによる影響であると考えられる．
- **F23-画像の左上の X 座標**: TITLE に属する画像はこの値が小さいことが多く，その平均は 46.2 であった．一方，全画像の平均は 314.1 であった．
- **F24-画像の左上の Y 座標**: MENU や TITLE に属する画像はこの値が小さいことが多く，その平均は 216.3, 20.0 であった．一方，全画像の平均は 603.4 であった．
- **F25-同じページ内の F23 が同じ画像の数**: ページにおいて特定の URL をもつ画像が複数回使われている場合，それぞれを 1 つの画像としてカウントする．SECTION, ITEM, TITLE, CONTENT, AD, LAYOUTER に属する画像はこの値が大きいことが多く，その平均は 13.8, 15.4, 6.1, 6.2, 6.8, 8.4 であった．LAYOUTER が大きいのは，縦に並べられている画像や文字列間の空白を作るために用いられることが多いからである．
- **F26-同じページ内の F24 が同じ画像の数**: ページにおいて特定の URL をもつ画像が複数回使われている場合，それぞれを 1 つの画像としてカウントする．MENU に属する画像はこの値が大きいことが多く，平均は 5.8 であった．一方，全ての画像の平均は 2.0 であった．
- **F27-同じページ内の幅と F23 が同じ画像の数**: ページにおいて同じ画像が複数回使われている場合，それぞれを 1 つの画像としてカウントする．SECTION, ITEM, CONTENT, AD に属する画像はこの値が大きいことが多く，その平均は 4.4, 11.2, 3.1, 2.9 であった．TITLE と LAYOUTER 画像は F25 は大きかったが，この値は小さくなった．
- **F28-同じページ内の高さと F24 が同じ画像の数**: ページにおいて同じ画像が複数回使われている場合，それぞれを 1 つの画像としてカウントする．MENU に属する画像はこの値が大きいことが多く，平均は 5.6 であった．一方，全ての画像の平均は 1.6 であった．
- **F29-ページの下部から画像の下部までの距離**: MENU に属する画像の一部はこの値が小さいことが多い．
- **F30-周辺文字列の位置**{UP, DOWN, LEFT, RIGHT, NONE}
- **F31-使用色数**
- **F32-同色領域数**
- **F33-近くにある画像との類似度**
- **F34-アニメーション GIF か否か**{TRUE, FALSE}: AD はアニメーション GIF であることが多く，実際 AD に属する画像のうち，14.29%の画像がアニメーション GIF となっていた．一方，AD 以外に属する画像のうち 0.36%がアニメーション GIF であった．
- **F35-角が丸まった矩形か否か**{TRUE, FALSE}
- **F36-文字列領域の占有率**
- **F37-文字列領域の数**

ここで，周辺文字列の取得方法について簡単に説明する．HTML ソースは基本的に木構造となっている．そこで，対象とする画像の IMG タグから兄弟ノードの文字列を探索した後，親ノードの兄弟ノードを探索する．この手続きを繰り返すことにより周辺文字列を取得する．兄弟を探索するときは，近い兄弟から順番に 4 つのノードを探索するようにしている．また，親を辿る回数に制限を与えている．以上のような簡単な文字列処理の手続きにより，周辺文字列の取得を実現している．

6. 評価実験

本章の評価実験では，トレーニングセットの画像を 11 のカテゴリに分類する決定木を 5. で示した画像特性から作成し，作成した決定木を用いてテストセットの分類を行った．決定木の作成には C4.5 [13] を用いた．40 のサイトから収集した画像から，1 つのサイトの画像をテストセットとし，残り 39 のサイトの画像をトレーニングセットとするような評価実験を，全てのサイトがそれぞれテストセットとなるよう 40 回行った．ここで，利用する画像特性を以下の条件のように変化させて実験を行った．

- **C1**: HTML ソースの解析により取得できる特性を用いる．つまり，F1〜F20 を用いる．
- **C2**: HTML ソースの解析により取得できる特性および Web サーバから取得できる特性を用いる．つまり，F1〜F22 を用いる．
- **C3**: HTML ソースの解析により取得できる特性，Web サーバから取得できる特性，および，DOM から取得できる特性を用いる．つまり，F1〜F30 を用いる．ただし，DOM からも Web サーバから取得できる特性を取得できるため，実際には Web サーバにアクセスする必要はない．
- **C4**: HTML ソースの解析により取得できる特性，Web

表 3　分類精度

C1	C2	C3	C4	C5
0.749	0.768	0.796	0.766	0.831

表 4　F　　値

	C1	C2	C3	C4	C5
MENU	0.854	0.886	0.852	0.826	0.876
SECTION	0.749	0.742	0.863	0.772	0.867
DECORATION	0.177	0.106	0.109	0.173	0.294
BUTTON	0.392	0.364	0.360	0.365	0.458
ITEM	0.578	0.652	0.800	0.567	0.834
ICON	0.388	0.477	0.647	0.539	0.673
TITLE	0.676	0.725	0.732	0.681	0.801
MAP	0.981	0.981	0.981	0.971	0.971
AD	0.716	0.744	0.657	0.705	0.694
CONTENT	0.888	0.882	0.906	0.911	0.911
LAYOUTER	0.746	0.761	0.787	0.844	0.880

サーバから取得できる特性，および，画像処理により取得できる特性を用いる．つまり，F1〜F22, F31〜F37 を用いる．画像処理を行うためには，画像をサーバからダウンロードする必要があるため，Web サーバから画像のサイズを個別に取得する必要はない．

- **C5**: 全ての特性を用いる．つまり，F1〜F37 を用いる．

C1 から C5 における分類精度を表 3 に示す．ソースの解析による特性のみを用いたときでも，75%もの画像が正しく分類されている．また，基本的に利用する属性が増えるほど，その精度は増加するが，C4 は C2 に比べて微小ではあるが逆に精度が低下している．C4 に比べて DOM から得た特性を利用する C5 では，その精度が 0.831 と大幅に向上している．一方，C5 は C3 に対して画像処理から得た特性を追加しており，C3 に比べても大幅に精度が向上している．これにより，画像処理は DOM と組み合わせることで，その効果を発揮することが分かる．また C2 は C1 に対して，2 つの特性を追加しただけであるが，その精度は 0.019 も向上している．2 つの特性を得るためには Web サーバにアクセスする必要があるものの，その効果は大きい．

また，C1 から C5 におけるカテゴリごとの F 値（F-measure）を表 4 に示す．ただし，F 値は再現率と適合率から以下の式で求める．ここでは，b を 1 としている．

$$F\text{-}measure = \frac{(b^2+1)\cdot Precision \cdot Recall}{b^2 \cdot Precision \cdot Recall}.$$

全ての結果において，DECORATION, BUTTON 画像の F 値が低いことが分かる．これは，それぞれのカテゴリにおいて，他のカテゴリとはっきり区別できる特性をもたないためである．しかし，利用する特性を増加させるほど，その精度は徐々に向上している．これは，他のカテゴリの特徴が特性を増加させるほど明確になるためであると考えられる．

ICON と AD 画像も全ての結果において，それほど高い精度をもっているとは言えない．ICON 画像も，他のカテゴリと明確に区別できる特性をもたない．しかし ICON 画像は，DECORATION や BUTTON 画像と同様に利用する特性を増加させるほど，その精度は徐々に向上している．AD 画像は C2 おいて，0.744 もの精度をもつが，C3 では精度を大幅に落としている．これは，F27 や F30 などから MENU 画像と混同されていることが多いためであるが，C5 において画像処理を用いることで，ある程度は回復している．

ITEM 画像は，DOM による特性を用いることで，大幅にその F 値が向上する．ITEM は，その形状が ICON や LAYOUTER 画像と混同することが多いが，F27 や F30 により分類が可能となるためである．

以上の結果から，C5 の特性を利用できる端末でも，DECORATION や BUTTON 画像を利用するアプリケーションへの適用は難しいが，1. や 3. で述べたようなほとんどのアプリケーションへの適用が可能である．また，画像処理を行うほどの性能はもたないが，Web ページのレンダリング機能をもつ端末においても，C3 の条件では DECORATION や BUTTON 画像以外はある程度の精度をもつため，レンダリングした Web ページの閲覧補助などのアプリケーションが実現可能である．C2 や C1 の条件において用いられる特性しか利用できない端末においても，画像の役割を用いた Web ページの再構成や個人化などの精度向上が可能である．

実験結果をさらに検証するため，C5 での 40 回の評価における混同行列（Confusion matrix）を重ね合わせた行列を表 5 に示す．MENU と AD 画像はこれまでにも述べたように，それぞれ似た特性をもつものが存在するため，互いに誤認識されていることが多かった．SECTION と AD 画像が互いに誤認識されていることも多い．しかし，F33 を用いることでその誤認識が改善していることを確認している．

ICON 画像は，その出現パターンに明確な法則がないため，LAYOUTER 画像と互いに混同しやすくなっている．さらに，大きめの ICON 画像は小さめの CONTENT 画像として誤認識されていることを確認しており，特に右に周辺文字列をもつ大きめの ICON は，小さめで写真でかつ GIF 形式の CONTENT との区別が難しかった．手動での分類の際には，ある程度小さくても写真の画像は CONTENT として分類した．

ICON と ITEM 画像は互いに誤認識されやすくなっている．これは，稀に F27 が大きい ICON 画像があり，それらと ICON 画像が混同しやすいためであった．

CONTENT 画像は MENU や SECTION 画像に間違えて分類されることが多くなっている．特に単純な文字列しか含まない画像が多く誤分類されていた．また，CONTENT 画像は AD 画像に間違えて分類されることが多い．これは，形状が AD 画像に似ているものが誤分類されていることが多かった．

LAYOUTER 画像は，CONTENT 画像に誤分類されることが多い．複雑な画像のレイアウトを実現するための，ある程度大きく使用色数の多い画像が誤分類されていた．

7. 考　　察

7.1 アプリケーションについて

画像分類における最大のオーバーヘッドは画像特性の取得である．アプリケーションによっては，特定のカテゴリへの分類のみを求めるものもあるため，最低限利用すればよい特性の数は異なる．例えば，文字列画像のみをそれぞれ区別せずに検出する必要があるものもあれば，CONTENT 画像と TITLE 画像のみをそれぞれ検出する必要があるものもある．これらには無数の組み合わせが考えられるため，本評価では最低限の分類精度を達成する 11 のカテゴリへの分類をおこなっている．つまりアプリケーションを実装する際には，評価結果を参考にどの特性を最低限利用すればよいか判断する必要がある．さらに，利用する特性を細かく選定するためには，決定木の内容まで解析する必要がある．

また，アプリケーションによっては本研究で定義している 11 のカテゴリよりさらに詳細な分類を要求するものも存在すると考えられる．例えば，CONTENT 画像をさらに写真画像とそ

表 5　C5 の結果における混同行列

IC1	IC2	IC3	IC4	IC5	IC6	IC7	IC8	IC9	IC10	IC11	Category
611	2	0	14	0	15	6	0	33	1	4	IC1: MENU
4	413	8	6	3	2	2	0	16	8	7	IC2: SECTION
0	21	16	2	8	7	3	0	3	1	8	IC3: DECORATION
16	5	4	41	0	8	1	0	9	3	0	IC4: BUTTON
0	1	0	0	252	46	2	0	0	0	10	IC5: ITEM
5	2	1	6	20	190	0	0	0	19	21	IC6: ICON
10	4	9	0	0	1	111	0	4	2	0	IC7: TITLE
0	2	0	0	0	0	0	50	0	1	0	IC8: MAP
40	13	0	20	0	1	7	0	235	11	2	IC9: AD
20	19	0	2	0	9	4	0	39	850	8	IC10: CONTENT
3	2	2	1	10	22	0	0	9	20	472	IC11: LAYOUTER

れ以外の画像に分類したり，文字列を含むものと含まないものを分類したりなどである．写真画像とそれ以外の画像を分類する研究 [5] は多くあり，決定木による分類結果に対してそのような手法を適用することで実現できる．文字列を含むものと含まないものを分類するためには，決定木による分類において用いる F36 を利用すればよい．このように，決定木による分類結果に対して新しく用意した特性を適用したり，すでに抽出されている特性を適用したりすることで，より細かい分類を容易に実現できる．

7.2　広告画像の定義について

ある画像の役割が AD であるかそれ以外の役割であるかどうかは，ユーザの視点により異なる場合がある．例えば，サイト製作者が CONTENT 画像としてページに埋め込んだ画像が，ユーザの視点から見ると単なる AD 画像である場合などである．このような現象は，ユーザにニュースなどの情報を無料で配信しており，その収入を Web の広告などで得ているサイト以外のサイトにおいて起こりやすい．つまり，そのページ自身のサービスによりユーザから収入を得ていたり，そのサイトが現実の業務に対する付随的なものであったりする場合である．このようなサイトに含まれる情報は，そのほとんどが，そのサイトの収入につながるものであったり，新しいサービスの紹介であったり，実際の業務の紹介や広告であったりするため，極端に言えば全ての情報がユーザにとって広告に見える可能性がある．そこで，AD 画像を，AD 画像のリンク先ページへと誘導するために作られた，そのページの紹介画像として定義する．そのような画像は，その性質上，数多くのページに貼り付けられることを前提としているため，その画像だけで完結している．また，同様の理由から周辺文字列を多くもたない．さらに，そのページの紹介をすることのみを考えて作成されたため，そのような画像が並べてレイアウトされていても画像同士にデザインの共通点は存在しないことがある．以上の三つの条件を基準として，本稿の準備実験では手動でカテゴライズしている．

8. む す び

モバイル端末による Web ページ閲覧において，Web ページ内の画像の役割を適切に把握することにより，小さいスクリーンサイズや貧弱な入力インタフェースによる制約を改善するさまざまなアプリケーションを提供できるものと考えられる．そこで，本研究では Web ページに含まれる画像の 11 のカテゴリを役割ごとに定義した．そして，定義したカテゴリへの自動分類を実現するために，40 のサイトから収集した 3901 の画像を手動で分類し，その結果から画像の 37 の特性を抽出した．それらの特性を用いて，最高で 83.1%の分類精度を実現した．さらに，画像の役割を考慮した新しい応用例について示した．

謝辞　本研究の一部は，文部科学省 21 世紀 COE プログラム「ネットワーク共生環境を築く情報技術の創出」の研究助成によるものである．ここに記して謝意を表す．

文　献

[1] C. R. Anderson, P. Domingos, and D. S. Weld, “Personalizing web sites for mobile users,” Proc. World Wide Web Conference (WWW’01), pp. 565–575, May 2001.

[2] T. W. Bickmore and B. N. Schilit, “Digestor: device-independent access to the world wide web,” Proc. World Wide Web Conference (WWW6), pp. 655–663, April 1997.

[3] O. de Bruijin, R. Spence, and M. Y. Chong, “RSVP browser: web browsing on small screen devices,” Personal and Ubiquitous Computing, vol. 6, issue 4, pp. 245–252, Sept. 2002.

[4] Y. Chen, W. Ma, and H. Zhang, “Detecting web page structure for adaptive viewing on small form factor devices,” Proc. World Wide Web Conference (WWW’03), pp. 225–233, May 2003.

[5] C. Frankel, M. J. Swain, and V. Athitsos, “WebSeer: an image search engine for the world wide web,” University of Chicago Technical Report, TR96-14, July 1996.

[6] J. Hu and A. Bagga, “Categorizing images in web document,”IEEE Trans. on Multimedia, vol. 11, issue 1, pp. 22–30, Jan./March 2004.

[7] W. Y. Ma, I. Bedner, G. Chang, A. Kuchinsky, and H. J. Zhang, “A framework for adaptive content delivery in heterogeneous network environments,” Proc. SPIE Multimedia Computing and Networking 2000, pp. 86–100, Jan. 2000.

[8] T. Maekawa, T. Hara, and S. Nishio, “A collaborative web browsing system for multiple mobile users,” Proc. IEEE Int’l Conf. on Pervasive Computing and Communications (PerCom2006), Mar. 2006 to appear.

[9] T. Maekawa, T. Hara, and S. Nishio: “Image classification for mobile web browsing,” Proc. Int’l World Wide Web Conference (WWW2006), May 2006 to appear.

[10] OPERA Software, “Opera for moblie,” http://www.opera.com/products/mobile/.

[11] S. Paek and J. R. Smith, “Detecting image purpose in worldwide web documents,” Proc. IS&T/SPIE Symp. Electronic Imaging: Science and Technology Document Recognition, Jan. 1998.

[12] S. J. Perantonis, B. Gatos, and V Maragos, “Text area identification in web images,” Proc. Hellenic Conference on AI, pp. 82–92, May 2004.

[13] WEKA Machine Learning Project, “WEKA 3,” http://www.cs.waikato.ac.nz/ ml/weka/.

[14] J. Zhou and D. Lopresti, “Extracting text from WWW images,” Proc. Int’l Conference on Document Analysis and Recognition (ICDAR), vol. 1, pp. 248–252, Aug. 1997.