

Web Refinding Support System Based on Process Recollective Activity

Masashi Toda

Center for Multimedia and Information Technologies, Kumamoto University, Kumamoto, Japan

Email: toda@cc.kumamoto-u.ac.jp

Abstract—The recent growth of search technology has enabled people to find information more easily. However, most people need to re-find information on a daily basis. Finding and re-finding are different activities and require different types of support. However, current re-finding support systems don't consider this point. This has caused several problems: PVR, loss of contextual information, and difference in search experiences. We discuss these problems and their solutions from a cognitive perspective. We propose a process-recollective re-finding support system based on this discussion. We demonstrate a novel approach to re-finding information on the web and a specific system as an example.

Index Terms—web re-finding, process-recollective, human-memory

1. Introduction

Recently, the growth of search technology has enabled people to find information more easily. However, at the same time, storage and organization of information is becoming a more significant problem[10]. The need to re-find information occurs on a daily basis.

To solve this problem, we use familiar re-finding tools such as bookmarks. Many studies have extended these tools to make them more convenient. However, the current re-finding support systems have significant common problems: PVR (Post-Valued Recall[18]), loss of context information, and so on. The most significant problem is an absence of perspective on the cognitive side.

Finding and re-finding require different types of support [5]. This is because re-finding depends heavily on recollection. This dependency causes crucial differences in user behavior. To design an effective re-finding support system, re-finding researchers should consider the difference between these activities. However, no existing system has thus far taken these ideas into account. In this study, we propose a process-recollective re-finding support system. "Process-recollective" means that a user completes re-finding by himself by repeating the previous finding process. This approach is fundamentally different from existing methods that are designed to provide direct access to re-finding targets.

In process-recollective re-finding, users can use rich resources embedded in the context for their re-finding activity. This approach also solves existing re-finding

problems, such as PVR and loss of contextual information.

This paper is organized as follows. Section 2 describes the current state and problems of the re-finding field. Section 3 discusses related work on solving the problems discussed in section 2. In section 4, we propose an effective re-finding support method from a social/cognitive viewpoint. Section 5 shows an outline of the system we propose. And finally, Section 6 concludes this study and discusses future work.

2. Background

As Pitkow's experiment shows[13], Bookmark is the most popular re-finding support method. This feature was supported as a browser function starting in the early stages of the web. However, many researchers (e.g. Abram[3]) have pointed out that Bookmark has problems, such as difficulty structuring information. Increasing the number of bookmarks makes these problems more complicated. Many systems (e.g. PowerBookmarks[11]) have extended Bookmark to solve these problems.

Jones et al.[10] discussed the difficulty of re-finding and people's daily re-finding activity. They called these re-finding activities, "Keeping Found Things Found (KFTF)". They showed that people use several familiar methods for increasing re-finding efficiency: mail, printouts, notes, web sites, and so on.

But most of these methods have three significant common problems:

- PVR (Post-Valued Recall)
- Loss of contextual information
- Difference of experience

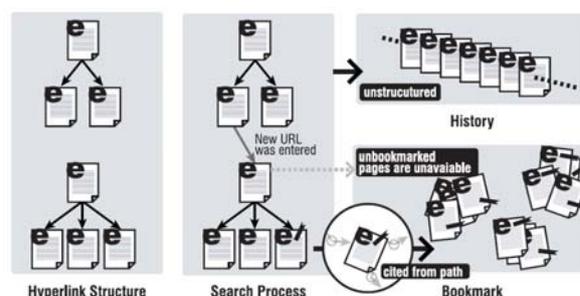


Figure 1: Characteristic differences between information spaces.

Wen pointed out that "a user may have trouble recalling information whose value is not recognized until some

time after its initial retrieval"[18]. He discussed this problem as PVR (Post-Valued Recall). Most current refinding methods, which need eclectic information, have this problem. For example, as figure 1 shows, Bookmark cannot refind target information that was not bookmarked.

A second problem is a loss of contextual information. This problem is caused by the data structure of current refinding systems. A user's experience of browsing a particular web page is embedded in a semantic/temporal context of web browsing. However, as figure 1 shows, current refinding support systems (e.g. bookmark and history) manage information in another, new information space. In other words, these systems remove this contextual information from the user's original search process. The important point is that this removed contextual information includes rich cues that help in refinding. The difficulty of refinding is directly affected by richness of these cues. Capra et al conducted refinding experiments[6] that showed that these cues for refinding (they called them "waypoints") helped users in most refinding test cases.

A third problem is difference of experience. This problem is caused by the interface of an existing refinding system. From the perspective of a user's activities, two experiences, previous finding from a WWW space and refinding from an eclectic information space, are fundamentally different[5]. In other words, the refinding context is different from the original finding context. This difference of context makes refinding more difficult[17]. Previous contextual information includes unconscious cues, which are not free-recalled but are recognized if the context is duplicated. According to the theory of context-dependency[9], even non-semantically related elements within target information play a role as cues for refinding. However, if the context was not duplicated, these unconscious / un verbalized cues do not help with refinding. This leads to a low rate of success in refinding.

Some of the problems we have discussed have already been resolved in related work. However, at the moment, no refinding support system solves all of these problems at the same time. The system we propose solves these problems with a directed graph data structure and process repeating.

3. Related Work

History was the only solution that didn't need eclectic information and that was able to solve the PVR problem. But History doesn't have a useful interface for refinding and doesn't maintain structure during the browsing process.

As figure1 shows, a browsing process has original information spaces, which are different from the Bookmark, History, and WWW Hyperlink structures. To improve the efficiency of refinding, the data structure of this process should, as much as possible, be retained.

To address this, some systems (e.g. WebNet[7] and MosaicG[4]), which are extensions of History, manage

history data as a graph structure. These systems solved problems such as PVR and loss of contextual information at the same time. However, even these systems cannot solve the problem of difference of experience.

Here we propose a process-recollective refinding support system. First, like the above-mentioned systems, our system uses a directed graph structure to manage history data. In our system though, refinding is done by the users themselves as a repeat of the previous finding process. This approach solves the problem of difference of experience. It also means that the role of this system differs greatly from that of current refinding support systems. Historically, these systems "get" cues for refinding from users and suggest target information directly. Our system, by contrast, "puts" cues for refinding to the users and suggests waypoints from the previous finding process.

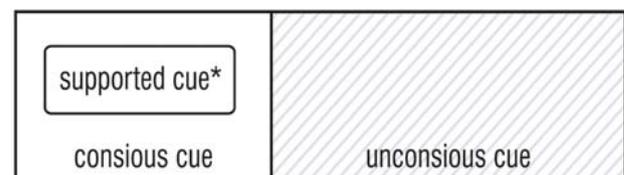
As table1 shows, our system solves all of the problems discussed in the previous section.

This process-recollective approach has not been addressed in the refinding field until now. This is because this approach seems to require more effort than traditional approaches. It's true that the effort required for process repeating is a problem. However, the purpose of this approach is not to replace existing approaches but to compensate for the weak points of traditional refinding systems. Users may use both approaches depending on the case.

Personalized search systems are similar to this approach. However, these systems support refinding only in their domains. In addition, these systems need to get keywords from a user to begin refinding support, but users do not always have keywords. Search engines were used in 44.60% of all tasks in an experiment by Capra et al[5] and keyword searches were used in 39% of the searches in a study done by the MIT Haystack group[16].

Basically, current refinding support systems need to get information from a user to begin refinding support. As figure2 shows, with these systems, available cues for refinding are limited to pieces the user is conscious of and that are supported by the system. In the process-recollective approach, by contrast, a user doesn't have to input information to the system explicitly. It enables a use of all refinding cues available to a user.

Like our approach, Remembrance Agent [14] focused attention on unconscious / un verbalized cues. This system suggests related information to the user automatically. However, it does not specialize in refinding specific information that a user needs.



*e.g. verbalized cue in keyword search

Figure 2: Classification of memorized cue.

	PVR	Loss of context information	Difference of experience
Bookmark			
History	✓		
Structured history	✓	✓	
Structured history + process-repetition	✓	✓	✓

Table 1: Response ability of refinding system

4. Design Theory

In the previous section, we pointed out advantages of a process-recollective refinding approach. In this section, we first discussed our design concept from a cognitive viewpoint. Then we explained the importance of Bookmark from a social viewpoint. This is why our system collaborates with social bookmark services.

4.1 Cognitive Aspects

Refinding is closely linked to human recollection [5]. Thus, in our design concept we take the following points from theories about recollection into account:

- Importance of context repetition
- Human-memory-like data structure
- Supporting Process Recollection

4.1.1 Importance of context repetition

The encoding specificity principle [17] is part of a theory that determines refinding success rates. According to this principle, whether a cue works effectively for refinding is determined by the existing context when user memorized the target information. If that context is repeated, refinding will be prompted by information included in the context. This explains a problem we described previously as difference of experience.

In traditional eclectic refinding, the context of previous searching (memorizing) is not repeated when a user refinds (recollection) particular information. This makes refinding more difficult. By contrast, in process-repetition refinding, the refinding context reproduces the previous context. This explains how process-repeating refinding supports refinding effectively from a cognitive perspective.

4.1.2 Human-memory-like data structure

In this subsection, we show how to reflect a recollection system to improve refinding support. To this end, we first compare the refinding process to the recollection process. The following are significant elements in the process of memorization and recollection:

- Memorization of target information is attended by encoding of contextual information
- The occurrence of a similar event encourages re-encoding of a previous memory
- Encoded contextual information works as a cue for refinding

As these points show, memorized information is reconstructive. Computer data storage, by contrast, has a static structure, which is not affected by reading. These differences in structure cause a gap between a user's expectations and the behavior of a system.

In some systems that have time-based interfaces (e.g. LifeStreams[8]), past data “floats” when a user is accessing it. This feature looks like reconstruction as it occurs in human memory. However, it is not a semantic reconstruction as human re-encoding is.

We use a directed graph structure of web browsing history as an imitation of a human memory system. Graph nodes correspond one-to-one with browsed web pages. Nodes are connected based on the page transitions in a user's browsing. If a user arrived at a page that was already shown in a previous session, these two sessions will be integrated by this common node. This sequential integration feature enables semantic structuring while retaining contextual information. Thus, in our system, all previous sessions are represented as a huge network.

As figure1 shows, this structured session data differs from the WWW hyperlink structure. The key difference is that one structure includes rich user-specific information and the other does not.

We sometimes confuse independent events that occurred at different moments. This is due to reconstruction of memory, which occurs with re-encoding. We explain this phenomenon with a simple example, as figure3 shows. One day, the user accessed pages A, B, and C, in that order. And at a later date, he accessed pages D, B, and E, in that order. However, he may recall page A as a cue of page E. This confusion is due to a duplication of page B. In a traditional refinding support system, he cannot refind page E if he recalls page A as a cue. This is because these systems managed multiple sessions independently or didn't retain the temporal context. By contrast, a directed graph structure, which is semantically and temporally structured, solves the problem of refinding.

As discussed in a previous section, this structure is already used in some systems, such as WebNet[7] and MosaicG[4]. The above discussion provided a new theory of availability to this data representation from a cognitive perspective. However, while traditional systems provide “wrong” paths, they don't provide cues to remind a user of the “right” path. Our system solves this problem by enabling a rich use of unconscious cues. The user will notice the right path using the cues embedded in the repeated finding process.

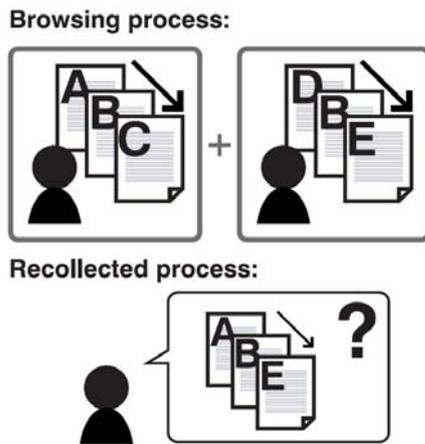


Figure 3: Example of memory confusion.

4.1.3 Supporting Process Recollection

A waypoint, as a cue for refinding, plays an important role in refinding. Our system artificially makes candidates of waypoints and presents them to the user as thumbnails. This helps the user's recollection process. To make candidates of waypoints, the system has to estimate what pages the user memorized clearly. We focused on the following four elements: bookmarks, the number of transit destinations from the page, traffic, and last access time.

The following paragraphs show how our system determines waypoints.

First, we identify bookmarked pages as waypoints. Bookmark is the clearest message that a user paid attention to the page. In addition, according to the theory of effect of self-choice, the act of bookmarking may promote memorization. For this reason, we identify bookmarked pages as waypoints. Why we used a social bookmark service will be explained later.

The second element, the number of transit destinations of from the page, is closely related to traffic, the third one. Having a lot of destinations indicates that there is also a lot of traffic. However, if multiple pages have the same traffic, the one with a lot of destinations may be memorized more clearly than the others. This can be seen from the following explanation. Having a lot of destinations also indicates that a user accessed the pages in several contexts. This leads to a great deal of re-encoding in different contexts. According to the theory of encoding specificity[17], re-encoding in different contexts increases the efficiency of memorization. Because of this, we judged the number of destinations to be a more important element than simple traffic.

If multiple pages have the same value for these three elements, we order them by last access time.

This was is due to the simple fact that older information is more likely to be forgotten.

The pages that were chosen as waypoints are represented to the user as thumbnails in our refinding process. According to the theory of pictorial superiority effect[12], visual information is recollected better than

text information. It means that thumbnails work well as cues for refinding.

4.2 Social Aspects

Bookmark has several weak points as a method for refinding. However, it is a familiar act that is part of everyday web browsing. Recently, the way bookmark is used has begun to change drastically. The core of the paradigm shift is represented by Social Bookmark services, such as Delicious[1]. The advantages of these services are that bookmarks can be used simultaneously in multiple computers and that they can flexibly classify information by tagging. The particularly explosive growth of Digg[2] indicates the possibility of "The Wisdom of Crowds"[15] and a novel propagation model for information. For users of these services, bookmarking is becoming sort of like easy voting. This change increased the number of bookmarks explosively. This indicates that Bookmark is beginning to take on new roles as a collaborative filtering mechanism and a life log on the web. Bookmark will be not only a private notation space but also an important element embedded naturally in daily web browsing activity.

Although section 2 shows the weak points of a traditional bookmark system, the above-mentioned background demonstrates the need to integrate refinding support systems and bookmarks. We integrate social bookmark services into our process-recollective refinding support system. It enables refinding that uses both artificial cues and cues that are embedded in context.

This indicates that our refinding support has two roles. The first one is as a non-traditional refinding support. The other is as an extension of bookmark that adds contextual information to finds bookmarks more easily. This role will be more important as the new use of bookmarking becomes more popular.

5. System Outline

Based on the previous discussion, we propose a process-recollective refinding support system. The following subsections explain data structure, system architecture, interface, and refinding interactions.

5.1 Data Structure

Our system stores the page transition information of web browsing as a directed graph structure. Graph nodes correspond one-to-one with browsed web pages. These nodes are connected according to the page transitions in the user's browsing activity. When the user enters a URL directly or opens the page from the bookmark, the system assumes that the user changes his or her browsing goal. In these cases, the system does not link nodes. There is an exception. If the user cuts text from a previous page and pastes it to the opened page, these two pages are linked in the system because the system assumes that these pages are related semantically. One

example of this occurs when the user finds an unknown keyword on a page and calls a search engine from a bookmark.

5.2 System Architecture

Figure 4 shows our system architecture. The system is used as a server application and a user script (Trixie / Greasemonkey / Opera User JavaScript) is used as a client. The following paragraphs describe the process of system behavior with reference to figure4.

First, process P1 and P2 are called at every page transition in a client's browser. The client sends information about a source page and a destination page to a server CGI (P1). The user sends the URL, title, and time. Then the CGI reads node-data that need to be updated from the Database. These nodes are updated (e.g. an update of an assessed value of the waypoint and an increment of the number of accessed pages) and stored in the database again (P2). If CGI cannot find node-data that are mapped to a received URL, a new node is created and added to a data structure.

The system imports the user's new bookmark from the social bookmark services at intervals. This is done by getting RSS from these services using information that is presented by the user (P3). Then the system updates an evaluated value of waypoint and stores the result in the database (P4).

When the user requests refinding support, the client sends the URL of the current browsing page to the CGI (P5). Then the CGI extracts and returns waypoint candidate URLs using the assessed values and received URL information (P6, P7).

After that, the client displays candidates to the user as thumbnails. The thumbnails are created by calling an external web service with received URLs.

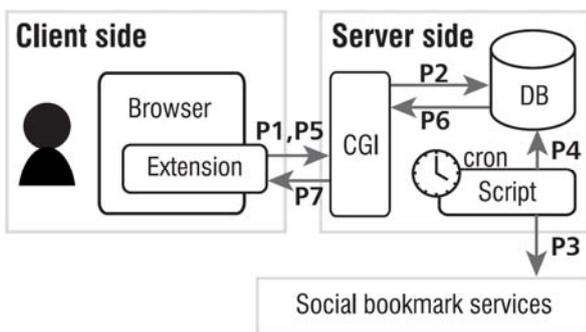


Figure 4: System architecture.

5.3 Interface

Figure5 shows an interface image of this system. Waypoint candidates are represented in the overlap area at the sides of the window. The right side displays forward candidates and the left side displays backward candidates. Backward candidates work well in particular refinding cases. For example, “In the previous section, I moved to the current page from a refinding target page”.

Waypoint candidates are represented as thumbnails. Basically, users repeat the previous browsing process with their own cues. However, based on their needs, they also can use these candidates to complement their cues. In addition, waypoint candidates have a role in shortcutting the repeating process.



Figure 5: System Interface image.

5.4 Example Interaction

In this subsection, we use a specific example to explain our refinding process. First, a user browses pages in previous sessions, as shown in figure6. In this case, the system's data structure is represented, as shown in figure7. After that, the user requires support to refind page D with the cue “I viewed page A before I viewed page D”. Then, he opens page A at the browser and enables our refinding support. Figure8 shows the screenshot at this point. In this figure, page B is chosen as a waypoint candidate because this page was viewed in the previous two sessions. Then the user clicks the thumbnail and the browser window appears as in figure9. Here page C is chosen as a primary candidate. Then the user clicks the thumbnail and the browser window appears as in figure10. In this figure, page B is displayed on the left side of the window as a backward candidate. Finally, the user finds the link to page D in the current browsing page.

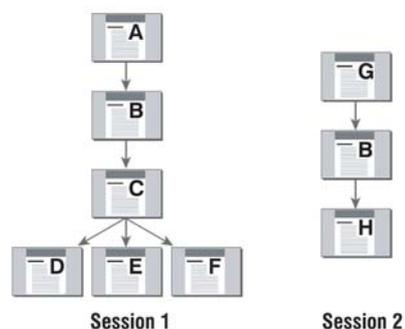


Figure 6: Page transitions.

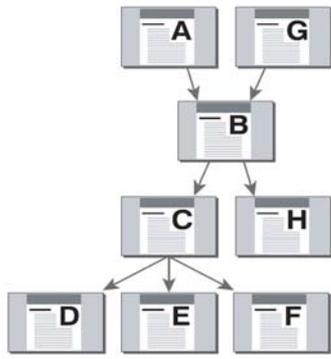


Figure 7: Data structure

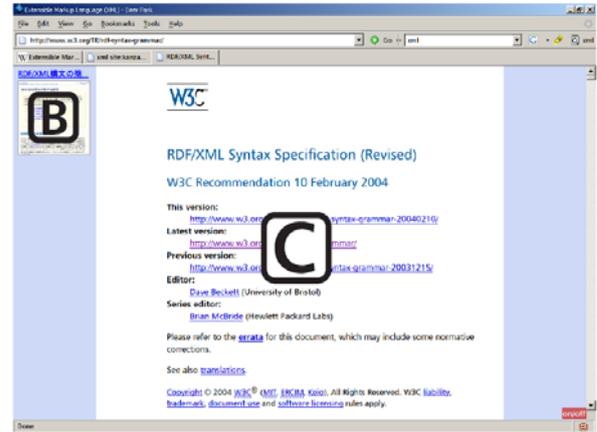


Figure 10: Refinding screenshots-3.



Figure 8: Refinding screenshots-1.

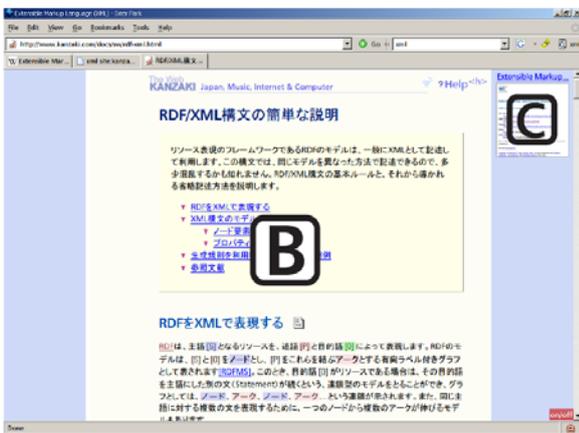


Figure 9: Refinding screenshots-2.

6. Conclusion

In this paper, we proposed a process-recollective refinding support system and provided a specific example of its use. Refinding has been the focus of a lot of recent attention. However, most traditional refinding support systems are based on an eclectic approach and don't take human cognition into account. We discussed the problems of traditional approaches and new theories on how to design refinding support in order to solve them. We expect our system to be used in future refinding studies.

The next focus of our work is to refine the system we proposed here. Currently, the following items are planned: adding new features that reflect other aspects of cognition, tuning-up the evaluation function, improving the sophistication of the interface, and collaborative refinding. An example of a new feature is process recollection support for false memory. By adding the feature, the semantic classifications of the bookmark will be integrated into the data structure. This will enable refinding support even if the user confuses multiple pages that are semantically or visually similar but are from independent contexts.

References

- [1] Del.icio.us, <http://del.icio.us/>
- [2] Digg, <http://digg.com/>
- [3] D. Abrams, R. Baecker, and M. Chignell. Information archiving with bookmarks: personalWeb space construction and organization. *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 41-48, 1998.
- [4] E. Ayers and J. Stasko. *Using Graphic History in Browsing the World Wide Web*. Graphics, Visualization & Usability Center, Georgia Institute of Technology, 1995.
- [5] R. Capra, M. Pinney, and M. P´erez-Quinones. Refinding Is Not Finding Again. Technical report, Technical Report TR-05-10, Computer Science, Virginia

Tech, 2005.

[6] R. Capra III and M. P´erez-Qui˜nones. Mobile refinding of web information using a voice interface: an exploratory study. *Proceedings of the 2005 Latin American conference on Human-computer interaction*, pages 88-99, 2005.

[7] A. Cockburn and S. Jones. Which way now? Analysing and easing inadequacies in WWW navigation. *International Journal of Human Computer Studies*, 45(1):105-129, 1996.

[8] E. Freeman and D. Gelernter. Lifestreams: a storage model for personal data. *ACM SIGMOD Record*, 25(1):80-86, 1996.

[9] R. Herz. The effects of cue distinctiveness on odor-based context-dependent memory. *Memory & Cognition*, 25(3):375-80, 1997

[10] W. Jones, H. Bruce, and S. Dumais. Keeping found things found on the web. *Proceedings of the tenth international conference on Information and knowledge management*, pages 119-126, 2001.

[11] W. Li, Y. Hara, R. Ito, Y. Kimura, K. Shimazu, Y. Saito, Q. Vu, E. Chang, D. Agrawal, K. Hirata, et al. PowerBookmarks: a system for personalizable Web information organization, sharing, and management. *Proceedings of the 1999 ACM SIGMOD international conference on Management of data*, pages 565-567, 1999.

[12] D. Nelson, V. Reed, and J. Walling. Pictorial superiority effect. *J Exp Psychol [Hum Learn]*, 2(5):523-8, 1976.

[13] J. Pitkow and C. Kehoe. Emerging trends in the WWW user population. *Communications of the ACM*, 39(6):106-108, 1996.

[14] B. Rhodes. The wearable remembrance agent: A system for augmented memory. *Personal Technologies*, 1(4):218-224, 1997.

[15] J. Surowiecki. *The wisdom of crowds: why the many are smarter than the few and how collective wisdom shapes business, economies, societies, and nations*. Doubleday, 2004.

[16] J. Teevan, C. Alvarado, M. Ackerman, and D. Karger. The perfect search engine is not enough: a study of orienteering behavior in directed search. *Proceedings of the 2004 conference on Human factors in computing systems*, pages 415-422, 2004.

[17] E. Tulving and D. Thomson. Encoding Specificity and Retrieval Processes in Episodic Memory. *Psychological Review*, 80(5):352-73, 1973.

[18] J. Wen. Post-Valued Recall Web Pages: User Disorientation Hits the Big Time. *IT & Society*, 1(3):184-194, 2003.

Masashi Toda was born in Hamamatsu, Shizuoka Pref., Japan in 1969. He received B.S. from the University of Tokyo, Japan in 1993, and M.S. and Ph.D. in electro-informatics engineering from Hokkaido University, Japan in 1995 and 1998. From 1998 to 2001, he was a researcher in IS Labo., SECOM Co., Ltd., Japan. From 2001 to 2005, he was an assistant professor in School of Systems Information Science in Future

University Hakodate, Japan. From 2005 to 2012, he was an associate professor in Future University Hakodate, Japan. Since 2012, he has been a professor in Center for Multimedia and Information Technologies, Kumamoto University, Japan. His main research interest is in sensing architecture. He is also interested in image processing technology, wearable computing, ubiquitous computing, information retrieval technology, and educational information system. He is a member of the Information Processing Society of Japan.