



Title	Research on Comfortable Web Browsing using Mobile Phones
Author(s)	荒瀬, 由紀
Citation	大阪大学, 2009, 博士論文
Version Type	VoR
URL	https://hdl.handle.net/11094/1934
rights	
Note	

The University of Osaka Institutional Knowledge Archive : OUKA

<https://ir.library.osaka-u.ac.jp/>

The University of Osaka

Research on Comfortable Web Browsing using Mobile Phones

Submitted to
Graduate School of Information Science and Technology
Osaka University

January 2010

Yuki ARASE

List of Publications

1. Journal Paper

1. Arase, Y., Maekawa, T., Hara, T., Uemukai, T., and Nishio, S.: An Adaptive Content Presentation System for Web Browsing using Cellular Phones, *IPSJ Journal*, Vol. 47, No. 12, pp. 3149–3164 (Dec. 2006) (in Japanese).
2. Arase, Y., Hara, T., Uemukai, T., and Nishio, S.: Nine Button Browsing System: A Web Browsing System for Cellular Phone Users, *DBSJ Letters*, Vol. 5, No. 4, pp. 9–12 (Mar. 2007) (in Japanese).
3. Arase, Y., Maekawa, T., Hara, T., Uemukai, T., and Nishio, S.: A Web Browsing System for Cellular Phone Users based on Adaptive Presentation, *Universal Access in the Information Society (UAIS)*, Vol. 6, No. 3, pp. 259–271 (Nov. 2007).
4. Arase, Y., Hara, T., Uemukai, T., and Nishio, S.: Web Page Presentation on Cellular Phone using Drive Metaphor, *DBSJ Journal*, Vol. 7, No. 1, pp. 103–108 (Aug. 2008) (in Japanese).
5. Komaki, D., Ohnishi, K., Arase, Y., Hara, T., Hattori, G., and Nishio, S.: Design and Implementation of a Click-Search Interface for Web Browsing using Cellular Phones, *International Journal of Web and Grid Services (IJWGS)*, Vol. 5, No. 1, pp. 66–84 (Mar. 2009).
6. Iwata, M., Arase, Y., Hara, T., and Nishio, S.: Web Browser for Children using a Bubble Metaphor, *International Journal of Web Information Systems (IJWIS)* (2010, to appear).

2. International Conference Paper

1. Arase, Y., Maekawa, T., Hara, T., Uemukai, T., and Nishio, S.: A Web Browsing System based on Adaptive Presentation of Web Contents for Cel-

- lular Phones, in *Proceedings of International Cross-Disciplinary Workshop on Web Accessibility (W4A 2006)*, pp. 86–89 (May 2006).
2. Arase, Y., Hara, T., Uemukai, T., and Nishio, S.: Nine-Button Web Browsing System for Cellular Phone Users, in *Proceedings of Innovations in Information Technology 2006*, CD-ROM (Nov. 2006).
 3. Arase, Y., Hara, T., Uemukai, T., and Nishio, S.: OPA Browser: A Web Browser for Cellular Phone Users, in *Proceedings of ACM Symposium on User Interface Software and Technology (UIST 2007)*, pp. 71–80 (Oct. 2007).
 4. Komaki, D., Ohnishi, K., Arase, Y., Hara, T., Hattori, G., and Nishio, S.: A Click-Search Interface for Web Browsing using Cellular Phones, in *Proceedings of International Wireless Internet Conference (WICON 2008)*, CD-ROM (Nov. 2008).
 5. Ohnishi, K., Arase, Y., Hara, T., Uemukai, T., and Nishio, S.: Preview Functions for Web Browsing using Cellular Phones, in *Proceedings of International Workshop on Data Management for Information Explosion in Wireless Networks (DMIEW 2009)*, pp. 1070–1075 (Mar. 2009).
 6. Arase, Y., Xie, X., Duan, M., Hara, T., and Nishio, S.: A Game Based Approach to Assign Geographical Relevance to Web Images, in *Proceedings of International World Wide Web Conference (WWW 2009)*, pp. 811–820 (Apr. 2009).
 7. Ohnishi, K., Arase, Y., Hara, T., Uemukai, T., and Nishio, S.: A Content Clipping System for Web Browsing using Cellular Phones, in *Proceedings of International Workshop on Data Management for Wireless and Pervasive Communications (DMWPC 2009)*, pp. 978–983 (May 2009).
 8. Arase, Y., Hara, T., Uemukai, T., and Nishio, S.: MotoBrowser: Enjoyable Browsing using Cellular Phones by “Motoring” Web Pages, in *Proceedings of International Conference on Mobile Data Management (MDM 2009)*, pp. 373–374 (May 2009).

9. Arase, Y., Hara, T., and Nishio, S.: User Profiling for Web Search Based on Biological Fluctuation, in *Proceedings of International Conference on Human-Computer Interaction (HCI International 2009)*, LNCS 5612, pp. 239–247 (July 2009).
10. Iwata, M., Arase, Y., Hara, T., and Nishio, S.: Investigation of Children’s Characteristics for Web Browsing, in *Proceedings of ACM International Workshop on Web Information and Data Management (WIDM 2009)*, pp. 91–94 (Nov. 2009).

3. Domestic Conference Paper (with peer-review)

1. Arase, Y., Hara, T., Uemukai, T., and Nishio, S.: Nine Button Browsing System: A Web Browsing System for Cellular Phones, in *Proceedings of DBWeb 2006*, pp. 111–118 (Nov./Dec. 2006) (in Japanese).
2. Ohnishi, K., Arase, Y., Hara, T., Uemukai, T., and Nishio, S.: A Web Browsing System using Acceleration Sensors for Cellular Phone Users, in *Proceedings of DEWS 2007* (Feb./Mar. 2007) (in Japanese).
3. Iwata, M., Arase, Y., Hara, T., and Nishio, S.: On a Web Page Presentation Method for Children using a Bubble Metaphor, in *Proceedings of DICO MO 2008*, Vol. 2008, No. 1, pp. 626–633 (July 2008) (in Japanese).
4. Komaki, D., Ohnishi, K., Arase, Y., Hara, T., Hattori, G., and Nishio, S.: Click-Search Interface for Web Browsing using Cellular Phones, in *Proceedings of DICO MO 2008*, Vol. 2008, No. 1, pp. 553–560 (July 2008) (in Japanese).
5. Iwata, M., Komaki, D., Arase, Y., Hara, T., and Nishio, S.: Evaluation of Web Browser for Children based on a Bubble Metaphor, in *Proceedings of DICO MO 2009*, Vol. 2009, No. 1, pp. 1415–1423 (July 2009) (in Japanese).
6. Komaki, D., Ohnishi, K., Arase, Y., Hara, T., Hattori, G., Takishima, Y., and Nishio, S.: Evaluation of a Click-Search Interface for Web Browsing

using Cellular Phones, in *Proceedings of DICO MO 2009*, Vol. 2009, No. 1, pp. 263–271 (July 2009) (in Japanese).

4. Domestic Conference paper

1. Arase, Y., Maekawa, T., Hara, T., Uemukai, T., and Nishio, S.: Adaptive Presentation of Web Contents for Mobile Web Browsing using Cellular Phones, in *Proceedings of IEICE SIG-WI2*, pp. 47–52 (Mar. 2006) (in Japanese).
2. Arase, Y., Hara, T., Uemukai, T., and Nishio, S.: Web Browsing Assistance for Cellular Phones by Motoring Metaphor, in *Proceedings of DEWS 2008* (Mar. 2008) (in Japanese).

5. Book Chapter

1. Arase, Y., Hara, T., and Nishio, S.: Web Page Adaptation and Presentation for Mobile Phones, Hu, W.-C. and Zuo, Y. Eds., *Handheld Computing for Mobile Commerce: Applications, Concepts and Technologies*, IGI Global (2010, to appear).

Abstract

Due to the advancement in data-processing and communication technologies, mobile phones are now widely used to access the Web. However, small screens and poor input capabilities of mobile phones, such as a direction pad and telephone keypad (that is laid out in a 3×4 grid and allocated numbers and letters for text messaging), bring difficulties on Web browsing since sizes of Web pages are much larger than those of mobile phones' screens. Therefore, users have to perform numerous operations to browse a Web page and are suffered by getting lost in the page, e.g., losing a sense which way to scroll.

In this thesis, to solve this problem and provide comfortable Web browsing experience using mobile phones, we propose new Web browsing techniques for mobile phones that adaptively present Web contents according to their characteristics. We design a Web browser that provides various presentation functions so that users can select ones adapted to their browsing situations to preserve enough cognitive resources required for interacting with the environment. We also propose Web page overview presentation methods on mobile phones, which is an essential function to prevent users from getting lost in the page and help them find out contents of their interest.

This thesis consists of five chapters. First we describe the research background, as the general situation of Web browsing using mobile phones and discuss prior research works sharing the similar goal with us in Chapter 1. In Chapter 2, we design and implement a Web browsing system that adaptively presents Web contents according to their characteristics for mobile phone users in a fixed situation, such as sitting on a chair at their home. It first shows a scaled-down Web page that fits a mobile phone's screen to help users look for their contents of interests. When a user specifies a content, our browser shows the content adapted to its characteristics, e.g., by automatically scrolling the screen and providing a function of cue images in the content.

In Chapter 3, we focus on variety of mobile phone usage scenarios and propose a Web browser that provides various presentation functions on mobile phone's telephone keypad. Using this browser, users can select suitable presentation

functions according to their browsing situations, e.g., their available cognitive resources for browsing, only by pressing a key. Although we assume that users browse a Web page using mobile phones in a fixed situation in Chapter 2, users' browsing situations are possibly dynamic because users always carry their mobile phones around with them. Since it is still difficult to accurately detect users' situations and suitable presentation styles on various browsing situations are user dependent, we let users select presentation functions by themselves. We conduct a user experiment in real situations by asking participants to use our browser for three days in the same way as their own phones in their daily life. The results show that users can select preferable presentation styles on our browser that satisfy the requirement of their cognitive resources, and comfortably browse Web pages interacting with their environment.

In Chapter 4, we further examine overview presentation of a Web page on mobile phones. Although participants to the experiments in Chapters 2 and 3 regard the overview function is essential, some of them comment that presenting a scaled-down page is not sufficient, since they cannot see the details of contents. Therefore, we propose novel styles of overview and conduct a user experiment to investigate an appropriate way for overview presentation and merits/demerits of each presentation style, which effectively helps users find their information of interest. The results show that presenting annotations, which indicate what kinds of information are around the currently displayed area, is effective, and automatic scrolling can greatly reduce the number of operations and help users grasp the page structure when they are lost within the page. Finally, Chapter 5 summarizes this thesis and discusses our future work.

Contents

1	Introduction	1
1.1	Background	1
1.2	Related Work	3
1.2.1	Browsers for Individual Mobile Web Browsing	4
1.2.2	Web Browsing Assistance for Mobile Phone Users	6
1.2.3	Collaborative Work using Mobile Phones	7
1.2.4	Other Recent Trends	8
1.3	Organization of Thesis	9
1.3.1	Adaptive Presentation of Web Contents	9
1.3.2	Mobile Web Browser to Reduce Users' Cognitive Load in Dynamic Situations	9
1.3.3	Annotation Presentation and Auto-scrolling for Web Page Overview	10
1.4	Web Page and Components	11
1.4.1	Component Extraction	11
1.4.2	Component Attribute	13
2	Adaptive Presentation of Web Contents	15
2.1	Background	15
2.2	Component Classification & Experiment	16
2.2.1	Component Classes	16
2.2.2	Component Classification Method	18
2.2.3	Preliminary Experiments	19
2.3	Design and Implementation of Proposed System	24
2.3.1	Design of System Structure	24

2.3.2	Web Page Overview	26
2.3.3	Adaptive Presentation Methods	26
2.3.4	Implementation of Prototype System	30
2.4	User Experiments Using Real Data	30
2.4.1	Accuracy of Component Classification	30
2.4.2	User Experiments	31
2.5	Summary	41
3	Mobile Web Browsing in Dynamic Situations	43
3.1	Background	43
3.2	OPA Browser and Presentation Methods	44
3.2.1	Fragmented Attention of Mobile Users in Dynamic Situations	45
3.2.2	Sample User Scenarios	49
3.2.3	Web Page Overview (Function 1)	52
3.2.4	Enlargement of Scroll Unit (Function 2)	54
3.2.5	Jump to Components (Functions 3, 4)	54
3.2.6	Cue Images (Function 5)	54
3.2.7	Magnifying Glass (Function 6)	56
3.2.8	Word Search (Function 7)	57
3.2.9	Detecting Relevant Components (Function 8)	58
3.2.10	Auto-scrolling (Function 9)	60
3.2.11	Implementation of Prototype System	61
3.3	User Experiment in Real Situations	62
3.3.1	Experimental Procedure	62
3.3.2	Result and Discussion	65
3.4	Summary	70
4	Web Page Overview for Mobile Phones	73
4.1	Background	73
4.2	Overview Presentation Methods	75
4.2.1	Scaled-down View	75
4.2.2	Auto-scrolling	76
4.2.3	Annotation	77
4.2.4	Combination	80

<i>CONTENTS</i>	ix
4.3 Implementation of Proposed Methods	84
4.4 Experiments	85
4.4.1 Accuracy of Annotation and Category Extraction	85
4.4.2 User Experiment	86
4.5 Summary	91
5 Conclusion	93
5.1 Summary of Thesis	93
5.2 Future Work	95
5.2.1 Dynamic Contents	95
5.2.2 Optimization of Auto-scrolling Speed	95
5.2.3 Appropriate Presentation on Smart Phones	96
5.2.4 Mobile Users' Locations	96
Acknowledgment	97

Chapter 1

Introduction

1.1 Background

The advancement of Internet technology has dramatically changed the way of information acquisition. In earlier days, people have to go to a library to obtain academic knowledge, consult with a travel agency to plan their summer vacation, and ask savvy friends which restaurant is the best to take guests from overseas. Internet technology has enabled people to do all of these stuff at home, only by querying search engines and browsing resulting Web pages. In addition, the interactive feature of the Web has added diversity to the information on the Web. For example, we can see different opinions about an incident, which gives chance for us to view the case from different aspects. Using a more casual example, we can find a restaurant for welcoming foreign guests that fits the foreigners' preferences using the Web instead of taking recommendations from locals that may mismatch our guests' expectations.

The rise of mobile Web, accessing the Web using mobile phones, is the next paradigm shift on the way of people's information acquisition. Now people even don't have to sit in front of a computer. People just take their mobile phones out of their pockets and access the Web to obtain information they need. Namely, people can obtain information anywhere and anytime when they need it. The popularization of Web browsing using mobile phones indicates that it is widely accepted by people. According to the investigation performed by Japanese Ministry of Internal Affairs and Communications, the number of Internet accesses

using mobile phones has exceeded the number of that using PCs in 2005 [42]. Additionally, In-Stat (<http://www.instat.com/>) reported that 975 million mobile Web users are expected by 2012 in the world [43].

Although such Web browsing using mobile phones has become popular, most existing Web pages are designed for desktop PCs, and thus it is inconvenient to browse such large sized pages using mobile phones. As Figure 1.1 shows, sizes of Web pages are far larger than mobile phone screens. For example, most mobile phones adopt the screen size of QVGA (Quarter VGA) that has 230×320 [pix] resolution, while the size of the top page of Yahoo! (<http://m.www.yahoo.com/>) is 982×1059 [pix] and Wired News (<http://www.wired.com/>) is 1036×5014 [pix]¹. Since mobile phones have only a small screen like this and poor input capabilities (like a direction pad and telephone keypad), users have to perform numerous operations to find the contents of interest from an entire page and read them in detail. The same problem is reported when viewing a large-sized photo using mobile phones [87], which represents that it is generally difficult to browse large information on small screens. To provide comfortable Web browsing experience using mobile phones, we propose several Web browsing techniques for mobile phones. We describe their designs and contributions that were evaluated by the user experiments.

In this thesis, we refer to a *mobile phone* as the one that only has a small screen and poor input capabilities such as a direction pad and telephone keypad. Although a group of more advanced phones, generally called as *smart phone* represented by iPhone [44] and BlackBerry [24], that has larger screen and rich input capabilities (such as a touch screen and gyro-sensor) is becoming conspicuous, most portable phones in the world are still the conventional ones, and thus we target on them in this thesis. Furthermore, even though smart phones have larger screens and richer input capabilities, they still suffer from the same issue due to the small screens limited by the nature of portable devices. Therefore, our works can contribute to information presentation on smart phones to some extent.

¹these sizes change daily according to their contents



Figure 1.1: Browsing a large Web page using mobile phone

1.2 Related Work

According to the growing attention to the mobile phone technology that enables users to connect to the Web anywhere and anytime, many research efforts have been made to realize comfortable Web browsing experience using mobile phones. We categorize these studies as follows based on their target scenarios.

1. Browsers for individual Web browsing using mobile phones
2. Web browsing assistance for mobile phone users
3. Collaborative work using mobile phones

Our research falls into the first category, as we aim to provide comfortable Web browsing experience using mobile phones performed by individuals. In the last of this section, we introduce the recent emerging trend of mobile phone oriented studies and services.

1.2.1 Browsers for Individual Mobile Web Browsing

As described in Section 1.1, due to their small screens and poor input capabilities, mobile phones can only display limited area of a Web page, and users have to perform numerous operations to browse the page. To solve this problem, past studies have focused on reducing users' scroll operations by deleting the part of contents or simplifying the contents of the Web page to display only important information.

PROTEUS [5,6] personalizes Web pages using users' recorded Web browsing histories. It highlights contents that users often read, and deletes and replaces rarely viewed contents with links, to display only important contents for the users. Bruijn et al. [27,28] propose a system that extracts and displays important images from a Web page in a sequential manner. This allows users to obtain the outline of a page. However, this approach is effective only for Web pages that contain many meaningful and large images associated with its content, such as online shopping sites. On the other hand, the Web browser proposed by Buyukkokten et al. [29,30,31] summarizes the text of a content within a Web page and then creates an index by removing all images within the page. When a user selects a content from the index of the page, the entire content is displayed. Billsus et al. [23] propose a method that modifies the layout of a Web page according to the user's preference. This method estimates what information the user needs from his/her reading behavior, and then arranges the content containing the corresponding information from the top of the page. Many commercial Web browsers for mobile phones, such as Opera for Mobile [67], NetFront [63], jig browser [45], and Scope [74], restructure a Web page so that users can read it using only vertical scrolling. However, it is difficult to properly restructure a complicated Web page, e.g., the one containing nested tables. In addition, restructured Web pages are often too long to scroll, requiring users to perform many operations to scroll to the end of the restructured page.

These prior works have a significant drawback in which they have to change the layout of a Web page by simplifying or deleting contents of the page. If the layout of a Web page is changed, users cannot use their past Web browsing experience on desktop PCs. For example, users may be used to the presence of

a menu list on the left side of a Web page. However, if the layout is changed, and this common feature is removed or is different, users might not be able to comfortably browse the page. In addition, while most prior approaches use HTML (hypertext markup language) [86] tag analysis to change the layout, the HTML tags can only determine the page layout but not semantically describe the content. Furthermore, alternation of the page layout may make the page inconsistent with the intentions of the Web page's author. For example, if the author writes "See the left figure" in the Web page's text and the layout is different, readers may not understand which figure the author means.

Therefore, it is the best to retain the original layout of a Web page in Web browsing using mobile phones, as well as to reduce users' scroll operations. There are several prior approaches that keep the layout of Web pages. WebThumb [83] first displays an overview of a Web page that is a scaled-down image of the page. When users select a content area from the overview, this content is displayed in a new application window at its original size. Collapse-to-Zoom technique [22] allows users to collapse areas deemed irrelevant from the overview of a Web page. Collapsing content causes all of the remaining content to be redrawn with more detail, which increases the users' chance of identifying relevant content. On the other hand, Baluja [21] proposes a system that divides a Web page into nine regions so that users can select and zoom in the region on the overview by pressing a corresponding key. Minimap proposed in [73] changes the width of text paragraphs and scales down images and tables, while preserving the layout as close as possible to the original layout of a Web page. Additionally, our research group proposed a method that automatically scrolls an entire Web page [56] to show users the entire structure of the page without any operations.

We also follow the policy to retain the original layout of a Web page and provide functions so that users can browse it without being annoyed by the limitations of mobile phones, i.e., small screens and poor input capabilities, which are described in the following chapters.

1.2.2 Web Browsing Assistance for Mobile Phone Users

In addition to leveraging Web page presentation on mobile phones described in Section 1.2.1, there are many works for Web browsing assistance. One of the most important services on the Web is Web search. Yahoo! (<http://yahoo.client.shareholder.com/press/overview.cfm>) and Google (<http://www.google.com/corporate/>) started Web search services targeting mobile phones. Yahoo! MOBILE [88] generates and shows an all-in page that summarizes search results into a simple shape so that users can quickly obtain the information they want. It also has a functionality of taking users' locations into account when a GPS receiver is available on their mobile phones, which returns the local information around users' location. Google Mobile [37] infers what kind of information, e.g., images and local information, a user needs from his/her input query. For example, when the user inputs "Toy poodle," Google Mobile understands the user might want images of toy poodle and returns the list of images first. These mobile search services are useful for mobile phone users to look for information they needs. Relating to the mobile search, our research group proposed a method to reduce users' burden to input a search query [46, 47, 48, 49]. Since mobile phones only have a telephone keypad, inputting query is demanding. Our method aims to help users query a term that they are interested in within his/her current browsing page without complicated operations. When the user wants to query a term, he/she presses a corresponding key and catch the term using a presented circle, and the term is automatically submitted to a search engine.

We also proposed preview functions [64, 65] that pre-fetch linked pages and show their previews by the small screenshots to assist users to determine which link to select for finding the information of interest. Tajima and Ohnishi [79] propose techniques that support better browsing of large HTML tables on small screens. Their methods render tables in the same way as on the PC screens and provide various functions for browsing the large tables, such as hiding unnecessary rows and columns. We proposed a Web browser function for mobile phones that enables users to save clips of browsed contents [66]. When a user presses a corresponding key, the function automatically detects candidate areas for clipping, and the user can easily select the content area to save as a clip. These

methods can reduce burdensome operations usually required for Web browsing using mobile phones.

1.2.3 Collaborative Work using Mobile Phones

According to the investigation performed by Japanese Ministry of Internal Affairs and Communications, the adoption rate of mobile phones became 89.3% in September of 2009 in Japan [41]. This situation, most people has their own mobile phones, arises a new way of mobile device usage in conjunction with the advancement of digital devices with screens, i.e., collaborative work using mobile phones with other users and devices, such as a Desktop PC, TV, and digital signage. Forecasting this trend, our research group has proposed a novel concept of Web browsing using mobile phones with multiple users as collaborative browsing [54, 58]. In collaborative browsing, users collaboratively browse web pages that are designed for desktop PCs using their mobile phones together. Since the number of handheld devices, their capabilities, and other conditions can vary widely amongst mobile users who want to browse a Web page, the system divides the page into multiple components and distributes them in an appropriate way. In this way, users can search and browse their target information by discussing and watching partial pages displayed on multiple devices.

The collaboration is not necessarily limited among mobile devices. Remote Display environment [57, 80, 81, 82] enables users to control near-by large displays by mobile phones. In this environment, mobile phone users make use of the large screen, such as displaying a large-sized Web page on it and control a pointer by their mobile phones, and users can also perform various collaborative works, such as playing online games together on the display. WebSplitter [39] provides a unified XML (extensible markup language) [85] framework that enables multi-device and multi-user Web browsing. WebSplitter splits a requested Web page and delivers the appropriate partial view of each page to a set of devices (e.g., laptops, PDAs, etc.). Multiple users can participate in the same browsing session, as in traditional conferencing groupware. WebGlance [69] leverages the advantages of both large displays and mobile devices. It enables multiple users to control a Web browser on a large, shared display using their own personal mobile devices,

such as a mobile phone, laptop, or wireless PDA. CoSearch system [3] focuses on collaboration on Web search. It enables co-located collaborative Web search by leveraging readily available devices such as mobile phones and extra mice of a desktop PC. Users can perform collaborative Web search using desktop PCs and mobile phones.

Such collaboration among users and devices will enhance the capability and convenience of mobile devices. Considering the fact that more and more displays are located around our lives, and the data processing and communication facilities of both displays around and mobile devices are advancing, this direction is becoming more important.

1.2.4 Other Recent Trends

As for the recent emerging trend of mobile phone oriented studies and services, researchers and service providers are paying attention to the mobile phone's property of always being with users. One is its ability to record people's life by a camera equipped with a mobile phone. People take photos when they find something attracts their interest with the camera and upload them to photo sharing sites on the Web, such as Flickr [36] and Picassa [71] to communicate with their friends and families by sharing scenes in their lives. We have performed a research focused on these photos, which decides relevance of these photos against the locations they were captured using a Web based game [18]². Crandall et al. [34] examine the relation between these photos and locations by data-mining these photos and extract popular sightseeing spots and their representative photos. Another characteristic drawing attention is a built-in GPS receiver that tells users' location information. Mobile social networking sites provide services that make use of users' locations. Brightkite [26] lets users "check in" at various locations out in the real world and see who else are there, have been there, and who are nearby. After checking in, users can post comments and also upload photos. Loopt [52] and GypSii [38] show friends' locations and their status, such as available and busy on a map. Loopt also has a function that sends an alert when a

²This work has been done during the author was with Microsoft Research Asia as a research intern.

friend comes nearby, and a function that sends e-mails to all friends being within a certain radius.

1.3 Organization of Thesis

In this thesis, we propose techniques to offer users comfortable Web browsing experience using mobile phones by approaching from the following aspects; characteristics of Web contents (Chapter 2), variety of mobile Web browsing situation (Chapter 3), and styles of overview to show overall structure of a Web page (Chapter 4).

1.3.1 Adaptive Presentation of Web Contents

Generally, mobile phone users in fixed situations, such as sitting on a chair at their home, browse a Web page with the following two steps. First, a user looks over a Web page to find contents that attract their interests, then reads information within the contents in detail. Due to the limited size of mobile phone's screen, the user has to scroll through the Web page for a long time to find the content that they want. The user again has to scroll within the content to read the information in detail. These tiresome scroll operations have to be reduced to provide a more comfortable Web browsing experience using mobile phones. To reduce users' scroll operations, Chapter 2, which is based on our works published in [8,9,12,15], describes the design and implementation of a Web browsing system that presents an overview of Web page and adaptively presents the contents according to their characteristics.

1.3.2 Mobile Web Browser to Reduce Users' Cognitive Load in Dynamic Situations

In Chapters 3, we focus on the dynamic usage scenarios of mobile phones, while in Chapter 2, we assume users browse Web pages in fixed situations. Considering that mobile phones are getting smaller and more portable, more dynamic browsing situations are possible, such as walking on a street and riding on a train

while commuting. Accordingly, preferable functions for Web browsing would be different based on users' situations, e.g., their available amount of cognitive resources. It is usually difficult to precisely detect or predict users' situations by using some sensing devices, because to fully equip a phone with accurate sensors is very costly. Furthermore, preferences of function on each browsing situation would be user dependent. Therefore, it is reasonable and effective to provide functions so that users can easily select preferable presentation styles by themselves according to their situations. In Chapter 3, which is based on our research works published in [10, 11, 13, 14], we propose a Web browser that allocates different functions for presenting Web pages on the telephone keypad of a mobile phone. Using this browser, users can select a presentation function adapted to their situations only by pressing a key so that they can comfortably interact with both the Web browser and environment.

1.3.3 Annotation Presentation and Auto-scrolling for Web Page Overview

In the user experiments in Chapters 2 and 3, we confirmed that showing the overview of a Web page is crucial for Web browsing using mobile phones. Since small screens can only present a portion of the Web page, users cannot grasp the entire page structure and foresee which direction to scroll to find their information of interest. Showing an overview can avoid this issue. The conventional way of overview is to show the scaled-down page that fits mobile phone's screen [33]. However, in our user experiments, some participants commented that the scaled-down page is insufficient as overview. Therefore, in Chapter 4, which is based on our research works published in [16, 17, 19, 20], we propose novel methods for presenting overview and compare with the conventional scaled-down page presentation to verify appropriate styles of overview that help users find the information of their interest and merits/demerits of each method.



Figure 1.2: Web page and components

1.4 Web Page and Components

Before we move to the details of the proposed Web browsing techniques for mobile phones, here we introduce the structural features of Web pages, which are the basis of our designs.

1.4.1 Component Extraction

Generally, a Web page is composed of a number of different components [32, 35, 90], each of which can be viewed as a block consisting of similar information, such as a site directory, news, and a search form located at the top page of a portal site. Figure 1.2 shows an example where each block enclosed with a thick rectangle is a component.

Many researchers have addressed component extraction from a Web page. Chen et al. [33] use a DOM (document object model) tree [84] to roughly divide a Web page, and then extracts components precisely by image processing. Embley et al. [35] detect HTML tag patterns from the HTML source of a Web page, then divide the page based on those patterns. Yang et al. [89] extract HTML tag

patterns as objects based on the frequencies of those patterns, and then merge similar objects into a component by text analysis.

These approaches analyze the structure of HTML tags and perform image processing and text analysis to precisely extract components. However, the size of components is not taken into account. We aim to show a Web page on a mobile phone screen, and thus we extract components so as to adjust to the size of mobile phone's screen because a very large component requires very long time for users to read the component. With large components, users have to scroll around within them to find useful information in the same way as on the entire page. Our method extracts components using a DOM tree, so that the size of all components fit within the objective size (width and height), which is defined between one to five times of the mobile phone's screen. In addition, it takes into account HTML tags to enhance the accuracy of extraction.

Figure 1.3 shows an example of component extraction. First, the whole page is set to the root node in the DOM tree, i.e., the whole page is set as one block. Then, the block size is compared to the objective size. If the size of the block is larger than the objective size, it is divided into a few blocks as child nodes in the DOM tree (see Figure 1.3). Again, the sizes of these new blocks are compared to the objective size. This process is recursively performed until the sizes of all blocks become the objective size. At this point, when a block is divided, blocks whose sizes are smaller than the objective size might be obtained. Such blocks are merged with a block corresponding to its brother node by considering the following HTML tags. However, if the size of the merged block becomes larger than the objective size, the merge operation is not performed.

- `<TABLE>`, `<TH>`, `<TD>`, `<DIV>`, `<P>`, `` tags

These tags themselves represent information blocks, thus, if the size of a block enclosed with these tags is within the objective size, the block is never combined with other blocks.

- `<HR>` tag

This tag is used only to separate components. Thus, a block enclosed with `<HR>` tags is never combined with others.

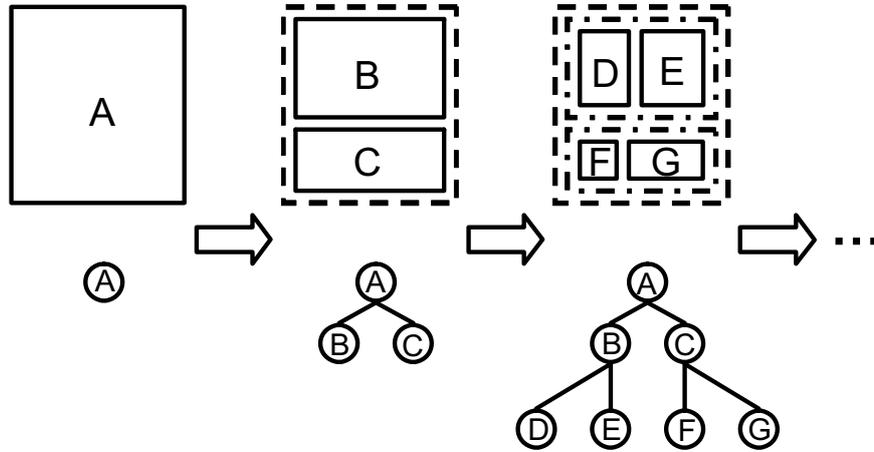


Figure 1.3: Component extraction from a Web page

1.4.2 Component Attribute

According to the Web page design theory, components have structural attributes, where components of the same attribute tend to share the same role in each Web page as follows. Here we define the metrics to determine the attribute based on our observation of a large number of Web pages.

- **HEADER**
 - The upper edge of the component is located in the area within 100[pix] from the top of the page, and the aspect ratio (height/width) is smaller than 0.25, e.g., the shape of the component is landscape.
 - A **HEADER** component usually contains the title image of the page.
- **FOOTER**
 - The bottom edge of the component is located in the area within 100[pix] from the bottom of the page, and the aspect ratio is smaller than 0.25.
 - A **FOOTER** component of public organizations (universities, companies, and governments) Web pages usually contains the help and

contact information, while most personal Web pages do not have a FOOTER component.

- LEFT(RIGHT)SIDE
 - The left (right) side edge of the component is located in the area within 30 [pix] from the left (right) side edge of the page, the width is smaller than 40% of the page width and the aspect ratio is larger than 3, e.g., the shape of the component is portrait.
 - A LEFT(RIGHT)SIDE component usually contains an advertisement, menu list and summary of the page or the site.
- BODY
 - Other components, i.e., those located near the center of the page.
 - A BODY component usually contains the main topics of the page.

These components and their attributes are the basis of our proposed Web browsing techniques for mobile phones described in the following chapters.

Chapter 2

Adaptive Presentation of Web Contents for Mobile Phone Users

2.1 Background

As we described in Section 1.4, a Web page is generally composed of a number of different components, each of which can be viewed as an information block. Due to such Web page structure, users in a fixed situation, such as sitting on a chair at their home, generally browse a Web page using the following two steps. First, a user looks over a page to find some desired component, then reads information within the component in detail. Due to the mobile phone's small screen, the user has to scroll through the Web page for a long time to find the objective component. The user then has to scroll within the component to read the information in detail. These tiresome scroll operations have to be reduced to provide a comfortable Web browsing experience using mobile phones.

In [33], Chen et al. proposed a system that provides users with an overview, which is a scaled-down image of a Web page. When a user selects a desired component from the overview, the system zooms into the selected component so that the user can read its details. This allows the user to find the objective component from the entire page with fewer number of scroll operations. However, the system proposed in [33] did not address reducing complicated scroll operations when reading specific information within the component. Furthermore, users' behaviors in reading Web components vary depending on each compo-

nent's characteristics, and thus, it is necessary to adapt each component to fit each user's reading behavior.

This chapter describes the design and implementation of a Web browsing system that adaptively presents the components of Web pages according to their characteristics to reduce users' scroll operations. The system displays the overview of a page as the scaled-down image that fits the mobile phone's screen to reduce time-consuming scroll operations for finding an objective component from the entire page. In addition, this system automatically adapts to component's characteristics to reduce tiresome scroll operations when reading the component.

The remainder of this chapter is organized as follows. Section 2.2 describes the component classification and the preliminary experiments. Section 2.3 describes the design and implementation of the system. Section 2.4 describes user experiments. Finally, we conclude this chapter in Section 2.5.

2.2 Component Classification and Preliminary Experiment

As described in Section 1.4, Web pages are composed of various components. This section explains the adopted component classification method and reports preliminary experimental results conducted to identify an appropriate presentation style for each component class.

2.2.1 Component Classes

Components of web pages have common characteristics according to their contents. We examined 60 typical Web sites, such as portal, news, and major companies' sites, and found that components could be classified into the following six categories according to their contents:

- *Text*: a component where the amount of text is larger than that of links.
- *Text & Image*: a component that contains text and images associated with the text, where the amount of text is larger than that of links. Figure 2.1(a) shows an example.



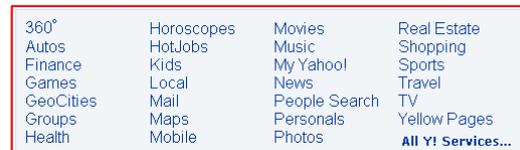
(a)Text & image



(c)Vertical link set



(b)Link & image



(d)Tiled link set

Figure 2.1: Examples of component classes

- *Image*: a component that contains only images.
- *Link & Image*: a component that contains a set of both links and images, where the amount of links is larger than that of text. Figure 2.1(b) shows an example.
- *Vertical link set*: a component that contains a set of vertically listed links. Figure 2.1(c) shows an example.
- *Tiled link set*: a component that contains a set of both vertically and horizontally listed links. Figure 2.1(d) shows an example.

In this chapter “link & image,” “vertical link set,” and “tiled link set” components are referred to as “link set” components. Users’ reading behaviors change according to the component’s class. For example, a user thoroughly reads each sentence in a “text” component, while he/she views images as well as reads texts in a “text & image” component. Furthermore, a user searches and selects an objective link or a link of interest in a “vertical link set” or “tiled link set” component.

2.2.2 Component Classification Method

After extraction of components as describe in Section 1.4, the system determines the component classes defined in Section 2.2.1.

First, a component that contains only images is classified as the “image.” Other components are classified as “text”/“text & image” or “link set,” based on the link occupation; the ratio of the height occupied by link strings to that of the entire component, which is a measure of how much amount link strings occupy within the component:

$$\begin{aligned}
 \text{Component class} &= \begin{cases} \text{Link set} & (\text{if link occupation ratio} \geq \gamma) \\ \text{Text or text\&image} & (\text{if link occupation ratio} < \gamma), \end{cases} \\
 \text{Link occupation} &= \frac{(\text{number of links}) \cdot (\text{height of link text [pix]})}{\text{component's height [pix]}}, \\
 \gamma &: \text{a threshold that determines the balance of amount of} \\
 &\quad \text{links in link set components and others.}
 \end{aligned}$$

To calculate the link occupation, we only use the heights of the component and links, since we observed that many components aimed to show a list of links when certain amount of lines (text and links) contained at least a link. Therefore, we suppose a component is filled with text or link lines and each line contains at most a link, and calculate the ratio of lines that contain a link. We empirically set the threshold γ to 0.5 based on our observation. The height of link text is the height of the strings that are embedded the links. Considering that it is becoming popular to embed links in a plain text to refer Web pages, such as blogs and Wikipedia (http://en.wikipedia.org/wiki/Main_Page), we should consider more

sophisticated method for determining the link occupation, e.g., layout analysis and machine learning, which is one of our future work.

If the link occupation is equal to or greater than γ , the component is classified as “link set.” Otherwise, the component is classified as “text” or “text & image;” if it contains images, it is classified as “text & image,” or otherwise, as “text.” Regarding a “link set,” if its height is larger than its width, it is classified as a “vertical link set.” Otherwise, if it contains images, it is classified as a “link & image.” If not, it is classified as a “tiled link set.” The system deals with image links in the same way as textual links, e.g., a component that contains a set of vertically listed image links is classified as a “vertical link set” component. Generally, image links are different from images without links, because most image links are used as decorated strings of the link texts, or as a banner briefly representing the contents of the linked pages. In addition, users may consider image links in the same way as text links, because their aim is searching and selecting an objective link, but not viewing the images themselves in detail. Therefore, image links are treated in the same way as text links.

2.2.3 Preliminary Experiments

Preliminary experiments were conducted to determine an effective way of presenting a component according to its class and characteristics. Auto-scrolling was considered as an effective approach, because it can reduce tiresome scroll operations within a component. In these experiments, the effectiveness of auto-scrolling and its appropriate setting was examined. Preliminary experiment 1 was targeted to check which component classes are suitable for auto-scrolling. Subsequently, preliminary experiment 2 examined the appropriate setting for auto-scrolling for those component classes that were evaluated as suitable for auto-scrolling in preliminary experiment 1.

Preliminary Experiment 1

We asked 16 subjects (14 males and two females ranging from 20 to 30 years old) to read six components in total, using auto-scrolling, where each component belonged to one of the component classes described in Section 2.2.1. Then sub-

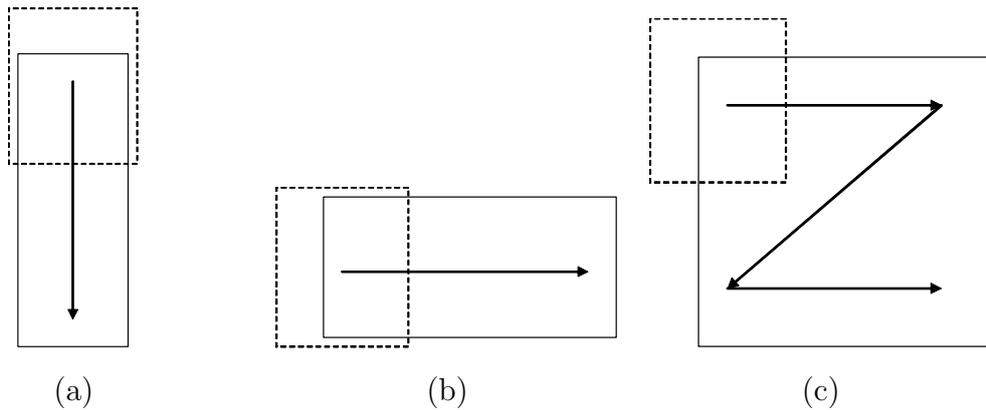


Figure 2.2: Scroll paths

jects answered a questionnaire survey that asked which classes were suitable for auto-scrolling.

For auto-scrolling, the scroll path was determined by the system for each component based on the component's shape as Figure 2.2 shows, in which the rectangle with a solid line represents a component's shape, the rectangle with a dotted line represents a mobile phone's screen, and the arrowhead represents the scroll path's direction. Specifically, when the component's height was higher than that of the mobile phone's screen and the component's width was narrower than that of the screen, the scroll path was set to the vertical direction, as shown in Figure 2.2 (a). On the contrary, when the component's height was lower than that of the screen and the component's width was broader than that of the screen, the scroll path was set to the horizontal direction, as shown in Figure 2.2 (b). When both the component's height and width were larger than those of the screen, the scroll path was set to zigzag, as shown in Figure 2.2 (c). In the experiment, the auto-scrolling speed was set to 50 pixels per second, which is based on a small test with five subjects where we asked their preferable auto-scrolling speeds from 25 [pix/sec], 50 [pix/sec], and 75 [pix/sec]. The experimental system displayed components on 240×240 [pix] screen of a mobile phone in the same fashion as on a PC screen. The components used in this experiment were extracted from the Web sites of MSN Japan (<http://jp.msn.com/>) and Osaka University (<http://www.osaka-u.ac.jp/en/guide/about/history.html>).

Experimental Results The result of the questionnaire survey confirmed that auto-scrolling is not suitable for “text” and “text & image” components. This is because people’s reading speeds strongly depend on individuals. Moreover, some subjects mentioned that while reading a “text & image” component using auto-scrolling, they could not look at images within the component. On the other hand, it was confirmed that auto-scrolling is suitable for “image,” “link & image,” “vertical link set,” and “tiled link set” components. However, some subjects said that it was hard to view these components when the scroll path is set to zigzag. Regarding “image” components, ten subjects stated that an auto-scrolling speed of 50 [pix/sec] was effective, while for the “link set” components, subjects’ opinions are divided; three subjects thought the auto-scrolling was too fast, and another three subjects thought it too slow.

Preliminary Experiment 2

Preliminary experiment 1 showed that auto-scrolling is suitable for “image” and “link set” components. It also showed that the auto-scrolling speed of 50 [pix/sec] tended to be accepted for the “image” components, and thus we decided to use this speed for “image” components. However, the proper speed for the “link set” components had not been determined in experiment 1. Therefore, in preliminary experiment 2, the most appropriate speed and zoom ratio of auto-scrolling were examined. We asked 20 subjects (16 males and four females ranging from 20 to 30 years old) to use auto-scrolling to read “link set” components that were extracted from five Web sites, including news and shopping sites. In the experimental system, the subjects can adjust the speed and zoom ratio of the auto-scrolling. Specifically, the basic auto-scrolling speed is set to 50 [pix/sec], and subjects can change it by 10 [pix/sec] increments/decrements. The basic zoom ratio is set to 1.0, and subjects can change it by 0.1 increments/decrements. Scroll paths are determined based on the components’ shapes in the same way with the experiment 1. In this experiment, each subject selected the speed and zoom ratio that he/she thought the best.

Experimental Results Regarding the zoom ratio, answers from the subjects were collected for different 320 cases, and these show that in 98.2% of cases, sub-

Table 2.1: Preferable zoom ratio

Zoom ratio	< 0.9	from 0.9 to 1.1	> 1.1
Number of answers	6	314	0

jects stated that the best zoom ratio was 0.9 to 1.1 (Table 2.1 shows the number of answers for the preferable zoom ratio to view a component). That is, most subjects liked the default zoom ratio. The reason is twofold; first, since subjects aimed to view components (which is different from to look for a component from a Web page), the zoom ratio needed to be preserved for visibility, i.e., subjects required the certain scales to view contents of the components. The other reason why the subjects disliked small or large zoom ratios is that, although JPEG images are not suitable for scaling up and down, the system makes a screenshot of a Web page in a JPEG format and displays it for auto-scrolling; therefore, the appearance (resolution) of these images is deteriorated.

Figures 2.3 and 2.4 show the preferable auto-scrolling speeds selected by the subjects. The appropriate scrolling speed, which users decided the best speed for auto-scrolling, in the vertical direction were about 60 [pix/sec] on average for all components. A t-Test was performed with a significance level of 5%, and it was found that there was no significant difference among each pair of components. Since letter strings within a component are displayed horizontally, the vertical auto-scrolling goes through the letter strings one by one, like the end credits of a movie. Thus, users could easily check each letter string one by one with auto-scrolling. This is why users' reading styles, i.e., favorite scrolling speeds, did not differ much among subjects.

On the other hand, for horizontal auto-scrolling, subjects' evaluations were divided. There were two groups of components; the favored scrolling speed in the horizontal direction was about 30 [pix/sec] on average for one group, and about 50 [pix/sec] for the other group was. Again, a t-Test with a significance level of 5% was performed, and it was found that there was a significant difference between the two groups. Since the auto-scrolling direction is same as that of letter strings within a component, many letter strings appear at the same time,

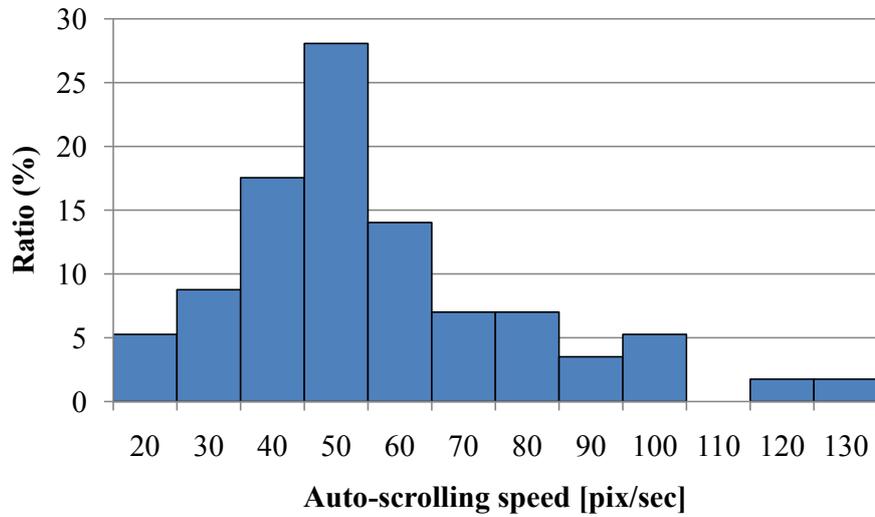


Figure 2.3: Preferable auto-scrolling speed for vertical scrolling

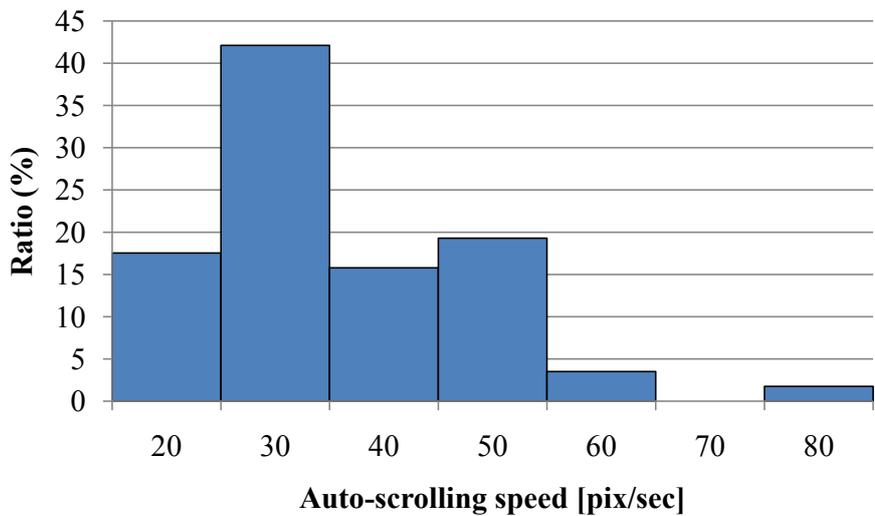


Figure 2.4: Preferable auto-scrolling speed for horizontal scrolling

which means users have to look at them one at a time. In this case, the difference in the favored scrolling speed may depend on whether or not a user can guess the following letter strings. For example, if a user browses a familiar page, the user can guess the next string from the currently displayed one or from part of the next

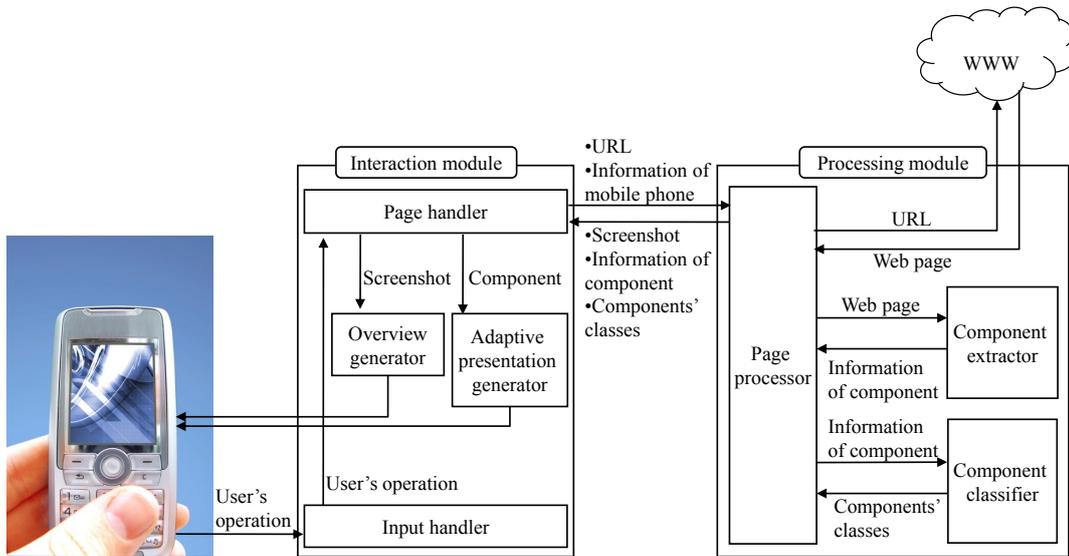


Figure 2.5: System structure

one, and thus, they prefer a faster speed because it is boring to keep watching known information with slow speed. Moreover, the preferred auto-scrolling speed depends on the user's reading style, e.g., whether the user thoroughly reads each string or skims through the component.

2.3 Design and Implementation of Proposed System

2.3.1 Design of System Structure

Based on the results of these preliminary experiments, a system was designed and implemented that offers adaptive component presentation. Figure 2.5 shows the system structure.

The interaction module is composed of an input handler, a page handler, an overview generator, and an adaptive presentation generator.

Input handler: This module handles a user's input events, and sends this information to the page handler.

Page handler: This module handles the screenshot of a Web page and information of components and their classes that were received from the page processor in the processing module. When it receives information of user's operations from the input handler, it sends the screenshot and the information of component selected by the user to the overview generator and adaptive presentation generator, respectively. In addition, it also handles the information of link selection by the user, and requests the corresponding Web page to the page processor in the processing module, by sending the URL of the link and information of the user's mobile phone's screen.

Overview generator: This module displays an overview of a Web page created from the screenshot received from the page handler.

Adaptive presentation generator: This module adaptively presents components according to their characteristics.

The processing module is composed of a page processor, a component extractor, and a component classifier, which can run as a proxy server.

Page processor: This module receives a Web page from a WWW server that corresponds to the URL sent by the page handler in the interaction module. It requests the component extractor to extract components from the page, and then requests the component classifier to determine each component's class. Lastly, it returns the screenshot of the Web page, information of the components, and their classes to the page handler.

Component extractor: This module extracts components from a Web page (that is described in Section 1.4), then sends the component information to the page processor.

Component classifier: This module determines each component's class (that is described in Section 2.2.2), then sends this information to the page processor.

When a user selects a link on the mobile phone, the interaction module sends the URL and information on the user's mobile phone's screen to the processing

module. In response to this request, the processing module receives the corresponding Web page from a WWW server, and makes a screenshot of the page. Next, the processing module extracts components from the page, determines the class of each component, and sends the screenshot, the components' information (location, height, and width) and the component classes to the interaction module. The interaction module displays an overview of the requested page, which is created from the screenshot and the component information received from the processing module. When the user selects a component from the overview, the interaction module performs an adaptive presentation of the component. The following sections describe the details of the interaction module and the implementation of the system.

2.3.2 Web Page Overview

A user can select a component that he/she wants to read from the Web page overview. An overview is a scaled-down page that fits the screen size of the user's mobile phone (Figure 2.6 (a)). The component that a user is currently reading, i.e., where the pointer is focused, is automatically zoomed in. The user can further zoom in the component by pressing a specific key on his/her mobile phone (Figure 2.6 (b)). In addition, the user can perform the following operations:

Component selection: A user can select a component to view by pressing a specific key on the mobile phone; the selected component is then presented adaptively.

Back function: A user can go back to the Web page viewed prior to the current page.

Forward function: A user can go forward to the Web page viewed before going back to the current page.

2.3.3 Adaptive Presentation Methods

This clause describes how the adaptive presentation generator adaptively presents components, which is determined based on our preliminary experiments.



(a) Overview

(b) Zooming in

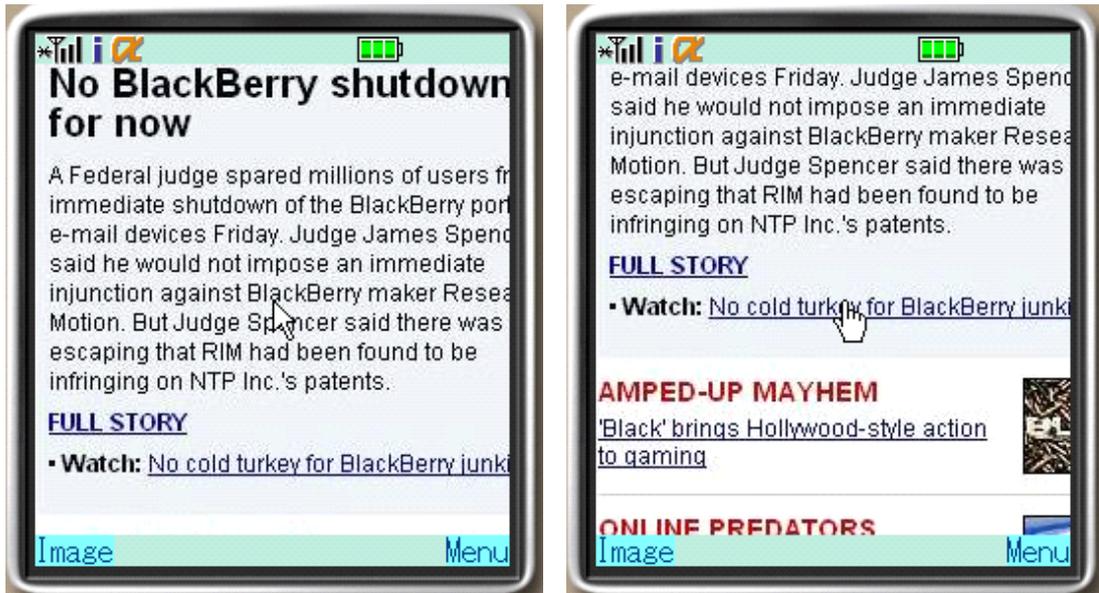
Figure 2.6: Overview of a Web page

Text Components

Users read “text” components by manual scrolling, because the preliminary experiments showed that people’s reading speeds strongly depend on individuals. Figure 2.7 (a) shows a screenshot of a mobile phone displaying a “text” component, where the arrowhead in the center is a pointer. The shape of the pointer changes when the pointer is on a letter string of a link as shown in Figure 2.7 (b), where the user can select the link. In addition, the system offers a “jump” function. The users can jump to the top of the next sentence by pressing a key on the mobile phone when they finish reading the current sentence.

Image Components

An “image” component is presented by auto-scrolling at a speed of 50 [pix/sec] as determined in the preliminary experiments. The preliminary experiments also showed that auto-scrolling in a zigzag manner is hard for users to follow. Therefore, in our system, if the size of a component is larger than that of the mobile phone’s screen, the component is zoomed out until its height or width fits the screen’s height or width. However, based on the results of the preliminary exper-



(a) Manual scroll

(b) Selecting a link

Figure 2.7: Adaptive presentation of a “text” component

iments, it preserves the zoom ratio from 0.9 to 1.1 to maintain the components’ resolution. In addition, users can stop auto-scrolling anytime, and they can scroll manually. They also can select a link in the same way as for “text” components.

Text & Image Components

The preliminary experiments showed that users cannot read “text & image” components comfortably with auto-scrolling because they cannot view images within the components as they wish. Therefore, in our system, users read sentences within “text & image” components by scrolling manually, and they can switch over images within the components by pressing a key when they want to change the image. That is, the users can switch over the images and sentences in rotation by consecutively pressing the key. In this way, they can view images associated with sentences, and then they can go back to the sentences. In addition, the system offers a “jump” function, and users can select a link in the same way as for “text” components.

Link Set Components

“Link set” (link & image, vertical link set, and tiled link set) components are presented using auto-scrolling because the preliminary experiments confirmed its effectiveness. The system determines the speed and zoom ratio of the auto-scrolling function according to the content of the component. Regarding the speed, the system should show an appropriate amount of information while auto-scrolling, which is the amount of information that a user can recognize while scrolling. Therefore, we determine the speed using the Equation (2.1), which is based on the definition of auto-scrolling speed proposed in [56]:

$$Speed[\text{pix}/\text{sec}] = \frac{\alpha}{ID \cdot Breadth}. \quad (2.1)$$

Here, α [characters/sec] is the number of characters per second that a human can recognize during auto-scrolling, and ID (information density) [characters/pix²] is the number of characters per unit area, i.e., 1 [pix²], supposing that all the characters within the component are uniformly distributed. $Breadth$ [pix] is the width (height) of the component in the orthogonal direction using auto-scrolling, which is represented by the following formula:

$$Breadth = \begin{cases} \text{component's width} & (\text{vertical scrolling}) \\ \text{component's height} & (\text{horizontal scrolling}). \end{cases}$$

Therefore, the auto-scrolling speed is set so that the number of characters that the user can recognize for a second is presented at the rate of 1 second intervals.

The preliminary experiments showed that when searching for an objective link, users prefer different auto-scrolling speeds depending on the component class and their individual reading styles. Therefore, the value of α is determined based on the results of the preliminary experiments to take into account the component classes and users' reading styles. Accordingly, users should select that appropriate value of α to adapt to individual reading styles so that they can set an appropriate scrolling speed.

Next, the path and zoom ratio of auto-scrolling are determined as follows. If the width (height) of the component is larger than the height (width) of mobile phone's screen, the path is set in the horizontal (vertical) direction. If the size of component is larger than the mobile phone's screen, the component is zoomed

out to avoid zigzag auto-scrolling. Based on the results of the preliminary experiments, it was determined that the zoom ratio should be limited from 0.9 to 1.1 to maintain the components' resolution.

The system also offers rewind and fast-forward functions. In addition, users can stop auto-scrolling and manually scroll anytime they want, and can select a link in the same way as for "text" components.

2.3.4 Implementation of Prototype System

We implemented the system based on the above design. Since mobile phones lack processing power at this time, the processing module was implemented on a PC with Windows XP, and coded with Visual C# and PHP (Hypertext preprocessor). The interaction module was implemented on a mobile phone (SH900i [75]) that was equipped with an i-appli developed by NTT Docomo, and coded in Java. Within the 240×320 [pix] screen of the mobile phone, i-appli can only use a 240×240 [pix] area.

Since the technical progress of mobile phones is remarkably rapid, it is expected that it will be possible to implement the processing module on a mobile phone in the very near future.

2.4 User Experiments Using Real Data

This section shows results of user experiments conducted to examine the effectiveness of the implemented system.

2.4.1 Accuracy of Component Classification

Before the user experiments, the accuracy of the component classification was first examined. Components were extracted from 25 popular Web pages such as Yahoo! Japan (<http://www.yahoo.co.jp/>) and CNN.co.jp (<http://www.cnn.co.jp/>), and each component's class was determined using the implemented system. The accuracy of the component classification was manually checked.

The system classified 218 components correctly, and 4 components incorrectly. Thus, the accuracy was about 98%, which is high enough for practical use. The

four miss-classifications include a case where a “tiled link set” component was classified as “link & image.” This happened because the system treats all images equally, e.g., a large image, which is the main content of a page and a small icon image. To solve this problem, a more sophisticated image classification method, such as [55], should be applied to the system.

2.4.2 User Experiments

We conducted two user experiments to examine the effectiveness of our system. For comparison, a system was also implemented that offers traditional style Web browsing on a mobile phone (SH900i), which was equipped with i-appli developed by NTT Docomo, using Java. This system is called as the “conventional system,” whereas our system described in Section 2.3 is referred to as the “adaptive presentation system.” The conventional system presents Web pages as they are, in the same way as those on PCs, within a 240×240 [pix] area of the mobile phone’s screen. A user can scroll a page by pressing the direction pad of the mobile phone. If a user presses the direction pad once, the page scrolls by 5 [pix], and if the user keeps pressing it, the page keeps scrolling up to a speed of 150 [pix/sec] so that the user can scroll through a part he/she wants to skip through. We set this speed based on the fact that users are used to scroll through an unwanted part of information by keeping pressing the direction pad through the experience of reading text messages by mobile phones, and the speed is usually faster than 50 [pix/sec] (which aims to let users read information). If we set the speed to 50 [pix/sec], users would feel irritated since the speed is much slower than their expectation. Prior to the experiment, we asked three subjects to adjust the speed of the conventional system, and decided to use 150 [pix/sec]. We also compare a commercial Web browser for mobile phones, which adopt the popular presentation style that restructures Web pages so that users can browse them by only using vertical scrolling (which is referred to as “commercial browser” in the following). The commercial browser presents the restructured pages also on a 240×240 [pix] area of the mobile phone’s screen. It also has the functionality that when the user keeps pressing a direction pad, the Web page keeps scrolling.

Experiment 1

Experimental Procedure We collected 30 subjects (16 males and four females from 20 to 30 years old) who are familiar with using mobile phones and divided them into three groups (ten subjects in each group), where group 1 used the adaptive presentation system, group 2 used the conventional system, and group 3 used the commercial browser. Before starting the experiments, the subjects in the first group were given an opportunity to read the KYODO NEWS (<http://www.kyodo.co.jp/>) to practice using the system. After that, the subjects in all groups were asked to search and select a particular link on a Web page about cooking (<http://www.ajinomoto.co.jp/recipe/??top3=recipeIndex2>) (*task 1*). Next, on the page linked from the page in task 1, the subjects were asked to read texts (*task 2-1*) and to select a link associated with a dish they like (*task 2-2*). The experiments involved situations where a user searches an objective link in a large Web page in task 1, where a user reads a “text & image” component in task 2-1, and where a user reads a “link set” component in task 2-2.

The subjects’ operation logs were recorded on the adaptive presentation and conventional systems. These logs contain information about each user operation (the key pressed by the user and the time) and the position of a Web page that was displayed on the mobile phone at the rate of every 0.1 secs. Regarding the commercial browser, since it is not possible to modify commercial products, we cannot automatically record logs. Therefore, we count subjects’ numbers of operations manually, which are double-checked using a video capturing each subject’s experiment scene. Additionally, orbits of use in commercial browser become only straight lines because of the reconstruction of Web pages, thus we might not get useful insights.

Results and Discussion Figure 2.8 shows the number of operations by each subject in task 1, where values are categorized based on used systems and shown in ascending order. The result shows that most subjects in group 1 (adaptive presentation system user) could find the objective link with fewer operations than those in group 2 (conventional system user) and 3 (commercial browser user). The medians of the number of operations on the adaptive presentation system, con-

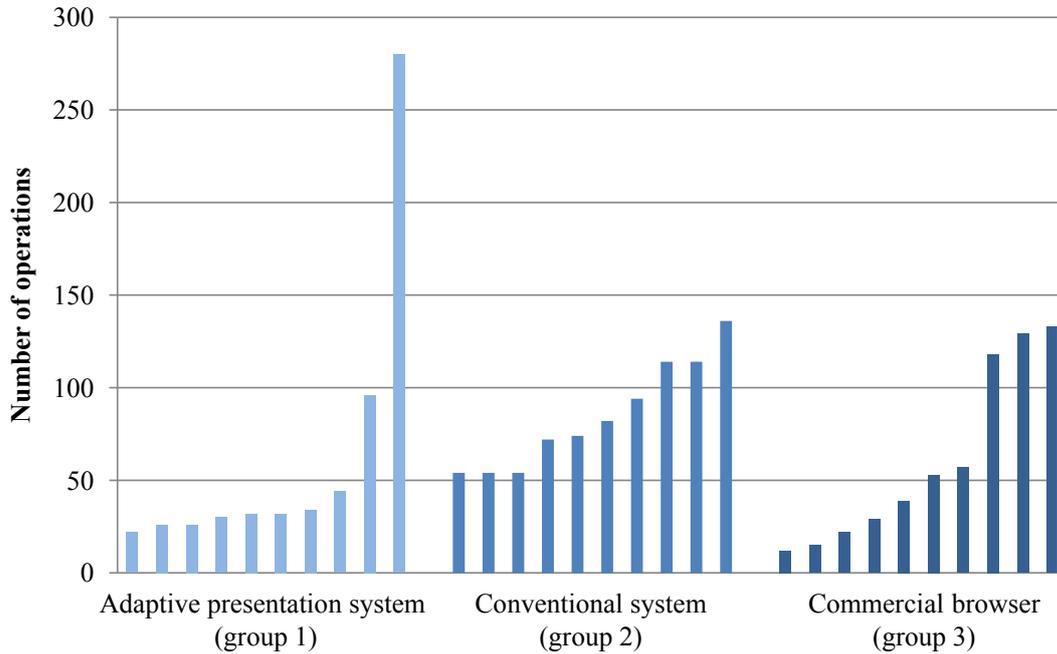


Figure 2.8: The number of operations in task 1

ventional system, and the commercial browser were 32, 78, and 46, respectively. A Mann-Whitney U test [59] with a significance level of 5% was performed for the experimental results, which is a widely used non-parametric statistical significance test that is suitable for data sets that contain large differences among individuals, like the case of the experiment reported here. As the result, it was found that there was a significant difference ($U=17.0$, $U\text{-table}(10,10)=23.0$) between the results in the adaptive presentation system and those in the conventional system, while there were no significant difference between the adaptive presentation system and the commercial browser. These results show that our adaptive presentation system costs operations of at most the same amount with the commercial browser that provides quite simple operations, e.g., vertical scrolling. We observed that the subjects used the commercial browser greatly reduced the number of operations by keeping pressing a direction pad, while it caused rapid scrolling in vertical direction. Some subjects commented that this high-speed scrolling disturbed them to browse the page.

We examined the scroll orbits of two subjects, who performed the most number of operations with the adaptive presentation system, and found that they repeated scrolling and zooming many times. After the experiment, we interviewed them and obtained comment that they could not find the component containing the objective link because the zoom ratio of the overview was too small, thus it was difficult to recognize the characters. Some other subjects who used the system also commented that the zoom ratio should be modified. This suggests the appropriate styles of overview can be improved, as we will describe in Chapter 4.

We further analyze subjects' behaviors using their browsing orbits on experimental pages. We should note here that we do not discuss the commercial browser, since we could not record logs on the commercial browser to avoid modifying commercial products. Also, orbits of use in commercial browser become only straight lines because of the reconstruction of Web pages, thus we might not get useful insights. Figure 2.9 shows a typical scroll orbit of a subject using the adaptive presentation system, and Figure 2.10 shows the same for the conventional system in task 2-1 and task 2-2. The orbit in the first case was much simpler; it was a purely combination of horizontal lines and vertical lines. Since the adaptive presentation system provides an overview of a Web page, where the boundary lines of components are clearly shown, and adaptive presentation is performed individually for each component, users can easily grasp the content and structure of the whole page. The shape of the component, such as horizontally and vertically long, enables subjects to recognize which direction they should scroll to, and hence, the scroll orbit was simple. On the other hand, the orbit using the conventional system was highly complex, with the subject traveling back and forth among different parts of a Web page many times. This is because the subjects using the conventional system were not given information of the structure of the page and their current position within the page, thus they scrolled back and forth to understand it. Moreover, since the conventional system does not show the boundaries among components, the subjects could not recognize which direction they should scroll to.

Figure 2.11 shows an area browsed by a subject engaged in task 2-2 using the conventional system. Specifically, the gray area indicates the area that was

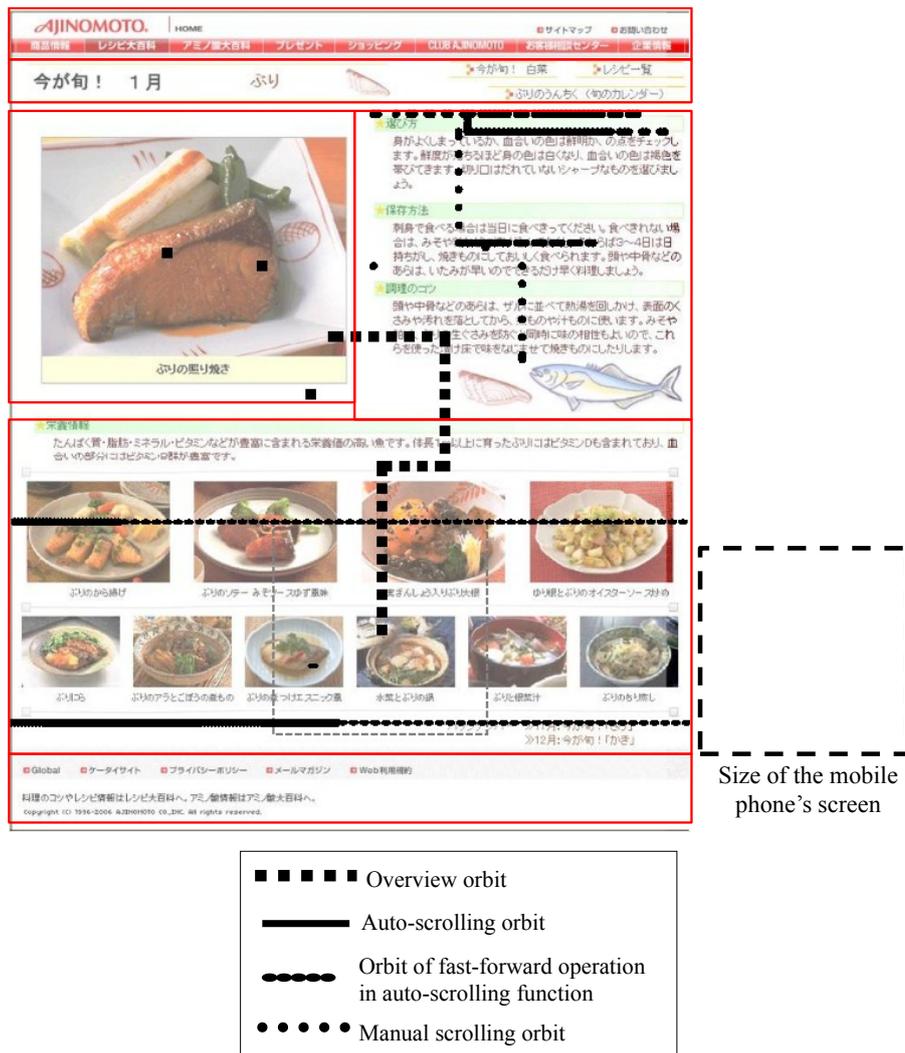


Figure 2.9: Scroll orbits of a subject using the adaptive presentation system (the thick rectangles show component boundaries)

displayed on the screen of the subject's mobile phone. It can be seen that within a component, there is a part that the subject did not browse. From this result, it is confirmed that in the conventional system, the subjects could not recognize either their current positions within a page or boundaries among components. This problem never happens when browsing Web pages on desktop PCs, and it is a serious issue for mobile phone users because they might not be able to browse full information on Web pages, i.e., they might miss beneficial information when browsing the pages by mobile phones. The adaptive presentation system

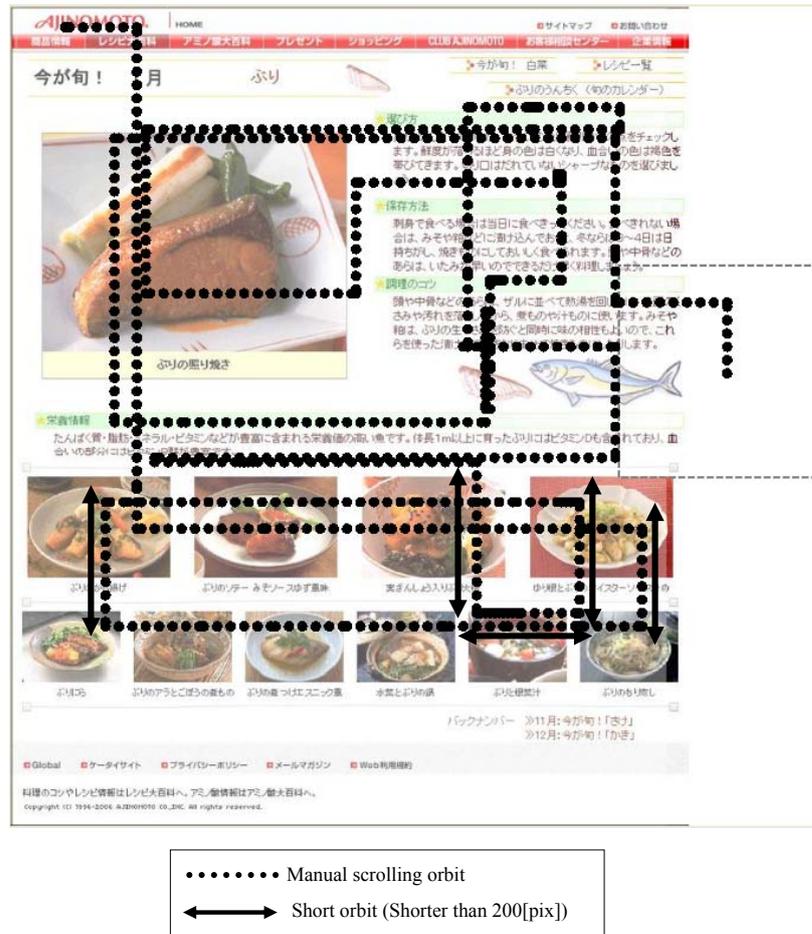


Figure 2.10: Scroll orbits of a subject using the conventional system

can avoid this problem by providing an effective overview and by individually presenting each component.

The scroll orbits of all subjects were analyzed. From this analysis, it is found that the subjects in group 1 engaged in task 2-1 effectively used the image switching function described in Section 2.3.3 to view images. Some subjects viewed images first, others viewed them after reading texts. On the other hand, it was also found that subjects using the conventional system viewed images when they first found them on the screen even while they were reading texts. Moreover, some of them missed some images. These subjects said that they did not notice these images because the images did not appear on the screen while scrolling. Since a mobile phone's screen can display only a part of a Web page, users cannot know



Figure 2.11: Area browsed by a subject using the conventional system

what contents are contained in the page until they scroll to the part where the contents exist.

The orbits of all the subjects engaged in task 2-2 using the conventional system contained many short line segments with lengths less than 200 pixels, as shown in Figure 2.10, where the average number of short line segments for each subject was 4.6. These short line segments indicate that the subjects often scrolled to one edge of a page to recognize their current position within at page, and they often scrolled to an adjacent component to establish the boundary with the component. Obviously, the shorter line segments appear, the more tiresome scroll operations

the user is forced to perform. Many subjects who used the conventional system said that during scrolling, they often got lost within the page and could not find out where to scroll. These subjects had to scroll around to recognize the current position, and they complained that these scroll operations were really frustrating. On the other hand, most orbits using the adaptive presentation system contained only long horizontal and vertical line segments, as shown in Figure 2.9. Since the subjects could view components by auto-scrolling, they did not lose their locations whilst scrolling. Consequently, this result confirms that auto-scrolling can reduce tiresome scroll operations.

Experiment 2

Experimental Procedure In this experiment, 18 subjects (14 males and four females from 20 to 30 years old) were asked to use all of the adaptive presentation system, conventional system, and commercial browser, and to answer a questionnaire to compare their usability. Specifically, the subjects were divided into three groups and asked to use each system in different orders for fair comparison. Before starting the experiment, all the subjects were given time to browse the Web site of KYODO NEWS (<http://www.kyodo.co.jp/>) to learn how to use experimental systems. After that, the subjects were asked to freely browse a page on each of the Web sites, Yahoo! Japan (<http://www.yahoo.co.jp/>), Yahoo! NEWS (<http://dailynews.yahoo.co.jp/>), and goo (<http://www.goo.ne.jp/>), and to select a link of interest. For each site, the subjects repeated this task sequence (free browsing and link selection) for three times. Such a Web browsing style, i.e., reading a page and selecting a link of interest on a popular portal or news site, are ordinary in daily life.

In the questionnaire survey, the subjects were asked two kinds of questions; scoring the usability of each system and free writing about the advantages and disadvantages of each system. Regarding usability scoring, the subjects were asked to respond to each of the following questions by scoring from -3 (not good at all) to 3 (very good). We set the questions so that the higher score means more positive impression.

- (i) How easy it was to find an objective link?
- (ii) How easy it was to read texts?
- (iii) How easy it was to view images?
- (iv) How easy it was to read a set of links?
- (v) How easy it was to grasp the current position within a page?
- (vi) How similar the experience was to Web browsing using PCs?
- (vii) How intuitive the system was for operation?
- (viii) How would you like to use the system after this?

Results and Discussion Table 2.2 shows the average scores and their standard deviations of the questionnaire survey. From the results, the proposed adaptive presentation system obtained very high scores for three questions: (iii) “How easy it was to view images?,” (iv) “How easy it was to read a set of links?,” and (v) “How easy it was to grasp the current position within a page?” This shows that the subjects who used the adaptive presentation system could grasp the structure of a Web page and the current position within the page using the overview. It is also shown that the subjects could effectively view images and a set of links by auto-scrolling, because they did not need manual scrolling. This fact was also confirmed from the results of the free writing question, where one subject mentioned that he could read components without manual scrolling, and it was really nice.

The adaptive presentation system also obtained a high score for question (i) “How easy it was to find an objective link?.” Generally, while searching an objective link, users judge the role of a link, and whether or not the link is related to a main content, a menu link and an advertisement, based on the structure of the page. The scoring result shows that the subjects using the adaptive presentation system could easily grasp the structure of the page from the overview and could comfortably read components owing to the adaptive presentation, i.e., could easily find an objective link.

Table 2.2: Average scores (standard deviations) aggregated all subjects

Question	Adaptive presentation system	Conventional system	Commercial browser
(i) How easy it was to find an objective link?	0.9 (1.2)	-0.8 (1.7)	-0.5 (1.8)
(ii) How easy it was to read texts?	0.7 (1.3)	-0.8 (1.9)	1.3 (1.6)
(iii) How easy it was to view images?	1.4 (1.4)	-0.3 (1.7)	0.2 (1.7)
(iv) How easy it was to read a set of links?	1.9 (0.9)	0.1 (1.3)	0.2 (1.9)
(v) How easy it was to grasp the current position within a page?	2.4 (0.9)	-0.9 (1.7)	-1.3 (1.3)
(vi) How similar the experience was to Web browsing using PCs?	-0.4 (1.9)	1.1 (1.5)	-1.1 (1.7)
(vii) How intuitive the system was for operation?	-0.8 (1.6)	1.1 (1.6)	1.2 (1.5)
(viii) How would you like to use the system after this?	1.8 (0.9)	-0.6 (1.8)	0.1 (1.9)

On the contrary, the conventional system obtained high scores for two questions (vi) “How similar the experience was to Web browsing using PCs?” and (vii) “How intuitive the system was for operation?,” because it presents a Web page in the same way as on desktop PCs, and requires more naive operations. However, the conventional system obtained very low scores for other questions because the subjects easily lost their current positions while scrolling and needed to perform many operations.

The commercial browser especially obtained high score for question (ii) “How easy it was to read texts?” This result shows that scroll operation in horizontal direction is burdensome for users when reading text. However, linearizing Web pages results in producing extremely long pages, and pages’ appearances are dramatically changed. Some subjects commented that they felt uneasy during scrolling through such long pages because they could not estimate how long a page would continue and where they were viewing. Furthermore, subjects said that they were confused when they saw a familiar page in totally different appearance.

Throughout this experiment, we conclude that the adaptive presentation system is effective for Web browsing using mobile phones, supported by the fact that it obtained high score for question (viii) “How would you like to use the system after this?”

2.5 Summary

This chapter proposed the adaptive presentation system for Web browsing using mobile phones. This system reduces successive scroll operations to find a component of interest by providing an overview of the page. In addition, the system adaptively presents the component according to its component class, and thus, users are required fewer operations to view it. The results of user experiments showed that the adaptive presentation system can reduce scroll operations to look for a component by showing the overview of the page and operations to read the component by adaptively presenting it according to its characteristics. It was also confirmed that the proposed system can reduce the possibility that users miss information in a Web page.

As part of future work, we further consider the distribution of contents, i.e., text, links, and images, in a component during auto-scrolling. Although we assumed that the characters are uniformly distributed in a component when determining the auto-scrolling speed, and the speed becomes constant in the component, it should be adjusted according to the actual character distribution, i.e., the speed becomes slower in an area where many characters densely exist. Furthermore, we consider more sophisticated way to set scroll paths. The preliminary experiment showed that users do not prefer a zig-zag scrolling, however, to keep

visibility of information, a zoom ratio has to be preserved to around 1.0. To reduce burden on the zig-zag scrolling, we can take the distribution of contents into account, i.e., setting scroll paths so that they are along with the direction that users view the contents.

Chapter 3

Design and Implementation of a Web Browser for Mobile Phones to Reduce Mobile Users' Cognitive Load

3.1 Background

In Chapter 2, we proposed a Web browsing system that presents components of a Web page adapted to their characteristics, where we assumed that users browse Web pages in somehow fixed situations, such as sitting on a chair at their home. In this chapter, we focus on users who are in various situations. Considering that mobile phones are getting smaller and more portable, it is natural that dynamic browsing situations are also possible, such as walking a street and riding a train while commuting. Oulasvirta et al. [68] showed that users' cognitive loads are different according to situations, which implies that Web browsing functions that satisfy users' available cognitive resources are different based on users' situations. For example, when users are searching a specific article from a Web page while walking outside, a presentation style that requires minimal operations is suitable, since users have to pay much attention to their environment compared to indoors. As we described in Section 1.2.1, although there are many studies that

aim to provide comfortable Web browsing experience using mobile phones, none of them considered the possibility that presentation functions that satisfy users' requirement are different according to users' browsing situations, which we focus on this chapter.

It is usually difficult to precisely detect or predict users' situations by using some sensing devices, because to fully equip a phone with accurate sensors is very costly. Furthermore, preferences of functions on each browsing situations would be user dependent. Therefore, we think that it is reasonable and effective to provide functions so that users can easily select a presentation style by themselves to adjust their distribution of cognitive resources according to their situations. In this chapter, we propose a novel Web browsing system called *OPA Browser (One Push Aegis Browser)*, in which keys on the telephone keypad of a mobile phone have different functions for presenting Web pages. We selected functions to reduce users' cognitive load on fundamental operations on Web browsing so that they can release enough cognitive resource to interact with their current environment while browsing Web pages comfortably, which is based on RCF (Resource Competition Framework) proposed in [68] and informal interviews of our previous user experiments. To our best knowledge, this is the first Web browser for mobile phones that considers the variety of users' Web browsing situations.

In the following, Section 3.2 explains the design of the proposed OPA Browser. Then Section 3.3 describes a user experiment that we conducted in a real situations by asking participants to use OPA Browser for three days in their daily life. Finally, we discuss concluding remarks and future work in Section 4.5.

3.2 OPA Browser and Presentation Methods

In this browser, we aim to provide various presentation functions so that users can easily select ones adapted to their situations to adjust their cognitive resource distribution. Since most mobile phones commonly have a direction pad and telephone keypad, and the direction pad is usually reserved for scrolling operation, we assigned different presentation functions on each key of telephone keypad of a mobile phone, and users can select a function by only pressing the corresponding

key. In the following, we first describe motivations on selecting functions to provide, which is based on users' cognitive resource distribution during Web browsing using mobile phones in various situations, and then introduce each function in detail.

3.2.1 Fragmented Attention of Mobile Users in Dynamic Situations

When mobile users are in a dynamic situation, their cognitive resources are reserved partly for monitoring and reacting environments, and partly for interacting with mobile devices. Kristoffersen and Ljungberg [50] describe that mobile devices reserve users' physical and attentional capabilities from other tasks required for mobility. They observe that mobile users need to "make a place" for the device in a taxing situation (e.g., drivers may need to stop the car to release enough resources to operate a phone). Similar observations are made by Pascoe et al. [70] and by Lumsden and Brewster [53], where they mention that there is often a conflict between interaction with environments and mobile devices.

Oulasvirta et al. [68] propose RCF (Resource Competition Framework) that relates mobile task demands to users' cognitive resources, which is our basis of the function design of OPA Browser. Table 3.1 shows cognitive faculties, their resources, and limitations that have practical significance for mobile Web browsing. In this manner, users' actions on Web browsing are defined as combinations of the cognitive faculties as follows:

- Scrolling: consists of cognitive faculties 1, 2, 3, 4, and 5
- Deciding scroll path: consists of cognitive faculties 4, 5, 6, 7, 8, and 9
- Searching information from screen: consists of cognitive faculties 2, 3, 4, and 6

They conduct user experiments with 28 subjects in typical mobile situations, and conclude that subjects' cognitive resource (percentage of time spent to attend the environment) dwells in the environment much more while outdoors than indoors, and more crowded places than calm places as Table 3.2 shows.

Table 3.1: Cognitive faculty, the provided resource, and known limitation in mobile Web browsing [68]

#	Cognitive faculty	Provided resource	Limitations
1	Motor control	Sequencing, timing, control, and finalization of motor actions	Fluent execution dependent on learned procedural skills.
2	Sensation	Intake of external stimuli	Acuity, accuracy
3	Perception	Organization of sense data	Uniformity of perception
4	Attention	Search, selection, and integration in and across modality-bound representations	Limited spatial span, limited object span
5	Central executive	Meta-level control of cognitive operations; selection, inhibition, updating, and shifting	Serial processing
6	Working memory	Retention of previously attended information	Short relation span, small capacity
7	Episodic memory	Mental “time travel”	Distortions, slow, and effortful access
8	Semantic memory	Fact knowledge	Susceptibility to interference
9	Thought	Conscious manipulation of Working memory materials; abstraction, inference, reasoning, and problem-solving	Uniformity of thought, reliance on heuristics

Based on these studies, we design functions to reduce the amount of cognitive resource that mobile Web browsing occupies in dynamic situations as listed in Table 3.2, since it becomes burden in cognitively busy situations. First, to reduce the cognitive load on scrolling, we further classify scroll operation into two:

Table 3.2: Percentage of time spent for attending the environment [68]

Situation	%
Quiet laboratory	5
Crowded metro car	14
Crowded bus	16
Crowded station	20
Escalator	20
Cafeteria	22
Quiet street	23
Metro platform	24
Busy street	51

scrolling on a Web page and on a component. To reduce the load on scrolling the page, it is effective to enable users to jump to components around, since components are information blocks of the same sort, and thus users can cue information blocks one by one using this function. In this manner, we can reduce the cognitive load of faculties 1, 4, and 5. On the other hand, to reduce the cognitive load on scrolling a component, the most naive way is to enlarge the unit of scrolling, where users can control scrolling operation. This reduces the cognitive load of faculty 1. We also provide the auto-scrolling function to further reduce the cognitive load on scrolling, in which users browse a component by totally passive manner. This greatly reduces the cognitive load of faculty 1, while increases the load of faculty 4, since users cannot control the movement of screen and passively view the automatically scrolled screen.

Second, to reduce the cognitive load on deciding a scroll path, showing the whole page structure is effective so that users can easily look for information of their interest from the entire page. For this aim, we decided to provide three kinds of functions, e.g., presenting the scaled-down page, dividing the screen into four parts and presenting contents on each of them (we refer this to “tile view” in the following), and providing a magnifying glass. Presenting the scaled-

down page is the conventional method to show the entire page on mobile phone screens at a glance, while its scale is sometimes too small to show details of contents. Therefore, we also provide the other two functions, where providing the magnifying glass is also a popular method to show the details of contents on scaled-down documents, images, and Web pages on PC screens [25], which is widely adopted by commercial products, such as Adobe Acrobat Reader [1]. The design of the tile view is based on the comments that were obtained from the informal interview of the user experiments conducted in Chapter 2. Some subjects mentioned that it would be great if they could view some components at a time with a bigger scale. These functions effective to reduce the cognitive load of faculty 4, where the reduction rate of each function would depend on users. In that informal interview, some subjects also mentioned that when they already have what information they want in mind, a word search function would be effective since it enables them to directly jump to the content. Although the cognitive load for inputting text using the telephone keypad would be large, the word search function reduces cognitive load of all other faculties required to decide a scroll path.

Finally, there is no direct solution to reduce the cognitive load on searching information on a small screen, since it is difficult to predict what information users are currently looking for on the Web page. We can, however, “assist” users to search information. There are basically three kinds of information on Web pages, e.g., text, links, and images, and thus we provide functions to assist users to search them. Specifically, we provide two functions; jumping to image and relevant component functions, which aim to assist to search images and text/links, respectively. In this manner, we can reduce the cognitive load of faculties 2 and 3. In the previous user experiments conducted in Chapter 2, we confirmed that the subjects could use the image cue function and effectively looked for and browsed images, which are possibly missed due to the limited presentation area. Therefore, we decided to provide the function in OPA Browser as well. As for jumping to a relevant component function, we aim to reduce the cognitive load after they find information of interest and further look for related information in the page, which results in reducing the cognitive load of faculty 6. These functions are also effective to enhance users’ Web browsing experience, which is possibly limited

since users can only see a small portion of the page due to the small screen of mobile phone and the entire view of the page is unavailable, as the experimental results in Chapter 2 showed.

Table 3.3 summarizes OPA Browser's functions assigned to telephone keypad. Here, if displayed contents change suddenly and drastically, users cannot understand the system's operations, and thus get confused in browsing. Therefore, OPA Browser uses auto-scrolling for "jump" in functions 3, 4, 5, and 8 so that users can intuitively understand the operation. As described in Table 3.3, overview and word search functions further provide different functionalities, and users can select one of them. As for the overview, users can select two different presentation styles; they first select key 1 (overview) and then selects one of the two presentation styles on the displayed menu. As for the word search, they first select key 7 (word search) and then select one of four search functions from the displayed menu, e.g., same word, synonym, antonym, and input word searches. Furthermore, the keys of 0, #, and * corresponds to bookmark, back, and forward functions to help users browse Web pages as with ordinary Web browsers.

We should note that the allocation of functions on keys of the telephone keypad is arbitrary. Although we allocated more fundamental functions on upper keys so that users can easily memorize and access them, optimal allocations would be different among users. Optimizing the allocations depending on users is our future work.

3.2.2 Sample User Scenarios

We describe examples of user scenarios and show functions that are expected to be selected by the user to browse Web pages adapted to his/her situation.

First of all, a user launches OPA Browser stored in his/her mobile phone and inputs the URL of a Web page that he/she wants to browse or selects a Web page from bookmarks. When the selected page is downloaded, the page is presented in the same way as on PCs. Then, he/she selects functions according to his/her situation and browses the page.

Suppose a user who have lunch with browsing Web pages for preparation of her afternoon meeting. She has to concentrate on browsing to some degree

Table 3.3: Functions assigned to telephone keypad

key	Function	Outline of the function	
		Menu	Outline
1	Overview	Scaled-down view	A scaled-down page that fits the screen size is displayed.
		Tile view	The screen is divided into four sub-screens and a part of components is displayed on each sub-screens.
2	Scrolling per screen size	–	User can scroll a page at the unit as the mobile phone's screen size.
3	Jump to the previous component	–	User can jump to the previous component (block of related information).
4	Jump to the next component	–	User can jump to the next component.
5	Jump to an image	–	User can jump to an image within a page sequentially.
6	Magnifying glass	–	User can magnify a content with a magnifying glass on the overview of a page.
7	Word search	Same word search	User can search words that match with a link that he/she now focuses on.
		Synonym search	User can search synonym words of a link that he/she now focuses on.
		Antonym search	User can search antonym words of a link that he/she now focuses on.
		Input word search	User can search words that match with a word that he/she input.
8	Jump to a relevant component	–	User can jump to a relevant component.
9	Auto-scrolling	–	User can browse a component using auto-scrolling.

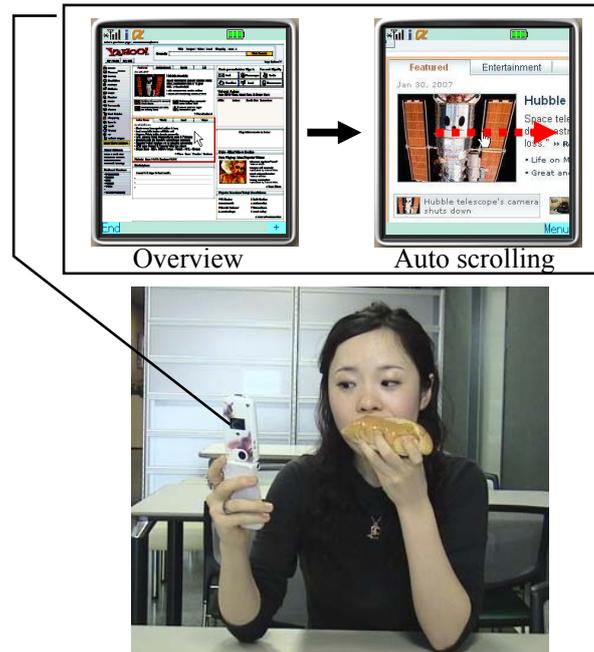


Figure 3.1: User browsing Web pages having lunch

while eating lunch. In this case, she first uses overview (function 1) to grasp the entire page structure and find a content quickly (see Figure 3.1). When she finds the content of interest, she switches to auto-scrolling (function 9). While auto-scrolling, she eats lunch and sometimes views the mobile phone's screen to check which part of the content is important for her. By auto-scrolling, she can view the content without manual scrolling, and thus, she can continue lunch uninterrupted. Finally when she finds the most important part within the content, she switches off all functions and reads the part carefully using manual scrolling.

On the other hand, suppose a user who rides on a train for a long time and has enough motivation to browse Web pages that he often checks using his PC. In his case, he does not need overview because he is already familiar with the whole page structure. Thus, he gets around using scrolling per screen size function (function 2) and jumping to the next/previous component function (functions 3 and 4) to reduce scrolling operations. When he finds an interesting content, he reads it carefully. If he wants to check a relevant content within the page, he uses the function of jumping to the relevant component (function 8).



(a) Scaled-down view

(b) Tile view

Figure 3.2: Two styles of overview

3.2.3 Web Page Overview (Function 1)

For mobile phone users, it is difficult to grasp the entire structure of a Web page because a mobile phone's screen presents only a small part of the page. Users usually recognize the role of each content area (e.g., a main content, a menu of the page, and an advertisement) based on the structure of the page. Users also decide in which direction to scroll from the structure of the page, and thus, if they cannot grasp the page structure, they often lose their way on the page. To solve this problem, presenting an overview of the page is effective. Our system provides the following two styles of overview.

Scaled-down View:

A scaled-down page that fits the screen size of users' mobile phones is displayed, so that they can grasp the structure of the whole page at a glance. Figure 3.2 (a) shows an example of a scaled-down page. In this image, there is a pointer similar to a mouse cursor. Users can move this pointer by operating the direction pad and scrolling the scaled-down page. When users select a content specified by the pointer, it is displayed on the mobile phone's screen in its original size. Here,



Figure 3.3: Zooming up

users cannot read the detail of the component because the zoom ratio of overview is often too small. Thus, OPA Browser provides here a function to zoom up a component. In the bottom of Figure 3.2 (a), there is a menu represented by a “+” mark. When users press a corresponding key (a soft key), the component that users now focus on is consecutively zoomed up (see Figure 3.3). As a result of the consecutive zooming up, the component is finally presented in its original size (if the size of the component is bigger than that of the screen, the component is scaled-down to fit its height or width with that of the screen). We designed this function to zoom up gradually so that users can easily understand the operation of OPA Browser, because if the component is suddenly zoomed up, users cannot connect the current screen with the previous presentation and might get confused.

Tile View:

This function aims to show more details of components, since the scaled-down page is sometimes too small. A mobile phone’s screen is divided into four parts (sub-screens), and a part of each component of a Web page is displayed on each sub-screen. Compared to the scaled-down view, users cannot see the whole structure at a glance from the tile view but can see components with their original sizes, and compare them all at the same time. When users press the direction pad downward, the next four components are displayed on each sub-screen. Figure 3.2 (b) shows an example of the tile view, where each sub-screen is allocated a number. If users select a certain sub-screen’s number by pressing the corresponding key, the component is displayed on the full screen with its original size.

3.2.4 Enlargement of Scroll Unit (Function 2)

This function aims to decrease the number of users' operations by simply enlarging the unit of scrolling. In this function, the unit of scrolling is set as three quarters of the screen's width/height. By doing so, previously displayed contents and the current ones overlap to some degree, and thus users can recognize the connections among displayed contents and to which direction they are now scrolling.

3.2.5 Jump to Components (Functions 3, 4)

To reduce the number of operations to scroll the entire page, functions 3 and 4 enable users to jump to the previous and next components from the currently read component by only pressing a single key. The approach of this function is based on the fact that components are blocks of relevant information, thus, users can easily decide which component to read from the fraction of the component presented while moving.

Function 2 only enlarges the unit of scrolling, but does not consider components' borders. On the other hand, functions 3 and 4 allow users to jump to the previous and the next component where the unit of jump is one component.

3.2.6 Cue Images (Function 5)

This function helps users browse images in a component, which are sometimes ignored since users do not notice their presence due to the limited presentation area as shown in the user experiments in Chapter 2. Using this function, users can jump to an image within a component sequentially. A Web page is basically composed of texts, links, and images, where images sometimes play important roles, e.g., depicting the main content of the page and representing the content of a linked page such as a banner. Therefore, the function to jump to an image within a component is useful for finding important information.

To determine the order to show images, their sizes and attributes (JPEG, GIF, and PNG) are useful properties that can be obtained with simple computation [55]. Textures and colors are also useful properties, however, it requires much

computational time, and thus it is not applicable at this moment. Therefore, OPA Browser determines priorities of images f_{image} based on Equation (3.1) and shows images according to their priorities in order.

$$f_{image} = Size + \alpha \cdot Dist + \beta \cdot W_{image}. \quad (3.1)$$

$Size$: size of the image [pix²]

$Dist$: Euclidean distance between the current position and the image

W_{image} : weight determined based on the image size and attribute

α, β : scaling parameters

Here, the weighing function W_{image} is determined based on image attributes, i.e., it becomes largest for JPEG images, since we confirmed in our previous work [55] that JPEG images regularly play the main part in a Web page, such as a photo associated with a headline news. In addition, the bigger the size of an image becomes, the higher its importance grows. Thus, W_{image} is set high for GIF and PNG images whose size is bigger than a certain threshold. We also confirmed in the previous work [55] that particular images are usually used as icons or bullet item icons, whose width and height are especially small and aspect ratio is nearly 1.0. Thus, in this function, such particular images are regarded as noise and ignored.

In this manner, users can first view an image that is located near the position they are focusing on and has greater importance than others. In the user experiments in Chapter 2, we observed that users prefer to first view images that are located near the current position. Additionally, images of higher importance in a component should be presented initially. When we implemented our browser for the first time, we used the distance (appearance order) computed in the HTML source instead of that in the rendered Web page. However, as a result of the user study using the prototype, many subjects mentioned that they could not understand intuitively which images would be displayed next. Thus, we decided to use the distance between images within the component currently focused on in the rendered page.



Figure 3.4: Magnifying glass

3.2.7 Magnifying Glass (Function 6)

While users can grasp the structure of a Web page from the overview, the zoom ratio of the overview is often too small, and it is difficult for users to read the contents of the page. To solve this problem, this function provides the magnifying glass functionality, which is one of the most common methods to show the details of contents on scaled-down documents, images, and Web pages on PC screens [1, 25]. The left side of the screenshot in Figure 3.4 shows an example of the magnifying glass function, where users can browse contents with the original size on the overview and read them. When users find a content that looks interesting, they exit from the function and read it with the original size.

Regarding the size of the magnifying glass, while users are reading a content, the large-sized magnifying glass is better for them. On the other hand, while users are scrolling, the narrow-sized magnifying glass is suitable because the large-sized magnifying glass covers its background (overview) and it is hard for users to grasp the distance they scrolled. Thus, OPA Browser determines the size of the magnifying glass Mag_{size} dynamically using Equation (3.2):

$$Mag_{size} = (W \times P) \times (H \times P). \quad (3.2)$$

W : Width of the screen
 H : Height of the screen

$$P = \begin{cases} 3/4 & \text{if } t < t_s \\ \{3 - (t - t_s)\}/4 & \text{else if } t_s \leq t < t_s + 600[\text{msec}] \\ 1/2 & \text{otherwise } t \geq t_s + 600[\text{msec}] \end{cases}$$

t : the current time
 t_s : time the user started scrolling

In this manner, when a user is viewing a content with the magnifying glass, its width and height is set to three quarter of those of the screen. When a user starts scrolling, the size of the magnifying glass becomes gradually small until its width and height becomes half of those of the screen. Regarding the size of the magnifying glass, it would be better to adjust the size based on the scroll speed, since users might not follow the scrolled distance when the speed is fast and the magnifying glass covers a large part of the screen, which is our future work.

The middle screenshot in Figure 3.4 is that of while scrolling. Furthermore, it is designed in the way that the decreasing and increasing of size of magnifying glass is smooth to let users understand the operation intuitively.

3.2.8 Word Search (Function 7)

As mentioned above, mobile phones can display only a small part of a Web page, thus, it is difficult for users to find the information of interest. Therefore, OPA Browser provides a word search function that enables users to directly find the information of interest in exchange of operations to specify query terms. Inputting a word using telephone keypad of a mobile phone requires many operations. Thus, we assigned three different search functions called “same word search,” “synonym search” and “antonym search,” in addition to the normal word search function (referred to as “input word search,” i.e., users input a word to search). In these three search functions, users can find words using a link on which they are presently focused. For example, in the “same word search,” users can find words that match with the link, while in synonym and antonym searches, they can find its synonyms and antonyms. By using synonyms as candidates for search, users can extend the search range, so that they can obtain results on more than the “same word search.” In addition, we expect the “antonym search” to be used only when users browse Web pages with enough time and look for interesting



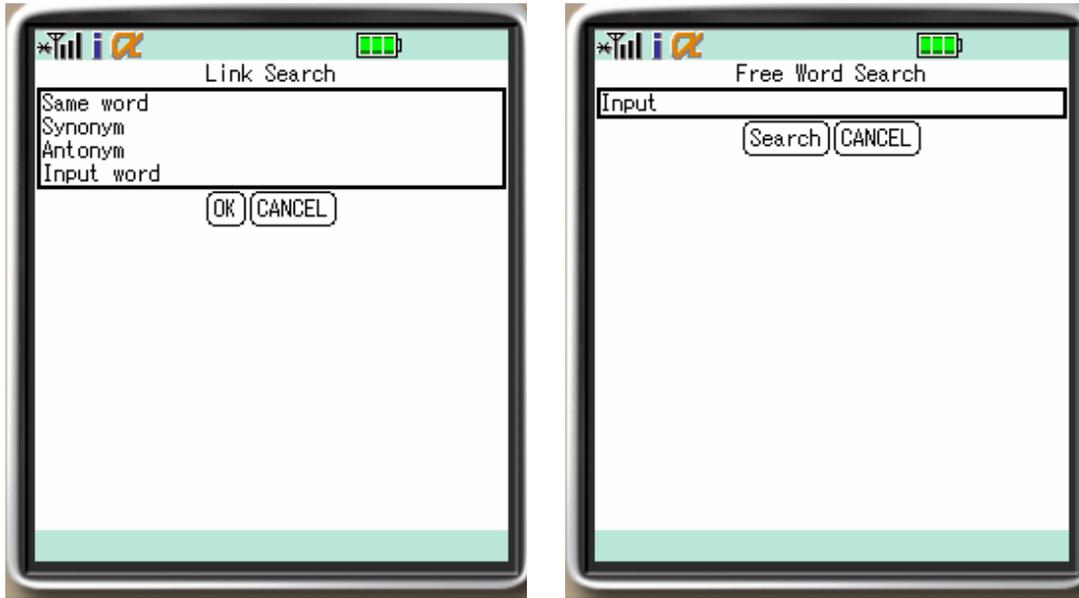
Figure 3.5: Example of word search

information through net-surfing. Due to the implementation issue, OPA Browser can use only link strings to the word search but cannot use those of plain text.

Figure 3.5 depicts the case where a user searches the word “video” by the word search function, in which the dotted arrows represent the moving direction of the mobile phone’s screen (these dotted arrows and word balloons are not presented in the real operation). In the case of the input word search, when users press key 7 and select “Input word” from the presented menu (see Figure 3.6 (a)), they input a word in the form and press the “search” button (see Figure 3.6 (b)). The screen of the mobile phone then moves by auto-scrolling to one of the link strings that contain the input word. When users press the center action button of the mobile phone, the screen moves the next resulting link string. In other searches, when users select one of the search functions from the menu and press the center action button on a link string that they want to search, the screen of mobile phone automatically moves to one of the resulting link strings.

3.2.9 Detecting Relevant Components (Function 8)

As mentioned, components are information blocks of the same sort, and Web pages presenting specific information, such as a news page of IT technology,



(a) Word search menu

(b) Input word search

Figure 3.6: Menu of word search function

contain some components that have relevant information. This function enables users to jump to among relevant components, which can enhance their Web browsing experience. We consider that this function is effective when users browse a Web page without a specific goals. If a user has specific objectives such as to read articles about a newly released mobile phone (he/she memorized the model number of the mobile phone) in a Web page, the user can easily find objective articles using the word search function (function 7). On the other hand, if a user wants to read articles about mobile phones (he/she does not have specific objectives), the user can find more articles using the function to jump to a relevant component after first reading one of the objective articles.

To determine relevant components, OPA Browser uses all text (both plain text and link strings) and images within a component to model them as a *feature vector*. OPA Browser scales a feature vector according to the frequencies of words and image attributes (e.g., JPEG, GIF, and PNG) within a component. The former is the number of times the word appears in the component, divided by the number of times the word appears in the whole page. The latter is the number of times image attribute appears in the component, divided by the number of times

image attribute appears in the whole page, e.g., when a component contains three JPEG images out of ten JPEG images in the whole page, the normalized frequency of JPEG image in the component is 3/10.

We calculate the similarity between two components $f_{similar}$ based on the cosine similarity as Equation (3.3) shows, which is a measure of similarity between two vectors by finding the cosine of the angle between them, and a common method to calculate the similarity between feature vectors [5]. Two components are regarded relevant when they have high similarity.

$$f_{similar} = \frac{\vec{x} \cdot \vec{y}}{|\vec{x}| \cdot |\vec{y}|}. \quad (\vec{x}, \vec{y} : \text{feature vector}) \quad (3.3)$$

The merit of using the cosine similarity is its simplicity of computation. Considering that the dimension of the feature vector here is small since the amount of text in a Web page is limited, the cosine similarity is enough to judge the similarity of components.

Primarily, we used only frequencies of words as a feature vector, however, we could not achieve sufficient accuracy. Because a Web page contains only a small number of words, it is difficult to determine the similarity between two components using only text. Thus, we added a structural aspect as the frequencies of image attributes, since we confirmed that images play different roles according to their attributes within a Web page [55], and components of similar appearance tend to have relevant information, such as a product list in online shopping sites. We examined the accuracy of relevance estimation using 30 various Web sites and verified that our method achieved 70% accuracy. To calculate the accuracy, we manually judged component pairs containing common topics as correct component pairs. Then, “accuracy” is defined as the ratio of the number of correct component pairs found by OPA Browser to that of all components considered relevant to it. We expect that higher accuracy can be achieved by augmenting the feature vector using more elements, and this remains as our future work.

3.2.10 Auto-scrolling (Function 9)

Users can view a component by auto-scrolling without conventional hand scrolling. We confirmed in our previous works [8,9,12,15,56] that auto-scrolling can reduce

users' scrolling operations and enables them to browse Web pages more comfortably. Since we aim to show a glance of the component here by auto-scrolling (which is different from the aim of Chapter 2, which is to show details of the component), OPA Browser determines the speed of auto-scrolling using Equation (3.4), which is based on our previous work [56]:

$$Speed[pix/msec] = \frac{c \cdot Attribute \cdot Area[pix^2]}{AoI[msec] \cdot Breadth[pix]}. \quad (3.4)$$

Attribute is the property score of the component, where the properties are HEADER, FOOTER and BODY. OPA Browser sets the speed faster for less important components e.g., HEADER and FOOTER, since they usually do not convey important information for users. *Area* is the area (*width* \times *height*) of the component, and *AoI* indicates the time period required for users to read the component, which is defined by a function of the amount of information within the component. *AoI* is determined based on the previous observation; humans, on average, need 1 [min] to read 280 words and 100 [msec] to view an image [40, 78]. *Breadth* is the width (or height) of the component that crosses the path of auto-scrolling. The Equation (3.4) is designed to show an appropriate amount of information for users at a time, otherwise users cannot follow the scrolling. Therefore, the speed is determined so that the amount of information in the scrolled area for a unit time matches the amount of information that general human can recognize for the unit time. To adjust the speed while viewing components, OPA Browser provides fast-forward and rewind functions.

3.2.11 Implementation of Prototype System

We implemented OPA Browser in a server/client architecture because the processing power of today's mobile phones is still limited. When the client sends a request for a Web page with the size of the screen, the server receives the corresponding page from a Web server and generates a screenshot of the page. Then, the server extracts components considering the size of the client's screen and determines relevant components. Furthermore, the server prepares two dictionaries, e.g., synonym and antonym dictionaries using link strings within the page to extract words from WordNet [62]. These dictionaries are specific to the

page because mobile phones could not have entire dictionary due to their limited memory. Finally, the server sends the screenshot, dictionaries and information of the components (their coordinates, sizes, information about relevant components, link strings, URLs, coordinates of links) to the client. The client displays the Web page and provides each function using the received information.

The server was implemented on a PC with Windows XP, and coded with Visual C# and PHP (Hypertext preprocessor). The client was implemented on a mobile phone (SH902iS [76]) that was equipped with an i-appli developed by NTT Docomo, and coded with Java. Within the 240×320 [pix] screen of the mobile phone, i-appli can only use a 240×240 [pix] area.

Since the progress of mobile phone technologies is remarkably rapid, we expect that it will be possible to implement the whole system (both the server and the client) on a mobile phone in the very near future.

3.3 User Experiment in Real Situations

3.3.1 Experimental Procedure

Setting:

We conducted a user experiment to verify whether users change presentation functions according to their browsing situations, e.g., the change of amount of cognitive resources that are required to interact with the environment. We used 18 subjects (five males and 13 females in their twenties) to browse Web pages for three days. Each subject is asked to use two browsers, OPA Browser and a commercial Web browser for mobile phones to make comparison. The commercial Web browser is a pre-installed Web browser for mobile phones (we refer it as “commercial browser” in the following). The commercial browser adopts a popular presentation style of Web pages among commercial products, which restructures pages so that users can browse them by only vertical scrolling. Since we confirmed that the presentation style of such a commercial browser is better than the conventional style (that presents a Web page as the same way with PCs) in our user experiments in Chapter 2, we decided to use the commercial browser for comparison in this experiment.

The 18 participants were volunteers from our laboratory. Seven participants have used commercial Web browsers for mobile phones several times, and they knew how to use them. The 11 other participants had no experience of using such browsers. Before the experiment started, we explained to the subjects how to use both browsers. Additionally, we gave time subjects to get used to the commercial browser, who have never used commercial Web browsers on mobile phones before.

The participants used an SH902iS phone for browsing over a W-CDMA network. The screen size of the SH902iS is 240×320 [pix], however, these browsers can only allow a 240×240 [pix] area for rendering. The main input control is the direction pad and center action button for selecting, and two soft keys.

Tasks:

We sent each experimental task by an e-mail pointing out which browser to use for the task. We selected 12 goal-oriented tasks (six tasks for each browser) that are to access many different types of Web pages, both textual and graphical, simple and crowded, small-sized and large-sized, and different layouts. We tried to select tasks that would be somehow interesting to participants, thus most pages were popular ones with recent contents. For example, “Please find a link that you get interested in on the BBC Science & Nature” and “Please find a CD you want to listen to on the Sony Music.” In addition, we selected each set of two tasks in 12 tasks at two Web sites that have the same kind of contents and similar structure and page size. We asked subjects to browse them using OPA Browser and the commercial browser in turn to avoid fixing browsers and experimental Web sites for the fairness of the experiment. Furthermore, we requested subjects to browse three Web sites freely using OPA Browser. When subjects finished a browsing task, they sent feedback via e-mail to report their browsing situations, subjective amount of operations, difficulty of the task, and comments.

Logs and Errors:

We also recorded subjects’ operation logs on OPA Browser. These logs contain information on each operation (the selected functions, the keys pressed by the

user, and its time) and the position in a Web page that was displayed on the mobile phone at the rate of every 0.1 secs to examine subjects' orbits of browsing. We could not record logs on the commercial browser because it is not possible to modify commercial products. Additionally, participants used our experimental mobile phones equipped with both browsers in a train, at home, etc. as the same way with their own phones in their real life. Therefore, we could not collect logs even manually. In compensation, we could learn users' impressions using them in real situations. Also, orbits when using the commercial browser become only straight lines because of the restructure of Web pages, thus we might not obtain useful insights.

We examined the subjects' browsing orbits and selected functions. Subjects' logs naturally contain errors (operations that they did not intend to do so) in any systems. However, it is not a trivial task to automatically determine which operation was an error or not. Therefore, we manually discarded operations that obviously looked as subjects missed. In addition, we also discarded logs that subjects unnaturally selected a large number of functions, since we regarded them as subjects just tried various functions, which are not meaningful as logs.

Questionnaire Survey:

Furthermore, to evaluate the overall impression towards OPA Browser, we conducted a questionnaire survey. When finishing the experiment, we asked participants to judge the usability score of each browser and freely write about which browser was comfortable to use and which functions or combinations of functions were useful for browsing. For the usability score, the subjects were asked to respond to each of following questions by scoring from -3 (strongly disagree) to 3 (strongly agree). We designed the questions so that the higher score means more positive impression.

- (i) How few was the amount of operations?
- (ii) How intuitive was the browser for operation?
- (iii) How convenient it was to find information of interest?

- (iv) How easy it was to grasp the current position within a page?
- (v) How would you like to use the browser after this?

3.3.2 Result and Discussion

Relation between Situations and Functions:

In this clause, we discuss the relation between selected functions and subjects' situations based on the collected logs to examine whether participants changed functions according to their browsing situations, e.g., the changes of their cognitive distributions to interact with the browser and environment.

We collected participants' browsing logs in 15 various situations, such as sitting on a chair, walking outside, riding on a train, and eating meals. The situation that occurred most often was sitting on a chair, for which we collected 82 logs. In 53 out of 82 browsing events, subjects first selected scaled-down view (function 1) or magnifying glass (function 6), then read components by manual scrolling. They usually moved among components using jumping to the previous/next component (functions 3, 4). This is because subjects could concentrate on their browsing, e.g., they could fetch much cognitive resources for browsing, namely, they primarily grasped the structure of the page and marked the component seeming to contain the information of interest using the scaled-down view or magnifying glass, then read the component in detail by manual scrolling.

The situation that occurred next often was walking outside, for which we collected 10 browsing logs. We confirmed that subjects seemed to select auto-scrolling (function 9) more frequently than other situations, which 5 out of 10 logs substantiated. This means that subjects could not fully concentrate on browsing while walking, thus, they moved to the component by scrolling per screen size function (function 2) or jumping to the previous/next component (functions 3, 4) then tried to find the information of interest by checking the automatically scrolled screen (by function 9) at intervals. We also observed such browsing styles in situations that participants browsed pages in parallel with other activities, such as eating meals and brushing teeth, in which they had to release their cognitive resources to handle other activities.

Furthermore, we confirmed based on 34 logs that subjects tended to select jumping to an image (function 5) and jumping to a relevant component (function 8) functions to browse information for fun in addition to information of interest when they could concentrate on browsing, such as while sitting on a chair or a seat on a train. Thus, in such situations, it is effective to provide functions to entertain users as well as to decrease operations. While in situations in which subjects could not concentrate on browsing, they were inclined to select fewer functions, i.e., they wanted to use more cognitive resources to interact with the environment.

As mentioned above, users can change functions according to their situations. In addition, we confirmed that users' preferences of functions also affect which function they will select. For example, some subjects selected the scaled-down view to grasp the structure of the page; another selected the magnifying glass even in the same situations; some subjects used the jumping to an image function to view images in the page; and other subjects found the locations of images with the scaled-down view, then moved there and viewed the images. In addition, four of 18 subjects preferred auto-scrolling to manual scrolling. Here, the subjects' browsing logs showed that subjects used auto-scrolling for a different purpose. When subjects concentrated on browsing, they used it to decrease the number of scroll operations while focusing on the screen. However, when subjects could not concentrate on browsing, they did not focus on the screen, e.g., they checked the automatically scrolled screen at intervals and found the information of interest.

We confirmed that users select functions adapted to their situations, i.e., their available cognitive resources that can be used for browsing, although they have a preference of functions. The logs show that 14 out of 18 participants changed functions according to their situations. The three subjects out of the remaining four browsed Web pages in only one situation, thus we could not observe their change of selection, and one subject said that the scaled-down view and zooming function were enough for him to browse pages in any situation. Moreover, 11 subjects used some functions in combination. For example, they used the magnifying glass and jumping to the previous/next component, and scaled-down view and jumping to an image functions together. One subject said that using a few functions in combination enhanced the usability.

Table 3.4: Average scores (standard deviations) aggregated all subjects

Question	OPA Browser	Commercial browser
(i) How few was the amount of operations?	1.9 (1.7)	-0.5 (1.7)
(ii) How intuitive was the browser for operation?	0.1 (1.3)	1.7 (1.3)
(iii) How convenient it was to find information of interest?	2.2 (0.7)	-1.2 (1.4)
(iv) How easy it was to grasp the current position within a page?	2.4 (1.2)	-2.3 (0.7)
(v) How would you like to use the browser after this?	2.3 (0.8)	-0.8 (1.6)

Hence, we conclude users of OPA Browser can browse Web pages comfortably by selecting functions and using them in combination adapted to their situations.

Ratings on Questionnaire Survey:

Table 3.4 shows the average scores and their standard deviations of the questionnaire survey, which indicates that participants felt OPA Browser clearly required fewer operations and made it easier to find the information of interest.

Regarding the amount of operations on the commercial browser, subjects pointed out that it was stressful to focus on the screen until the information of interest appeared, and to go back to that information after scrolling to the bottom of the page. Additionally, some subjects said that they could not re-find the information of interest on their way from the bottom. Other subjects said that they felt uneasy because they could not estimate how long the page would continue. On the other hand, although OPA Browser requires the operation of selecting a function, it could decrease subjects' burden by enabling them to select presentation functions adapted to their situations.

The most important usability factor in using large Web pages on small screens is how to locate the information of interest. The results for convenience of finding the information of interest on the pages show that subjects could find it efficiently using functions that matched with their available cognitive resources in OPA Browser. While using the commercial browser, subjects could find information of interest easily only when it was located near the top of the page.

On the other hand, the commercial browser obtained a higher score for operability (question (ii), asking “How intuitive was the browser for operation?”) than OPA Browser because it requires only vertical scrolling to browse pages, while OPA Browser requires users to learn the functions. However, seven subjects said that once they memorized the functions of OPA Browser, they preferred it to the commercial browser. We think that the questionnaire result was affected by initial impressions, thus, subjects gave the commercial browser a higher score. One subject said that she could browse Web pages with fun owing to the variety of functions on OPA Browser.

From the result of the free writing about which browser was comfortable to use, all 18 subjects chose OPA Browser. This is because the commercial browser provides only a single function, which is not sufficient to browse Web pages using mobile phones in dynamic situations. However, five subjects commented that the commercial browser was more comfortable on textual pages than OPA Browser. We think that it is better to restructure textual content so that users can read it with only vertical scrolling as the Minimap [73] does.

Subjective Amount vs. Actual Number of Operations:

After executing each task, participants gave their feedback via e-mail. Figure 3.7 shows the ratio of participants’ subjective amount of operations. They felt that operations were “much” or “many” in 50% of the commercial browser browsing, while in OPA Browser browsing, none of them felt “much” and only 21% responded “many.”

Regarding the commercial browser browsing, subjects had to focus on the screen all the time because they could not grasp the approximate location of the information of interest due to the restructure of the page. Therefore, subjects felt easy to find the information of interest only when it was located near the top

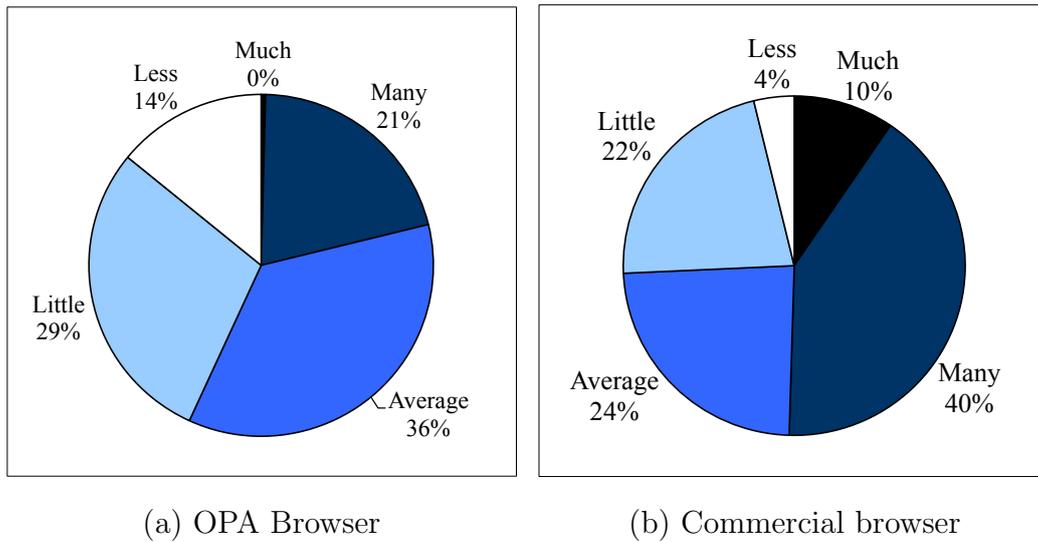


Figure 3.7: Participants' subjective amount of operations

of the page. When the information of interest was located at the bottom of the page, they felt great stress focusing on the screen for a long time.

Figure 3.8 shows the relations between participants' subjective amount and actual number of operations in OPA Browser. In this figure, we categorized each task result according to the participant's subjective amount of operations ("Many," "Average," "Little" and "Less"), and then, sorted the results in each category in ascending order of the actual number of operations. Thus, in this graph, the horizontal axis indicates task results classified by the categories and the vertical axis indicates the actual number of operations for the corresponding task result. From Figure 3.8, it can be seen that actual numbers of operations vary even if subjects felt them as the same. By analyzing subjects' orbits of browsing, we confirmed that they felt the amount of operations as few when they could find the information of interest soon, even if the actual number of operations was large. One subject said that scrolling by continuously pressing a key to find the information of interest was stressful.

This result shows that we cannot judge the effectiveness of browsers only by the number of operations because users' subjective judgment does not always correspond to the actual amount of operations. Thus, we should judge it from both subjective and objective sides; that is, from the actual amount of effort (the number of operations, time needed to complete tasks, time used to operate

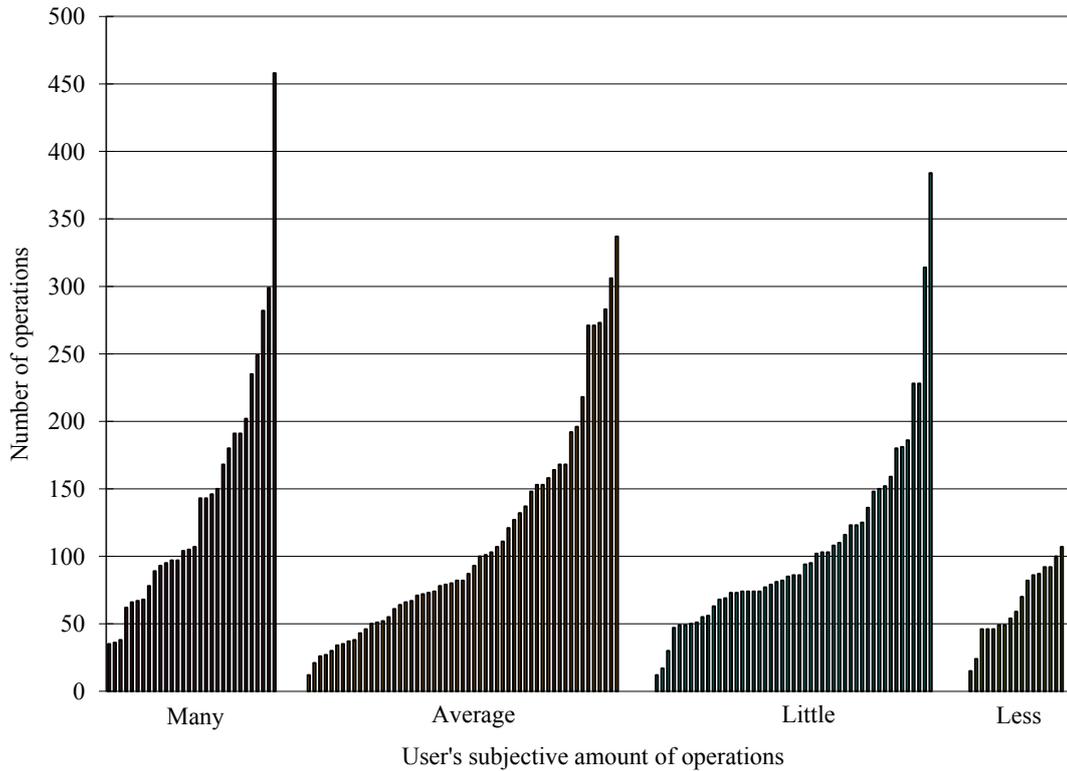


Figure 3.8: Comparison between participants' subjective amount and actual number of operations; participants' actual number of operations varied even though they felt the operations were the same amount.

keys, etc.) and users' subjective satisfaction level by questionnaire survey and/or interview.

3.4 Summary

Since mobile phone users browse Web pages in various situations, comfortable browsing styles for these situations are different from each other, according to the available cognitive resources for browsing Web pages. In this chapter, we proposed and implemented OPA Browser, in which keys of the telephone keypad of a mobile phone have different functions for presenting Web pages. We conducted a user experiment in which we asked participants to use our OPA Browser for three days as the same way with their own mobile phones in their daily life. The results showed that OPA Browser enables users to change the presentation

styles by pressing a single key and adapts to their various situations. Furthermore, we confirmed that we cannot judge the effectiveness of browsers only by the number of operations because users' subjective impression does not always correspond to the actual amount of operations. Thus, we should judge it from both subjective and objective sides; that is, from the actual amount of effort and users' subjective satisfaction level by questionnaire survey and/or interview.

Users' preferences of functions were revealed by the experiment. Thus, it is clear that appropriate arrangements between functions and keys are different for each user. Though we have fixed the arrangement of functions in the current OPA Browser, we think that it is effective to enhance the usability of OPA Browser by enabling users to customize the assignment of functions on telephone keypad.

In OPA Browser, we allocated each of nine presentation functions on telephone keypad of a mobile phone. In the experiment, we observed that the participants got used to the allocation of functions soon through the practice that we asked prior to starting the experiment, and could use the functions smoothly. However, this might be because the participants were all young and familiar with mobile phone operations. It can be expected that older users have difficulty to remember the functions and their allocation. This problem, how many functions can be accepted by users, is beneficial on designing not only Web browsers for mobile phones but general applications with other purposes. We would like to conduct a large scale experiment to tackle with this problem.

Chapter 4

Annotation Presentation and Auto-scrolling for Web Page Overview on Mobile Web to Find Contents Out

4.1 Background

As we described in previous chapters, we have proposed Web browsing techniques for mobile phones that adapt to characteristics of Web contents and users' browsing situations in order to provide them comfortable Web browsing experience. In the user experiments using these Web browsing techniques, we confirmed that overview presentation is an essential function for Web browsing using mobile phones. Since mobile phones have small screens, users can only view small portion of a Web page at a time, and thus they cannot grasp the entire page structure. This feature causes users to get lost in the page, as they scroll around and get confused where they are viewing in the entire page. Overview presentation is the direct solution to this problem.

In these techniques proposed in Chapters 2 and 3, we have used a naive method for presenting the overview of a Web page, as to show the scaled-down page that fits a mobile phone's screen [33]. This method is effective to some

extent, in which users can view the overall structure of the page at a glance. However, since it scales down the page as it fits the mobile phone's screen, it is too small to show details of each content. As we reported in the previous user experiments, some participants commented that they could not see the details of the contents in Web pages, which disturbed them to look for the information they needed.

Therefore, in this chapter, we aim to find more effective way to show the overview of a Web page. The main obstacles on Web browsing using mobile phones are the limited screen sizes and input capabilities, and thus we need to solve them on presenting overview as well. To compensate for the limited screen sizes, we propose to show annotations on the overview, which indicate what kinds of information are in the focused contents. Since the amount of text included in each content is small, it is impossible to extract reasonable annotations only using the text inside. Therefore, we collect linked pages and use the TF/IDF scheme to detect characteristic terms of each content, and present them as annotations. As for the limited input capabilities, the auto-scrolling method [56] can greatly reduce the number of operations by automatically scrolling the mobile phone's screen on Web pages. Auto-scrolling also has an effect to show the overview of the page, since it makes paths that scrolls the entire page through and users can passively view the page structure.

We conduct a user experiment with 20 participants (17 males and three females from 20 to 30 years old) to compare the following overview presentation methods, including the conventional method that shows the scaled-down page and is widely used on previous studies, to see the characteristics and merits/demerits of each overview method.

1. Scaled-down view: the traditional overview presentation method that scales down a Web page so that it fits a mobile phone's screen.
2. Auto-scrolling: automatically scrolls the entire page and users passively view the entire page structure.
3. Annotation: extracts characteristic terms of each content and show them as annotations so that users can see what kinds of information are around.

4. Combination: combines above three methods, e.g., users can switch to the scaled-down view and auto-scrolling, on which annotations are presented.

The experimental results show that annotations are informative for users as an overview to grasp which direction to scroll to find the information of interests. Although auto-scrolling is effective to reduce the number of operations, it turned out to be insufficient by itself, since users have to follow the moving screen and read each text inside a content carefully. The questionnaire survey shows that auto-scrolling can be used to compensate annotations, e.g., users can choose auto-scrolling when they get lost in the page and feel tired on manual scrolling.

The remainder of this chapter is organized as follows: Sections 4.2 and 4.3 describe the design and implementation of the conventional and the proposed overview presentation methods. In Section 4.4, we describe experiments conducted to verify the merits and demerits of each overview presentation method. Finally, we give our conclusion of this chapter in Section 4.5.

4.2 Overview Presentation Methods

In this section, we describe conventional and the proposed overview methods that we compare in the experiments to verify their merits/demerits for users.

4.2.1 Scaled-down View

This method is the most naive approach to show the entire structure of a Web page at a glance, and adopted by many conventional Web browsers for mobile phones [21, 22, 33, 83]. As the scaled-down view, we use the same design with the conventional methods as described in the Section 3.2.3. We show the scaled-down Web page so that its width fits with that of the mobile phone's screen. Additionally, we also provide the functionality that zooms in a component. Specifically, when a user presses a soft-key, the focused component is scaled up until its width (or height if the height is smaller than the width) fits with that of the mobile phone's screen (see Figure 3.3). The user can switch off these zooming and scaled-down view presentation by pressing the other soft key and view the page at the original scale.

As we described in Chapter 3, the magnifying glass also has a face as an overview. However, we saw that the majority of experimental participants preferred the scaled-down view than the magnifying glass function and the basic feature of presenting a scaled-down page is common between them, we decided to use the scaled-down view.

4.2.2 Auto-scrolling

The auto-scrolling method we described in Chapters 2 and 3 is designed for scrolling a single component. Here, we need to scroll the entire page so that users can passively grasp the page structure, e.g., only by viewing the automatically scrolled screen. Therefore, we adopted the auto-scrolling method proposed by Maekawa et al. [56]. The basic scrolling speed and the scroll path inside the component is determined by the same way with the previous chapters based on the amount of text and images, and the shape of the component. Additionally, this method also automatically scrolls between components. The scroll path between components i and j is set as to connect the end point of the scroll path of component i and the starting point of the scroll path of component j (see Figure 4.1, where dotted rectangles show component boundaries, and arrows of dotted line represent auto-scrolling paths of inner components and those of solid line represent paths of inter-components). The scroll speed between components is empirically fixed to 400 [pix/sec] [56]. The order of components visited by auto-scrolling is determined based on the order of appearances of their HTML elements in the whole HTML source.

To add flexibility on auto-scrolling, we provide the following functions:

- key 1** Start auto-scrolling: starts auto-scrolling from the beginning
- key 2** Go back to the previous component: goes back to the previous component and starts auto-scrolling from the head of the previous component
- key 3** Go to the next component: goes to the next component and starts auto-scrolling from the head of the next component
- key 4** Resume auto-scrolling: restarts auto-scrolling from the current point

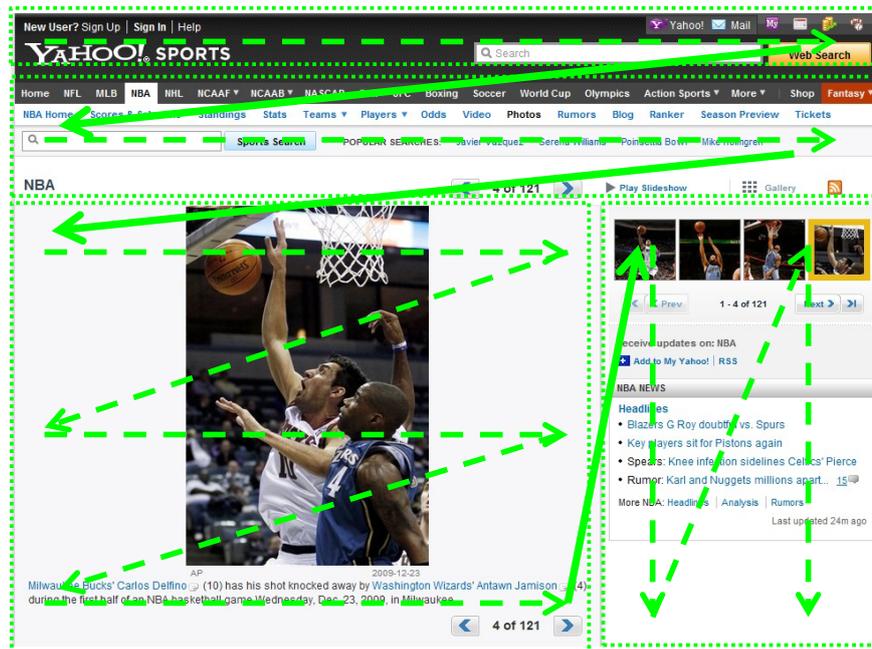


Figure 4.1: Auto-scrolling paths on entire page; dotted rectangles show components, and arrows of dotted line represent auto-scrolling paths of inner components and those of solid line represent paths of inter-components

The allocation of these functions on the telephone keypad is determined according to the functions' priorities, however, it is our future work to investigate comfortable design policies of allocation of functions. In addition to the above functions, users can also use the fast-forward and rewind functions while auto-scrolling. When a user finds the information of interest while auto-scrolling, he/she can stop auto-scrolling by pressing the direction pad, and browse it by manual scrolling in detail. He/she can resume auto-scrolling from the current point by pressing the corresponding key (key 4) of telephone keypad.

4.2.3 Annotation

Since the previous user experiments showed that the scaled-down image of a Web page is often too small to show details of the contents, we propose a method to show an overview without scaling down the page. For this aim, we show characteristic terms of each component as annotations to give hints to users so



Figure 4.2: Annotation method

that they can infer which direction to scroll to find out the information of interest. While a user manually scrolls the page and his pointer enters a component, its characteristic terms are presented to show what kinds of information are in the component as Figure 4.2 shows, where terms that are presented on the rectangle are annotations.

Since a component usually does not contain enough text to extract annotations using conventional text processing approaches, such as TF/IDF and Lexical Compounds [7] that assume the presence of enough large amount of documents, it is difficult to automatically generate annotations only using the text inside the component.

As related work, there are some prior studies that aim to summarize a Web page. They take into account characteristics of Web pages such as link structures and layouts. The method proposed by Shen et al. [77] extracts the main topic of a Web page by page-layout analysis, and then uses the sentences within the main topic as the summary of the page. InCommonSense system [4] looks for pages that have a link pointing to the target page of summarization. Then the system extracts sentences around the link to the target page, because these are likely descriptions of the page. However, these approaches cannot be used in

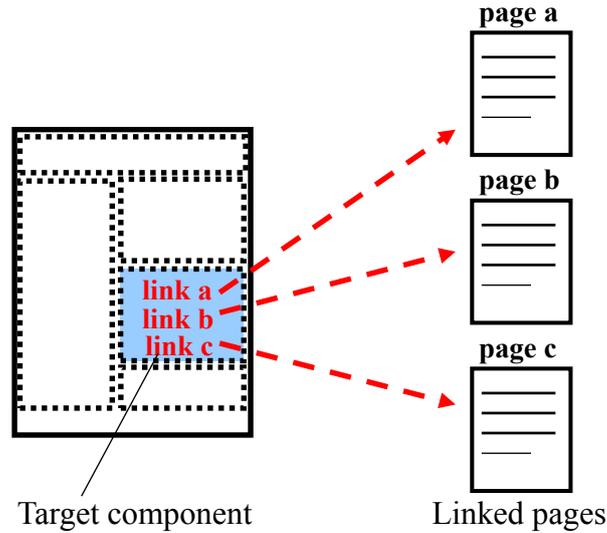


Figure 4.3: Linked pages from a component

our case, e.g., extracting annotations for each component, because a component contains even fewer texts than the entire page. Additionally, the approach in [4] that uses links to the target page is not applicable because links are usually directed to pages but not to components. Buyukkokten et al. [29,30,31] proposed five methods for summarizing parts of Web pages, where the core algorithm is to compute a word’s importance using the approximate dictionary obtained by analyzing word frequencies over 20 million Web pages. Although this approach might work well for ours as well, it would be quite costly to keep the approximate dictionary latest.

Based on above discussion, we decided to collect linked pages from a component to gain more source to analyze. Web page authors add links to refer pages containing related information, and thus we can expect to collect related texts with the one in the target component. Specifically, we collect all linked pages of the same domain with the page as Figure 4.3 shows (to reduce noise, since Web pages of outside domains are often advertisements) and perform a morphological analysis and use the TF/IDF method to extract characteristic terms from the texts as follows. The TF/IDF method is one of the most popular methods to extract characteristic terms from a group of documents with very simple computation, and thus we adopt it to our problem.

First, we extract the text from the target component ($text_{component}$) and all texts from the linked pages ($text_{link}$). Next, we perform a morphological analysis on both texts to pick up only nouns because verbs and adjectives are not suitable for annotations. To treat a compounder, we combine consecutive nouns as a term. Additionally, we also combine prefixes with the following nouns and suffixes with the precedent nouns. Then we remove all stop words from the resulted terms. After that, we compute the weight of each term t based on the TF/IDF method as Equation (4.1) shows. Finally, we select the top 3 terms as annotations of the component.

$$\begin{aligned}
 w(t) &= TF(d, t) \cdot IDF(t). & (4.1) \\
 TF(d, t) &= \frac{\text{number of occurrence of the term } t}{\text{number of terms in } d}. \\
 d &: text_{component} \cup text_{link} \\
 IDF(t) &= \text{Log}\left(\frac{N}{df(t)}\right). \\
 df(t) &= \text{number of texts containing the term } t. \\
 N &: |text_{component} \cup text_{link}| \text{ (each text extracted from linked pages} \\
 &\quad \text{and component is counted as one)}
 \end{aligned}$$

4.2.4 Combination

In this method, we combine the above scaled-down view, auto-scrolling, and annotation methods, i.e., provide auto-scrolling with annotation and scaled-down view with annotation methods. We modified each method to utilize mutual advantages as following paragraphs show. We allocated keys 1 and 2 of telephone keypad for auto-scrolling with annotation, and key 3 for scaled-down with annotation methods:

key 1 Start auto-scrolling: starts auto-scrolling from the beginning

key 2 Resume auto-scrolling: resumes auto-scrolling from the point that the user stopped the auto-scrolling

key 3 Scaled-down view: shows the scaled-down view with annotations



(a) Annotation presented at an intersection (b) Auto-scrolling (c) Annotation presented while auto-scrolling

Figure 4.4: Auto-scrolling with annotation (auto-scrolling is proceeding in order of (a), (b), then (c))

To switch auto-scrolling with annotation and scaled-down view with annotation, users can press key 0 to end the current method, then press the corresponding key to start the other method.

Auto-scrolling with Annotation

This method shows annotations that indicate what kinds of information are contained in neighboring components while auto-scrolling. The scroll path used for auto-scrolling in Section 4.2.2 goes through inside a component, and thus if we adopt this auto-scrolling path, we can only show the annotations of the currently scrolling component. Therefore, in this method, the scrolling path is set to thread its way through components, e.g., the path is set along the border of components to show annotations of neighboring components. Here we show two kinds of annotations: one shows the characteristic terms in the component to show what kinds of information are contained there (as described in Section 4.2.3) and another shows categories of information in the component as its abstract, such as “sports” and “politics” (as described later).

When a user presses key 1, a Web page is automatically scrolled along with paths on the page (see Figure 4.4, where scroll paths are represented by the thick gray lines). While auto-scrolling, annotations that show what information

a neighboring component contains is displayed (see Figure 4.4 (b) and (c)). When the screen reaches an intersection on the current scroll path, auto-scrolling stops and waits for the user's selection of which direction (path) he/she wants to go. Additionally, at an intersection, annotations with a white arrow is displayed to show what kinds of contents are ahead (see Figure 4.4 (a)). The user can choose one direction using his/her direction pad considering the information provided by the annotations. Then, auto-scrolling is resumed. If the user finds a content of interest while auto-scrolling, he/she can stop auto-scrolling by pressing any keys, and can read the component in detail by manual scrolling. The user presses key 2 when he/she wants to go back to the scrolling, and can restart auto-scrolling along the path that the user stopped auto-scrolling before.

Scroll Path After component extraction, we determine scroll paths so that they thread their ways through components, to show annotations of neighboring components. The scroll paths are set by the following steps:

step 1 Sets paths along with the four sides of each component (For HEADER and FOOTER components, paths along with only bottom and top sides are respectively set since these components are head and tail of the page, as the left image of Figure 4.5 shows.)

step 2 Joins paths lining vertically and horizontally (see the middle image of Figure 4.5)

step 3 Merges neighboring paths

step 4 Joins vertical and horizontal paths in order not to leave isolated paths (see the right image of Figure 4.5)

Annotation of Category In this method, we show categories of information contained in components ahead at an intersection of scroll paths as an abstracts of their information. Many components have headers that indicate categories of information inside the components as Figure 4.6 shows, which are usually tagged by specific HTML tags. We checked 50 Web sites of various kinds (news, company, and online shopping sites), and confirmed that most components in

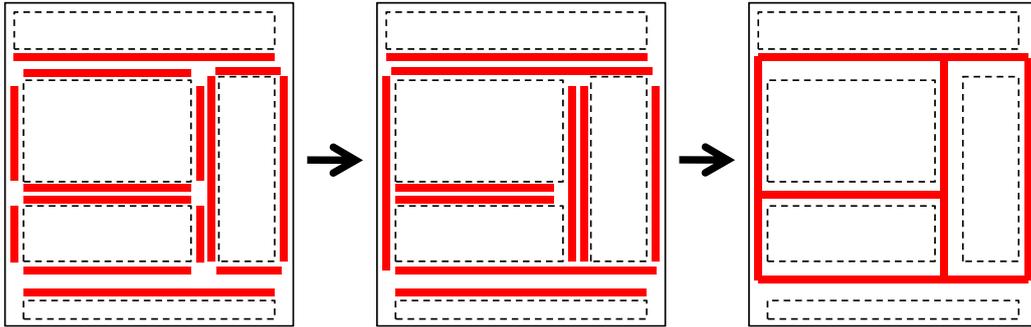


Figure 4.5: Scroll path setting for the combination method (dotted rectangles represent components and thick solid lines represent paths)



Figure 4.6: Category headers in components

each page have titles emphatically written using particular HTML tags, such as `<H1>` and ``.

Therefore, we extract sentences and words that are emphasized by particular tags, i.e., from `<H1>` to `<H6>`, ``, ``, ``, `<I>`, `<U>`, `<SMALL>`, `<BIG>`, and `` tags, and then, perform a morphological analysis to extract nouns as categories. As same with annotation extraction in Section 4.2.3, we merge consecutive nouns, prefixes, and suffixes as a compounder. If a component contains multiple emphasized terms, we give higher priorities to terms enclosed by tags listed in higher positions in the above tag list, which emphasize words more strongly than latter ones. If the component does not have terms tagged by emphasizing HTML tags, we show its characteristic terms extracted by the method described in Section 4.2.3 instead as annotations.



(a) Scaled-down page

(b) Annotation on scaled-down page

Figure 4.7: Scaled-down view with annotation

Scaled-down View with Annotation

This method shows annotations on a scaled-down page. When a user presses key 3, the scaled-down page that fits the mobile phone's screen is presented (see Figure 4.7 (a)), on which annotations of the component specified by the user's pointer is presented (see Figure 4.7(b), where terms presented in the white rectangle are annotations). Here we show the categories of information contained in the component to show the abstract of the component as annotations. The user presses key 0 to switch off the method.

4.3 Implementation of Proposed Methods

We implemented each overview presentation method as a server/client architecture. When the client sends a request for a Web page with the information of the size of the screen, the server receives the corresponding page from a Web server and generates a screenshot of the page. Next, the server extracts components considering the size of the client's screen. Then the server extracts and adds meta-data required by each overview method and return the screenshot and

the meta-data to the client. For the overview methods providing annotations (the annotation and combination methods), the server fetches Web pages that are linked from each component to extract annotations, and looks for HTML tags used for emphasizing terms to extract categories of each component. For the overview methods providing auto-scrolling (the auto-scrolling and combination methods), the server calculates the amount of information of components and determines scroll paths and speeds. The client presents the screenshot and presents the corresponding overview using the received meta-data.

The server was implemented on a PC with Windows XP, and coded with Visual C# and PHP (Hypertext preprocessor). The client was implemented on a mobile phone (SH902iS [76]) that was equipped with i-appli developed by NTT Docomo, and coded with Java. Within the 240×320 [pix] screen of the mobile phone, i-appli can only use a 240×240 [pix] area. For the later user experiment with Japanese participants, we implemented the system to handle Japanese pages. Therefore, though examples of annotations here are in English, the real ones are in Japanese. We used MeCab [60] for Japanese morphological analysis.

4.4 Experiments

First we conducted an experiment to evaluate how accurate the extracted annotations and categories are. Then, we conduct a user experiment to find out the effective styles of overview presentation on small screens and see merits/demerits of each method.

4.4.1 Accuracy of Annotation and Category Extraction

We selected 30 Web sites of various kinds (news, company, and online shopping sites; each of them is a popular site such as Amazon.com) and tested how accurate the extracted annotations and categories were. Since it is difficult to think all possible annotations, we chose to judge whether the extracted annotations and categories were correct or not. Therefore, we define the accuracy as the ratio of the annotations judged as accurate among all extracted annotations. Here, one of our research team members judged the extracted annotations.

As a result, our method achieved 82% accuracy for extracting characteristic terms from components, while achieved 83% accuracy for extracting categories. As for the annotations of characteristic terms extracted using the link structure of a component, we think that this result is acceptable considering the limitation of available information to extract annotations. Most error terms came from advertisements included in the linked pages. To exclude these noisy terms, we need to analyze texts first considering page structures and link anchors in components.

Although the category extraction method is very simple, it also achieved high accuracy for extracting components' categories. However, because of its simplicity, our method cannot detect categories that are not tagged by emphasizing HTML tags. To further improve the accuracy, we can apply conventional techniques of detailed HTML source analysis, such as HTML tag pattern recognition. Such an extension is open to our future work.

4.4.2 User Experiment

Experimental Procedure

We conducted a user experiment to find out appropriate methods to show the overview of a Web page, which effectively help users to find out the information of their interest, and merits and demerits of each method. We collected 20 participants, who are in their twenties and consist of 17 males and three females.

The aim of presenting the overview of a Web page is to show an entire structure of the page and help users look for information of their interest. Therefore, we evaluate how users can effectively find out a content they need. In the experiment, we asked them to look for and click a link text in a Web page that we specified as an experimental task, such as looking for a link text of "Subprime lending" in the world news page of Yahoo! News. As experimental tasks, we set eight tasks in total and asked participants to perform two tasks for each overview method. To keep fairness of the experiment, the order of trying overview methods and allocation of tasks are randomized. We explained how to use each overview method and asked participants to practice to use it until they got used to its operation. The overall experimental time ranged from about 20 to 40 minutes per participant.

Table 4.1: Experimental logs

Log entity	Description	Example
Key	The key that the user operated	up
Operation	Record whether the user pressed or released the key	press
Time	The time stamp the user operated the key	1500

As experimental logs, we recorded every key operation and its time, as Table 4.1 shows. As evaluation metrics, we compare the number of operations, the time to achieve a task, and the operation time that is the total time a user operated keys as Equation (4.2) shows.

$$\begin{aligned}
 \text{Operation time} &= \sum_{i=1}^N (\text{time}_{\text{key } i \text{ released}} - \text{time}_{\text{key } i \text{ pressed}}). \quad (4.2) \\
 N &: \text{Number of operated keys}
 \end{aligned}$$

The number of operations and operation time measure participants' burden, while the task completion time measures the efficiency to find out a content.

We also conducted a questionnaire survey to evaluate participants' subjective impression against the overview methods. We asked participants to score each overview method regarding the following aspects using 5 scale points, where score -2 means strongly disagree while score 2 means strongly agree. We set the questions so that the higher score means more positive impression.

- (i) How helpful was the method to find the link text?
- (ii) How intuitive and easy was the method to use?
- (iii) How would you like to use the overview presentation method after this?

Results

Table 4.2 shows the average number of operations, average time for task completion, and average operation time. We performed the Kruskal Wallis Test [51] with a significance level of 5% to examine whether these values were significantly

Table 4.2: Average number of operations, average time for task completion, and average operation time (the number in parenthesis is its standard deviation)

Overview method	Average number of operation	Average time for task completion [sec]	Average operation time [sec]
Scaled-down view	38.7 (24.3)	43.9 (26.2)	0.44 (0.51)
Auto-scrolling	23.1 (11.9)	54.3 (30.7)	0.23 (0.32)
Annotation	36.1 (29.8)	39.1 (24.9)	0.49 (0.51)
Combination	30.2 (14.8)	58.1 (34.7)	0.23 (0.29)

different among overview methods. Since users' operations tend to vary among individuals, parametric tests like ANOVA are not suitable. Therefore, we decided to use a non-parametric test, e.g., the Kruskal Wallis Test. As a result, p-values of number of operations, time for task completion, and operation time were smaller than 0.05 and we confirmed that the resulted values were significantly different based on overview presentation methods that the participants used.

As we expected, the number of operations was smallest on the Auto-scrolling method, since after starting auto-scrolling, participants basically did not need to operate keys and passively viewed the automatically scrolled screen. However, auto-scrolling needs a certain time to scroll through a Web page, and thus the average task completion time was smaller on the scaled-down view and annotation methods (the average task completion time was smallest on the annotation method). These methods require users to manually scroll the page (in other words, users can control the whole operation), however, if users can infer where their information of interest locates, they can take the shortest path to scroll to there, which resulted in the smallest task completion time on the annotation method. Although the scaled-down view method enables users to zoom in a component, it is still not sufficient to show which component contains the information of interest, which resulted in the larger number of operations and task completion time. The result of the average operation time was also reasonable, which showed

Table 4.3: Characteristics of overview presentation methods

Overview method	Burden of operation	Users' independence	Visibility
Scaled-down view	++	++	-
Auto-scrolling	-	-	+
Annotation	++	++	++
Combination	+	+	++

that users kept pressing the direction pad to scroll on the scaled-down view and annotation methods to reduce the number of operations.

The combination method showed a middle result between two methods; the annotation and auto-scrolling methods. When using the combination method, users can decide which path to scroll considering the presented annotations. Therefore, the average number of operations was between the auto-scrolling method and the scaled-down view/annotation methods, and the average task completion time was the largest since the combination method needed time for both auto-scrolling and users' operations. We summarize the characteristics of the overview methods in Table 4.3.

Table 4.4 shows the average scores and their standard deviations of each overview presentation method regarding questions, where the higher score means the evaluation is more positive. As for question (i), which asked how the overview method was helpful to find a target link text, the annotation and combination methods outperformed others. This result shows that annotations presented by these methods were informative as participants could infer which direction to scroll to find information they needed. The lowest score of the scaled-down view method shows that participants could not decide a component to look for the target link text because its visibility was poor. As for question (ii), which asked how intuitive was the operation of overview method, the annotation method obtained the best score followed by the scaled-down view method. This result corresponds to the number of functions they provide, e.g., users only need to scroll and annotations are presented on the annotation method, and users only zoom in and out on the scaled-down view method. As for question (iii), which asked whether

Table 4.4: Average scores (standard deviations) aggregated all subjects

Question	Scaled-down view	Auto-scrolling	Annotation	Combination
(i) How helpful was the method to find the link text?	-0.1 (1.0)	0.4 (1.3)	0.6 (0.9)	0.6 (1.0)
(ii) How intuitive and easy was the method to use?	1.1 (0.8)	0.6 (0.9)	1.7 (0.7)	0.2 (1.0)
(iii) How would you like to use the overview method after this?	-0.1 (1.3)	0.3 (1.1)	0.8 (1.2)	0.6 (0.9)

participants want to use the overview presentation method in the future, the annotation method obtained the best score followed by the combination method. Again this result shows that the annotations are informative for users. Although the combination method can reduce the number of operations compared with the annotation method, its score was slightly lower than that of annotation method. This is because the combination method needs several operations to switch the auto-scrolling with annotation and scaled-down view with annotation methods. Additionally, it stops auto-scrolling at an intersection of scroll paths to ask users' decision of the next path to go, which seemed to be confusing for participants, since it mixes an automatic operation and users' manual operation. It would be better to separate these two to keep consistency in an interaction, as users do not need to do anything while automatic operations. During the experiment, five participants commented that they got lost in a Web page when using the annotation method because it could only show small portion of the page. Auto-scrolling also has a role to show the entire page structure by automatically scrolling the page along the paths, and thus it would be effective to add the auto-scrolling method in the annotation method so that users can select it adaptively when they get lost or get bored on manual scrolling.

4.5 Summary

In this chapter, we examined the effective way of showing the overview of a Web page on mobile phones. We conducted a user experiment with 20 participants and compared the conventional scaled-down view and auto-scrolling methods with the annotation based method and their combination. Experimental results showed that the visibility of the scaled-down view is poor to look for the information of interest, while annotations are informative and thus users can decide a direction to scroll by referring annotations that show what kinds of information are around the displayed area. We also confirmed that mixing automatic operation and users' manual operation is confusing for users, and thus it is better to separate these two. Considering five participants commented that they got lost when using the annotation method, adding the auto-scrolling method and enabling users to use it adaptively are effective to show the overview of the page on mobile phone screens.

Chapter 5

Conclusion

5.1 Summary of Thesis

In this thesis, we have performed studies under the goal of providing comfortable Web browsing experience using mobile phones. We tackled this challenge from the three aspects: presenting Web contents adapted to their characteristics, providing various presentation functions on telephone keypad of mobile phones so that users can select ones adapted to their browsing situations to preserve enough cognitive resources required there, and designing effective overview presentation of a Web page that help users find out their information of interest.

In Chapter 1, we first described the privilege that mobile Web brings to people's life, and then open problems caused by the limited screen sizes and input capabilities of mobile phones. Then we introduced series of research works that share the similar goal with us, as well as works targeting to assist users on Web browsing, and those targeting to realize a novel browsing situation, as collaboration with other mobile users and available digital devices in the environment. We also described the recent emerging trends of studies and services targeting mobile phones. Next we described the organization of our thesis, followed by the introduction of general structure of Web pages that is the basis of our technique designs.

In Chapter 2, we proposed a Web browsing system that adaptively presents Web contents according to their characteristics. Generally, users in a fixed situation, such as sitting on a chair at their home, browse a Web page using mobile

phones following the two steps: first they look for a content of interest from the entire page and then read the content in detail. Due to the limitation of mobile phones' screen sizes and input capabilities, users have to perform numerous operations on these steps. Therefore, we designed a Web browsing system that shows the scaled-down page that fits the mobile phone's screen to reduce scroll operations to find out a content of interest. Our system then shows the content adapted to its characteristics to reduce operations to read it in detail. The experimental results showed that our system can reduce the number of operations to find contents of interest and futile scrolls to read them. The results also showed that our system enables users to cover the entire page, e.g., reduces the possibility of their missing information on the page.

In Chapter 3, we focused on mobile phone users who are in dynamic situations and proposed a Web browser that provides various presentation functions so that users can select ones adapted to their browsing situations, i.e., users can flexibly distribute their cognitive resources for Web browsing and interacting with the environment. While in Chapter 2, we assumed a fixed Web browsing situation, considering the trend that people always carry their mobile phones around with them, we think that users' browsing situations can dynamically vary. Since it is usually difficult to accurately detect users' situations and the suitable presentation styles on various browsing situations are user dependent, we decided to let users select ones easily, only by pressing a key. We have conducted a user experiment in real situations, as asked participants to use our browser for three days as the same way with their own mobile phones in their daily life. The experimental results showed that participants selected presentation styles according to their browsing situations to adjust their distributions of the cognitive resources on both the Web browser and environment.

In Chapter 4, we focused on presentation of the Web page overview. The user experiments in Chapters 2 and 3 showed that the overview greatly helps users to look for a content of interest from a large Web page. However, several participants commented that a scaled-down page was not sufficient as overview, since they could not see details of the contents. Therefore, we proposed novel ways to show the overview of the page to users and conducted a user experiment to verify effective overview presentation methods that help users find out the information

of interest and their merits/demerits. The experimental results showed that presenting annotations, which indicate what kinds of information are contained in the contents around, is effective and auto-scrolling can reduce the number of operations and help users grasp the page structure when they are lost within the large page. Furthermore, we confirmed that mixing automatic functions and users' manual operations is confusing for users. It is important to separate these two for simplicity of interaction.

In summary, the techniques proposed in this thesis could successfully reduce users' burden on Web browsing using mobile phones, which admit to take mobile Web technology one step further. Furthermore, our user experiments revealed interesting characteristics of users' behaviors on Web browsing and their reactions towards different presentation styles on small screens, which would be a beneficial reference for other researchers working for information presentation on mobile devices.

5.2 Future Work

Through the research works in this thesis, the following topics are open to our future work.

5.2.1 Dynamic Contents

In our current implementations, we cannot present dynamic contents, such as CGI [72], Flash [2], and Silverlight [61] since we show the screenshot of a Web page. Considering that these dynamic contents are getting proactively used for their interactivity and ability on expression of beautiful designs, we should modify our implementation platform to be able to handle them.

5.2.2 Optimization of Auto-scrolling Speed

Although we have conducted an initial experiment to determine the basic speed of auto-scrolling, the experimental subjects were lopsided to youngers. Therefore, we need to conduct larger scale experiments to investigate the optimal auto-scrolling speed, since it would differ among individuals depending on several

factors, including age (or eyesight), familiarity with Web pages, etc. In the experiments in Chapter 2, subjects were males and females ranging from 20 to 30 years old, and thus the age effects on the optimal auto-scrolling speed could not be observed. However, generally speaking, as people get older, their eyesight becomes weaker, thus the optimal auto-scrolling speed is likely to become slower.

5.2.3 Appropriate Presentation on Smart Phones

In this thesis, we focused on mobile phones, however, smart phones also share the common problems caused by the limited screen sizes. Our presentation methods are applicable to smart phones to some extent, however, richer interaction is possible due to their advanced input capabilities, such as a touch screen and gyro sensor. It is interesting direction to consider appropriate presentation methods with rich interaction for smart phones.

5.2.4 Mobile Users' Locations

As we described in Section 1.2.4, more and more mobile phones are getting equipped with a GPS receiver that can tell users' locations. By combining users' locations with other services, we can greatly enrich the services so that they match with users' contexts, such as recommending Web pages that users usually browse when they are at home, and finding out online discount tickets when they are at a shopping mole. Additionally, users' location history is an interesting data source, from which we can extract crowd's knowledge, such as which restaurants and shops are currently popular among local people. We would like to do research for this direction in the future.

Acknowledgment

Upon the completion of this thesis, I would like to borrow this opportunity to express my sincerest gratitude to those who have done their best to offer me assistances to the thesis.

First and foremost, I am deeply indebted to my honors supervisor, Prof. Shojiro Nishio, who gave me insightful guidance to my research work and great encouragement to overcome all the obstacles I encountered. This thesis would never have been completed without his unfailingly wise advice and support.

Meanwhile, it is my pleasure and fortunate that Associate Prof. Takahiro Hara has become my direct advisor, assisting me to overcome the limitation of my knowledge and converting me into a researcher in this exciting research field, as well as providing me valuable advice and guidance throughout my undergraduate, master, and doctoral research work. I would especially appreciate him hereby, for what I have learned from him is far beyond what this thesis shows.

I would also like to acknowledge the committee members of my thesis, Prof. Norihisa Komoda, Associate Prof. Yuichi Itoh, Prof. Toru Fujiwara, and Prof. Fumio Kishino at the Department of Multimedia Engineering of the Graduate School of Information Science and Technology of Osaka University. Their insightful and constructive comments considerably improved the quality of the thesis.

I would like to express my appreciation to Prof. Shinji Shimojo at Osaka University (Cybermedia Center) / Executive Researcher at National Institute for Information and Communications Technology, Dr. Yoshinobu Tonomura, Director of NTT Communication Science Laboratories, and Principal Researcher Mr. Toshiyasu Nakao at NEC Corporation for their valuable advices and discussions as doctoral advisories.

I am very grateful for the help and support from Associate Prof. Yuichi Teranishi, Associate Prof. Tomoki Yoshihisa, and Associate Prof. Kaname Harumoto at Osaka University, Associate Prof. Tsutomu Terada at Kobe University, Lecturer Takefumi Ogawa at the University of Tokyo, and Assistant Prof. Akimitsu Kanzaki at Osaka University, Assistant Prof. Kotaro Nakayama at the University of Tokyo, and Dr. Yoshimasa Ishi at Osaka University.

I sincerely appreciate Dr. Gen Hattori and Dr. Toshiaki Uemukai at KDDI R&D Laboratories Inc. for their continuous support throughout our joint work, Dr. Takuya Maekawa at NTT Communication Science Laboratories for supervising me at the very beginning of my research career, and Dr. Xing Xie at Microsoft Research Asia for his supervision and assistance during and after my internship, which had strong impact on my career design.

I also would like to thank my team members, Masahiro Itoh, Maike Erdmann, Daijiro Komaki, Masumi Shirakawa, Tomoyuki Michishita, Mayu Iwata, Azusa Oku, and Akiyoshi Suzuki, including the graduate student Kenji Ohnishi, with whom I had many discussions concerning the research and made good progress on research work.

With pleasure of studying and working with a number of very talented and warm hearted people at Nishio laboratory, I would like to express thanks to them.

Last but by no means least it gives me immense pleasure to offer my hearty thanks to my family and friends. No words can express my appreciation to their support and love in my life. I would like to specially thank Mamiko Yamagishi, who is my best friend of more than 10 years and has been sharing difficulties and happiness with me, without her support and encouragement, I could not have finished this thesis. I also express my best appreciation for the late Hiroyuki Takemoto, who taught me the preciousness of life. The days with him gave me strength to survive in tough situations. I am proud of his bravery and positive attitude toward living his life from bottom of my heart.

Finally, with all my best wishes to those wonderful people who ever gave me assistances, supports and encouragements throughout my research experience.

Reference

- [1] Adobe Acrobat Reader: <<http://www.adobe.com/products/reader/>>.
- [2] Adobe Flash: <<http://www.adobe.com/products/flash/>>.
- [3] Amershi, S. and Morris, M. R.: CoSearch: A System for Co-located Collaborative Web Search, in *Proceedings of Annual SIGCHI Conference on Human Factors in Computing Systems (CHI 2008)*, pp. 1647–1656 (Apr. 2008).
- [4] Amitay, E. and Paris, C.: Automatically Summarizing Web Sites - Is There a Way around It?, in *Proceedings of ACM Conference on Information and Knowledge Management (CIKM 2000)*, pp. 173–179 (July 2000).
- [5] Anderson, C. R., Domingos, P., and Weld, D. S.: Personalizing Web Sites for Mobile Users, in *Proceedings of International World Wide Web Conference (WWW 2001)*, pp. 565–575 (May 2001).
- [6] Anderson, C. R., Domingos, P., and Weld, D. S.: Adaptive Web Navigation for Wireless Devices, in *Proceedings of International Conference on Artificial Intelligence (IJCAI 2001)*, pp. 704–712 (Aug. 2001).
- [7] Anick, P. G. and Tipirneni, S.: The Paraphrase Search Assistant: Terminological Feedback for Iterative Information Seeking, in *Proceedings of Annual ACM SIGIR Conference (SIGIR 1999)*, pp. 153–159 (Aug. 1999).
- [8] Arase, Y., Maekawa, T., Hara, T., Uemukai, T., and Nishio, S.: Adaptive Presentation of Web Contents for Mobile Web Browsing using Cellular Phones, in *Proceedings of IEICE SIG-WI2*, pp. 47–52 (Mar. 2006) (in Japanese).

- [9] Arase, Y., Maekawa, T., Hara, T., Uemukai, T., and Nishio, S.: A Web Browsing System based on Adaptive Presentation of Web Contents for Cellular Phones, in *Proceedings of International Cross-Disciplinary Workshop on Web Accessibility (W4A 2006)*, pp. 86–89 (May 2006).
- [10] Arase, Y., Hara, T., Uemukai, T., and Nishio, S.: Nine-Button Web Browsing System for Cellular Phone Users, in *Proceedings of Innovations in Information Technology 2006*, CD-ROM (Nov. 2006).
- [11] Arase, Y., Hara, T., Uemukai, T., and Nishio, S.: Nine Button Browsing System: A Web Browsing System for Cellular Phones, in *Proceedings of DBWeb 2006*, pp. 111–118 (Nov./Dec. 2006) (in Japanese).
- [12] Arase, Y., Maekawa, T., Hara, T., Uemukai, T., and Nishio, S.: An Adaptive Content Presentation System for Web Browsing using Cellular Phones, *IPSSJ Journal*, Vol. 47, No. 12, pp. 3149–3164 (Dec. 2006) (in Japanese).
- [13] Arase, Y., Hara, T., Uemukai, T., and Nishio, S.: Nine Button Browsing System: A Web Browsing System for Cellular Phone Users, *DBSJ Letters*, Vol. 5, No. 4, pp. 9–12 (Mar. 2007) (in Japanese).
- [14] Arase, Y., Hara, T., Uemukai, T., and Nishio, S.: OPA Browser: A Web Browser for Cellular Phone Users, in *Proceedings of ACM Symposium on User Interface Software and Technology (UIST 2007)*, pp. 71–80 (Oct. 2007).
- [15] Arase, Y., Maekawa, T., Hara, T., Uemukai, T., and Nishio, S.: A Web Browsing System for Cellular Phone Users based on Adaptive Presentation, *Universal Access in the Information Society (UAIS)*, Vol. 6, No. 3, pp. 259–271 (Nov. 2007).
- [16] Arase, Y., Hara, T., Uemukai, T., and Nishio, S.: Web Browsing Assistance for Cellular Phones by Motoring Metaphor, in *Proceedings of DEWS 2008* (Mar. 2008) (in Japanese).
- [17] Arase, Y., Hara, T., Uemukai, T., and Nishio, S.: Web Page Presentation

- on Cellular Phone using Drive Metaphor, *DBSJ Journal*, Vol. 7, No. 1, pp. 103–108 (Aug. 2008) (in Japanese).
- [18] Arase, Y., Xie, X., Duan, M., Hara, T., and Nishio, S.: A Game Based Approach to Assign Geographical Relevance to Web Images, in *Proceedings of International World Wide Web Conference (WWW 2009)*, pp. 811–820 (Apr. 2009).
- [19] Arase, Y., Hara, T., Uemukai, T., and Nishio, S.: MotoBrowser: Enjoyable Browsing using Cellular Phones by “Motoring” Web Pages, in *Proceedings of International Conference on Mobile Data Management (MDM 2009)*, pp. 373–374 (May 2009).
- [20] Arase, Y., Hara, T., and Nishio, S.: Web Page Adaptation and Presentation for Mobile Phones, *Hu, W.-C. and Zuo, Y. Eds., Handheld Computing for Mobile Commerce: Applications, Concepts and Technologies*, IGI Global (2010, to appear).
- [21] Baluja, S.: Browsing on Small Screens: Recasting Web-page Segmentation into an Efficient Machine Learning Framework, in *Proceedings of International World Wide Web Conference (WWW 2006)*, pp. 33–42 (May 2006).
- [22] Baudisch, P., Xie, X., Wang, C., and Ma, W.-Y.: Collapse-to-Zoom: Viewing Web Pages on Small Screen Devices by Interactively Removing Irrelevant Content, in *Proceedings of ACM Symposium on User Interface Software and Technology (UIST 2004)*, pp. 91–94 (Oct. 2004).
- [23] Billsus, D., Brunk, C. A., Evans, C., Gladish, B., and Pazzani, M.: Adaptive Interfaces for Ubiquitous Web Access, *Communications of the ACM*, Vol. 45, Issue 5, pp. 34–38 (May 2002).
- [24] BlackBerry: <<http://na.blackberry.com/eng/>>.
- [25] Blenkhorn, P., Evans, D. G., King, A., Kurniawan, S. H., and Sutcliffe, A.: Screen Magnifiers: Architecture, Evolution and Evaluation, *IEEE Computer Graphics and Applications*, Vol. 25, pp. 54–61 (Sept./Oct. 2003).
- [26] Brightkite: <<http://brightkite.com/>>.

- [27] Bruijn, O. and Spence, R.: Rapid Serial Visual Presentation: A Space-time Trade-off in Information Presentation, in *Proceedings of Working Conference on Advanced Visual Interfaces (AVI 2000)*, pp. 189–192 (May 2000).
- [28] Bruijn, O., Spence, R., and Chong, M. Y.: RSVP Browser: Web Browsing on Small Screen Devices, *Personal and Ubiquitous Computing*, Vol. 6, Issue 4, pp. 245–252 (Sept. 2002).
- [29] Buyukkokten, O., Garcia-Molina, H., and Paepcke, A.: Power Browser: Efficient Web Browsing for PDAs, in *Proceedings of Annual SIGCHI Conference on Human Factors in Computing Systems (CHI 2000)*, pp. 430–437 (Apr. 2000).
- [30] Buyukkokten, O., Garcia-Molina, H., and Paepcke, A.: Accordion Summarization for End-game Browsing on PDAs and Cellular Phones, in *Proceedings of Annual SIGCHI Conference on Human Factors in Computing Systems (CHI 2001)*, pp. 213–220 (Mar. 2001).
- [31] Buyukkokten, O., Garcia-Molina, H., and Paepcke, A.: Seeing the Whole in Parts: Text Summarization for Web Browsing on Handheld Devices, in *Proceedings of International World Wide Web Conference (WWW 2001)*, pp. 652–662 (May 2001).
- [32] Cai, D., Yu, S., Wen, J.-R., and Ma, W.-Y.: Extracting Content Structure for Web Pages based on Visual Representation, in *Proceedings of Asia Pacific Web Conference (APWeb 2003)*, pp. 406–417 (Apr. 2003).
- [33] Chen, Y., Ma, W., and Zhang, H.: Detecting Web Page Structure for Adaptive Viewing on Small Form Factor Devices, in *Proceedings of International World Wide Web Conference (WWW 2003)*, pp. 225–233 (May 2003).
- [34] Crandall, D., Backstrom, L., Huttenlocher, D., and Kleinberg, J.: Mapping the World’s Photos, in *Proceedings of International World Wide Web Conference (WWW 2009)*, pp. 761–770 (May 2009).
- [35] Embley, D. W., Jiang, Y., and Ng, Y. K.: Record-boundary Discovery in

- Web Documents, in *Proceedings of ACM SIGMOD Conference (SIGMOD 1999)*, pp. 467–478 (June 1999).
- [36] Flickr: <<http://www.flickr.com/>>.
- [37] Google Mobile: <<http://www.google.co.jp/mobile/>>.
- [38] GypSii: <<http://www.gypsii.com/>>.
- [39] Han, R., Perrett, V., and Naghshineh, M.: WebSplitter: A Unified XML Framework for Multi-Device Collaborative Web Browsing, in *Proceedings of ACM Conference on Computer Supported Cooperative Work (CSCW 2000)*, pp. 221–230 (Dec. 2000).
- [40] Healey, C. G., Booth, K. S., and Enns, J. T.: Highspeed Visual Estimation using Preattentive Processing, *ACM Transactions on Computer-Human Interaction (TOCHI)*, Vol. 3, Issue 2, pp. 107–135 (June 1996).
- [41] Information on subscribers of telephone, Pager and PHS (Personal Handy-Phone System) in Japan:
<<http://www.soumu.go.jp/johotsusintokei/field/data/gt01020101.xls>> (in Japanese).
- [42] Information & Communications Statistics Database:
<http://www.soumu.go.jp/johotsusintokei/tsusin_riyou/data/eng_tsusin_riyou02_2005.pdf>.
- [43] In-Stat Press Releases:
<<http://www.instat.com/press.asp?ID=2321&sku=IN0804034MCM>>.
- [44] iPhone: <<http://www.apple.com/iphone/>>.
- [45] jig Browser: <<http://br.jig.jp/pc/>>.
- [46] Komaki, D., Ohnishi, K., Arase, Y., Hara, T., Hattori, G., and Nishio, S.: Click-Search Interface for Web Browsing using Cellular Phones, in *Proceedings of DICO 2008*, Vol. 2008, No. 1, pp. 553–560 (July 2008) (in Japanese).

- [47] Komaki, D., Ohnishi, K., Arase, Y., Hara, T., Hattori, G., and Nishio, S.: A Click-Search Interface for Web Browsing using Cellular Phones, in *Proceedings of International Wireless Internet Conference (WICON 2008)*, CD-ROM (Nov. 2008).
- [48] Komaki, D., Ohnishi, K., Arase, Y., Hara, T., Hattori, G., and Nishio, S.: Design and Implementation of a Click-Search Interface for Web Browsing using Cellular Phones, *International Journal of Web and Grid Services (IJWGS)*, Vol. 5, No. 1, pp. 66–84 (Mar. 2009).
- [49] Komaki, D., Ohnishi, K., Arase, Y., Hara, T., Hattori, G., Takishima, Y., and Nishio, S.: Evaluation of a Click-Search Interface for Web Browsing using Cellular Phones, in *Proceedings of DICOMO 2009*, Vol. 2009, No. 1, pp. 263–271 (July 2009) (in Japanese).
- [50] Kristoffersen, S. and Ljungberg, F.: “Making Place” to Make It Work: Empirical Exploration of HCI for Mobile CSCW, in *Proceedings of ACM SIGGROUP Conference on Supporting Group Work (GROUP 1999)*, pp. 276–285 (Nov. 1999).
- [51] Kruskal, W. H. and Wallis, W. A.: Use of Ranks in One-Criterion Variance Analysis, *Journal of the American Statistical Association*, Vol. 47, No. 260, pp. 583–621 (Dec. 1952).
- [52] Loopt: <<http://www.loopt.com/>>.
- [53] Lumsden, J. and Brewster, S.: A Paradigm Shift: Alternative Interaction Techniques for Use with Mobile & Wearable Devices, in *Proceedings of Conference of the Centre for Advanced Studies on Collaborative Research (CASCON 2003)*, pp. 197–210 (Oct. 2003).
- [54] Maekawa, T., Hara, T., and Nishio, S.: A Collaborative Web Browsing System for Multiple Mobile Users, in *Proceedings of IEEE International Conference on Pervasive Computing and Communications (PerCom 2006)*, pp. 22–35 (Mar. 2006).

- [55] Maekawa, T., Hara, T., and Nishio, S.: Image Classification for Mobile Web Browsing, in *Proceedings of International World Wide Web Conference (WWW 2006)*, pp. 43–52 (May 2006).
- [56] Maekawa, T., Hara, T., and Nishio, S.: Two Approaches to Browse Large Web Pages using Mobile Devices, in *Proceedings of International Conference on Mobile Data Management (MDM 2006)*, pp. 52–55 (May 2006).
- [57] Maekawa, T., Uemukai, T., Hara, T., and Nishio, S.: A Java-Based Information Browsing System in a Remote Display Environment, in *Proceedings of IEEE International Conference on E-Commerce Technology (CEC 2004)*, pp. 342–346 (July 2004).
- [58] Maekawa, T., Uemukai, T., Hara, T., and Nishio, S.: Content Description and Partitioning Methods for Collaborative Browsing by Multiple Mobile Users, in *Proceedings of International Workshop on Mobility in Databases and Distributed Systems (MDDS 2005)*, pp. 1068–1072 (Aug. 2005).
- [59] Mann, H. B. and Whitney, D. R.: On a Test of Whether One of Two Random Variables is Stochastically Larger than the Other, *Annals of Mathematical Statistics*, Vol. 18, No. 1, pp. 50–60 (Mar. 1947).
- [60] MeCab: Yet Another Part-of-Speech and Morphological Analyzer: <<http://mecab.sourceforge.net/>>.
- [61] Microsoft Silverlight: <<http://www.microsoft.com/Silverlight/>>.
- [62] Miller, G. A.: WordNet: A Lexical Database for English, *Communications of the ACM*, Vol. 38, Issue 11, pp. 39–41 (Nov. 1995).
- [63] NetFront: <<http://www.access-netfront.com/>>.
- [64] Ohnishi, K., Arase, Y., Hara, T., Uemukai, T., and Nishio, S.: A Web Browsing System using Acceleration Sensors for Cellular Phone Users, in *Proceedings of DEWS 2007* (Feb./Mar. 2007) (in Japanese).

- [65] Ohnishi, K., Arase, Y., Hara, T., Uemukai, T., and Nishio, S.: Preview Functions for Web Browsing using Cellular Phones, in *Proceedings of International Workshop on Data Management for Information Explosion in Wireless Networks (DMIEW 2009)*, pp. 1070–1075 (Mar. 2009).
- [66] Ohnishi, K., Arase, Y., Hara, T., Uemukai, T., and Nishio, S.: A Content Clipping System for Web Browsing using Cellular Phones, in *Proceedings of International Workshop on Data Management for Wireless and Pervasive Communications (DMWPC 2009)*, pp. 978–983 (May 2009).
- [67] Opera for Mobile: <<http://www.opera.com/products/mobile/>>.
- [68] Oulasvirta, A., Tamminen, S., Roto, V., and Kuorelahti, J.: Interaction in 4-Second Bursts: The Fragmented Nature of Attentional Resources in Mobile HCI, in *Proceedings of Annual SIGCHI Conference on Human Factors in Computing Systems (CHI 2005)*, pp. 919–928 (Apr. 2005).
- [69] Paek, T., Agrawala, A., Basu, S., Drucker, S., Kristjansson, T., Logan, R., Toyama, K., and Wilson, A.: Toward Universal Mobile Interaction for Shared Displays, in *Proceedings of ACM Conference on Computer Supported Cooperative Work (CSCW 2004)*, pp. 266–269 (Nov. 2004).
- [70] Pascoe, J., Ryan, N., and Morse, D.: Using while Moving: HCI Issues in Fieldwork Environments, *ACM Transactions on Computer-Human Interaction (TOCHI)*, Vol. 7, No. 3, pp. 417–137 (Sept. 2000).
- [71] Picassa: <<http://picasa.google.com/>>.
- [72] RFC3875 The Common Gateway Interface (CGI) Version 1.1: <<http://www.ietf.org/rfc/rfc3875.txt>>.
- [73] Roto, V., Popescu, A., Koivisto, A., and Vartiainen, E.: Minimap: A Web Page Visualization Method for Mobile Phones, in *Proceedings of Annual SIGCHI Conference on Human Factors in Computing Systems (CHI 2006)*, pp. 35–45 (Apr. 2006).
- [74] Scope: <<http://www.programmer.co.jp/>>.

- [75] SH900i:
<<http://www.nttdocomo.co.jp/support/utilization/product/sh900i/index.html>>.
- [76] SH902iS:
<<http://www.nttdocomo.co.jp/support/utilization/product/sh902is/index.html>>.
- [77] Shen, D., Chen, Z., Yang, Q., Zeng, H.-J., Zhang, B., Lu, Y., and Ma, W.-Y.: Web-Page Classification through Summarization, in *Proceedings of Annual ACM SIGIR Conference (SIGIR 2004)*, pp. 242–249 (July 2004).
- [78] Spence, R.: Rapid, Serial and Visual: A Presentation Technique with Potential, *Information Visualization*, Vol. 1, Issue 1, pp. 13–19 (Mar. 2002).
- [79] Tajima, K. and Ohnishi, K.: Browsing Large HTML Tables on Small Screens, in *Proceedings of ACM Symposium on User Interface Software and Technology (UIST 2008)*, pp. 259–268 (Nov. 2008).
- [80] Uemukai, T., Hagino, H., Hara, T., Tsukamoto, M., and Nishio, S.: A Remote WWW Display Environment using a Cellular Phone WWW Service, in *Proceedings of Asian International Mobile Computing Conference (AMOC 2000)*, pp. 131–139 (Oct. 2000).
- [81] Uemukai, T., Hagino, H., Hara, T., Tsukamoto, M., and Nishio, S.: An Access Control Mechanism in a Remote Display Environment, in *Proceedings of IASTED International Conference on Advances in Communications (AIC 2001)*, pp. 178–183 (July 2001).
- [82] Uemukai, T., Hara, T., Tsukamoto, M., and Nishio, S.: A Remote Display Environment: An Integration of Mobile and Ubiquitous Computing Environments, in *Proceedings of IEEE Wireless Communications and Networking Conference (IEEE WCNC 2002)*, Vol. 2, pp. 618–624 (Mar. 2002).
- [83] Wobbrock, J., Forlizzi, J., Hudson, S., and Myers, B.: WebThumb: Interaction Techniques for Small-Screen Browsers, in *Proceedings of ACM*

- Symposium on User Interface Software and Technology (UIST 2002)*, pp. 205–208 (Oct. 2002).
- [84] World Wide Web Consortium, Document Object Model (DOM) Home Page: <<http://www.w3.org/DOM/>>.
- [85] World Wide Web Consortium, Extensible Markup Language (XML) Home Page: <<http://www.w3.org/XML/>>.
- [86] World Wide Web Consortium, HyperText Markup Language (HTML) Home Page: <<http://www.w3.org/MarkUp/>>.
- [87] Xie, X., Liu, H., Goumaz, S., and Ma, W.-Y.: Learning User Interest for Image Browsing on Small-Form-Factor Devices, in *Proceedings of Annual SIGCHI Conference on Human Factors in Computing Systems (CHI 2005)*, pp. 671–680 (Apr. 2005).
- [88] Yahoo! MOBILE: <<http://mobile.yahoo.com/search>>.
- [89] Yang, G., Tan, W., Mukherjee, S., Ramakrishnan, I. V., and Davulcu, H.: On the Power of Semantic Partitioning of Web Documents, in *Proceedings of Information Integration on the Web (IIWeb-03)*, pp. 39–46 (Aug. 2003).
- [90] Yang, Y. and Zhang, H.: HTML Page Analysis based on Visual Cues, in *Proceedings of International Conference on Document Analysis and Recognition (ICDAR 2001)*, pp. 859–864 (Sept. 2001).