DEWS2005 6-A-03

タグの深さに基づくコンテンツ間距離を利用したWebページの自動分割方式

服部　元† 松本　一則† 菅谷　史昭†

† 株式会社 KDDI 研究所
埼玉県上福岡市大原 2-1-15
E-mail: †gen@kddilabs.jp

あらまし インターネット上で公開されている Web ページのほとんどは PC で閲覧することを想定して作成されており，携帯端末で閲覧可能な Web ページはごく一部である．そのため，携帯端末を利用して一般の Web ページを閲覧する要求が高まっているが，携帯端末は画面が小さい等のユーザインタフェースの制限があるため，一般の Web ページを PC のように容易に閲覧できないことが課題となっている．すでに Web ページを小分割することでこの課題を解決する方式が提案されているが，この方式は HTML (Hyper Text Markup Language) の厳密なタグ構造を解析して分割点を導出する方式であるため，タグが省略される等の厳密に記述されていない HTML には対応できない問題がある．そこで本稿では，Web ページの HTML タグの深さに基づくコンテンツ間距離を利用して局所的な解析を可能とすることで，厳密な HTML の記述を要求条件としない Web ページの自動分割方式を提案する．
キーワード コンテンツ間距離，Web ページ自動分割，HTML 解析，WWW

Auto Web Page Distilling Scheme Using Content Distance Based on Depth of Tag Hierarchy

Gen HATTORI†, Kazunori MATSUMOTO†, and Fumiaki SUGAYA†

† KDDI R&D Laboratories Inc.
2-1-15 Ohara Kamifukuoka Saitama, Japan
E-mail: †gen@kddilabs.jp

Abstract Most of general Web pages in the Internet are produced for PC users, then mobile terminal users cannot retrieve enough information from Web. Although the demand of retrieving information from general Web pages using a mobile terminal is increasing, there are problems of restrictions of a user interface, which are the screen of the terminal is small, and so on. Therefore, although the scheme which divides a Web page small was examined, there was a problem that the conventional scheme cannot adapt to the HTML to which one or more tags were abbreviated, since it is a strict scheme which derives division points by analyzing the strict tag construction of the HTML. In this paper, we propose a new automatic Web page distilling scheme which does not require the condition of the strict HTML construction by using the distance between contents based on the depth of the HTML tags in Web pages.
Key words Contents Distance, Auto Distilling of Web Page, HTML Analysis, WWW

1. はじめに

近年，携帯電話や PDA 等の小型画面の携帯端末の普及が進んでいる．これらの多くは Web にアクセスする機能を持っており，乗り換え案内や天気予報，最新ニュース等の情報収集が可能である．しかしながら，Web で公開されている膨大な Web ページのほとんどは，画面のレイアウトや情報量を PC ユーザ向けに調整して作成されており，PC のユーザは比較的大きな画面と豊富なユーザインタフェースを活用して容易に情報を取捨選択しながら閲覧することができる．一方，携帯端末向けの Web ページの数は PC 向けと比較してはるかに少なく，また携帯端末の場合は狭小な画面と自由度の低い入力デバイス等，ユーザインタフェースの制限があることから携帯端末向けの Web ページは情報量が少ない．そのため，携帯端末から情

報量が豊富な一般の Web ページを効率的に閲覧する要求が高まっている．このときユーザが必要とする情報は Web ページの一部のみである場合も多いことから [1] [2]，ユーザが情報を容易に選択して閲覧できることが重要である．

PC 向けの Web ページを携帯端末で閲覧するための従来方式として，端末の画面サイズに合わせて Web ページのレイアウトを変更する方式がある [3]．ただしこの方式では縦長の Web ページにレイアウトされるため，ユーザが見たい情報を探し出すために時間を要する場合もあり，携帯端末で容易な情報の選択と閲覧を実現しているとはいえない．他の従来方式として，DOM(Document Object Model) パーサを利用して対象となる Web ページ全体のタグの階層構造を分析し，階層構造に基づき小さい Web ページに分割する方式がある．分割した情報を 1 単位としてユーザが取捨選択できるように工夫することで，携帯端末のような小さい画面でも容易に閲覧する環境を実現している．しかしながら，タグが省略されている等の正則でない HTML (Hyper Text Markup Language) には対応できない問題があり，また分割時に必要なパラメタの設定方法に関する検討はしていない．

そこで本稿では，Web ページの HTML タグの深さを利用した情報の距離に基づき，Web ページを分割する方式を提案する．本方式では HTML のタグ構造を全体的に解析するのではなく局所的に解析することが可能であり，HTML の正則でない記述の影響が部分的である特徴を持つ．また評価実験を行い，統計的な手法により分割時に必要なパラメタを動的に決定する方式について検討する．

以降，2 章では，携帯端末を利用して Web ブラウジングを行うシステムの機能用件と関連研究について述べる．3 章では，タグの深さに基づく Web ページの分割方式を提案する．4 章では，評価実験の結果と分割のパラメタの動的設定方法を述べる．最後に，5 章でまとめを述べる．

2. Web ページの分割によるブラウジングと従来研究

2.1 携帯電話による Web ブラウジングの課題

現在，日本における携帯電話の契約台数は 2004 年 9 月時点で 8300 万台を超え，人口普及率は 66%を超えている．このうちのほとんどの携帯電話は Web ブラウザ機能を搭載しており，利用可能な HTML タグに制限があるものの Web 上の情報を閲覧可能である．しかしながら，Web ページのほとんどは PC 向けに作成されたものであり，またそれらの情報量は携帯電話向けの Web ページと比較して豊富である．そのため，携帯電話から一般の Web ページを閲覧することが望まれるが，PC と比較して画面が極端に小さいことから 1 画面で表示可能な情報量が限られることや，入力デバイスの自由度が低い等の課題がある．これは携帯電話だけでなく PDA(Personal Digital Assistance) 端末のような携帯端末についても同様である．情報量をコントロールすることにより携帯端末でもユーザが情報を容易に取捨選択できる等，効率的な Web 情報の閲覧を可能にするシステムが必要である．

このようなシステムの機能要件として，次の 2 点が挙げられる．

(1) 1 ページの情報量を小さくすること
一般の Web ページは情報量が豊富であるがユーザインタフェースが貧弱で自由度が低いため，一度に表示する情報量を制限する必要がある．また副次的な利点として，閲覧する情報を選択する手段があれば無駄な通信を削減することができる．

(2) あらゆる HTML に対応できること
閲覧の対象となる Web ページは，巧みにタグを構成して複雑なレイアウトを実現しているものや，簡易なリスト形式のものなど多種にわたる．また HTML は XML (Extended Markup Language) のような厳密な DTD (Document Type Definition) ではないため，タグの省略や未知のタグの挿入等が多い．

2.2 関 連 研 究

従来の方式は大きく 2 通りに分類できる．1 つは Web ページのレイアウトを携帯端末向けに変更する方式であり，もう 1 つは Web ページを関連するコンテンツの小さな集合であるコンテンツオブジェクトに分割する方式である．ここでコンテンツとは，画像やハイパーリンク，テキストのような Web ブラウザ上で可視的な情報を指す．

前者の方式として，携帯端末の画面の狭い横幅に合わせてブラウザが Web ページのレイアウトを縦長に変更する方式がある [3]．ただしこの方式では，ユーザの見たい情報が下のほうに配置された場合にはスクロールが必要となるため，探し出して閲覧するまでに時間がかかる等，容易に情報の取捨選択ができない問題がある．表形式の情報を携帯端末向けにレイアウト変更を行う方式があるが [4] [5]，これらについてもユーザが情報を容易に取捨選択する助けにはならないことや，表形式以外の情報には適用できない問題がある．また，Web のインデックスページを解析して携帯端末向けのメニューを生成する方式や [6] [7]，Web ページ中のリンクの重要度に基づく情報のメニューリストを生成する方式がある [8] [9]．これらの方式では，ユーザが情報を取捨選択するための補助にはなるが，インデックスのページの存在が必須であることや，選択したリンク先のページを閲覧するためには結局広い画面が必要となる問題がある．

後者の方式として，HTML の階層構造を解析して Web ページを小分割し，同類のコンテンツの集合 (以下，コンテンツオブジェクトと呼ぶ) を生成する方式がある [10] [11]．この方式では，ユーザがコンテンツオブジェクト毎に情報を取捨選択可能であり，大量の情報を持つ Web ページでも容易に閲覧することができる特徴がある．具体的には，まず HTML のタグ構造を DOM (Document Object Model) パーサで解析してツリーを生成し，次にツリーの階層構造を利用して上位の階層から下位の階層へ再帰的に分割していく．ただし，HTML 全体を解析してツリーを生成する必要があるため，HTML の DTD に従い HTML が厳密に記述されていることが必要条件となる．

しかしながら一般の Web ページは HTML の終了タグが省略されていることや，DTD には定義されていない不明なタグが挿入されていることも多いため，適用できない Web ページが多いことが課題となっている．ユーザのアクセス履歴等から嗜好情報を抽出し，単語の出現頻度をカウントしてユーザのプロファイルを作成することで，ユーザの嗜好に合わない部分を削除して小さな HTML を再構成する方式がある [12]．この方式は，ユーザが情報を取捨選択する点で機能要件に合致するが，再構成した HTML が携帯端末で容易に閲覧可能であるくらいに十分に小さくなるとは限らない問題がある．また，複数の携帯端末の画面を利用して 1 つの Web ページを閲覧することを目的とした方式は [13]，文書を重み付きの完全グラフに変換して小分割することを特徴としているが，XML 文書を対象としており正則でない HTML には適用できない問題がある．

そこで本稿では，Web ページを携帯端末向けに小分割する後者の分類に属する方式として，正則でない HTML にも対応可能な Web ページ分割方式の検討を目的とする．

3. 提 案 方 式

3.1 方式の概要

HTML タグが省略されている等の厳密に記述されていない HTML にも対応可能とするため，HTML を局所的に解析してコンテンツ間の距離を算出し，複数のコンテンツオブジェクトに分割する方式を提案する．この方式は，必要なタグの省略や未知タグの挿入等の正則でない HTML に対する耐性が高い方式であり，文献 [13] で検討されているような Web ページ全体のツリー構造が正しいことを前提としてコンテンツ間の距離を求めている方式ではない．

本稿ではコンテンツを，(ア)HTML 中の `<a>` タグで指定されているアンカー，(イ)`<img>` タグで指定されている画像，(ウ)テキスト，の 3 種類と定義する．

提案方式を利用したアプリケーションイメージについて以下に述べる．なお，本方式はクライアント側，あるいは Web のプロキシサーバのいずれかに実装する．

(1) 図 1 に示すような Web ページを携帯端末で閲覧しようとした場合，コンテンツの量が多く 1 画面では表示できない．この例では，3 つのコンテンツオブジェクトに分割することが望ましい．

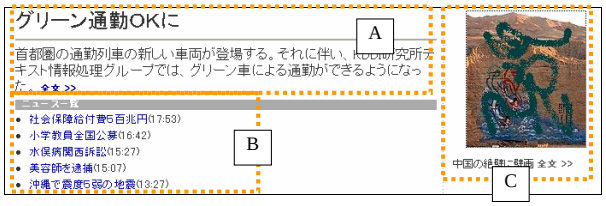

図 1 対象とする Web ページの例
Fig. 1 Example of Web page

(2) 図 1 の HTML ソースを図 2 に示す．コンテンツ間の距離が大きい部分をコンテンツオブジェクトに分割する分割点であるとする．図 2 においてコンテンツオブジェクト間距離として示している部分が分割点である．

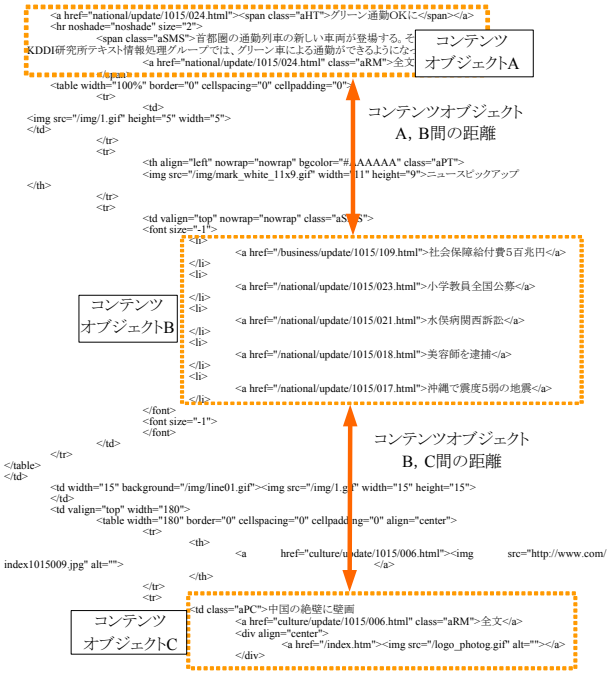

図 2 コンテンツ間距離
Fig. 2 Content Distances

(3) コンテンツオブジェクト毎に HTML を分割して小さな HTML を再構成し，携帯電話で表示する．表示例を図 3 に示す．

図 3 携帯端末の画面イメージ
Fig. 3 Display Image of Mobile Terminal

コンテンツ間距離を利用した Web ページの分割方式は次の (A)〜(C) の 3 つのステップからなる．ここで，ステップ (A) と (B) は SAX (Simple API for XML) パーサが出力順に処理ができるため，高速な処理が可能である．

(A) コンテンツの抽出

HTML から 3 種類のコンテンツを抽出する．フォントタグ (`<font>`) やその他レイアウト用のタグ (`<table>`) 等，コンテンツの修飾や配置のためのタグは携帯端末上で正しく再生できない場合も多いため，ここで削除する．`<table>` タグを利用した表形式の情報については配置が重要となるが，項目行を自動識別してレイアウトを最適化する従来方式 [4] [5] を利用すれば対応可能である．

(B) コンテンツ間距離の算出

隣接するコンテンツの間に挿入されているタグの数とその深さに基づき，コンテンツ間距離を算出する．タグの深さとは，HTML の最初のタグ (<html>) の深さを 0 とし，開始タグであれば直前のタグの深さに +1 し，終了タグがあれば −1 として算出した値である．

(C) コンテンツオブジェクトへの分割

コンテンツ間距離の大小を比較して分割点を再帰的に導出し，Web ページを複数のコンテンツオブジェクトに分割する．

各ステップの詳細について次の各節で述べる．

3.2 コンテンツの抽出

HTML を先頭から順にタグを検索し，HTML 中の <a> タグで指定されているアンカー，<img> タグで指定されている画像，およびテキストを抽出する．

(ア) アンカーの抽出

開始タグ <a> と終了タグ で囲まれた部分を 1 コンテンツとする．ここには <img> タグで指定した画像とテキストを含む場合もあるが，まとめて 1 つのコンテンツとする．

(イ) 画像の抽出

(ア) に含まれる <img> タグを除く画像をコンテンツとする．ただし，<img> タグは本来の画像として表示する以外にも，小さなスペースを作るための利用や記事のセパレータとしての利用が多い．このような可視的でない画像をコンテンツと見なさないため，<img> タグの画像をコンテンツとするための条件を以下の「(a) または (b)」とする．

(a) 画像のサイズが縦 x (pixels)，かつ横 y (pixels) 以上であること

(b) <img> タグの属性値”alt”で指定された代替テキストが記述されていること

(ウ) テキストの抽出

(ア) に含まれるテキストを除き，すべてのタグに挟まれているテキストをコンテンツとする．

3.3 コンテンツ間距離の算出

以下の方式によりコンテンツ間距離を算出する．コンテンツ間距離とは，HTML ソース上で隣接するコンテンツの近接度を表す．言語を限定する場合は自然言語処理により近接度を求める方法も考えられることができる．しかしながら，WWW は文字通り World Wide な環境であるため，言語に依存しない方法が望ましい．

一般の Web ページは HTML のタグを利用して見た目のレイアウトを制御しており，近接しないコンテンツ間では，HTML の構造が大きく変化している．HTML の構造が大きく変化する部分はタグの深さが大きく変化する部分であることから，(1) コンテンツ間にあるタグの数と (2) それらのタグの深さ変化の度合いが，コンテンツ間距離の指標となる．そこで，図 4 に示すようにコンテンツ間にあるタグとコンテンツの深さの大きい方の値で囲まれた部分の面積を算出し，これをコンテンツ間距離と定義する．

コンテンツ A，B 間の距離 S_{ab} は図 5 に示すフローにより算出する．ここで，$y = f(x)$ はタグの出現順序 (x) に対する深さ (y) を表す．また，S_a と S_b を算出してその最大値を選択しているのは，図 4 に示すような $y = f(x)$ が谷の場合だけでなく，$y = f(x)$ が山の場合についても対応するためである．

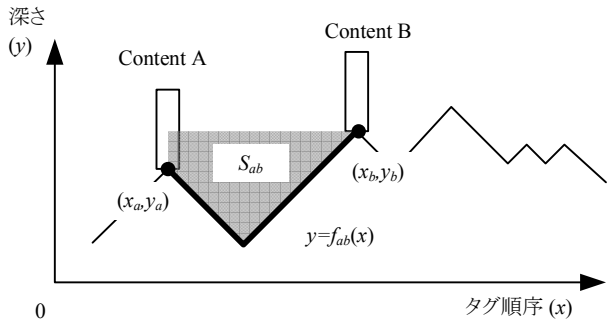

図 4 タグの深さとコンテンツ間距離
Fig. 4 Tag Depth and Content Distance

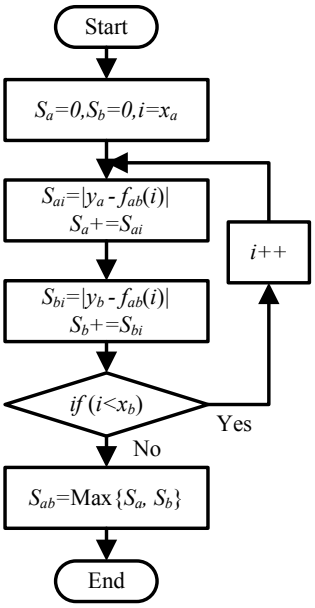

図 5 コンテンツ間距離の算出手順
Fig. 5 Calculation of Content Distance

3.4 コンテンツオブジェクトへの分割

算出したコンテンツ間距離に基づき分割点を決定し，Web ページをコンテンツオブジェクトに分割する．基本的にはコンテンツ間距離が大きい点を分割点とする．手順を図 6 に示し，手順の概要を以下に述べる．

(1) Web ページ全体を 1 つのコンテンツオブジェクト ($ObjectID$ =root) とする．

(2) コンテンツオブジェクト内のコンテンツ間距離の最大値 ($S_{\max}$) が，コンテンツオブジェクト内のコンテンツ間距離の平均値 (S_{average}) の N_1 倍以上であれば，$S_{\max}$ の位置を分割点とする．

(3) (2) が真でない場合，S_{average} の N_2 倍以上かつ分割した場合のコンテンツ数の最小値 ($C_{\mathrm{distilled}}$) が N_3 以上であれば $S_{\max}$ の位置を分割点とする．

(4) 分割した場合は分割結果の左のコンテンツオブジェクトに移動し (2) へ．そうでなければ (5) へ．

(5) 左のコンテンツオブジェクトであった場合は，右のコンテンツオブジェクトに移動し，(2) へ．

(6) 右のコンテンツオブジェクトであり，かつ $ObjectID$ <> ($root$) の場合は，親コンテンツオブジェクトに移動し，(5) へ．

(7) 終了．

```
[Distilling Algorithm]

#define LEFT = 0;
#define RIGHT = 1;
i = 0; objectID = (root); x = LEFT;
D_{objectID,x} = (Whole of HTML);
Distill(D_{objectID,x}){
  S_max = max{S_{i,i+1} : i ∈ D_{objectID,x}};
  S_average = average{S_{i,i+1} : i ∈ D_{objectID,x}};
  C_distilled = (Minimum Number of Tags in a Group
                 when it is Distilled)
  if(S_max>N1 * S_average){
    (Distill at the S_max);
    ChildObjectID = (objectID of created object by distilling);
    Distill(D_{ChildObjectID,LEFT});
  };
  else if(S_max>N2 * S_average & C_distilled>N3){
    (Distill at the S_max);
    ChildObjectID = (objectID of created object by distilling);
    Distill(D_{ChildObjectID,LEFT});
  };
  Label(*)
  if(x = LEFT){
    Distill(D_{objectID,RIGHT});
  };
  else if(x == RIGHT & objectID ≠ (root)){
    ParentObjectID = (objectID of parent object of this object);
    objectID = ParentObjectID;
    Go to Label(*)
  };
  else{
    END;
  };
};
```

図 6 分割アルゴリズム

Fig. 6 Distilling Algorithm

4. 実験による評価と考察

4.1 正則でない HTML の割合

予備実験として，W3C の Web サイト [14] と，文献 [10] の評価実験で使用している Web ページのうち現時点でアクセス可能な Web ページ (39 件) に対し，Apache XML Project が開発している DOM パーサ (Xerces) [15] でツリーの構築を試みた．その結果，W3C を除く全てのページでツリー構築エラーとなり正則でない HTML であると判明した．正則でない HTML のパターンには以下のような場合があった．

(1) 終了タグが必要であるにも関わらず記述していない

(2) タグの入れ子構造がクロスしている

(3) タグ名のスペルミス，不明なタグ

但し，処理時間が余分にかかる問題はあるが，ある程度の不足やミスであれば HTML の修復ツールを利用して自動的に修復可能である．そこで HTML Tidy Library Project が開発した HTML の修復ツールである TIDY [16] を利用した場合について，TIDY でも修復不能であった致命的なエラーのある HTML について検証した．対象とした Web ページは，Yahoo!Japan のカテゴリ情報 [17] から，ニュース，料理，自然，エンタテイメントのカテゴリに属する Web ページからそれぞれ 70 ページ程度を任意に選択したものとした．結果を表 1 に示す．

表 1 エラーのある Web ページ数と割合

Table 1 Number and Ratio of Error Web Pages

カテゴリ	Web ページ数	エラー数	割合 (%)
ニュース	106	9	8.5
料理	70	19	27.1
自然	71	14	20.0
エンタテイメント	71	7	9.9

ニュース情報を提供している Web ページは 8.5%程度であり，エンターテイメントのカテゴリでは約 10%であった．これにに対し，料理や自然のカテゴリではそれぞれ 27.1%，20.0%となり，エラーのある Web ページの割合がかなり高い結果となった．カテゴリによりばらつきはあるが，10%〜30%程度は含まれていると考えられる．これらの Web ページでは HTML 修復ツールを利用してもツリーを正しく作成することはできない．

4.2 有効性の評価

タグの深さを利用した Web ページ分割方式を実装し，有効性の評価を行う．ニュースを提供している 106 のサイトを対象とし，各 Web サイトについて 3. 4 節に記述した分割パラメタ N_1 および N_2 を変化させてそれらの最適値を求めた．なお $N_3 = 2$ とした．最適値は人の目で見て最も適切な位置でコンテンツオブジェクトに分割できた値とした．結果を図 7 に示す．

図 7 において縦方向が N_1 の設定値，横方向が N_2 の設定値であり，該当する Web ページ数を各セルに記述している．$N_1 = 6, N_2 = 4$ の場合が最も多い 54 となり，全 106 サイトのうちの約 51%を占めた．よって，対象としたニュースサイトに対しては，$N_1 = 6, N_2 = 4$ と設定することにより約 51%の Web サイトに対して最適な分割が可能であった．

4.3 分割パラメタの動的設定方法

より多くの Web ページに適応可能とするため，パラメタの動的設定方法について検討する．例えば図 7 において 2 番目に

N_1 \ N_2	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
0																	
1																	
2																	
3			6														
4			2	16													
5				1	3												
6				2	54	1											
7						1	3										
8					1		5										
9								2									
10									2								
11								1		1	3						
12																	
13																	
14																	
15																	
16											1						

図 7　N_1, N_2 の最適値の分布
Fig. 7　Distribution of Optimum Value for N_1 and N_2

多いのは $N_1 = 4, N_2 = 3$ の場合の 16 であり，全体の約 15%であった．この分割パラメタが最適である Web ページに対応することで合計約 66%に対応可能となる．そこで $N_1 = 6$ と $N_1 = 4$ の場合について Web ページの特徴を比較した．結果を表 2 に示す．また，表 2 の項目名の概要は次のとおりである．

平均：タグの深さの平均値
標準偏差：タグの深さの標準偏差の平均値
コンテンツ数：コンテンツ数の平均値 (個)
ファイルサイズ：HTML ファイルサイズの平均値 (Kbytes)

表 2　$N_1 = 6$ と $N_1 = 4$ における Web ページの特徴比較
Table 2　Comparison of Features of $N_1 = 6$ and $N_1 = 4$

	平均	標準偏差	コンテンツ数	ファイルサイズ
$N_1 = 6$	9.5	23.7	1488.5	38.2
$N_1 = 4$	4.7	10.9	1026.4	36.7

表 2 より，ファイルサイズがほぼ同等であるにも関わらず，いずれの数値も $N_1 = 6$ の方が大きな値となった．また図 7 における N_1 と N_2 のピアソンの積率相関係数を算出すると 0.94 となり，強い相関が確認できる．このことから，これらの値を利用して N_1 および N_2 を動的に決定可能であると考えられる．

4.4　システム実現例

提案方式を実装した携帯電話向け Web 閲覧システムを図 8 に示す．本システムは Web ページ自動分割部とスコア算出部からなる．ここでスコア算出部は，ユーザがコンテンツを効率的に選択するための情報を付与する機能であり，これにより分割することに加えてより効率的な Web ページの閲覧を実現する．

Web ページ自動分割部は，(1) ユーザからの要求を受け付ける要求受付 I/F，(2) 要求に応じて対象 Web ページの HTML を取得する HTML 取得 I/F，(3) 取得した HTML を提案方式により分割する分割処理機能，(4) 分割処理結果を再構成して新たな HTML を生成する HTML 再構成機能の 4 つから成る．スコア算出部は，(1) カテゴリ情報を利用して分割処理結果の各コンテンツオブジェクトのカテゴリを解析するカテゴリ解析機能，(2) ユーザ情報を利用して各コンテンツオブジェクトについてユーザが嗜好するカテゴリとの一致度 (以下，スコアと呼ぶ) を算出するスコア算出機能の 2 つから成る．なお，カテゴリの解析方法の具体例を付録 1. に，スコアの算出方法の具体例を付録 2. にそれぞれ記述した．

図 9 の Web ページ (`http://newstopics.dion.ne.jp/pubnews/photonews/`) に対して本システムを適用すると，例えば破線で囲まれた部分をコンテンツオブジェクトとして認識し，図 10 に示すように携帯電話で閲覧することができた．また，ユーザの嗜好情報に基づくスコアをコンテンツの傍らに付与することにより情報の選択が容易になり，効率的な Web の情報閲覧を実現できたといえる．

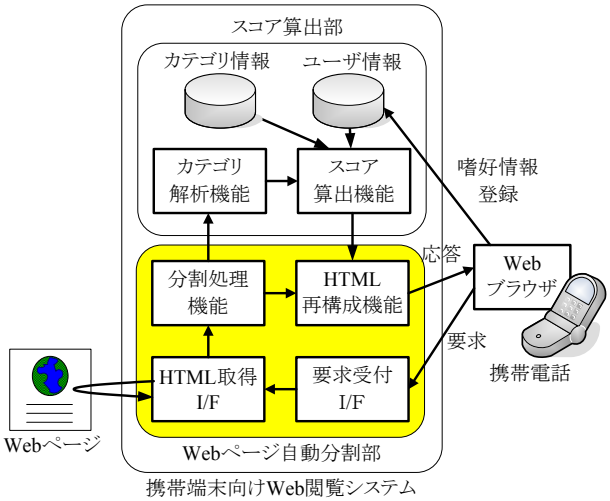

図 8　携帯電話向け Web 情報閲覧システム
Fig. 8　Web Information Display System for Mobile Phone

図 9　Web ページ例
Fig. 9　Example of Web Page

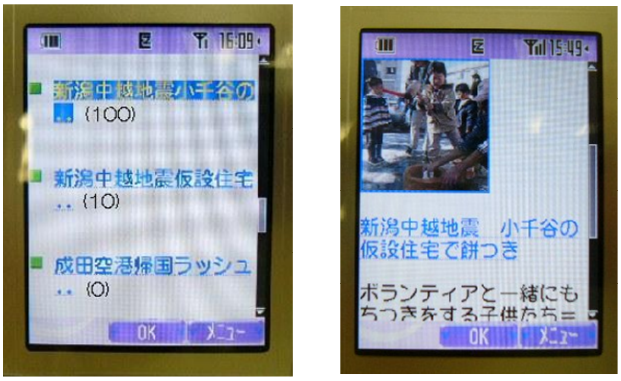

図 10　コンテンツオブジェクトリスト (左) とコンテンツオブジェクトの表示 (右)

Fig. 10　List and Display of Contents Objects

5. おわりに

本稿では，一般の Web ページを分割して画面の小さい携帯端末で閲覧するための新たな方式を提案した．提案方式は，コンテンツ間の距離を局所的に解析して算出し，その大小を区切りのパラメタとすることが特徴であり，これにより従来方式では対応できなかった正則でない HTML にも対応可能とした．評価実験を行い，正則でない HTML がカテゴリに応じて 10%〜30%程度存在することを示し，提案方式が約 51%の Web サイトに対して最適な分割が可能であることを示した．さらに Web ページのいくつかの統計的な特徴から分割のパラメタを推定する方法により，分割可能な Web ページの割合を増やすことができる見通しを得た．また，提案方式を適用した Web 閲覧システムを構築し，そこにユーザが嗜好するカテゴリ情報に基づくスコアを付与することにより，効率的な Web の閲覧環境を携帯端末上に実現した．今後の課題として，分割パラメタの推定方式の検討ならびに既存の方式との性能比較を行うことが挙げられる．

謝辞　日頃ご指導頂く KDDI 研究所浅見代表取締役所長，および中島執行役員に深く感謝致します．

文　献

[1] 服部元，松本一則，菅谷史昭，”表形式情報集約のための連想性の高いオブジェクトラベルの自動抽出方式,” 3rd Joint Agent Workshops & Symposium, 2004.

[2] 山田誠二, 中井有紀, ”対話的分類学習による Web ページの部分更新モニタリング,” 人工知能学会論文誌 17 巻 5 号, 2002.

[3] Small Screen Rendering (Opera Software ASA), http://www.opera.com/products/mobile/smallscreen/.

[4] Yu Chen, Wei-Ying Ma, Hong-Jiang Zhang, ”Improving Web Browsing on Small Devices Based on Table Classification,” The Twelfth International World Wide Web Conference, 20-24, May 2003.

[5] 増田英孝, 塚本修一, 安富大輔, 中川裕志, ”HTML の表形式データの構造認識と携帯端末表示への応用,” 情報処理学会論文誌: データベース, Vol.44, No.12, 2003.

[6] George Buchanan, Sarah Farrant, Matt Jones, and Harold Thimbleby, ”Improving Mobile Internet Usability,” Proc. 10th International World Wide Web Conference, Hong Kong, China, 2001.

[7] Jones, M., Buchanan, G., Thimbleby, H., ”Sorting out Searching on Small Screen Devices,” Conference on Mobile HCI, 2002.

[8] O. Buyukkokten, H. Garcia-Molina, and A. Paepcke, ”Seeing the whole in parts: Text summarization for web browsing on handheld devices,” Proc. 10th International World Wide Web Conference, 2001.

[9] O. Buyukkokten, H. Garcia-Molina, A. Paepcke, and T. Winograd, ”Power browser: Efficient web browsing for PDAs,” Proc. Human-Computer Interaction Conference 2000, 2000.

[10] Y. Chen, W. Ma, and H. Zhang, ”Detecting web page structure for adaptive viewing on small form factor devices,” in Proc. World Wide Web Conference 2003, 2003.

[11] 前川卓也，原隆浩，西尾章治郎，”複数のモバイルユーザのための Web ページ分割を用いた協調 Web ブラウジングシステム,” 情報処理学会モバイルコンピューティングとユビキタス通信研究会, Vol.2004, No.114 (MBL 31), 2004.

[12] Anderson, C.R., Domingos, P., and Weld, D.S., ”Personalizing web sites for mobile users,” Proc. 10th International World Wide Web Conference, 2001.

[13] 前川卓也，上向俊晃，原隆浩，西尾章治郎，”複数のモバイル端末による協調ブラウジングのための木構造型コンテンツ記述方式と分割方式,” 情報処理学会論文誌: データベース, Vol.45, No.SIG7(TOD 22), 2004.

[14] World Wide Web Consortium, http://www.w3.org/.

[15] Apache XML project Xerces2 Java Perser, http://xml.apache.org/xerces2-j/.

[16] HTML Tidy Library Project, http://tidy.sourceforge.net/.

[17] Yahoo! Japan カテゴリ, http://www.yahoo.co.jp/.

[18] 松本裕治，北内啓，山下達雄，平野善隆，松田寛, 高岡一馬, 浅原 正幸, ”日本語形態素解析システム『茶筌』 version 2.2.1 使用説明書,” 2000.

付　録

1. カテゴリ解析について

一般的な検索サイトは Web サイトを内容に適合するカテゴリに分類しており，各カテゴリには数箇所から数百箇所の Web サイトを登録している．この分類作業は人手で行っており，分類の信頼性は高いといえる．そこで，スコアの算出方法として，まず検索サイトを利用してカテゴリ階層の自動収集と登録 Web サイトの分析を行い各カテゴリの特徴量を抽出し，この特徴量を利用してコンテンツ毎に内容が類似するカテゴリを導出してユーザが嗜好するカテゴリとの一致度の総計をスコアとして算出する．具体的には，各カテゴリに属する Web ページにおける名詞の出現回数をカウントし，その上位にランクされる名詞をそのカテゴリの特徴を現す語として蓄積する処理を行う．ここでは (1) カテゴリ情報の抽出処理と (2) カテゴリ情報の生成処理について具体例を述べる．

1.1　カテゴリ情報の抽出処理

トップカテゴリから 4 階層下のカテゴリ情報までを抽出する．Yahoo! Japan [17] の場合を例に，カテゴリ情報の抽出手順について以下に述べる．

(ア)　<!---カテゴリ-->と<!---/カテゴリ--> の間に含まれる <a> タグの URL を 1 階層下のカテゴリとする．

(イ)　(ア) の URL が 0 個であれば最下層と判定する．

(ウ)　<!---登録サイト-->と<!---/登録サイト--> の間の <a> タグは，そのカテゴリに対するカテゴリ情報の生成処理 (1.2 節) の対象となる Web ページとする．

(エ) カテゴリを一意に識別可能な ID(以下，カテゴリ ID と呼ぶ) を設定する．カテゴリ ID とは例えば 1.2.10.3.110 のような".”をセパレータとして階層と順序を数値で表す．また，カテゴリ名として HTML からタイトルとなるテキストを抽出する．カテゴリの詳細へのリンクを表す <a> タグ内のテキスト，あるいはリンク先の HTML のタイトルが対象となり，具体的には「エンターテイメント」や「占い」のようなカテゴリ名を抽出する．

1.2 カテゴリ情報の生成処理

カテゴリ毎に登録されている複数の URL の HTML に含まれる名詞を抽出してその出現割合を算出し，それをカテゴリ情報とし，データベース (以下，カテゴリ DB と呼ぶ) に格納する．以下に算出手順を示す．

(ア) 付録 1.1 節に該当する URL の最大 R_1 件の HTML ソースを取得する．($R_1 = 20$ 程度とする)

(イ) 茶筌 [18] を利用して形態素解析を行い，HTML に含まれる名詞のうち「名詞・一般」と「名詞・固有名詞」と判定した語のみを収集する．

(ウ) 名詞の出現数のランキングを算出する．(イ) の結果をカテゴリ毎に総計し，出現数上位 N_2 位の名詞をそのカテゴリのキーワードとする．($R_2 = 10$ 程度とする)

(エ) tf・idf 法に基づく式 A·1 に従い，各キーワードの出現回数を R_2 種類のキーワードの出現回数の総計で割った値をそのキーワードの重みとする．

$$W_i = \frac{E_i}{\sum_{i=1}^{R_2} E_i} \times \left(\log \frac{N}{N_i} + 1\right) \qquad \text{(A·1)}$$

W_i : i 番目のキーワードの重み
E_i : i 番目のキーワードの出現数
N : カテゴリ総数
N_i : i 番目のキーワードが出現するカテゴリ数

(オ) カテゴリ DB 内のデータと比較を行う．ここで初期状態として，カテゴリ DB に前回のカテゴリ解析処理データが記録されているとする．

(1) 前回の処理データと今回の処理データを比較する．新たなカテゴリがあれば追加する．
(2) 消滅したカテゴリについては残したままとする．
(3) 前回と今回の処理データをカテゴリ毎に比較し，新たなキーワードが入っていれば最新キーワードとしてフラグを立てる．最新キーワードは重みを R_3 倍にする．(ここで $R_3 = 2$ 程度とする)
(4) 前回の処理データについて最新キーワードのフラグを全て削除し，定番キーワードとする．最新キーワードの重みを $1/R_3$ にする．
(5) カテゴリ情報を保存する．保存する内容は，カテゴリ ID，カテゴリ名，カテゴリの URL，キーワードベクトル (キーワード，重み，最新キーワードのフラグ)，登録日時とする．

2. スコアの算出

グループ内のコンテンツ毎に以下の項目に基づくスコアを算出する．算出手順の具体例を以下に述べる．

(ア) コンテンツが属するカテゴリを決定する

(1) 付録 1.2 におけるカテゴリのキーワードの抽出の場合と同様に茶筌を利用して形態素解析を行い，コンテンツのキーワードを抽出する．重みについても同様に算出する．
(2) コンテンツがテキストの場合は，コンテンツのテキストに含まれる名詞の上位 R_4 位のキーワードベクトルを生成する ($R_4 = 10$ 程度とする)．コンテンツがリンクの場合は，リンク先の HTML に含まれる名詞の上位 R_4 位をキーワードとする．また，コンテンツが画像のみの場合は <img> タグの alt 属性のテキストに含まれる名詞をキーワードとする．
(3) カテゴリのキーワードベクトルとコンテンツのキーワードベクトルの重みの内積を算出し，これを一致度とする．
(4) 一致度が上位 R_5 位のカテゴリをコンテンツの属するカテゴリとする ($R_5 = 3$ 程度とする)．

(イ) コンテンツとユーザの嗜好カテゴリとの一致度 (スコア) を算出する

(1) 全てのコンテンツについて，ユーザが嗜好するカテゴリとコンテンツが属するカテゴリが一致する場合は，その重みをコンテンツのスコアとする．
(2) ユーザが嗜好するカテゴリが複数ある場合はすべてのカテゴリについて (1) を実施し，スコアはその総和とする．