

# モバイル端末のための Web ページ自動スクロール方式

## A Web Page Automatic Scrolling Method for Mobile Devices

前川 卓也<sup>▼</sup> 原 隆浩<sup>▲</sup> 西尾 章治郎<sup>▲</sup>

Takuya MAEKAWA Takahiro HARA  
Shojiro NISHIO

携帯電話などの小型ディスプレイを備えた端末では、デスクトップPCなどの大画面を想定して作成されたコンテンツの閲覧は困難である。例えば、携帯電話を用いてデスクトップPC用に作成されたWebページを閲覧する場合、Webページが必要とする表示サイズは、携帯電話のディスプレイサイズに比べて圧倒的に大きいので、携帯電話が備える貧弱な入力インタフェースでは快適に閲覧することができない。そこで本論文では、Webページ内を自動でナビゲート(スクロール)することにより、ユーザによる煩雑な操作なしにページを閲覧する方式を提案する。提案方式では、Webページの構造やWebページに含まれる情報量などを考慮してスクロールの順序や速度を決定する。またユーザの操作などに応じて、スクロールの速度やページの表示倍率を変更する。

It is difficult for users equipped with small screen devices such as cellular phones to browse contents designed for desktop PCs with large screens. For example, a user equipped with a cellular phone cannot comfortably browse a Web page designed for desktop PCs because not only the size of the Web page is much larger than that of the cellular phone's display but also the cellular phone has a poor input interface. In this paper, we propose a method to browse a Web page without users' complicated operations by automatically navigating in the Web page. Our proposed method determines the order of scrolling contents and the scrolling speed considering the structure of the Web page and amount of information included in the Web page. The scrolling speed and zoom ratio can also be regulated according to users' operations.

### 1. まえがき

モバイル端末の小型化や通信技術の発展により、携帯電話などを用いたWebブラウジングが一般的に行われるようになってきている。また、ディスプレイの高精細化や通信速度の高速化により、デスクトップPCでの閲覧を想定して作成されたWebページを携帯電話で閲覧できるブラウザも登場している[7]。

しかし、携帯電話のディスプレイサイズは通常のWebページのサイズに比べて圧倒的に小さく、大画面での閲覧を前

提としたWebページの閲覧は困難である。Chenら[2]は、Webページの縮小画像をオーバービューとして利用しており、ユーザはオーバービューから閲覧したい部分を拡大できる。この手法は、ユーザがデスクトップPCで普段からよく閲覧しており、その構造を熟知しているページにおいては非常に有効である。しかし、そうでないページでは、オーバービュー内の小さな文字から情報をほとんど得ることができない。

ここで、ユーザが携帯電話を用いて通常の大きなWebページを閲覧するとき、縦と横の2次元でページをスクロールする必要がある。しかし、携帯電話の入力インタフェースは依然貧弱なままであり、ほとんどがダイヤル用ボタンと上下左右の4方向ボタンしか備えていないため、ユーザがページ内の情報をくまなく見るためには非常に手間がかかる。灘本ら[6]は、Webページからカラーセルと呼ばれる画像とテキストからなるコンポーネントの集合を抽出し、カラーセルを順番に提示する手法を提案している。このとき画像はディスプレイに表示し、テキストは音声読み上げを行っているため、ユーザの操作が軽減されている。Opera for Mobile[7]などのブラウザでは、ディスプレイの幅に合わせてページのレイアウトを再構成することにより、1次元の縦方向スクロールのみでページを閲覧することができる。しかし、複雑な構造をもつテーブルなどの再構成は困難である。また、一般的なWebページの両端にはそのサイトのメニューなどが配置されていることをユーザは経験的に知っているが、レイアウトが変更された場合、ユーザはこのような経験に基づいた閲覧ができなくなる。さらにある文章において、その文章の左に配置されている図に対して、“左図を参照”などのような位置を指定した参照があるとき、そのレイアウトを再構成するのは適切でない。

以上に述べたように、ユーザが大きなWebページをモバイル端末で閲覧する際には、できるだけWebページのレイアウトを変更せずに、かつユーザによる最低限の操作で、ユーザにWebページ内の情報をできるだけ多く提供することが重要である。そこで本研究では、レイアウトを変更していないWebページ内を自動でスクロールさせることにより、ユーザにWebページ内をナビゲート(案内)する手法を提案し、その手法を用いたアプリケーションの実装を行う。以下では、ページ内をナビゲートすることをページツーリング(page touring)と呼び、ページ内のナビゲート自体をツアー(tour)と呼ぶ。ツアーの内容は、ページ構成やページ内の情報量によって動的に決定し、ユーザが操作して速度などを変更することも可能である。

### 2. ページツーリングの概要

ページツーリングでは、Webページ内を2次元で自動スクロールすることにより、Webページのレイアウトを変更せずに、ユーザにページ内の情報を提供できる。図1に、大阪大学大学院情報科学研究科のWebページをツーリングしている概念図を示す。図1内の矢印は携帯電話の画面の中心が辿るツアーのパスとなっており、この例ではヘッダ部を通った後、左サイドメニューを通り、最後にメインのコンテンツであるボディ部を通っている。このとき、ボディ部は携帯電話の画面サイズに比べて幅も高さも大きいので、ツアーはボディ内を折り返して通過している。このようにページツーリングでは、Webページを構成するコンポーネント内を順番に通過していく。以下では、コンポーネント内のツアーをコ

<sup>▼</sup> 学生会員 大阪大学大学院情報科学研究科博士後期課程  
[t.maekawa@ist.osaka-u.ac.jp](mailto:t.maekawa@ist.osaka-u.ac.jp)

<sup>▲</sup> 正会員 大阪大学大学院情報科学研究科  
[fhara.nishio@ist.osaka-u.ac.jp](mailto:fhara.nishio@ist.osaka-u.ac.jp)

ンポーネント内ツアーと呼び、あるコンポーネント内ツアーの終点から次のコンポーネント内ツアーの始点までのツアーをコンポーネント間ツアーと呼ぶ。またページツーリングでは、ユーザがツアーを早送り、および、巻戻しすることが可能である。

以上のようなページツーリングを用いた Web ページ閲覧の手順について説明する。まずユーザは、ツアーによりページの内容を見る。このときユーザは、興味のない部分を早送りできる。また、巻戻しを用いて気になった部分に戻ることができる。そしてページの中から興味のある情報や目的の情報を見つけたら、ツアーを中断してその部分を閲覧する。ツアーを中断してからのスクロールは、携帯電話の4方向キーを用いて行う。ツアーは中断した点やツアーの開始点からいつでも再開できる。以上のようにページツーリングの目的は、ユーザにページの内容を速覧させ、興味のある情報や目的の情報を見つけさせる点にあり、Web ページの内容を最初から最後までじっくりと読ませるといった目的ではない。Web ページ内の情報を高速に提示する研究としては、Bruijin らの RSVP Browser[1]があり、Web ページを RSVP (Rapid Serial Visual Presentation) の Keyhole と呼ばれる手法を用いてスライドショーのように表示する。このとき、各スライドは、現在のページのリンク先のページをよく表す画像とテキストを、現在のページから抽出して作成したものである。しかし、Web ページを提示する場合にはもとのレイアウトを保つことが重要と考えるため、本研究ではスクロールの手段を用いている。次章では、ユーザにページ内を速覧させるためのツアーの構成手法について述べる。

### 3. ツアーの構成手法

2章で述べたように、ツアーはコンポーネント内ツアーとコンポーネント間ツアーの繰り返しから構成される。ツアーの構成手順は下記の通りである。

1. Web ページからコンポーネントを抽出する。
  2. 抽出したコンポーネントを辿る順序を決定する。
  3. それぞれのコンポーネントにおいて、コンポーネント内ツアーを決定する。
  4. コンポーネント間ツアーを決定する。
- 各手順の詳細について以下で説明する。

#### 3.1 コンポーネントの抽出

コンポーネントは、Web ページ内の関連する情報の集合であり、例えば Yahoo! Japan や goo といったポータルサイトのトップページにおける、検索フォーム、最新のトピックやニュース、ディレクトリなどである。Web ページの構造を解析してコンポーネントを抽出する研究は多くあり[2][3]、そのほとんどが HTML タグの解析を行っているが、本研究ではコンポーネントのサイズを考慮して Web ページを再帰的に細分化することにより、コンポーネントを抽出する。具体的には、全てのコンポーネントのサイズが指定する範囲内となるように細分化する。図2に、その例を示す。まず Web ページ全体をコンポーネントと見立て、そのサイズを判定する。この例ではコンポーネント A が指定する範囲より大きかったため B と C のコンポーネントに分割している。分割には、DOM ツリーを用いる。この例では、コンポーネント A の DOM ツリーにおける子ノードがコンポーネント B と C になっているため、コンポーネント A をコンポーネント B と C に分割している。そして、コンポーネントに対してサイズの判定と分割を再帰的に行うことで、指定した範囲内のコ



図1 ページツーリングの概念図  
Fig.1 Basic Concept of the Page Touring

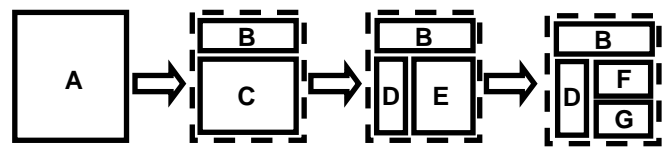


図2 コンポーネントの抽出手順  
Fig.2 Procedure of Component Extraction

ンポーネントを抽出する。ただし、あるコンポーネントを分割した結果、いずれかの子のサイズが指定した範囲より小さくなるときはその分割を行わない。本研究ではサイズの範囲を携帯電話の画面サイズの1から5倍としている。

#### 3.2 ツアー順序の決定

まず順序を決定するために、抽出したコンポーネントの属性付けを行う。属性は、HEADER、FOOTER、LEFTSIDE、RIGHTSIDE、BODYの5つを定義している。

HEADERは、ページの上部に位置し、横長の形状をもつ。具体的には、コンポーネントの上端がページ上端から100pix以内にあり、コンポーネントの高さと幅の比が0.25未満である。FOOTERは、ページの下部に位置し、横長の形状をもつ。具体的には、コンポーネントの下端がページ下端から100pix以内にあり、コンポーネントの高さと幅の比が0.25未満である。LEFTSIDE(RIGHTSIDE)は、ページの左(右)に位置し、幅はページ幅よりある程度小さい。具体的には、コンポーネントの左(右)端がページの左(右)端から30pix以内にあり、コンポーネントの幅がページ幅の40%以下である。BODYは、上記以外のコンポーネントで、ページの中央に位置する。

本研究では、コンポーネントのレイアウト的特徴からその属性を決定し、属性の内容的特徴からコンポーネントを辿る順序を決定する。具体的には、まずタイトルであるHEADERからLEFTSIDE、メインコンテンツであるBODYをたどり、RIGHTSIDE、FOOTERを辿る順序を選択する。このとき、同じ属性をもつ複数のコンポーネントが存在することが考えられるが、そのときはHTMLソースにおいて先に記述されているコンポーネントを優先する。またHEADERの次にBODYを辿ることで、メインコンテンツをユーザに早く提示する方法も考えられる。さらに、最初にBODYを辿る方法も考えられ、そのポリシーはユーザによってさまざまである。そこで、辿る順序はユーザが任意に切り替えられるようシステムを設計する。その詳細については3.5節で述べる。

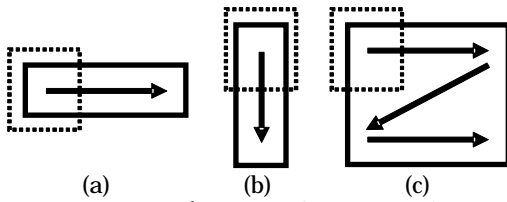


図3 コンポーネント内ツアーのパス  
Fig.3 Path of "Tour in Component"

### 3.3 コンポーネント内ツアー

まず携帯電話の画面サイズとコンポーネント  $i$  のサイズから,  $i$  のコンポーネント内ツアーのパスを決定する. その方法を図3に示す. 実線の矩形はコンポーネントを, 点線の矩形は携帯電話の画面を, 矢印はコンポーネント内ツアーのパスを表す. このとき携帯電話の画面の中心が矢印を通過するようスクロールが行われる. 画面の高さがコンポーネントの高さより大きいときは, 図3(a)のように左から右へスクロールするようパスを設定する. 画面の幅がコンポーネントの幅より大きいときは, 図3(b)のように上から下へスクロールするようパスを設定する. 画面の高さと幅がコンポーネントの高さと幅よりそれぞれ小さいときは, 図3(c)のように折り返してスクロールするようパスを設定する.

このコンポーネント内ツアーにおけるスクロール速度は, 以下の式で決定する.

$$Speed_i (pix / msec) = \frac{c \cdot Attribute_i \cdot Area_i (pix^2)}{AoI_i (msec) \cdot Breadth_i (pix)}$$

$Attribute_i$  はコンポーネント  $i$  の属性に関係し, 以下のよう  
に与えられる.

$$Attribute_i = \begin{cases} 1.0, & \text{HEADER} \\ 1.5, & \text{FOOTER} \\ 1.2, & \text{LEFTSIDE} \\ 1.2, & \text{RIGHTSIDE} \\ 1.0, & \text{BODY} \end{cases}$$

これにより, あまり重要ではないと考えられる FOOTER, LEFTSIDE, RIGHTSIDE の属性をもつコンポーネント内ツアーのスクロール速度を早くしている.

$Area_i$  はコンポーネント  $i$  の面積である.  $AoI_i$  はコンポーネント  $i$  に含まれる情報量を表し, ユーザがコンポーネント  $i$  を閲覧するためにかかる時間として定義している. 具体的には, コンポーネント  $i$  内の文字列を読むためにかかる時間と, コンポーネント  $i$  内の画像を認識するためにかかる時間の和である. 一般的に日本人の読書の速さは, 1分あたり 400 から 600 文字程度と言われており, 画像を認識するのに要する時間は最低で 100msec と言われている[4][8]. そこで,  $AoI_i$  を以下の式で与える.

$$AoI_i = \frac{TextLength_i}{500} + \sum_{k=0}^m 100 \cdot Th(Width_k, Height_k)$$

ここで,  $TextLength_i$  はコンポーネント  $i$  に含まれる文字列長を表し, それを 1 ミリ秒あたりの文字列の読解速度で割っている.  $Width_k$  と  $Height_k$  はコンポーネント  $i$  に含まれる画像  $Image_k$  の幅と高さのピクセル値で,  $Th$  は画像の大きさを判定するステップ関数である.  $Th$  は以下の式で与える.



(a) (b)  
図4 クライアントシステムの画面例  
Fig.4 Screenshots of Client System

$$Th(W, H) = \begin{cases} 1, & \text{if } W > 20 \text{ and } H > 20 \\ 0, & \text{else} \end{cases}$$

$Breadth_i$  はコンポーネント  $i$  をスクロールする際の, スクロール方向と直交するコンポーネント  $i$  の幅(高さ)であり, 以下の式で表す.

$$Breadth_i = \begin{cases} Width_i, & \text{if 縦にスクロール} \\ Height_i, & \text{if 横にスクロール} \\ ScreenHeight, & \text{if 折り返してスクロール} \end{cases}$$

$Width_i$  と  $Height_i$  は, コンポーネント  $i$  の幅および高さを,  $ScreenHeight$  は携帯電話の画面の高さをそれぞれ表す. これにより, 例えば横にスクロールするとき, コンポーネントの高さが小さいほど,  $Breadth_i$  は小さくなり, スクロール速度は速くなる.

また, 折り返してスクロールするとき, 左下方向へのスクロール速度は 400pix/sec としている.

### 3.4 コンポーネント間ツアー

コンポーネント  $j$  をツアーにおけるコンポーネント  $i$  の次のコンポーネントとすると,  $i$  から  $j$  へのコンポーネント間ツアーのパスは,  $i$  のコンポーネント内ツアーの終点から,  $j$  のコンポーネント内ツアーの始点をつなぐ直線である. コンポーネント間ツアーの速度は 400pix/sec としている.

### 3.5 システムの実装

前章までの設計に基づいて, ページツーリングを用いたシステムの実装を行った. NTT Docomo の  $i$  アプリ対応携帯電話 SH900i を用いて, Java 言語でクライアントシステムを実装した. SH900i の画面解像度は, 240 × 320(pix) であるが,  $i$  アプリで利用可能な画面は 240 × 240(pix) である. Windows XP 搭載の PC 上で, Visual Studio .NET Visual C# 言語および PHP(Hypertext preprocessor) 言語を用いてサーバシステムを実装した. サーバシステムは, クライアントシステムが指定する Web ページからコンポーネントを抽出し, さらに, そのページ内の位置や, 幅, 高さ, 属性, 情報量などを算出してクライアントシステムに返す. クライアントシステムは, サーバシステムから受信したこれらの情報からツアーを決定し, ユーザに提供する.

図4(a)に図1に示すページにおいて, コンポーネント内ツアーを行っている際のキャプチャー画面を示す. 下方向の矢印はスクロールの進行方向を示しており, 実際の画面に表示されるわけではない. 図4(b)にコンポーネント間ツアーを行っている際のキャプチャー画面を示す. このコンポーネント

間ツアーでは、ズームアウトをしている。

クライアントシステムは、通常の Web ブラウザとしての機能以外に以下の 3 つの機能をユーザに提供する。

- **ツアーの早送り・巻戻し:** ユーザは任意のときに、ツアーの早送りと巻戻しをすることができる。早送りではツアーの速度が 2 倍になる。つまり、コンポーネント内ツアーとコンポーネント間ツアーの速度をそれぞれ 2 倍にする。巻戻しではツアーを逆方向に送り、その速度は早送りのときと同様である。また早送りと巻戻しの際、ページを縮小して表示する。ここで、スクロール速度にしたがって倍率を変更する研究としては文献[5]がある。この研究では、縦長の大量のドキュメントをユーザがマウスで上下にスクロールする際に、そのスクロール速度に応じて倍率を変更する。具体的にはユーザがスクロールを早くすればするほど、ドキュメントが縮小される。本研究では Web ページ内を複雑にスクロールするため、縮小しすぎるとユーザはシステムが現在注目しているコンポーネントを把握できない。したがって、倍率を 0.77 程度に抑えている。
- **ツアーの順序の変更:** 3.2 節で述べたように、ツアーの順序を決定する方法はさまざまである。実装したシステムでは、HEADER, LEFTSIDE, BODY, RIGHTSIDE, FOOTER の順序で送る方法、HEADER, BODY, LEFTSIDE, RIGHTSIDE, FOOTER の順序で送る方法、BODY, HEADER, LEFTSIDE, RIGHTSIDE, FOOTER の順序で送る方法の 3 つをユーザが選択できる。
- **ツアーの再開:** ページツーリングでは、ユーザがツアー中に興味のある情報を見つけるとツアーを中断し、手操作によるスクロールを用いてその情報を閲覧する。その後ユーザがツアーを再開するときは、ツアーを初めから開始、中断した点から再開、現在地点に最も近いツアーのパス上の点からの再開の 3 つの方法を選択できる。

クライアントシステムにおける通常のスクロールは、携帯電話の 4 方向キーを用いて行う。

#### 4. まとめと今後の課題

本論文では、携帯電話などの画面が小さいモバイル端末上で Web ページを閲覧するための、自動スクローリング方式を提案した。提案方式では、Web ページから情報の集合であるコンポーネントを抽出し、コンポーネントのレイアウトやコンポーネント内の情報量などからスクロールの方法を決定する。自動スクローリングにより、ユーザは貧弱な入力インタフェースしかもたない携帯電話などのモバイル端末において、最低限の操作で目的の情報を探ることができる。

今後は、コンポーネント内の情報量の算出方法や、コンポーネント間ツアーの際の表示倍率の選択方法などについて研究を進める予定である。

#### [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] O. de Bruijn, 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.

- [2] 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.
- [3] D.W. Embey, Y. Jiang, and Y. -K. Ng: Record-boundary discovery in web documents, Proc. ACM SIGMOD'99, pp. 467 478, May/June 1999.
- [4] C.G. Healey, K.S. Booth, and J.T. Enns: Highspeed visual estimation using preattentive processing, ACM Transactions on Computer-Human Interaction, vol. 3, issue 2, pp. 107 135, June 1996.
- [5] T. Igarashi and K. Hinckley: Speed-dependent automatic zooming for browsing large documents, Proc. ACM Conference on User Interface Software and Technology (UIST'00), pp. 139 148, Nov. 2000.
- [6] A. Nadamoto, H. Kondo, and K. Tanaka.: WebCarousel: Automatic presentation and semantic restructuring of web search result for mobile environments, Proc. the 12th International Conference on Database and Expert Systems Applications (DEXA 2001), pp. 712 722, Sept. 2001.
- [7] OPERA Software: Opera for Mobile, <http://www.opera.com/products/mobile/>.
- [8] R. Spence: Rapid, serial and visual: a presentation technique with potential, Information Visualization, vol. 1, issue 1, pp. 13 19, Mar. 2002.

#### 前川 卓也 Takuya MAEKAWA

2003 年大阪大学工学部情報システム工学科卒業。2004 年同大学院情報科学研究科博士前期課程修了。現在、同大学院情報科学研究科博士後期課程在学中。日本データベース学会学生会員。

#### 原 隆浩 Takahiro HARA

1995 年大阪大学工学部情報システム工学科卒業。1997 年同大学院工学研究科博士前期課程修了。同年、同大学院工学研究科博士後期課程中退後、同大学院工学研究科情報システム工学専攻助手、2002 年より同大学院情報科学研究科マルチメディア工学専攻助手、2004 年より同大学院情報科学研究科マルチメディア工学専攻助教授となり、現在に至る。工学博士。IEEE、電子情報通信学会、日本データベース学会の各会員。

#### 西尾 章治郎 Shojiro NISHIO

1975 年京都大学工学部数理工学科卒業。1980 年同大学院工学研究科博士後期課程修了。工学博士。京都大学工学部助手、大阪大学基礎工学部および情報処理教育センター助教授、大阪大学大学院工学研究科情報システム工学専攻教授を経て、2002 年より同大学院情報科学研究科マルチメディア工学専攻教授となり、現在に至る。2000 年より大阪大学サイバーメディアセンター長、2003 年より大阪大学大学院情報科学研究科長を併任。この間、カナダ・ウォータールー大学、ビクトリア大学客員。データベース、マルチメディアシステムの研究に従事。現在、Data & Knowledge Engineering, Data Mining and Knowledge Discovery などの論文誌編集委員。情報処理学会フェローを含め、ACM、IEEE など 9 学会の会員。