社団法人 情報処理学会 研究報告
IPSJ SIG Technical Reports

2008－DD－69（3）
2008 / 11 / 27

モバイル端末のためのWebコンテンツ閲覧支援環境とその応用

近藤 圭佑† 浅見 昌平† 大囿 忠親† 新谷 虎松†

† 名古屋工業大学大学院情報工学専攻
E-mail: †kondo@toralab.ics.nitech.ac.jp

あらまし 本論文では，通信速度が制約されたモバイル端末における Web コンテンツ表示の課題（表示速度，フォント，スタイルなど）に関連して，効果的な閲覧支援環境について述べる．ここでは，サーバ側での 画像化コンテンツのレンダリング手法を実現することにより，モバイル端末での効果的な閲覧性の向上を可能とする．また，サーバ上で作成した画像化コンテンツを再利用することで，画像化コンテンツを作成する時間を短縮し，高速な画像化コンテンツ配信を可能にする．

キーワード モバイル端末，Web ブラウザ，コンテンツ配信

On a Web Contents Browsing Support Environment for Mobile Devices and Its Applications

Keisuke KONDO†, Shouhei ASAMI†, Tadachika OZONO†, and Toramatsu SHINTANI†

† Dept. of Computer Science and Engineering, Graduate school of Engineering, Nagoya Institute of Technology
E-mail: †kondo@toralab.ics.nitech.ac.jp

Abstract We propose an effective web browsing support environment . We focus on some problems for displaying web pages on mobile devices in which the network speed is restricted . Effective web browsing with mobile devices is enabled by realizing a rendering technique of web pages at a server side. The system can speed up web browsing by using web contents by reusing existing web contents made on the server . The experimental results show how the system can be used effectively for speeding up web browsing on mobile devices .

Key words Mobile Devices, Web Browser, Contents Delivery

1. はじめに

本研究では，モバイル端末上でより高速に Web ページを表示するために，Web ページをサーバ上で適切なコンテンツに変換するシステムを構築する．ここでは，モバイル端末上で行うレンダリング処理をサーバ上で行い配信用コンテンツへ変換することで，Web ページのデータ容量削減とモバイル端末上での処理時間を短縮する．さらに，そのシステムで作成したコンテンツを再利用することでコンテンツを作成する時間を省略し，より高速な Web ページ配信を実現する方法を提案する．

近年，携帯電話やモバイルパソコンなどのモバイル端末が広く普及し，その機能は多岐にわたり，急速に発展してきている．また，携帯電話やパソコンだけではなく，音楽プレイヤーや携帯ゲーム機などでもインターネットが利用可能となってきている．さらに，iPhone のようにフルブラウザを搭載したモバイル端末が登場したことで，いつでもどこでもインターネットを利用できるという環境が整いつつある．2008 年の総務省の調査によると，個人がインターネットを利用する際に使用する端末は，パソコンよりもモバイル端末のほうが多いという調査結果も出ている．この調査結果からもわかるように，モバイル端末からのインターネットアクセスの重要性が増している．

しかし，モバイル端末では，CPU，メモリなどのシステムリソース，ネットワーク帯域，ディスプレイサイズ，入出力などの環境がパソコンと比べて劣っている．これらの制約によって，ユーザがモバイル端末上でパソコン用の Web ページを快適に閲覧することは難しいと言える．ここでは Web ページを快適に閲覧するための要素として表示速度，閲覧性，操作性の３点を考慮する．ここでは，表示速度は，Web ページにアクセスしてからロードが完了しレンダリングが終了するまでの時間，閲覧性とは Web ページを表示したときに内容を読みやすいかどうか，操作性はユーザの操作する頻度や操作しやすいかどうかとする．一般的にモバイル端末は，表示速度の点では，

ネットワーク帯域が狭いためロードに時間がかかり，処理速度が遅いためレンダリングに時間がかかってしまう．表示における待ち時間が長くなることはユーザにとって快適だとは言えない．モバイル端末の閲覧性の点では，ディスプレイサイズが小さいため，レンダリングが崩れたり，長い文章が読みにくくなる場合がある．また，モバイル端末が指定されているフォントに対応していない場合なども考えられる．

最近では，パソコン用の Web ページを閲覧するためにフルブラウザを搭載した携帯電話や Safari を搭載した iPod touch などは，閲覧性，操作性の面でこれらの問題に対応してきている．しかし，表示速度に関してはまだ快適だとは言いがたく，これらの問題を解決するためのアプローチが必要である．

2. モバイル端末のための Web コンテンツ開発支援

携帯電話にパソコン用に作られた Web ページを配信する研究は広く行われている．情報量の多い Web ページをモバイル端末上に表示することを目的とした従来の研究では，閲覧する Web ページの簡略化や Web ページのインデックス作成，要約，分割，サムイル画像の利用などによる Web ページの閲覧性の向上を目指している．

文献 [1] [2] は Web サイトのインデックスページにあるリンクを解析することで，文字列のみからなるサイトのツリー型メニューを作成する．そのメニューを用いることにより，目的とするページを容易に探し出すことができる．文献 [3] では，Web ページの構造や内容を解析して，小さい画面のデバイスのために，要約した Web ページを表示するシステムを提案している．文献 [4] では，Web ページの要素に対して，Google のページランクに似たランク付けアルゴリズムを利用することで，小さい画面のデバイスに対しての Web ページを生成している．具体的には，元となる Web ページを要素化して，ランク付けアルゴリズムを使い，要素をツリー状に構造化させる．文献 [5] は，Web ページのコンテンツの中で重要であるリストを利用して，Web ページのコンテンツ中のリスト部分を小さいデバイスに表示するシステムを提案している．リストのみの表示は簡潔であり，小さい画面のデバイスには適している．文献 [8] では，Web ページを分割し，ユーザが指定したコンテンツに対して必要な情報を前後関係から発見する．文献 [9] は RSS に着目して Web ページのガイドラインをつくりコンテンツに誘導するものである．しかし，これらはテキストや画像を配信するものでレイアウト情報などは保たれていないので Web ページの製作者が意図するものと異なってしまう．文献 [7] [6] は，Web ページを意味のある単位で分割しておき，Web ページのサムネイル画像を表示し，ユーザが選択した部分のコンテンツを表示させる．

現在では，携帯電話用フルブラウザなどに対する取り組みがいくつか行われており，代表的な実例としてクライアント独立方式の Opera および NetFront，サーバ・クライアント協調方式の jig，SiteSneaker，ibisBrowser および Scope などがあり，問題に対して様々なアプローチをとっている．また，モバイルパソコンや iPod touch では，パソコンと同等のフルブラウザが搭載されている．

クライアント独立方式の Web ブラウザでは，単に Web サーバと通信を行うため，圧縮によるデータ通信量の削減ができず，クライアントのリソース消費も大きい．サーバ・クライアント協調方式では，ブラウザの機能の一部をサーバに持たせることで，クライアントでの処理を軽量化する．また，サーバ上で Web ページを圧縮することでデータ通信量を削減することができる．サーバ・クライアント協調方式に着目して文献 [10] では，サーバ上でレンダリングを行い画像化して配信する形式をとっている．しかし，i アプリとして作られているので機種に依存する場合や，Javascript や動画，サウンドなど未対応な部分が多く，サーバでの処理にも時間がかかるため多くの改善の余地がある．

3. Web ページ変換に基づく開発支援

本研究では，1 章であげた問題点を解決するために，モバイル端末上での処理を軽減し，ロードする Web ページの容量を減らすことで，Web ページの表示速度を向上させることを目標とする．モバイル端末上の処理を減らすために，本システムは Web ブラウザの機能をサーバ上とモバイル端末上に分け，本来はモバイル端末上で行う Web ページへのアクセスおよびレンダリング処理をサーバが代行する．そして，レンダリングした Web ページを画像に変換してモバイル端末のフルブラウザに配信する．ここでは，モバイル端末上で画像を表示するだけの単純な処理ですむため，処理時間を短縮することができる．また，モバイル端末よりも処理能力が高いサーバ上でレンダリングを行うことでより高速な Web ページの表示が可能となる．さらに，サーバ上で画像を圧縮することでデータ転送量を削減することもできる．

しかし，Web ページを画像化して配信するだけではリンクやフォーム，Flash，Javascript などの機能を表現することができない．そこで，本システムは，レンダリングした結果から DOM 情報を取得し，リンクやフォーム，Flash の座標や url などの値を抜き出す．その情報を HTML 形式で Web ページの画像とともに配信する．以降，本システムが変換して配信する HTML および Web ページの画像を画像化コンテンツと呼ぶこととする．Web ページを画像化してしまうと Javascript を利用した動的に変化する Web ページに対応できない．そこで，ユーザが Javascript を動かす場合，本システムはもとの Web ページを配信する．本システムを経由する形式に Web ページのリンク先を書き換えることで配信した Web ページは本システムの一部として動作する．

本システムの構成は図 1 のようになっている．本システムは Mac OS X 上で開発されている．使用言語として主に，perl，python，javascript を使用している．システムは，画像化コンテンツ配信機構，Web ページ画像生成機構，DOM 情報取得機構およびデータベースで構成されている．

以降は，システムの処理の流れを説明したあと，それぞれの機構について詳細に述べる．

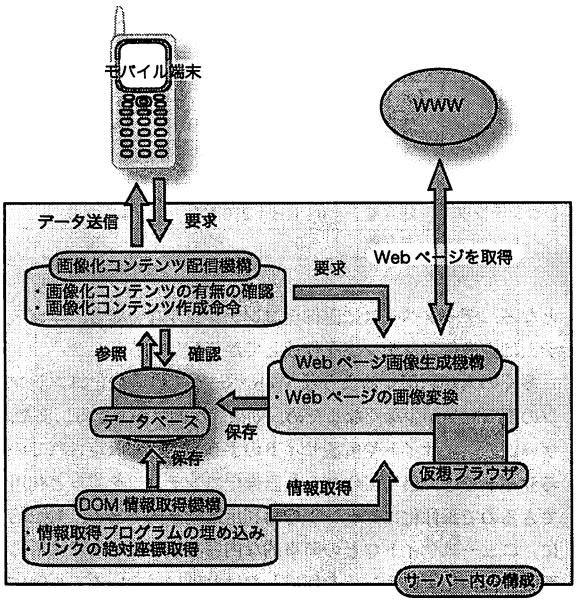

図 1　システム構成図

3.1　配信する画像化コンテンツの構成

本システムが配信する画像化コンテンツは，Web ページ画像，Web ページの機能として必要なリンクやフォーム，Flash などを記述した HTML，そして画像化コンテンツを管理するためのプログラムで構成されている．メインコンテンツである Web ページ画像，入力が必要となるフォーム，コンテンツの表示が必要な Flash はそれぞれ img タグ，form タグ，embed タグを埋め込み絶対座標指定することで表現している．リンクに関しては，サーバ上で管理し，クリックされたときに通信してサーバ上でリンクを照合して対象の Web ページの画像化コンテンツを配信する．クリックした後はサーバにアクセスして他の Web ページに移るので，リンク情報は HTML には含めない．

3.2　Web ページ画像生成機構

Web ページ画像生成機構とは，与えられた URL の Web ページを読み込んで画像化するシステムである．Web ページの画像化には仮想ブラウザを用いる．仮想ブラウザとは，サーバ上で稼動する HTML レンダリングエンジンである．

Web ページを画像に変換することで，モバイル端末へのデータ転送量を削減可能であり，モバイル端末が Web ページをロードする時間の短縮が可能になる．本システムでは，Web ページを画像化する際の画像形式に JPEG を採用している．JPEG の画質を下げ，画像を縮小化することでデータ通信量の削減を行っている．JPEG の画質を下げすぎるとブロックノイズができ視認性が落ちるので，文字が視認できるレベルまで画質を落とした．

本アプローチでの特筆すべき点は，サーバでのレンダリング結果をそのまま配信することができるためレイアウトが崩れない点，また，モバイル端末上で利用できないフォントを使用できるという点である．

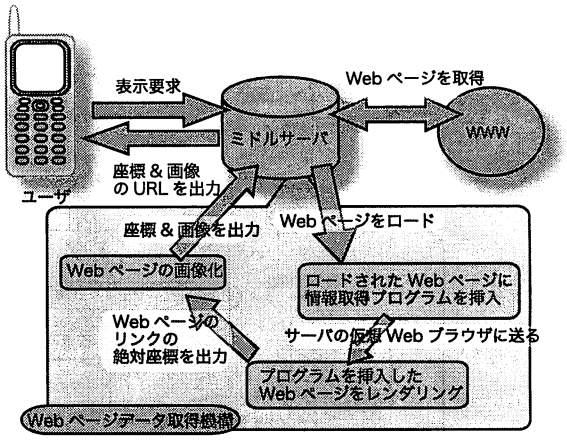

図 2　Web ページのリンクの絶対座標取得

3.3　DOM 情報の抽出と機能の実現

ここでは画像化コンテンツにおいて，リンク，フォーム，Flash を画像上で実現するためにこれらの情報を抽出する DOM 情報抽出機構について述べる．

例としてリンク情報の抽出について述べる．リンク情報はリンクの絶対座標と大きさ，リンク先の url といった属性値である．Web ページのリンクの絶対座標と大きさは，閲覧している Web ブラウザの種類や Web ブラウザのウィンドウのサイズなどによって変化するので，HTML ソースからでは取得することができない．そこで，DOM 情報取得機構がリンク情報を取得するプログラムをレンダリングした Web ページに対して実行することでリンク情報を取得する．図 2 にリンク情報取得の手順を示す．まず，Web ページ画像生成機構が対象の Web ページをロードする．次に，DOM 情報取得機構がロードした Web ページにプログラムを埋め込む．そして，仮想ブラウザが仮想的にレンダリング処理を行う．最後に，レンダリング処理の結果に対して仮想ブラウザが埋め込まれたプログラムを実行する．埋め込まれたプログラムは Web ページが表示されたときに実行され，表示したときの Web ブラウザの種類，ウィンドウのサイズに基づいて DOM 情報取得機能がリンク情報を取得することを可能にする．動的に表示されるようなリンク情報も，仮想ブラウザ上でイベントを発生させてから同様の手順で実行させてからリンク情報を取得するプログラムを実行させることで取得できる．

このように，仮想ブラウザのレンダリング結果に対して自分が用意したプログラムを実行させることで自由に Web ページを変更したり，情報を取り出すことができる．また，HTML を解析する場合とは異なり，リンクの位置情報のように Web ページを表示しないと取得できない情報でも得ることできる．この技術は，いろいろな場面で利用でき，Web ページを利用したアプリケーションを作る際に応用が利く手法である．本システムは，フォームや Flash の情報取得にもこの技術を利用している．また，フォントサイズを指定すれば，Web ページの文字を

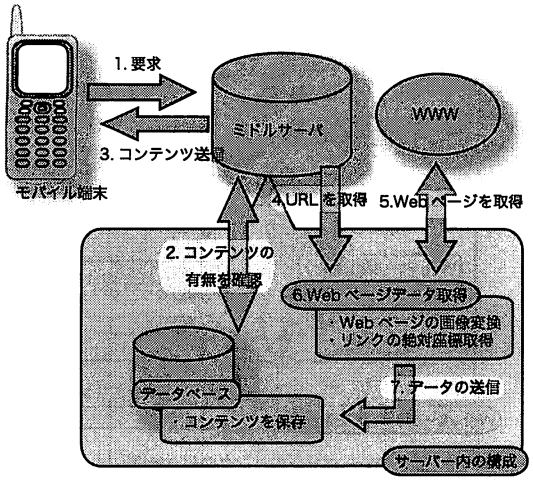

図 3　画像化コンテンツの再利用手順

大きくすることもできる．

次に，リンク情報を実現する方法について述べる．本システムはリンク情報をサーバ上で管理し，モバイル端末には配信しない．ユーザがリンクをクリックしたときに，サーバと通信を行い，サーバ上のリンク情報とクリックした座標を照合する．該当する URL があった場合，本システムはその Web ページの画像化コンテンツをユーザに配信する．

フォームと Flash は配信する画像化コンテンツに HTML タグとして記述し，スタイルで絶対座標を記述して画像の上に重ねて表示する．

3.4　画像化コンテンツ再利用による高速化

本節では，表示速度をさらに高速化するために本システムで行っている画像化コンテンツの再利用について述べる．

画像化コンテンツを作成する処理は，ある程度時間がかかる処理であり，アクセスするたびに画像化コンテンツを作成してから配信していては時間がかかる．しかし，製作者が Web ページを更新していなければ，ユーザがアクセスするたびに画像化コンテンツを作成してモバイル端末に配信するという処理は，同じ画像化コンテンツを何度も作ることになる．そこで，本研究は画像化コンテンツを再利用することで Web ページの配信を高速化する方法を提案する．図 3 に画像化コンテンツの再利用手順を示す．本システムは作成した画像化コンテンツをサーバ上のデータベースに保存する．ユーザが同じ Web ページへのアクセスするたびに同じ画像化コンテンツを作成するのではなく，画像化コンテンツ配信機構はまず前回作成した画像化コンテンツを配信する．そして，画像化コンテンツ配信機構は画像化コンテンツを配信したあとで，対象の Web ページの更新時間を確認し，前回の画像化コンテンツを作成した時間から更新がある場合，画像化コンテンツを作り直してデータベースに保存する．更新された画像化コンテンツは次回からまた利用される．このように，前回作成した画像化コンテンツを再利用することで画像化コンテンツを作成する時間を省略することができるため，本システムの処理時間を大幅に短縮することが可能となる．データベースに画像化コンテンツがない場合，本システムは画像化コンテンツを作成してから配信する．

多くの人が利用するような Web ページでは画像化コンテンツの更新間隔が少なくなるため，効果的な閲覧ができる．また，ゲームの攻略サイトや解説サイトのような内容が固定されているサイトでは，最初に作成した画像化コンテンツをずっと利用できるので画像化コンテンツの再利用は効果的である．対照的に，ニュースサイトなどの時事的な内容を扱うサイトでは常に新しい画像化コンテンツを作成しなければいけないので画像化コンテンツの再利用は効果的ではないといえる．最新の画像化コンテンツがほしい場合は，同じ画像化コンテンツにもう一度アクセスするか，常に画像化コンテンツを作成する機能をオンにする方法を用意している．

表 1　Web ページと画像化コンテンツの比較

		Google	Yahoo!	Infoseek ニュース
表示領域 (高さ)		600 px	2400 px	4800 px
元の Web ページ	ファイル数	5	32	61
	総容量	57 KB	341 KB	543 KB
	タグ数	138	434	608
画像化コンテンツ	ファイル数	3	6	11
	総容量	63 KB	270 KB	445 KB
	タグ数	5	8	13

4.　評価実験

本章では，前章で構築したモバイル端末のための画像化コンテンツ配信システムについて実験および評価する．本研究は，モバイル端末上での快適な Web ページの閲覧のために表示速度の面での改善を目的としている．評価には，実際にモバイル端末のフルブラウザ上で Web ページを表示した場合と本システムで作成した画像化コンテンツを表示した場合の表示速度を比較する．モバイル端末は Apple 社の iPod touch による WiFi 通信と 3G 携帯電話の SoftBank 911T を使用した．iPod touch では，標準で装備されているフルブラウザ Safari 上で元の Web ページと変換済みの画像化コンテンツを表示する．携帯電話では，標準で装備されている PC サイトブラウザ上で元の Web ページと変換済みの画像化コンテンツを表示する．サーバは CPU dual 4GHz，メモリ 4GB の Mac OS X Server を使用した．

4.1　Web ページと画像化コンテンツの比較

まず，実験に使用する Web ページの容量を比較する．Web ページの表示領域の大きさと構成する画像化コンテンツの多さを考慮し，本研究では Google，Yahoo!，Infoseek ニュースの TOP ページを対象としてデータを計測した．表 1 に元の Web ページと本システムが作成した画像化コンテンツの総容量と構成ファイル数，表示領域の大きさを示す．Web ページの画像が大きすぎると表示できない可能性があるため，本システムは表示領域が大きい Web ページ画像を分割して配信している．本システムはテキストコンテンツも画像データとして変換しているが，画像データを圧縮して容量を削減している．

Google ではコンテンツが少ないため，変換後の 6KB 大きくなっている．Yahoo! と Infoseek ニュースは Web ページを構成している画像ファイル数や HTML の容量が大きいが，本システムが画像に変換したことで Yahoo! は 164KB，Infoseek ニュースは 163KB 容量が削減されている．画像化コンテンツのタグ数はサーバ上で作られた Web ページ画像を表示するタグと script タグしかないためかなり減っている．これによりモバイル端末が行うレンダリング処理が削減されている．

4.2 画像化コンテンツの表示時間

実機により，表 1 に示した Web ページをモバイル端末が表示する時間を計測した．Web ページの表示時間はリクエスト開始から Web ページ全体の表示が完了するまでの時間を計測し，表 2 に iPod touch が各 Web ページを表示した時間の比較，表 3 に携帯電話が各 Web ページを表示した時間の比較を示している．

Google は表示領域が小さく画像もテキストも少ない．iPod touch は，Google のページを画像化コンテンツに変換するよりも Google のページをそのまま表示するほうが 0.93 秒速く表示された．しかし，Google のページを変換済みの画像化コンテンツを再利用した場合と Google のページをそのまま表示した場合では，表示時間にほとんど差がなかった．携帯電話では，Google のページをそのまま表示する時間と画像化コンテンツに変換してから表示する時間はほとんど差がないが，変換済みの画像化コンテンツを再利用して表示する場合はそのまま表示するよりも 3.73 秒早く表示できた．Yahoo! や Infoseek ニュースのように表示領域が広く構成ファイルが多い．iPod touch は，Yahoo! のページを画像化コンテンツに変換してから表示した場合は Yahoo! のページをそのまま表示する場合と表示する時間にほとんど差がない．しかし，Yahoo! のページを変換済みの画像化コンテンツを再利用した場合，Yahoo! のページをそのまま表示した場合より 3.73 秒速く表示された．携帯電話は，Yahoo! のページを画像化コンテンツに変換してから表示する場合，Yahoo! のページをそのまま表示するよりも 4.58 秒早く表示された．Yahoo! のページを変換済みの画像化コンテンツを再利用した場合，Yahoo! のページをそのまま表示した場合より 7.76 秒速く表示された．iPod touch は，Infoseek ニュースのページを画像化コンテンツに変換してから表示した場合，Infoseek ニュース のページをそのまま表示する場合より 4.10 秒早く表示された．Infoseek ニュースのページを変換済みの画像化コンテンツを再利用した場合，Infoseek ニュースのページをそのまま表示した場合より 9.95 秒速く表示された．携帯電話は，Infoseek ニュースのページを画像化コンテンツに変換してから表示する場合，Infoseek ニュースのページをそのまま表示するよりも 6.48 秒早く表示された．Infoseek ニュースのページを変換済みの画像化コンテンツを再利用した場合，Infoseek ニュースのページをそのまま表示した場合より 12.70 秒速く表示された．

4.3 サーバ上での処理時間

表 4 にサーバが表示速度の計測に使用した Web ページを画像化コンテンツに変換する際に必要な処理時間を示す．サーバ

表 2 iPod touch 上での表示時間の比較

	Google	Yahoo!	Infoseek ニュース
Safari	1.91 s	6.05 s	13.48 s
画像化コンテンツ作成後に表示	2.98 s	5.93 s	9.38 s
画像化コンテンツを先に表示	1.89 s	2.32 s	3.53 s

表 3 携帯電話上での表示時間の比較

	Google	Yahoo!	Infoseek ニュース
PC サイトブラウザ	5.05 s	14.6 s	25.83 s
画像化コンテンツ作成後に表示	5.17 s	10.02 s	19.35 s
画像化コンテンツを先に表示	1.48 s	6.84 s	13.13 s

表 4 サーバでの処理時間

	Google	Yahoo!	Infoseek ニュース
Web ページのロード	0.41 s	0.74 s	2.89 s
Web ページのレンダリング	0.05 s	0.38 s	0.50 s
プログラムの実行	0.01 s	0.13 s	0.66 s
Web ページの保存	0.08 s	0.21 s	0.45 s
合計	0.55 s	1.46 s	4.50 s

上の処理は，Web ページのロードに最も時間がかかっていて，Web ページの容量や構成ファイル数に応じてロードする時間が増えている．Web ページのレンダリング処理は，HTML のタグ数が多いとサーバが処理する時間が長くなっている．また，表示領域の広さに応じてサーバが画像を保存する時間が増加している．つまり，Yahoo! や Infoseek ニュースのように表示領域が広い Web ページは，本システムがサーバ上で行っている画像処理にかかる時間が増加している．しかし，モバイル端末が表示領域が広い Web ページをレンダリングする場合，処理時間が長くなっていたため表 2 や表 3 のような結果が得られた．また，HTML のタグが多くなるとリンクなどの情報を抽出するプログラムを実行する時間が長くなっていることも影響している．

5. 考　　察

本システムを利用して Web ページを画像化コンテンツに変換した後に iPod touch の Safari 上で画像化コンテンツを表示する場合，Google では 0.93 秒遅いが，Yahoo! で 0.8 秒，Infoseek ニュースでは 4.1 秒早く表示することができた．また，携帯電話のフルブラウザ上で Web ページと変換済みの画像化コンテンツを表示した場合，Google では 0.12 秒遅く，Yahoo! で 4.587 秒，Infoseek ニュースでは 5.48 秒早く表示することができた．また，変換済みの画像化コンテンツを再利用した場合，表示領域が広い Infoseek ニュースの表示時間は Web ページをそのまま表示した場合と比べ iPod touch で 9.95 秒，携帯電話で 12.7 秒早く表示することができた．

これは，表 1 で示したように，Web ページをサーバ上で画像化コンテンツに変換したことで，画像化コンテンツのタグ数と構成ファイル数がかなり減少したので，モバイル端末のレンダリング処理が少なくなったことが大きく影響していると思われる．本システムは，初回のアクセス時にはサーバ上に画像化

コンテンツがないため，通常の Web ページを表示する場合と同程度の時間で画像化コンテンツを表示する．そして，次回のアクセスからは常に前回作成した画像化コンテンツを再利用するので，サーバが画像化コンテンツを作成する時間を省くことができ，ユーザはより高速に画像化コンテンツを表示することができる．よって，本システムは表示速度の面ではとても有効である．

どのような Web ページが，変換済みの画像化コンテンツを再利用する手法に適しているかを考察する．本システムは，一人のユーザが対象の Web ページにアクセスすることで，その Web ページを画像化コンテンツに変換してユーザに配信する．以降，他のユーザが同じ Web ページにアクセスした場合に，以前作成した画像化コンテンツを再利用することでサーバでの処理時間を短縮する．サーバは，バックグラウンドで Web ページの更新を確認して最新の画像化コンテンツを作成する．よって，より多くのユーザが利用する Web ページは，サーバで画像化コンテンツを更新する間隔が短くなるので，本システムで画像化コンテンツを再利用することは有効であるといえる．

また，ポータルサイトやゲームの攻略サイト，趣味の解説サイトなど内容が固定されているサイトは，前回作成した画像化コンテンツを再利用することは効果的である．対照的に，ニュースサイトなどの時事的な内容を扱うサイトは，常に最新の情報を提示する必要がある．前回作成した画像化コンテンツを配信しているので，配信される画像化コンテンツは常に最新であるとはいえないため，変換済みの画像化コンテンツの再利用は効果が薄い．

しかし，評価実験の結果 (表 2，表 3) を見ると，アクセスがあったときに Web ページを画像化コンテンツに変換する場合でも，そのまま Web ページを表示した場合より画像化コンテンツに変換したほうが表示時間が短いことがわかる．よって，最新の画像化コンテンツが必要な場合は，新しい画像化コンテンツを作成しても通常の Web ページを表示する時間より早く表示できるので，十分ニュースサイトなどにも対応できると考えている．本システムでは，最新の画像化コンテンツが必要な場合に，ユーザが同じ画像化コンテンツにもう一度アクセスする，または，サーバの常に画像化コンテンツを作成する機能をオンにする，ユーザが CGI に引数を与えるという方法を用意している．

6. おわりに

本論文では，モバイル端末における Web ブラウジングの問題点を挙げ，その問題点を解決する方法を提案した．ユーザがモバイル端末上での Web ページ閲覧をより高速に行うために，本研究では Web ページの画像化をするための Web ページ変換技術を提案した．また，その高速化手法として本研究では，変換済みの画像化コンテンツを再利用する手法を提案した．

モバイル端末の処理を減らすために，Web ブラウザにおけるレンダリング処理をサーバ上とモバイル端末上に分離し，本来はモバイル端末上で行う Web ページへのアクセスおよびレンダリング処理をサーバが代行する．そして，サーバサイドでサーバはレンダリングした Web ページを画像に変換してモバイル端末のフルブラウザに配信する．モバイル端末は画像を表示するだけの単純な処理ですむため，処理時間を短縮することができる．また，モバイル端末よりも処理能力が高いサーバ上がレンダリングを行うことでより高速な Web ページの表示が可能となる．さらに，サーバで画像を圧縮することでデータ転送量を削減することもできる．

評価実験の結果、本システムを利用した場合は標準で搭載されている Web ブラウザで表示した場合よりも表示が最大 6.48 秒速いという実験結果が出た．変換済みの画像化コンテンツを再利用すると，サーバにおける画像化コンテンツ作成時間の分だけ表示時間がさらに短縮され，最大 12.7 秒早いという結果も得られた．変換済みの画像化コンテンツを再利用するとユーザはリアルタイムに新しい情報を得られるとは限らないが，画像化コンテンツの再利用により本システムは時事的な内容を扱うニュースサイトなどでの閲覧でも十分に効果的に機能することができる．ポータルサイトや攻略サイトなどの内容が固定されているようなサイトでは，本システムは十分に効果的な手法であるといえる．

文　献

[1] M. Jones, G. Marsden, N. Mohd-Nasir, and G. Buchanan : "A site-based outliner for small screen web access", in Proc. World Wide Web Conference (WWW99), May,1999.

[2] O. Buyukkokten, H. Garcia-Molina, A. Paepcke, and T. Winograd : "Power browser: Efficient web browsing for PDAs", in Proc. Human-Computer Interaction Conference 2000 (CHI 2000), vol. 2, issue 1, pp. 430-437, Apr, 2000.

[3] A. F. R. Rahman, H. Alam, R. Hartono and K. Ariyoshi : "Automatic Summarization of Web Content to Smaller Display Devices", in Post Presentations of 6th International Conference on Document Analysis and Recognition, Seattle, The United States, Sept. 10-13, 2001.

[4] Xinyi Yin, Wee Sun Lee: "Using Link Analysis to Improve Layout on Mobile Devices", in Proc. of World Wide Web Conference (WWW ' 04), NewYork, May, 2004.

[5] Hassan Alam, Fuad Rahman, Yuliya Tarnikova and Aman Kumar : "When is a List is a List?: Web Page Re-authoring for Small Display Devices", in Proc. of World Wide Web Conference (WWW ' 03), May, 2003.

[6] Heidi Lam, Patrick Baudisch : "Summary Thumbnails : Readable Overviews for Small Screen Web Browsers .", in Proc. of the SIGCHI conference on Human factors in computing system , 2005 .

[7] Shumeet Baluja : "Browsing on Small Screens : Recasting Web-Page Segmentation into an Efficient Machine Learning Framework" , in Proc. World Wide Web Conference (WWW'06), pp. 33-42, May, 2006.

[8] Yevgen Borodin, Jalal Mahmud, I.v.Ramakrishnan : "Context Browsing with Mobiles - When Less is More", in Proc. of the 5th Intl. Conf. on Mobile System , Applications and Services (MOBYSYS '07), pp.3-15, June, 2007.

[9] John Garofalakis, Vassilios Stefanis : " Using RSS feeds for effective mobile web browsing", in Proc. Information Society Journal, Volume 6, Number3/Novenmer, pp.239-257, 2007.

[10] 近藤圭佑，森重賢二，伊藤正都，新谷虎松，大圍忠親："携帯電話の組み込み機能を利用可能な Web アプリケーションの実現とその応用", 第 70 回情報処理学会全国大会論文集, Mar. 2008.