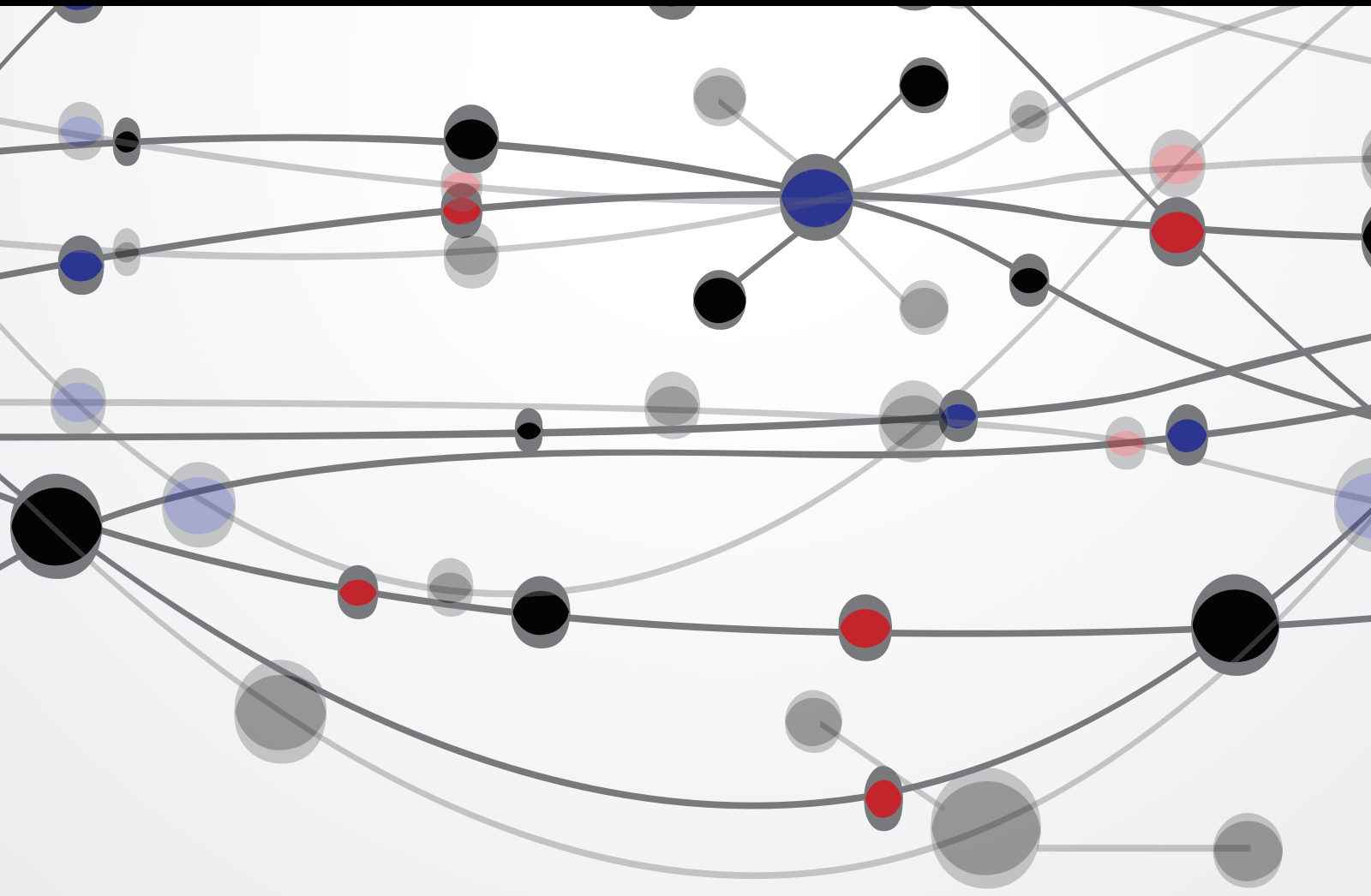
The Scientific World Journal

Intelligent User Interface for Interactive Multimedia: Emerging Techniques and Services

Guest Editors: Young-Sik Jeong, Jason C. Hung, and Mohammad S. Obaidat



Intelligent User Interface for Interactive Multimedia: Emerging Techniques and Services

The Scientific World Journal

Intelligent User Interface for Interactive Multimedia: Emerging Techniques and Services

Guest Editors: Young-Sik Jeong, Jason C. Hung, and Mohammad S. Obaidat

Contents

Hindawi Publishing Corporation
The Scientific World Journal
Volume 2014, Article ID 434326, 19 pages
http://dx.doi.org/10.1155/2014/434326

Research Article

Heuristic Evaluation on Mobile Interfaces: A New Checklist

Rosa Yáñez Gómez, Daniel Cascado Caballero, and José-Luis Sevillano

Department of Computer Technology and Architecture, ETS Ingenieria Informatica, Universidad de Sevilla, Avenida Reina Mercedes s/n. 41012 Seville, Spain

Correspondence should be addressed to José-Luis Sevillano; sevi@atc.us.es

Received 1 June 2014; Accepted 4 August 2014; Published 11 September 2014

Academic Editor: Mohammad S. Obaidat

The rapid evolution and adoption of mobile devices raise new usability challenges, given their limitations (in screen size, battery life, etc.) as well as the specific requirements of this new interaction. Traditional evaluation techniques need to be adapted in order for these requirements to be met. Heuristic evaluation (HE), an Inspection Method based on evaluation conducted by experts over a real system or prototype, is based on checklists which are desktop-centred and do not adequately detect mobile-specific usability issues. In this paper, we propose a compilation of heuristic evaluation checklists taken from the existing bibliography but readapted to new mobile interfaces. Selecting and rearranging these heuristic guidelines offer a tool which works well not just for evaluation but also as a best-practices checklist. The result is a comprehensive checklist which is experimentally evaluated as a design tool. This experimental evaluation involved two software engineers without any specific knowledge about usability, a group of ten users who compared the usability of a first prototype designed without our heuristics, and a second one after applying the proposed checklist. The results of this experiment show the usefulness of the proposed checklist for avoiding usability gaps even with nontrained developers.

1. Introduction

Usability is the extent to which a product can be used with effectiveness, efficiency, and satisfaction in a specified context of use [1]. While usability evaluation of traditional browsers from pc environments—desktop or laptop—has been widely studied, mobile browsing from smartphones, touch phones, and tablets present new usability challenges [2]. Additionally, mobile browsing is becoming increasingly widespread as a way of accessing online information and communicating with other users. Specific usability evaluation techniques adapted to mobile browsing constitute an interesting and increasingly important study area.

Usability evaluation assesses the ease of use of a website's functions and how well they enable users to perform their tasks efficiently [3]. To carry out this evaluation, there are several usability evaluation techniques.

Usability evaluation techniques can be classified as shown in Figure 1 [4–8]. Over real systems or prototypes, the best alternatives are evaluations conducted by experts, also known as Inspection Methods, or evaluations involving users, which are divided into inquiry methods and testing methods depending on the methodology adopted. With a more academic focus, predictive evaluation offers some predictions over the usability of a potential and not-yet-existent prototype.

Heuristic evaluation (HE) is an inspection method based on evaluation over real system or prototype, conducted by experts. The term "expert" is used as opposed to "users" but in many cases evaluators do not need to be usability experts [9, 10]. In HE, experts check the accomplishment of a given heuristic checklist. Due to its nature, this inspection cannot be performed automatically.

HE, like other usability assurance techniques, has to take into account the fact that usability is not intrinsically objective in nature but is rather closely intertwined with an evaluator's personal interpretation of the artefact and his or her interaction with it [11]. But, evaluations can be designed to compensate for personal interpretation as much as possible.

Moreover, inspection methods are often criticized for only being able to detect a small number of problems in total together with a very high number of cosmetic ones [12].

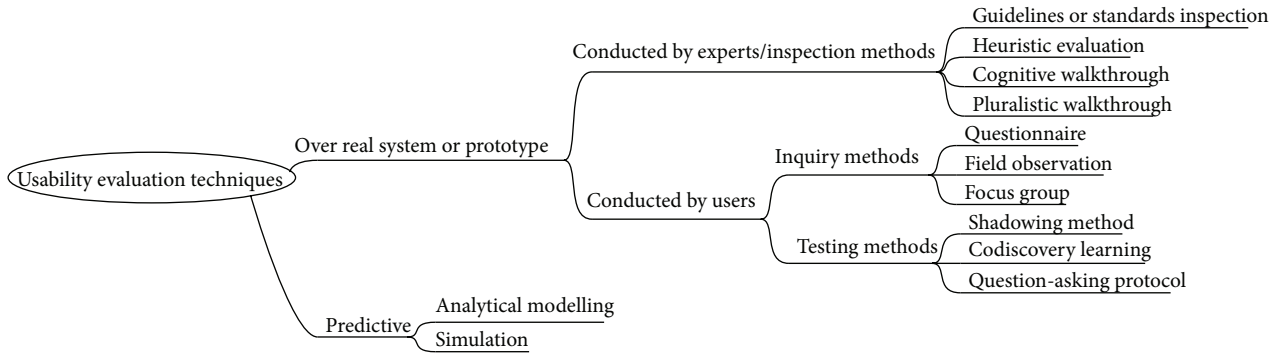


Figure 1: Classification of some usability evaluation techniques.

But, HE presents several advantages over other techniques: its implementation is easy, fast, and cheap, and it is suitable for every life-cycle software phase and does not require previous planning [7]. Furthermore, it is not mandatory for evaluators to be usability experts [9, 10]. It is possible for engineers or technicians with basic usability knowledge to drive an evaluation. Furthermore, regarding the number of evaluators, Nielsen demonstrated empirically that between three and five experts should be enough [13].

Because of all these advantages, HE is a convenient usability evaluation method: the worst usability conflicts are detected at a low cost. But, traditional HE checklists are desktop-centred and do not properly detect mobile-specific usability issues [2].

In this study, we propose a heuristic guideline centred in mobile environments based on a review of previous literature. This mobile-specific heuristic guideline is not only an evaluation tool but also a compilation of recommended best-practices. It can guide the design of websites or applications oriented to mobile devices taking usability into account.

The following section describes the methods followed to define the mobile heuristic guideline. Then, Results and Discussion section is divided according to the steps defined in the methodology. We have included a brief discussion of the results for each task. The final sections include Conclusions and Future Work, Acknowledgments, and References.

2. Methods

To obtain a heuristic guideline centred in mobile environments and based on a review of previous literature, we will follow a six-step process.

(1) A clear definition of the problem scope is necessary as a first step to define and classify the special characteristics of mobile interaction.

(2) Next, we rearrange existing and well-known heuristics into a new compilation. We can reuse heuristic guidelines from the literature and adapt them to the new mobile paradigm because heuristic checklists derive from human behaviour, not technology [14]. This heuristics is general checks that must be accomplished in order to achieve a high level of usability.

(3) After building this new classification of heuristics, we will develop a compilation of different proposed subheuristics. "Heuristic" in this paper refers to a global usability issue that must be evaluated or taken into account when designing. In contrast, the term "subheuristic" refers to specific guidelines items. The main difference between the two concepts lies in the level of expertise required of the evaluator and the abstraction level of the checklist. The resulting selection of subheuristics in this step takes into account some of the mobile devices restrictions presented in the first step. But, the result of this stage does not include many mobile specific questions, as they are not covered in traditional heuristic guidelines.

(4) The fourth step in this work consists of enriching the list with mobile-specific subheuristics. This subheuristics is gleaned from mobile usability studies and best practices proposed in the literature.

(5) One further step is required to homogenize the redaction and format of subheuristics in order to make it useful for nonexperts.

(6) Finally, we conduct an evaluation of the usefulness of the tool as an aid in designing for mobile.

This process differs slightly from the methodology proposed by Rusu et al. [15], but we can subsume their phases when establishing new usability heuristics in our proposed method.

It is worth remarking that popular mobile operating systems are now providing usability guidelines [16, 17] which focus mainly on maintaining coherent interaction and presentation through applications over the whole platform. These guidelines could in some cases enrich certain aspects of our proposal, although we have opted to keep it essentially agnostic of specific platforms aesthetics or coherence-determined restrictions.

Additionally, interfaces for mobile are mainly divided into web access and native applications. We do not restrict our study to a specific kind of interface. Again, the goal is to elaborate a guideline which is independent of specific technologies. The interaction between users and mobile interfaces is similar regardless of the piece of software they are using.

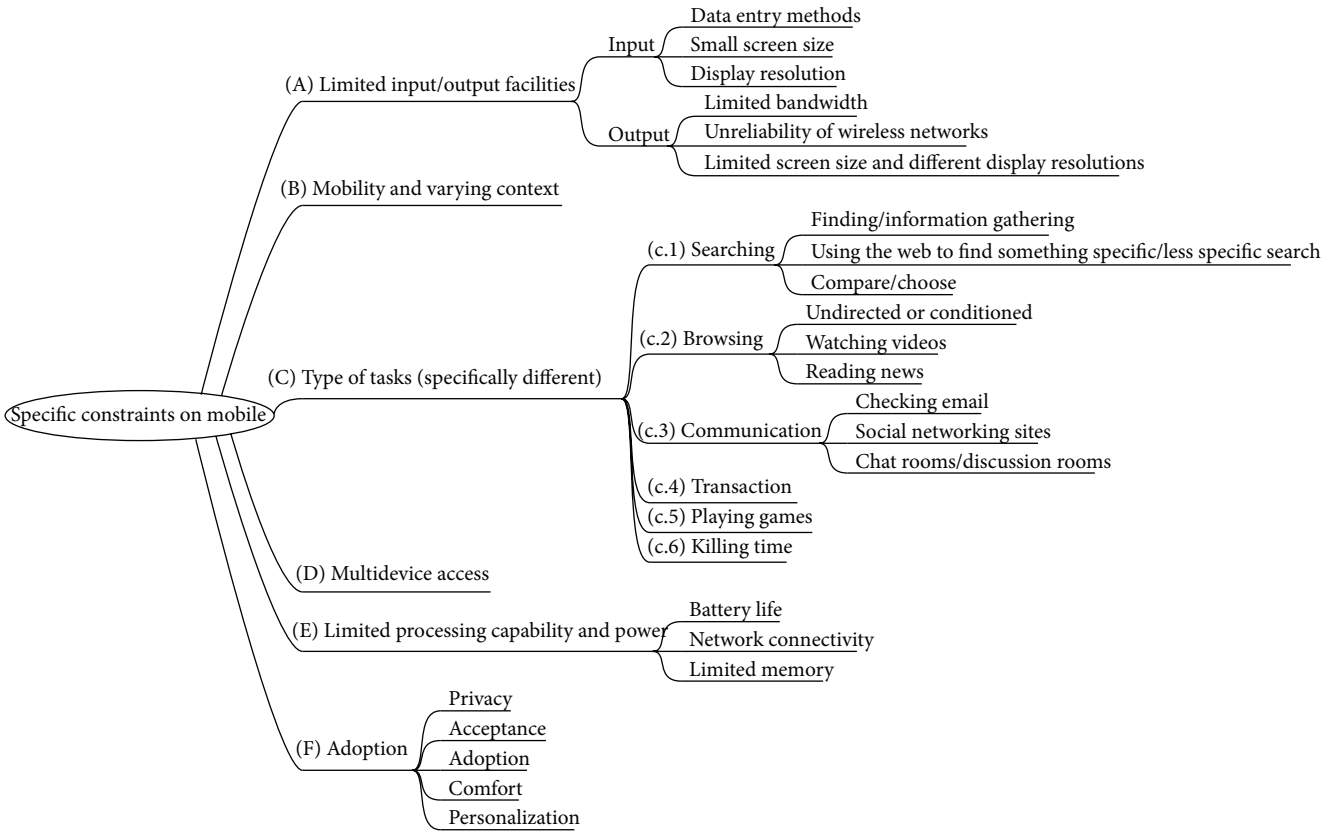


Figure 2: Specific constraints on mobile.

3. Results and Discussion

3.1. Problem Scope Definition. Users are increasingly adopting mobile devices. According to statistics of Pew Internet & American Life Project [18], only in the USA 35% of adults own smartphones and 83% of adults own a cell phone of some kind. Additionally, 87% of smartphones owners access the Internet or email on their handheld—68% on a typical day. A further 25% say that they mostly go online using their smartphone, rather than a computer. This survey shows that phones operating on the Android platform are currently the most prevalent type, followed by iPhones and Blackberry devices.

Mobile usability involves different kind of devices, contexts, tasks, and users. The compilation of a new heuristic guideline needs a restriction and definition of the scope of the user-interface interaction.

Devices can be divided in three types [19]:

(i) feature phones: they are basic handsets with tiny screens and very limited keypads that are suitable mainly for dialing phone numbers;

(ii) Smartphones: phones with midsized screens and full A–Z keypads;

(iii) touch phones/touch tablets: devices with touch-sensitive screens that cover almost the entire front of the phone.

In our study, we have ruled out feature phones because the interaction and interface design are deeply restricted and they are gradually being abandoned by a wide range of users. We have also ruled out smartphones because interaction is dramatically different due to the keyboard and they are commonly constrained to enterprise use. This study focuses on the ubiquitous touch phones and touch tablets. In this work, we use the term "touch phones" to refer to both phones and tablets because they share a similar interaction paradigm and the constraints we describe in Figure 2.

Mobile interactions define a new paradigm characterized by a wide range of specific constraints: hardware limitations, context of use, and so forth. All these restrictions have been studied in the bibliography in order to define the issues that must be overcome to improve usability. According to the literature, the main constraints when designing for mobile devices are (Figure 2):

(A) *limited input/output facilities* [20–24]: these limitations are imposed by data entry methods, small screen size, display resolution, and available bandwidth, as well as unreliability of wireless networks;

(B) *mobility and varying context* [20–23]: traditional usability evaluation techniques have often relied on measures of task performance and task efficiency. Such evaluation approaches may not be directly applicable to the often unpredictable, rather opportunistic and relatively unstable mobile settings. Mobile

devices use is on-the-run and interactions may take from a few seconds to minutes, being highly context-dependent. Environmental distractions have a significant effect on mobile interfaces usability and hence they need to be taken into account [25]. Context of use involves background noise, ongoing conversations, people passing by, and so on. Distractions can be auditory, visual, social, or caused by mobility.

The context of use is so influent in the interaction that many authors propose testing in the field as indispensable to study interaction with mobile devices [26]. Laboratory testing seems incapable of completely assuring usability in this mobile paradigm. Some attempts to cover this contextual information have been documented in the literature: Po et al. [27] proposed inclusion of contextual information into the heuristic evaluation proposed by Nielsen and Molich [9]; Bertini et al. [28] discussed the capacity of expert-based techniques to capture contextual factors in mobile computing. Indeed, it is not trivial to integrate real-world setting/context into inspection methods which are conceived as laboratory testing techniques. In any case, laboratory testing and expert-based techniques are complementary. Both approaches can be used in preliminary analysis and design of prototypes but, even more in mobile than when dealing with old desktop interaction paradigms, they need to be complemented with users-based testing.

(C) *Type of Tasks*: in mobile environments, typical tasks are relatively different from traditional desktop devices. From the origins of mobile devices, concepts such as "personal space extension" [29] previewed new uses of mobile terminals.

The literature has tended to classify mobile tasks on the basis of searching/browsing categories and also according to management of known information or new information. It is important to note that pre-2007 literature does not widely consider touch terminals which incorporate new tasks. Having taken all this into account, we can classify tasks as follows:

(i) search [29–35]:
  (a) information gathering [33];
  (b) using the web to more or less specific search;
  (c) compare/choose [32];

(ii) browsing [30, 31, 33–35]:
  (a) undirected or conditioned browsing [31];
  (b) watching videos [14];
  (c) reading news [14];

(iii) communication [14, 33, 35]:
  (a) checking email [14];
  (b) social networking sites [14];
  (c) chat rooms/discussion rooms [33];

(iv) transaction [29, 33, 34]: although it is more common to use a mobile device to browse the web or to perform some shopping-related search than shopping in earnest [14].

(v) playing games [14];

(vi) killing time [14].

Some literature includes other kinds of task like "maintenance" [35] or "housekeeping" [33] that have not been included in our classification because the frequency of realization is too low and these kinds of tasks do not define new kinds of interactions.

(D) *Multidevice access*: user's familiarity with a web page [34] helps them to construct a mental model based on the structural organization of the information, such as visual cues, layout, and semantics. When a site is being designed for multidevice access, a major concern is to minimize user effort to reestablish the existing mental model. This new way of working around structured information that must be delivered through so many different interface restrictions has been studied as a new paradigm known as Responsive Design [36].

(E) *Limited processing capability and power* [21–24]: these limitations include battery life, network connectivity, download delays, and limited memory.

(F) *Adoption* [22]: adoption of mobile technology by users is based on perceived privacy, acceptance of technology, comfort, and capacity of personalization. Different levels of adoption determine different group of users interacting in a very different way with the interface. This may not seem to be a mobile-specific restriction but the wide variety of mobile devices, touchable or keyboard-based, with different sizes and presentation models, makes the range of users requiring different approaches much broader.

3.2. Rearrangement of Traditional Heuristics. The first rearrangement of traditional heuristics in this step is mainly based on the review of the literature by Torrente [7] where the author selected the most influent heuristics guidelines [9, 37–44]. This compilation gives a total of 9 heuristics guidelines consisting globally of 83 heuristics and 361 subheuristics. We need to rearrange this list of items into a new classification which is coherent to our purpose. In this step, we only take into account "heuristics" and no "subheuristics" and this gives us the heuristic list shown in Figure 3.

Our rearrangement focuses on literature coincidences (i.e., when the same concept or category is included in different works in the literature, perhaps under different names) and tries to propose a coherent, exhaustive, and complete framework of heuristics that could be used to arrange further identified subheuristics. Literature coincidences for each heuristics are as follows:

(1) visibility of system status [9, 38, 44]: other bibliography references include this concept as "Track State" [41], "Give feed-back" [37], or "Feedback" [40];

(2) match between system and the real world [9, 38];

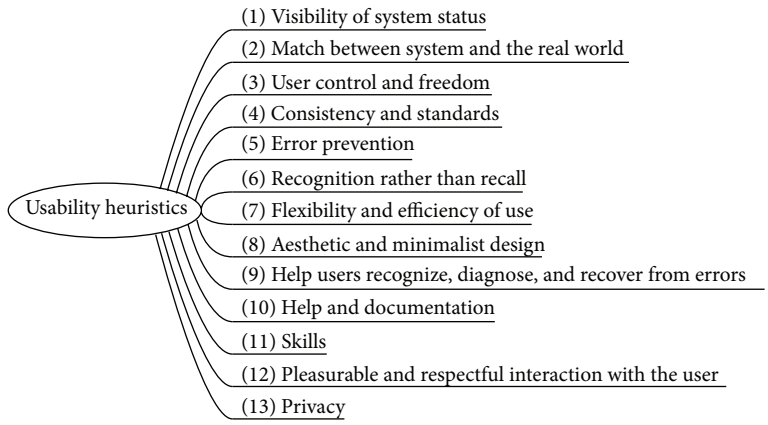


Figure 3: Proposed heuristic list.

(3) user control and freedom [9, 38, 42, 43]: other bibliography references include this concept as "Support the user control" [37], "The feedback principle" [39], "Autonomy" [41], or "Visible Navigation" [41];

(4) consistency and standards [9, 38, 41, 44] also cited as "Maintain Consistency" [37], "Structure Principle" [39], "Reuse Principle" [39], "Consistency" [40], or "Learnability" [41];

(5) error prevention [9, 38, 40] also cited as "Tolerance Principle" [39];

(6) recognition rather than recall [9, 38] also cited as "Reduce recent memory load for users" [37], "Structure principle" [39], "Reuse principle" [39], "Minimize the users' memory load" [40], or "Anticipation" [41];

(7) flexibility and efficiency of use [9, 38] also cited as "Simplicity principle" [39] or "Look at the user's productivity not the computer's" [41];

(8) aesthetic and minimalist design [9, 38];

(9) help users recognize, diagnose, and recover from errors [9, 37, 38] also cited as "Good error messages" [40] or "In case of error, is the user clearly informed and not over-alarmed about what happened and how to solve the problem?" [42];

(10) help and documentation [9, 38, 40, 42–44];

(11) skills [38] also cited as "Prepare workarounds for frequent users" [37], "Shortcuts" [40], or "Readability" [41];

(12) pleasurable and respectful interaction with the user [38] also cited as "Simplicity principle" [39], "Simple and natural dialog," or "Speak the user's language" [40]: this point also includes any accessibility questions that could enrich usability allowing a more universal access, such as "Color blindness" [41];

(13) privacy [38].

3.3. Compilation of Subheuristics from Traditional General Heuristic Checklists. As defined before, "heuristic" in this paper refers to a global usability issue which must be evaluated or taken into account when designing. In contrast, the term "subheuristic" refers to specific guidelines items. In this third step, we focus on locating subheuristics from the literature.

The first group of potential heuristics is the 361 subheuristics proposed in the 9 references selected by Torrente [7]. Among these sub-heuristics we exclude those that do not fit well with the previously described mobile constraints. For example, subheuristics referred to desktop data entry methods is obviously discarded. In contrast, this referring to screen use optimization is particularly relevant. Other discarded amounts of subheuristics include some proposed [38] with specific response times which do not apply in a mobile and varying context. We also discard coincidences between different authors proposals.

Thus, from a total of 361 amounts of subheuristics proposed by the 9 references [9, 37–44] selected by Torrente [7], in this study, we obtain a first selection of 158 subheuristics.

In order to maintain consistency in our classification, some subheuristics has been moved from their original heuristic parents, and new subcategories have been added so that semantically related amounts of subheuristics are grouped together. The final framework, shown in Figure 4, builds on that presented in the previous section.

It is also important to recall that at this stage, subheuristics redactions have been kept unchanged from their corresponding references. In the final compilation, these redactions will be modified in order to homogenize the whole guideline as we planned for step 4 in our methodology.

The final list of subheuristics is as follows:

(1) visibility of system status:

system status feedback:

(1) is there some form of system feedback for every operator action? [38]

(2) if pop-up windows are used to display error messages, do they allow the user to see the field in error? [38]

(3) in multipage data entry screens, is each page labeled to show its relation to others? [38]

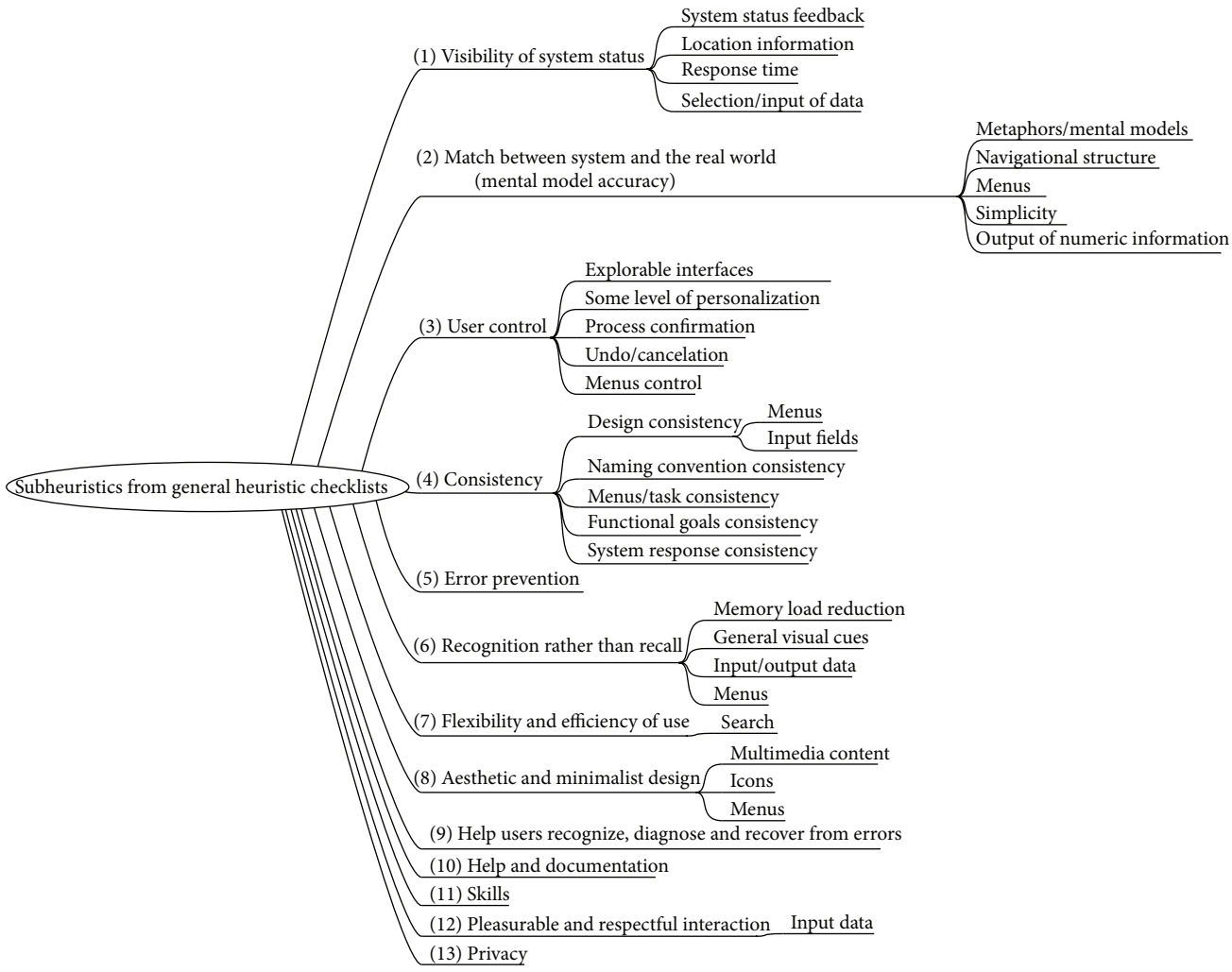


Figure 4: First framework for classification of detected subheuristics.

(4) are high informative contents placed in high hierarchy areas? [42]

location information:

(5) is the logo meaningful, identifiable, and sufficiently visible? [42]

(6) is there any link to detailed information about the enterprise, website, webmaster . . . ? [42]

(7) are there ways of contacting with the enterprise? [42]

(8) in articles, news, reports . . . are the author, sources, dates, and review information shown clearly? [42]

response times:

(9) are response times appropriate for the users cognitive processing? [38]

(10) are response times appropriate for the task? [38]

(11) if there are observable delays (greater than fifteen seconds) in the system's response time, is the user kept informed of the system progress? [38]

(12) latency reduction [41];

Selection/input of data:

(13) is there visual feedback in menus or dialog boxes about which choices are selectable? [38]. We will merge this statement with the following: "Do GUI menus make obvious which item has been selected?" [38], "Do GUI menus make obvious whether deselection is possible?" [38], "Is there visual feedback in menus or dialog boxes about which choice the cursor is on now?" [38], and "If multiple options can be selected in a menu or dialog box, is there visual feedback about which options are already selected?" [38]

(14) is the current status of an icon clearly indicated? [38]

(15) is there visual feedback when objects are selected or moved? [38]

(16) are links recognizable? Is there any characterization according to the state (visited, active,. . .)? [42]

(2) match between system and the real world (Mental model accuracy):

metaphors/mental models:

(17) use of metaphors [41];

(18) are icons concrete and familiar? [38]

(19) if shape is used as a visual cue, does it match cultural conventions? [38]

(20) do the selected colours correspond to common expectations about color codes? [38]

navigational structure:

(21) if the site uses hierarchical structure, are depth and height balanced? [42]

(22) navigation map [44], also known as site map or table of contents;

menus:

(23) are menu choices ordered in the most logical way, given the user, the item names, and the task variables? [38]

(24) do menu choices fit logically into categories that have readily understood meanings? [38]

(25) are menu titles parallel grammatically? [38]

(26) in navigation menus, are the number of items and terms by item controlled to avoid memory overload? [42]

simplicity:

(27) do related and interdependent fields appear on the same screen? [38]

(28) for question and answer interfaces, are questions stated in clear, simple language? [38]

(29) is the language used the same target users speak? [42]. We will merge this statement with the following: "Is the menu-naming terminology consistent with the user's task domain?" [38]

(30) is the language clear and concise? [42]. We will merge this statement with the following: "Does the command language employ user jargon and avoid computer jargon?" [38]

(31) does the site follow the rule "1 paragraph = 1 idea"? [42]

output of numeric information:

(32) does the system automatically enter leading or trailing spaces to align decimal points? [38]

(33) does the system automatically enter a dollar sign and decimal for monetary entries? [38]

(34) does the system automatically enter commas in numeric values greater than 9999? [38]

(35) are integers right-justified and real numbers decimal-aligned? [38]

(3) user control:

explorable interfaces:

(36) can users move forward and backward between fields or dialog box options? [38]

(37) if the system has multipage data entry screens, can users move backward and forward among all the pages in the set? [38]

(38) if the system uses a question and answer interface, can users go back to previous questions or skip forward to later questions? [38]

(39) clearly marked exits [40];

(40) is the general website structure user-oriented? [42]

(41) is there any way to inform user about where they are and how to undo their navigation? [42]

some level of personalization:

(42) can users set their own system, session, file, and screen defaults? [38]

process confirmation:

(43) when a user's task is complete, does the system wait for a signal from the user before processing? [38]

(44) are users prompted to confirm commands that have drastic, destructive consequences? [38]

undo/cancelation:

(45) can users easily reverse their actions? [38] Also found as "Do function keys that can cause serious consequences have an undo feature?" [38] and "Is there an "undo" function at the level of a single action, a data entry, and a complete group of actions?" [38]

(46) can users cancel out of operations in progress? [38]

menus control:

(47) if the system has multiple menu levels, is there a mechanism that allows users to go back to previous menus? [38]

(48) are menus broad (many items on a menu) rather than deep (many menu levels)? [38]

(49) if users can go back to a previous menu, can they change their earlier menu choice? [38]

(4) consistency:

designing consistency:

(50) are attention-getting techniques used with care? [38]
(51) intensity: two levels only [38];
(52) color: up to four (additional colors for occasional use only) [38];
(53) are there no more than four to seven colors, and are they far apart along the visible spectrum? [38]
(54) sound: soft tones for regular positive feedback, harsh for rare critical conditions [38];
(55) if the system has multipage data entry screens, do all pages have the same title? [38]
(56) do online instructions appear in a consistent location across screens? [38]
(57) have industry or company standards been established for menu design, and are they applied consistently on all menu screens in the system? [38]
(58) are there no more than twelve to twenty icon types? [38]
(59) has a heavy use of all uppercase letters on a screen been avoided? [38]
(60) is there a consistent icon design scheme and stylistic treatment across the system? [38]

menus:

(61) are menu choice lists presented vertically? [38]
(62) if "exit" is a menu choice, does it always appear at the bottom of the list? [38]
(63) are menu titles either centered or left-justified? [38]

input fields:

(64) are field labels consistent from one data entry screen to another? [38]
(65) do field labels appear to the left of single fields and above list fields? [38]
(66) are field labels and fields distinguished typographically? [38]

naming convention consistency:

(67) is the structure of a data entry value consistent from screen to screen? [38]
(68) are system objects named consistently across all prompts in the system? [38]
(69) are user actions named consistently across all prompts in the system? [38]

menu/task consistency:

(70) are menu choice names consistent, both within each menu and across the system, in grammatical style and terminology? [38]
(71) does the structure of menu choice names match their corresponding menu titles? [38]
(72) does the menu structure match the task structure? [38]
(73) when prompts imply a necessary action, are the words in the message consistent with that action? [38]

functional goals consistency:

(74) where are the website goals? Are they well defined? Do content and services delivered match these goals? [42]
(75) does the look & feel correspond with goals, characteristics, contents and services of the website? [42]
(76) is the website being updated frequently? [42]

system response consistency:

(77) is system response after clicking links predictable? [42]
(78) are nowhere links avoided? [42]
(79) are orphan pages avoided? [42]

(5) error prevention:

(80) are menu choices logical, distinctive, and mutually exclusive? [38]
(81) are data inputs case-blind whenever possible? [38]
(82) does the system warn users if they are about to make a potentially serious error? [38]
(83) do data entry screens and dialog boxes indicate the number of character spaces available in a field? [38]
(84) do fields in data entry screens and dialog boxes contain default values when appropriate? [38]

(6) recognition rather than recall:

memory load reduction:

(85) high levels of concentration are not necessary and remembering information is not required: two to fifteen seconds [38];
(86) are all data a user needs on display at each step in a transaction sequence? [38]
(87) if users have to navigate between multiple screens, does the system use context labels, menu maps, and place markers as navigational aids? [38]

(88) after the user completes an action (or group of actions), does the feedback indicate that the next group of actions can be started? [38]
(89) are optional data entry fields clearly marked? [38]
(90) do data entry screens and dialog boxes indicate when fields are optional? [38]
(91) is page length controlled? [42]

general visual cues:

(92) for question and answer interfaces, are visual cues and white space used to distinguish questions, prompts, instructions, and user input? [38]
(93) does the data display start in the upper-left corner of the screen? [42]
(94) have prompts been formatted using white space, justification, and visual cues for easy scanning? [38]
(95) do text areas have "breathing space" around them? [42]
(96) are there "white" areas between informational objects for visual relaxation? [42]
(97) does the system provide visibility; that is, by looking, can the user tell the state of the system and the alternatives for action? [38]
(98) is size, boldface, underlining, colour, shading, or typography used to show relative quantity or importance of different screen items? [38]
(99) is colour used in conjunction with some other redundant cue? [38]
(100) is there good colour and brightness contrast between image and background colours? [38]
(101) have light, bright, saturated colours been used to emphasize data and have darker, duller, and desaturated colours been used to deemphasize data? [38]
(102) is the visual page space well used? [42]

input/output data:

(103) on data entry screens and dialog boxes, are dependent fields displayed only when necessary? [38]
(104) are field labels close to fields, but separated by at least one space? [38]

Menus

(105) is the first word of each menu choice the most important? [38]
(106) are inactive menu items grayed out or omitted? [38]
(107) are there menu selection defaults? [38]
(108) is there an obvious visual distinction made between "choose one" menu and "choose many" menus? [38]

(7) flexibility and efficiency of use:

search:

(109) is the searching box easily accessible? [42]
(110) is the searching box easily recognizable? [42]
(111) is there any advanced search option? [42]
(112) are search results shown in a comprehensive manner to the user? [42]
(113) is the box width appropriated? [42]
(114) is the user assisted if the search results are impossible to calculate? [42]

(8) aesthetic and minimalist design:

(115) Fitt's Law [41]: the time to acquire a target is a function of the distance to and size of the target;
(116) is only (and all) information essential to decision making displayed on the screen? [38]
(117) are field labels brief, familiar, and descriptive? [38]
(118) are prompts expressed in the affirmative, and do they use the active voice? [38]
(119) is layout clearly designed avoiding visual noise? [42]

multimedia content:

(120) does the use of images and multimedia content add value? [42]
(121) are images well sized? Are they understandable? Is the resolution appropriate? [42]
(122) are cyclical animations avoided? [42]

icons:

(123) has excessive detail in icon design been avoided? [38]
(124) is each individual icon a harmonious member of a family of icons? [38]
(125) does each icon stand out from its background? [38]
(126) are all icons in a set visually and conceptually distinct? [38]

menus:

(127) is each lower-level menu choice associated with only one higher level menu? [38]
(128) are menu titles brief, yet long enough to communicate? [38]

(9) help users recognize, diagnose and recover from errors;
(10) help and documentation:

(129) are online instructions visually distinct? [38]

(130) do the instructions follow the sequence of user actions? [38]

(131) if menu choices are ambiguous, does the system provide additional explanatory information when an item is selected? [38]

(132) if menu items are ambiguous, does the system provide additional explanatory information when an item is selected? [38]

(133) is the help function visible, for example, a key labeled HELP or a special menu? [38, 42]

(134) is the help system interface (navigation, presentation, and conversation) consistent with the navigation, presentation, and conversation interfaces of the application it supports? [38]

(135) navigation: is information easy to find? [38]

(136) presentation: is the visual layout well designed? [38]

(137) conversation: is the information accurate, complete, and understandable? [38]

(138) is the information relevant? ([38], Help and documentation) [42] It should be relevant in the following aspects [38]: goal-oriented (what can I do with this program?), descriptive (what is this thing for?), procedural (how do I do this task?), interpretive (why did that happen?), and navigational (where am I?);

(139) is there context-sensitive help? [38, 42]

(140) can the user change the level of detail available? [38]

(141) can users easily switch between help and their work? [38]

(142) is it easy to access and return from the help system? [38]

(143) can users resume work where they left off after accessing help? [38]

(144) if a FAQs section exists, are the selection and redaction of questions and answers correct? [42]

(11) skills:

(145) do not use the word "default" in an application or service; replace it with "Standard," "Use Customary Settings," "Restore Initial Settings," or some other more specific terms describing what will actually happen [41];

(146) if the system supports both novice and expert users, are multiple levels of error message detail available? [38]

(147) if the system supports both novice and expert users, are multiple levels of detail available? [38]

(148) are users the initiators of actions rather than the responders? [38]

(149) do the selected input device(s) match user capabilities? [38]

(150) are important keys (e.g., ENTER, TAB) larger than other keys? [38]

(151) does the system correctly anticipate and prompt for the user's probable next activity? [38]

(12) pleasurable and respectful interaction:

(152) protect users' work [41], also as "For data entry screens with many fields or in which source documents may be incomplete, can users save a partially filled screen?" [38]

(153) do the selected input device(s) match environmental constraints? [38]

(154) are typing requirements minimal for question and answer interfaces? [38]

(155) does the system complete unambiguous partial input on a data entry field? [38]

(13) privacy:

(156) are protected areas completely inaccessible? [38]

(157) can protected or confidential areas be accessed with certain passwords [38]

(158) is there information about how personal data is protected and about contents copyright? [38]

3.4. Compilation of Mobile-Specific Subheuristics. The fourth step in this work is to enrich the list with mobile-specific subheuristics. The subheuristic list obtained in the previous section does not include many mobile specific questions because, as mentioned before, traditional heuristics does not usually cover these issues. New mobile-specific questions have been added into this list, taken from mobile usability studies and best practices that actually do not provide HE. Our approach allows us to include these new items into their corresponding categories, enriching the heuristic with mobile-specific issues. Some new categories had to be added to the original heuristic framework to include new mobile-specific subheuristics. The final framework is shown in Figure 5.

As we mentioned earlier, not all mobile devices have been considered; we discarded featured phones because they are rarely used for tasks other than phone calls and short message services (SMS) and they are gradually being abandoned apart from specific groups of users such as elderly or cognitively impaired people. We also discarded smartphones (phones with midsized screens and full A–Z keypads) because the interactivity with these devices is dramatically different from that of touch phones and they are commonly constrained to enterprise use. This study is centred in touch phones and tablets which are very popular nowadays and similar from a usability point of view.

This fourth step adds 72 new subheuristics to the compilation:

(1) visibility of system status:

System status feedback:

(1) All the items on a list should go on the same page: if the items are text-only and if they are

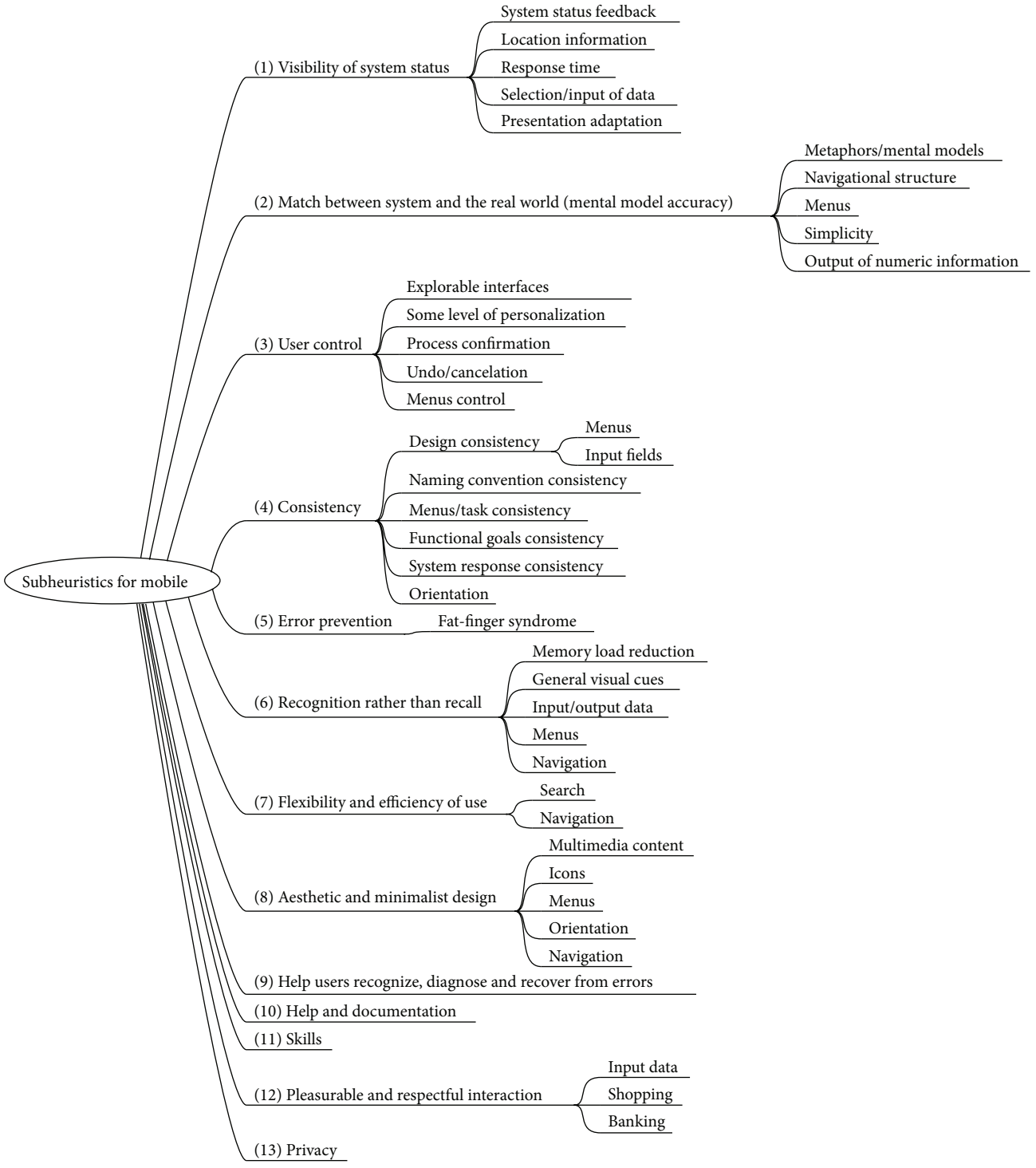


Figure 5: Second framework for classification of detected subheuristics.

sorted in an order that matches the needs of the task [24];

(2) if a list of items can be sorted according to different criteria, provide the option to sort that list according to all those criteria [24];

(3) if a list contains items that belong to different categories, provide filters for users to narrow down the number of elements that they need to inspect [24];

(4) if the list contains only one item, take the user directly to that item [24];

(5) if the list contains items that download slowly (e.g., images), split the list into multiple pages and show just one page at a time [24];

(6) if an article spans several pages, use pagination at the bottom. Have a link to each individual page, rather than just to the previous and the next ones [24];

location information:

(7) whenever you have physical location information on your website, link it to a map and include a way of getting directions [24];

response time:

(8) splash screens too long [14];
(9) download time [14]: "Progress bar is preferable" and "Alternative entertainment if download time is greater than 20 seconds";

selection/input of data:

(10) low discoverability (active areas that do not look touchable): users do not know that something is touchable unless it looks as if it is [14];
(11) swiping [14]: swiping is still less discoverable than most other ways of manipulating mobile content, so we recommended including a visible cue when people can swipe. And swipe ambiguity should be avoided: the same swipe gesture should not be used to mean different things on different areas of the same screen:
(12) expandable menus should be used sparingly. Menu labels should clearly indicate that they expand to a set of options [14];

presentation adaptation:

(13) detect if users are coming to your site on a mobile phone and direct them to your mobile site [24];
(14) include a link to your mobile site on your full site. It can direct mobile users who were not redirected to your mobile site [24];
(15) include a link to the full site on the mobile page [24];

(2) match between system and the real world:

navigational structure:

(16) too much navigation (TMN) [14];

(3) user control and freedom:

explorable interfaces:

(17) accidental activation (lack of back button) [24];
(18) include navigation on the homepage of your mobile website [14];

(4) consistency and standards:

orientation:

(19) about constraining orientation: users tend to switch orientation when an impasse occurs and, if the application does not support them, their flow is going to be disrupted, and they are going to wonder why it is not working [14];
(20) navigation (horizontal and vertical) must be consistent across orientations. Some applications use a different navigation direction in the two orientations; for instance, they use horizontal navigation in landscape and use vertical navigation in portrait [14];
(21) inconsistent content across orientations [14]: "Same content," "Keep location," and "If a feature is only available in one orientation, inform users";

(5) error prevention

(22) accidental activation (lack of back button) [14];

fat-finger syndrome:

(23) touchable areas are too small [14]. Research has shown that the best target size for widgets is 1 cm $\times$ 1 cm for touch devices [14];
(24) crowding targets: another fat-finger issue that we encountered frequently is placing targets too close to each other. When targets are placed too close to each other, users can easily hit the wrong one [14];
(25) padding: although the visible part of the target may be small, there is some invisible target space that if a user hits that space, their tap will still count [14];
(26) when several items are listed in columns, one on top of another (see the time example below), users expect to be able to hit anywhere in the row to select the target corresponding to that row. Whenever a design does not fulfil that expectation, it is disconcerting for users [14];
(27) do not make users download software that is inappropriate for their phone [24];
(28) JavaScript and Flash do not work on many phones; do not use them [24];

(6) recognition rather than recall:

Memory load reduction:

(29) the task flow should start with actions that are essential to the main task. Users should be able to start the task as soon as possible [14];
(30) the controls that are related to a task should be grouped together and reflect the sequence of actions in the task [14];

navigation:

(31) use breadcrumbs on sites with a deep navigation structure (many navigation branches). Do not use breadcrumbs on sites with shallow navigation structures [24];

(7) Flexibility and efficiency of use:

search:

(32) a search box and navigation should be present on the homepage if your website is designed for smartphones and touch phones [24];
(33) the length of the search box should be at least the size of the average search string. We recommend going for the largest possible size that will fit on the screen [24];
(34) preserve search strings between searches. Use autocompletion and suggestions [24];
(35) do not use several search boxes with different functionalities on the same page [24];
(36) if the search returns zero results, offer some alternative searches or a link to the search results on the full page [24];

navigation:

(37) use links with good information scent (i.e., links which clearly indicate where they take the users) on your mobile pages [24];
(38) use links to related content to help the user navigate more quickly between similar topics [24];

(8) aesthetic and minimalist design:

(39) recognizable application icons to be found in the crowded list of applications [14];

multimedia content:

(40) getting rid of Flash content [14];
(41) carousels [24]: avoid using animated carousels, but if they must be used, users should be able to control them;
(42) do not use image sizes that are bigger than the screen. The entire image should be viewable with no scrolling [24];
(43) for cases where customers are likely to need access to a higher resolution picture, initially display a screen-size picture and add a separate link to a higher resolution variant [24];
(44) when you use thumbnails, make sure the user can distinguish what the picture is about [24];
(45) use captions for images that are part of an article if their meaning is not clear from the context of the article [24];
(46) do not use moving animation [24];
(47) if you have videos on your site, offer a textual description of what the video is about. [24];
(48) clicking on the thumbnail and clicking on the video title should both play the video [24];
(49) indicate video length [24];
(50) specify if the video cannot be played on the user's device [24];
(51) use the whole screen surface to place information efficiently [14]: "Popovers for displaying information restricts size of frame where information will be shown" and "Small modal views present the same size constraints";

orientation:

(52) desktop websites have a strong guideline to avoid horizontal scrolling. But for touch screens, horizontal swipes are often fine [19];

navigation:

(53) do not replicate a large number of persistent navigation options across all pages of a mobile site [24];

(9) Help users recognize, diagnose, and recover from errors:

(54) To signal an input error in a form, mark the textbox that needs to be changed [24];

(10) help and documentation:

(55) focus on one single feature at a time. Present only those instructions that are necessary for the user to get started [14];

(11) skills:
(12) pleasurable and respectful interaction:

input data:

(56) users dislike typing. Compute information for the users. For instance, ask only for the zip code and calculate state and town; possibly offer a list of towns if there are more under the same zip code [14];
(57) be tolerant of typos and offer corrections. Do not make users type in complete information. For example, accept "123 Main" instead of "123 Main St." [14];
(58) save history and allow users to select previously typed information [14];
(59) use defaults that make sense to the user [14];
(60) If the application does not store any information that is sensitive (e.g., credit card), then the user should definitely be kept logged in (log out clearly presented) [14];

(61) minimize the number of submissions (and clicks) that the user needs to go through in order to input information on your site [24];

(62) When logging in must be done, use graphical passwords at least some of the time, to get around typing [24];

(63) Do not ask people to register on a mobile phone; skipping registration should be the default option [24];

(64) When logging in must be done, have an option that allows the user to see the password clearly [24];

shopping:

(65) when you present a list of products, use image thumbnails that are big enough for the user to get some information out of them [24];

(66) on a product page, use an image size that fits the screen. Add a Link to a higher resolution image when the product requires closer inspection [24];

(67) offer the option to email a product to a friend [24];

(68) offer the option to save the product in a wish list [24];

(69) on an e-commerce site, include salient links on the homepage to the following information: locations and opening hours (if applicable), shipping cost, phone number, order status, and occasion-based promotions or products [24];

banking and transactions:

(70) whenever users conduct transactions on the phone, allow them to save confirmation numbers for that transaction by emailing themselves. If the phone has an embedded screen-capture feature, show them how to take a picture of their screen [24];

(13) privacy:

(71) for multiuser devices, avoid being permanently signed in on an application [14];

(72) If the application does store credit card information, it should allow users to decide if they want to remain logged in [24]. Ideally, when the user opts to be kept logged in, he/she should get a message informing of the possible risks

3.5. Final New Mobile-Specific Heuristics. The final compilation of heuristics and subheuristics, which is shown in Appendix A (Supplementary Material available online at http://dx.doi.org/10.1155/2014/434326), gives a total of 13 heuristics and 230 subheuristics (158 + 72). In this final compilation, we have omitted intermediate classifications introduced during the discussion. Also, semantically related items have been merged into a single item following the most common presentation of heuristics guidelines in literature. Wording has been corrected to offer a homogeneous collection of heuristics questions.

This final mobile heuristics can be used as a tool to evaluate usability of mobile interfaces. In its current version, possible answers for the proposed questions are "yes/no/NA." The number of "yes" answers provides a measure of the usability of the interface. Other approaches in the literature include more elaborates ratings that have to be agreed between evaluators [45].

3.6. Empirical Test of the New Mobile-Specific Heuristics. The goal of our test was to perform an evaluation of the usefulness of the proposed heuristics as a tool for designers and software engineers with no specific knowledge and experience of usability.

The use case design was as follows: two software engineers without any specific knowledge about usability were asked to design an interface for a tablet application having a functional description in a low-fidelity prototype designed for a desktop version of the application. Over their proposed interface design they used our heuristics as an evaluation and reflexion tool. In view of the results of the evaluation, they were asked to develop a new prototype. Finally, both interfaces were tested with a small group of users to compare their usability.

This empirical test of usefulness of the proposed usability list was divided into the following phases:

(1) prototype 1: developing an interface prototype oriented to tablet access from a given PC-desktop low-fidelity functional design (prototype 1, P1);

(2) HE of P1 using the proposed heuristics as the basis for an oriented discussion between designers;

(3) prototype 2: evolution of P1 fixing usability gaps detected in phase 2 (prototype 2, P2);

(4) Empirical comparison of prototypes: users' testing of P1 and P2.

3.6.1. Prototype 1 Developing. The functional description of the desktop version used to build the prototypes evaluated in this testing was provided by Project PROCUR@ [46], an e-care and e-rehabilitation platform focused on neurodegenerative diseases patients, their carers, and health professionals. The project is based on the deployment of three social spaces for research and innovation (SSRI) [47] in the three validation scenarios: Parkinson's disease SSRI, acquired brain damage (ABD) SSRI, and Alzheimer's disease SSRI. The functional description corresponds to this latter SSRI and provides five low-fidelity interface descriptions from the point of view of five profiles: patients, relatives, doctors, caregivers, and sanitary personnel, respectively.

The subjects of these experiments were two software engineering students preparing their end of degree project. They had never been trained in usability but had knowledge about software life-cycles and design techniques. P1 was the result of a first tablet-interface adaptation without usability training. The tablet format was imposed because a bigger

screen size is specially convenient for the target users (i.e., elderly people with low vision capability and motor control).

This first adaptation included two main groups of changes: functional refinement and new interface adaptation. Functional refinement required changes that were not particularly relevant to this work. However, adaptations to the new interface involved decisions adopted by designers without knowledge of usability, guided only by their common sense. At a later stage, some of these decisions were confronted with the HE new tool and not all of them were maintained. These decisions are described in Figure 6.

Figure 7 shows an example of the interface change.

3.6.2. Prototype 1: Heuristic Evaluation. Once Prototype 1 was designed, the next step was to evaluate its usability. The objective was not the evaluation itself but how the designers reflected on its usability.

When performing a HE using such tool, one has to make certain decisions about the scoring of each subheuristics. In this case, the experts were asked to use a ponderation which would allow the prioritization of heuristic item relevance for the specific evaluated interface. Experts are marked with values from 1 to 4: 1 for accomplished heuristic items, 2 for those corresponding to usability gaps, 3 for heuristic items which were not evaluable in the actual software life-cycle phase, and 4 for questions not applicable to the interface.

Applying a Delphi-based [48] approximation, both experts were asked to independently evaluate the interface using the list. Afterwards, the results of the evaluations were confronted and the experts had to agree in the case of items with different scorings.

In the independent evaluation, the level of coincidence of the experts was moderate and in the final HE scoring, where both experts agreed, the results were as follows: 68 items scored 1, 33 items scored 2, 41 scored 3, and 98 scored 4. This final result established a huge number of items as "not applicable." This may have been because the heuristics was intended to be as general as possible, not focusing on any specific kind of application, and it therefore included an exhaustive list of checks.

The most important result from this evaluation was that experts were forced to reflect on each item in the heuristic guideline. For each not accomplished question, they learnt which usability gaps had to be avoided in the interface design. This learning provided a wider knowledge background when it came to designing next prototype.

3.6.3. Prototype 2 Building. Prototype 2 was not only a series of modifications to Prototype 1 but also a complete revision of the whole interface concept. This global reflexion was guided by the expert discussion from the previous section.

The most specific changes which fix detected usability gaps are shown in Figure 8 but, as mentioned, the overall appearance and design have changed dramatically (Figure 9).

3.7. Empirical Comparison of Prototypes. The empirical comparison of the two prototypes was intended to evaluate whether P2 designed using the proposed HE tool was better in any way than P1.

This empirical study involved users so the experiment had to be designed carefully to obtain valid results. The approach included a test design, a pilot phase to check the test design, the execution of the test itself, and a phase to analyze the collected data.

Several decisions were taken in the design phase.

(1) Wizard of Oz [49] (WO) was chosen as the evaluation technique because the prototypes are developed on paper and are well suited to presentation through human intervention to the users.

(2) To develop WO technique, users were asked to perform a task-guided interaction. The experts selected three functional tasks that users had to carry out interacting with the interface. The tasks were representative enough to be useful in this test. They were briefly described to the users so that they were able to accomplish them by exploring the interface without step-by-step guides.

(3) Ten users were selected with the characteristics shown in Table 1.

(4) The experimentation adopted an inner group [50] design: half of the users interacted with P1 first and the other half with P2 first. This was to avoid as much as possible correlations due to learning of the interfaces.

(5) Lastly, users were asked to give feedback about their overall feelings about each interface to provide us with some conclusions related to user experience beyond usability.

The pilot phase consisted of a simulation of the final experiment using two dummy users. This phase was very useful for consolidating of task description and helping to improve Wizard's skills managing prototypes and for the whole experiment.

Final test execution detected 6 serious usability gaps in P1 and 3 serious gaps in P2 (which are also included in the first 6) as can be seen in Table 2. When asked about general satisfaction, 100% of users stated they were more satisfied with P2 prototype interaction.

4. Conclusions and Future Work

In this paper, we have presented a compilation of heuristic evaluation checklists readapted to mobile interfaces. We started our work by reusing heuristics from desktop heuristics evaluation checklists, which is allowed because "heuristic checklists change very slowly because they derive from human behaviour, not technology" [14]. In fact, in the final proposal of this work, the amount of reused heuristics from the literature is 69% of the total proposed subheuristics. The rest are best-practices and recommendations for mobile interfaces not initially conceived as part of a usability tool.

In the final collection of 13 heuristics, the most influent author is Nielsen [9, 37–44]. While it is not a long list of heuristics, it is exhaustive enough to be a useful categorization for further research. However, in our work, Nielsen's

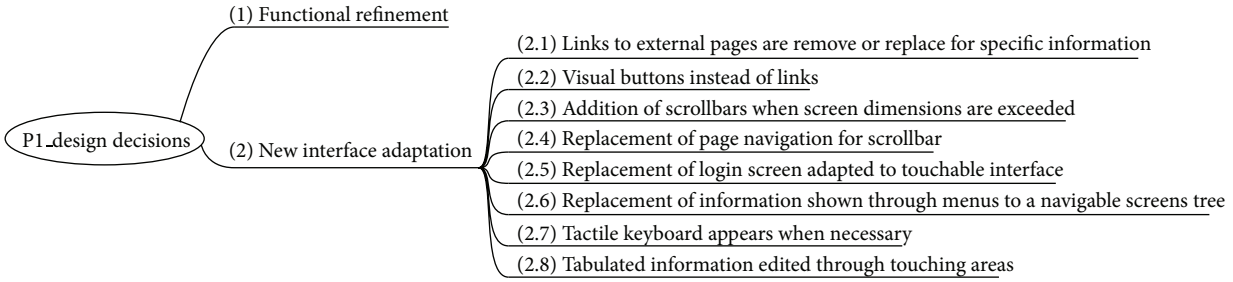


Figure 6: Design decisions in prototype 1.

Figure 7: Prototype 1 from desktop version description.

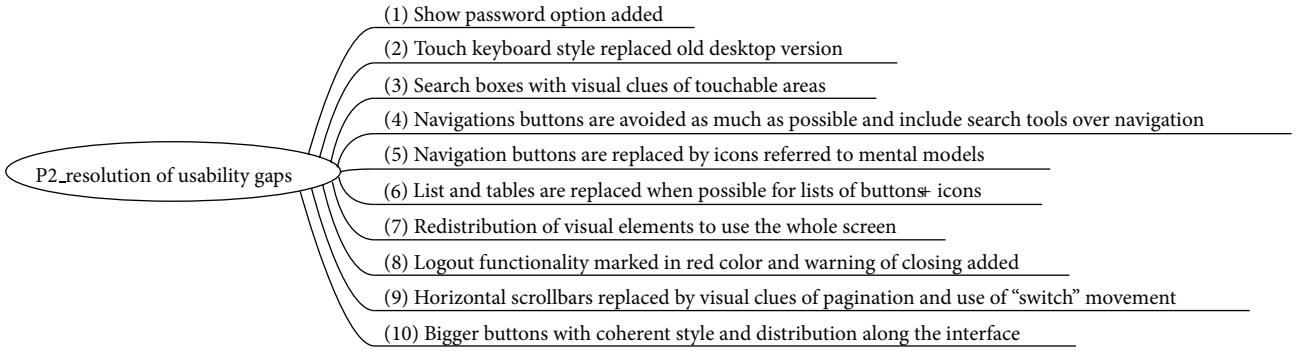


Figure 8: Prototype 2 main changes.

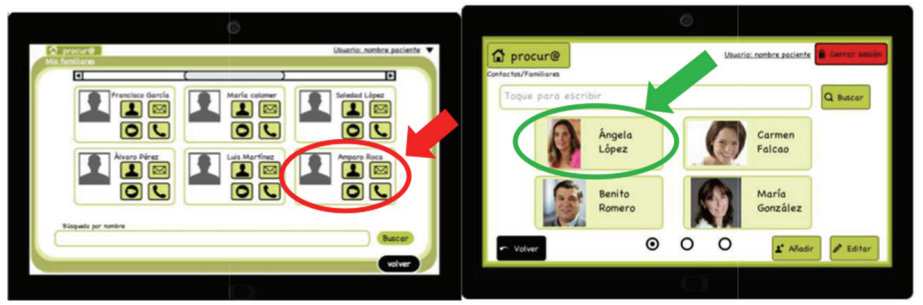


Figure 9: Prototype 2 global concept changes from the first version.

Table 1: Users of the experiment.

	Gender	Age	Kind of mobile devices they are used to	Adoption of technology
USER 1	M	40–50	Touch phone	Basic
USER 2	F	40–50	Smartphone	None
USER 3	M	40–50	Smartphone	None
USER 4	F	40–50	Touch phone	Basic
USER 5	F	40–50	Touch phone	Basic
USER 6	F	40–50	Touch phone	Basic
USER 7	M	40–50	Touch phone	Basic
USER 8	F	40–50	Touch phone	Basic
USER 9	M	50–60	None	None
USER 10	F	50–60	Touch phone	Basic

Table 2: Results of empirical user-based evaluation of prototypes.

	Prototype 1. Usability gaps	Prototype 2. Usability gaps	Description
1	Authentication method inappropriate for the targeted users	Authentication method inappropriate for the targeted users	The boxes "user" and "password" should appear independently
2	Information screen confusing	Information screen confusing	It was maintained because the functional description included it
3	Chatting function not localizable	Returning to main menu not localizable	Even after changing the graphical clue
4	Personal profile function not localizable		
5	Returning to main menu not localizable		
6	Close session function not localizable		

heuristics has been rearranged taking into account other proposals in the literature which emphasize concepts such as skills adaptation and pleasurable and respectful interaction with the user and privacy, elevating them to the category of heuristic item.

The added mobile-specific subheuristics in this proposal focus specifically on overcoming specific constraints on mobile such as limitations in input/output, limited processing capabilities, and power. Additionally, it focuses on favouring usual tasks in mobile and issues related to the adoption of this kind of devices (privacy, acceptance, comfort, personalization…).

The main original contributions of our work include (a) rearrangement of existing desktop heuristics into a new compilation, including detailed subheuristics, adapted to the new mobile paradigm; (b) enriching the list with mobile-specific subheuristics, mainly taken from mobile usability studies and best-practices proposed in the literature; (c) homogenization of the redaction and format of subheuristics in order to make it a useful and comprehensive tool for nonexperts; and (d) user-evaluation of the usefulness of the tool as an aid in designing for mobile.

Future work includes mobility and varying context and multidevice access, constraints that are not considered with enough detail in this work. Indeed, these two questions constitute specific areas of study. The typical mobility and varying context of this kind of devices highlight the limitations of laboratory testing: to fully test mobile interfaces, some field-testing is required. Multidevice access questions deal with Responsive Design [36], a discipline that manages access to a given source of information from different devices in a coherent and comprehensive manner.

Regarding rating, in this study, no weighting for categories was established. We mentioned the nonnegligible amount of items scored as nonapplicable in our experiment. Weighting specific categories or subsets of subheuristics according to the kind of application being evaluated represents a highly interesting area for future work and one which is closely related to certain advances in the work of Torrente [7].

The heuristic checklist we have proposed needs to be thoroughly validated in future research in relation to different aspects. The preliminary test and results obtained in this work appear to indicate that the proposed HE guideline is a useful tool for engineers, designers, and technicians with no specific knowledge in usability. A first hypothesis to explain this result is that more specific heuristic guidelines named subheuristics in this work are easier to manage for nonexpert evaluators.

The specificity of the items collected in the tool means that it can be used as a reference guide to help conceive more usable interfaces and not just as a reactive evaluation tool for existing prototypes. Future work should look into this to confirm this partial result.

Furthermore, other aspects related to the suitability of this guideline need to be validated. For instance, an experts-guided review could evaluate the completion, coherence, and adequacy of the heuristic checklist. This review could be carried out through questionnaires, experts panels, or some kind of Delphi-based surveys [48]. Another highly interesting question is the empirical comparison of general heuristics and this mobile-specific heuristics when analysing mobile interfaces.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This work has been supported by project PROCUR@-IPT-2011-1038-900000, funded by the program INNPACTO of MINECO and FEDER funds and by the Telefonica Chair "Intelligence in Networks" of the Universidad de Sevilla, Spain. The authors would like to thank the members of project PROCUR@, engineers Anabel Hernández Luis and Lidia Romero Martínez who participated in the empirical evaluation and Dr. Víctor Díaz Madrigal and Dr. José Mariano González Romano for their support in the initial phases of this work.

References

[1] International Organization for Standardization (ISO), "ISO 9241-11:1998 Ergonomic requirements for office work with visual display terminals (VDTs)—part 11: guidance on usability," http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=16883.

[2] R. Inostroza, C. Rusu, S. Roncagliolo, C. Jiménez, and V. Rusu, "Usability heuristics for touchscreen-based mobile devices," in *Proceedings of the 9th International Conference on Information Technology (ITNG '12)*, pp. 662–667, Las Vegas, Nev, USA, April 2012.

[3] J. Wang and S. Senecal, "Measuring perceived website usability," *Journal of Internet Commerce*, vol. 6, no. 4, pp. 97–112, 2007.

[4] J. Karat, "User-centered software evaluation methodologies," in *Handbook of Human-Computer Interaction*, M. Helander, T. K. Landauer, and P. Prabhu, Eds., pp. 689–704, Elsevier, New York, NY, USA, 1997.

[5] D. Zhang, "Overview of usability evaluation methods," http://www.usabilityhome.com.

[6] J. Heo, D. Ham, S. Park, C. Song, and W. C. Yoon, "A framework for evaluating the usability of mobile phones based on multi-level, hierarchical model of usability factors," *Interacting with Computers*, vol. 21, no. 4, pp. 263–275, 2009.

[7] M. C. S. Torrente, *Sirius: Sistema de evaluación de la usabilidad web orientado al usuario y basado en la determinación de tareas críticas [Ph.D. thesis]*, Universidad de Oviedo, Oviedo, España, 2011.

[8] M. Y. Ivory and M. A. Hearst, "The state of the art in automating usability evaluation of user interfaces," *ACM Computing Surveys*, vol. 33, no. 4, pp. 470–516, 2001.

[9] J. Nielsen and R. Molich, "Heuristic evaluation of user interfaces," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '90)*, J. C. Chew and J. Whiteside, Eds., pp. 249–256, ACM, New York, NY, USA, 1990.

[10] J. Nielsen, "Guerrilla HCI: Using Discount Usability Engineering to Penetrate the Intimidation Barrier," Nielsen's Alertbox, 1994, http://www.nngroup.com/articles/guerrilla-hci/.

[11] R. Agarwal and V. Venkatesh, "Assessing a firm's Web presence: a heuristic evaluation procedure for the measurement of usability," *Information Systems Research*, vol. 13, no. 2, pp. 168–186, 2002.

[12] C.-M. Karat, R. Campbell, and T. Fiegel, "Comparison of empirical testing and walkthrough methods in user interface evaluation," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '92)*, pp. 397–404, ACM, New York, NY, USA, May 1992.

[13] J. Nielsen, "Heuristic evaluation," in *Usability Inspection Methods*, J. Nielsen and R. L. Mack, Eds., John Wiley & Sons, New York, NY, USA, 1994.

[14] R. Budiu and N. Nielsen, *Usability of iPad Apps and Websites*, Nielsen Norman Group, 2nd edition, 2011.

[15] C. Rusu, S. Roncagliolo, V. Rusu, and C. Collazos, "A methodology to establish usability heuristics," in *Proceedings of the 4th International Conferences on Advances in Computer-Human Interactions (ACHI '11)*, pp. 59–62, IARIA, 2011.

[16] *Android Design Guidelines*, https://developer.android.com/design/index.html.

[17] iOS Human Interface Guidelines. Designing for iOS7, 2014, https://developer.apple.com/library/ios/documentation/user-experience/conceptual/MobileHIG/index.html.

[18] *Smartphone Adoption and Usage*, http://www.pewinternet.org/2011/07/11/smartphone-adoption-and-usage/.

[19] J. Nielsen, *Mobile Usability Update*, Nielsen's Alertbox, 2011, http://www.nngroup.com/articles/mobile-usability-update/.

[20] M. Dunlop and S. Brewster, "The challenge of mobile devices for human computer interaction," *Personal and Ubiquitous Computing*, vol. 6, no. 4, pp. 235–236, 2002.

[21] D. Zhang and B. Adipat, "Challenges, methodologies, and issues in the usability testing of mobile applications," *International Journal of Human-Computer Interaction*, vol. 18, no. 3, pp. 293–308, 2005.

[22] R. Looije, G. M. te Brake, and M. A. Neerincx, "Usability engineering for mobile maps," in *Proceedings of the 4th International Conference on Mobile Technology, Applications, and Systems and the 1st International Symposium on Computer Human Interaction in Mobile Technology (Mobility '07)*, pp. 532–539, ACM, New York, NY, USA, 2007.

[23] D. Zhang, "Web content adaptation for mobile handheld devices," *Communications of the ACM*, vol. 50, no. 2, pp. 75–79, 2007.

[24] R. Budiu and J. Nielsen, *Usability of Mobile Websites: 85 Design Guidelines for Improving Access to Web-Based Content and Services Through Mobile Devices*, Nielsen Norman Group, 2008.

[25] A. S. Tsiaousis and G. M. Giaglis, "An empirical assessment of environmental factors that influence the usability of a mobile website," in *Proceedings of the 9th International Conference on*

Mobile Business and 9th Global Mobility Roundtable (ICMB-GMR '10), pp. 161–167, June 2010.

[26] H. B. Duh, G. C. B. Tan, and V. H. Chen, "Usability evaluation for mobile device: a comparison of laboratory and field tests," in *Proceedings of the 8th International Conference on Human-Computer Interaction with Mobile Devices and Services (MobileHCI '06)*, pp. 181–186, ACM, Helsinki, Finland, September 2006.

[27] S. Po, S. Howard, F. Vetere, and M. Skov, "Heuristic evaluation and mobile usability: bridging the realism gap," in *Mobile Human-Computer Interaction—MobileHCI 2004*, vol. 3160 of *Lecture Notes in Computer Science*, pp. 49–60, Springer, Berlin, Germany, 2004.

[28] E. Bertini, S. Gabrielli, and S. Kimani, "Appropriating and assessing heuristics for mobile computing," in *Proceedings of the Working Conference on Advanced Visual Interfaces (AVI '06)*, pp. 119–126, ACM, New York, NY, USA, May 2006.

[29] Y. Cui and V. Roto, "How people use the web on mobile devices," in *Proceeding of the17th International Conference on World Wide Web 2008 (WWW '08)*, pp. 905–914, New York, NY, USA, April 2008.

[30] L. D. Catledge and J. E. Pitkow, "Characterizing browsing strategies in the World-Wide web," *Computer Networks and ISDN Systems*, vol. 27, no. 6, pp. 1065–1073, 1995.

[31] C. W. Choo, B. Detlor, and D. Turnbull, "A behavioral model of information seeking on the web," in *Proceedings of the ASIS Annual Meeting Contributed Paper*, 1998.

[32] J. B. Morrison, P. Pirolli, and S. K. Card, "A Taxonomic analysis of what world wide web activities significantly impact people's decisions and actions," in *Proceeedings of the Conference on Human Factors in Computing Systems (CHI EA '01)*, pp. 163–164, ACM, New York, NY, USA, April 2001.

[33] A. J. Sellen, R. Murphy, and K. L. Shaw, "How knowledge workers use the web," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '02)*, pp. 227–234, ACM, New York, NY, USA, 2002.

[34] B. MacKay, C. Watters, and J. Duffy, "Web page transformation when switching devices," in *Mobile Human-Computer Interaction—MobileHCI 2004: Proceedings of 6th International Symposium, MobileHCI, Glasgow, UK, September 13–16, 2004.*, vol. 3160, pp. 228–239, 2004.

[35] M. Kellar, C. Watters, and M. Shepherd, "A goal-based classification of web information tasks," *Proceedings of the American Society for Information Science and Technology*, vol. 43, pp. 1–22, 2006.

[36] E. Marcotte, "A list apart," Responsive Web Design, 2010, http://alistapart.com/article/responsive-web-design/.

[37] B. Shneiderman and C. Plaisant, *Designing the User Interface: Strategies for Effective Human-Computer Interaction*, Addison-Wesley, Reading, Mass, USA, 1987.

[38] D. Pierotti, "Heuristic evaluation—a system checklist," Tech. Rep., Xerox Corporation, Society for Technical Communication, 2005.

[39] L. Constantine, *Devilish Details: Best Practices in Web Design*, Constantine and Lockwood, 2002.

[40] K. Instone, "Usability engineering for the web," W3C Journal, 1997.

[41] B. Tognazzini, "First principles of interaction design," Interaction Design Solutions for the Real World, 2003, http://www.asktog.com/.

[42] Y. Hassan Montero and F. J. Martín Fernández, "Guía de Evaluación Heurística de sitios web," 2003, http://www.nosolousabilidad.com/articulos/heuristica.htm.

[43] Y. Hassan, M. Fernández, J. Francisco, and G. Iazza, *Diseño Web Centrado en el Usuario: Usabilidad y Arquitectura de la Información*, vol. 2, 2004, http://www.upf.edu/hipertextnet/.

[44] L. Olsina, G. Lafuente, and G. Rossi, "Specifying quality characteristics and attributes for websites," in *Web Engineering, Software Engineering and Web Application Development*, S. Murugesan and Y. Deshpande, Eds., pp. 266–278, Springer, London, UK, 2001.

[45] J. Nielsen, "Severity ratings for usability problems," Nielsen's Alertbox, 1995, http://www.nngroup.com/articles/how-to-rate-the-severity-of-usability-problems.

[46] November 2013, http://innovation.logica.com.es/web/procura.

[47] 2013, http://www.researchspaces.eu.

[48] H. A. Linstone and M. Turoff, *Delphi Method: Techniques and Applications*, Addison-Wesley, Reading, Mass, USA, 1975.

[49] N. O. Bernsen, H. Dybkjær, and L. Dybkjær, "Wizard of Oz prototyping: when and how?" Working Papers in Cognitive Science and HCI, 1993.

[50] J. Lazar, J. H. Feng, and H. Hochneiser, *Research Methods in Human-Computer Interaction*, John Wiley & Sons, New York, NY, USA, 2010.

Hindawi Publishing Corporation
The Scientific World Journal
Volume 2014, Article ID 905269, 15 pages
http://dx.doi.org/10.1155/2014/905269

Research Article

Comparative Study of Human Age Estimation with or without Preclassification of Gender and Facial Expression

Dat Tien Nguyen, So Ra Cho, Kwang Yong Shin, Jae Won Bang, and Kang Ryoung Park

Division of Electronics and Electrical Engineering, Dongguk University, Seoul 100-715, Republic of Korea

Correspondence should be addressed to Kang Ryoung Park; parkgr@dgu.edu

Received 16 June 2014; Revised 31 July 2014; Accepted 9 August 2014; Published 9 September 2014

Academic Editor: Young-Sik Jeong

Age estimation has many useful applications, such as age-based face classification, finding lost children, surveillance monitoring, and face recognition invariant to age progression. Among many factors affecting age estimation accuracy, gender and facial expression can have negative effects. In our research, the effects of gender and facial expression on age estimation using support vector regression (SVR) method are investigated. Our research is novel in the following four ways. First, the accuracies of age estimation using a single-level local binary pattern (LBP) and a multilevel LBP (MLBP) are compared, and MLBP shows better performance as an extractor of texture features globally. Second, we compare the accuracies of age estimation using global features extracted by MLBP, local features extracted by Gabor filtering, and the combination of the two methods. Results show that the third approach is the most accurate. Third, the accuracies of age estimation with and without preclassification of facial expression are compared and analyzed. Fourth, those with and without preclassification of gender are compared and analyzed. The experimental results show the effectiveness of gender preclassification in age estimation.

1. Introduction

With the development of smart devices, such as smart phones and smart televisions, natural user interfaces (NUIs) become increasingly attractive. In addition, with the vigorous research on three-dimensional (3D) video processing techniques on 3DTV [1], 3DTV NUIs can be also considered. NUIs offer the advantage of natural interaction with a system using predefined actions and/or physical human characteristics. For example, people have been using hand gestures, speech [2], biosignals [3], and mobile device gestures [4] to interact with machines such as computers and smart devices. In previous research [5], finger-triggered virtual musical instruments have also been proposed based on a finger data-glove system. Along with gesture and speech, the human face, which contains substantial information about a person, has also been widely used for human-machine interaction in many applications including face-based human identification, gender classification, age estimation, facial expression recognition, and race classification. Among these, age estimation using facial images is becoming increasingly important. In previous studies [6–10], age synthesis and estimation have been studied with many real-world applications, such as forensic art, surveillance monitoring, biometrics, cosmetology, and entertainment. In addition, age estimation is used for face recognition invariant to age progression [11, 12]. Because of the changing of facial characteristics, such as facial shape and skin detail, according to age progression, face recognition performance is less effective if it neglects age effects. In this case, age estimation can serve as a complement to the primary biometric feature of face. The age estimation and synthesis have been also used for finding lost children. Using age estimation and synthesis, the updated face appearance of lost children after several years can be predicted. Age estimation can be used to prevent children from accessing adult websites and restricted videos and from buying tobacco from automated vending machines.

Previous age estimation methods based on facial images can be divided into two categories: active appearance model- (AAM-) based and non-AAM-based methods [13]. AAMs are statistical face models and have been used for age estimation; they involve the modeling of face's shape and

appearance [6–8]. With a training data set, models of face shape and appearance are generated based on multiple feature points. Then, age is considered as a function of feature vectors learned from AAMs. Although the AAM-based method can produce high age estimation accuracy, its performance is strongly affected by the problem of fitting multiple feature points. Further, it involves substantial processing time to locate the feature points. Thus, the AAM-based method is difficult to use in real-time systems.

Another approach to age estimation is non-AAM-based methods. Choi et al. used the high frequency components of selected skin regions for age estimation [9]. In this study, the high frequency components are measured by using high-pass filters, such as a Sobel filter, image differences between an original and its smoothed images, an ideal high-pass filter (IHPF), a Gaussian high-pass filter (GHPF), and wavelet transforms. The work in [10] used the local binary pattern (LBP) operator to extract age features from skin regions of a face. Although these approaches can be used for age estimation, they still have limitations. The estimation performance of the work in [9] is strongly dependent on the method of choosing skin regions (where the frequency components are extracted), which are determined by the facial feature points. The performance enhancement of the work in [10] is limited by the use of a single-level LBP operator. To overcome this problem, the work in [13] proposed an age estimation method based on a multilevel LBP (MLBP) and support vector regression (SVR). However, they did not consider the local features for age estimation.

Based on human perception, we can see that there are some differences between men and women in terms of producing facial age features. For example, an adult man can have a beard and rough skin surface, whereas a woman does not have a beard and tends to have smoother skin compared to a man. This suggests that gender can have effects on age estimation. In previous studies, gender is recognized by voice, 3D body shape, or face image. In different ways, they show that gender recognition accuracy is affected by age.

Another factor that can produce negative effects on age estimation accuracy is facial expression. Humans of the same age can show different age features of textures, among different facial expressions. There have been many previous studies of facial expression recognition based on both the shape and/or texture appearance of a face image. In different aspect, they showed the effect of age on facial expression recognition performance.

Most of previous researches did not consider the effect of facial expression and gender on the age estimation system. Considering the limitations of previous research, we propose a new age estimation method. In addition, the effects of gender and facial expression on age estimation are investigated. The accuracies of age estimation using LBP and MLBP are compared, and MLBP shows better performance as the extractor of texture features globally. In addition, we compare the accuracies of age estimation using global features extracted by MLBP, local features extracted by Gabor filtering, and the combination of the two methods. Results showed that the third approach is superior. In addition, the accuracies of age estimation with and without preclassification of facial expression and gender are compared. We implemented an enhanced age estimator compared to that of [13], and the novelties of our research (as shown in Abstract) are different from those in [13].

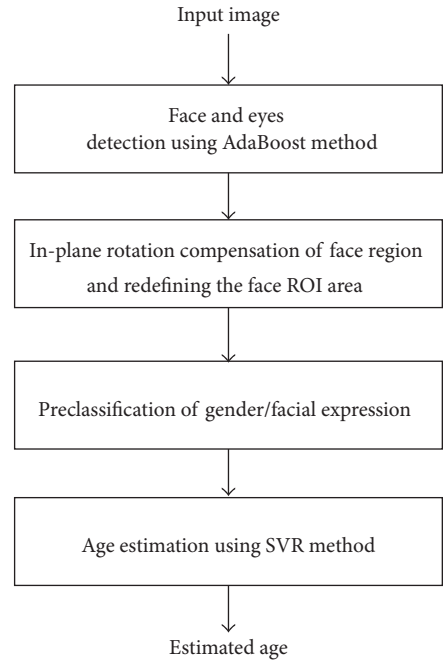


Figure 1: Overview of an age estimation system that considers the effects of gender and facial expression.

The remainder of this paper is organized as follows. In Section 2, an overview of age estimation system is provided. In Section 3, experiments are conducted using the proposed method with PAL aging database to determine the effectiveness of the proposed method and verify the effects of gender and facial expression on age estimation. Based on the experimental results, we discuss the effects of gender and facial expression on age estimation in Section 4. Finally, conclusions are presented in Section 5.

2. Age Estimation System: Overview

2.1. Overview of Proposed Age Estimation System. An overview of our age estimation system, which considers the effects of gender and/or facial expression, is depicted in Figure 1. For the preprocessing step, the face and eye positions are first detected from the input image using adaptive boosting (AdaBoost) method [13, 14]. Then, our system conducts in-plane rotation to align the face based on the detected positions of the two eyes [13].

As shown in Figure 1, to consider the effects of gender or facial expression on age estimation, preclassification of gender/facial expression is conducted before age estimation. The gender or facial expression preclassification procedure allows us to deal with each gender (male and female) and facial expression (neutral, happy, surprised, and the like) separately for age estimation. In our research, the automatic

gender and facial expression recognition algorithm is not implemented but is left for future work. For the initial research, we divide the experimental data manually according to gender and facial expression to measure the effects of gender or facial expression on age estimation. In our experiments, we compared the accuracies of age estimation without preclassification of gender and facial expression, with preclassification of gender, and with preclassification of facial expression. Finally, our system estimates the age of an input face using SVR [13].

2.2. Face Detection and In-Plane Rotation Compensation. Typically, an input facial image contains both the facial and background regions. Therefore, the first step of the age estimation system is to localize the facial region in the input image. There have been many previous studies of face detection [15–17], and we use the AdaBoost method [13, 14]. With AdaBoost, the facial region and the positions of the eyes could be detected efficiently by constructing a strong facial classifier from several weak facial classifiers. In the actual system, there typically exists an in-plane face rotation in the captured image, which degrades the performance of the age estimation system. Previous work [18] proved that age estimation systems could be affected by misalignment of the face region. Thus, we use the detected positions of the eyes to compensate for the in-plane rotation of the face [13]. In detail, within the detected face region, the positions of right and left eyes are located as (R_x, R_y) and (L_x, L_y), respectively. Then, the angle of in-plane rotation is calculated by (1), and the input face region is rotated by the angle θ:

$$\theta = \tan^{-1}\left(\frac{R_y - L_y}{R_x - L_x}\right). \quad (1)$$

Figure 2 shows an example of our in-plane rotation compensation method.

Although AdaBoost has been widely used to detect a face from a face image, it cannot localize the face region correctly, as shown in Figure 3(b). To estimate the face region more accurately, our system performs a redefinition step based on the positions of the eyes, as shown in Figures 3(a) and 3(c).

Suppose that the distance between the eyes is measured as a value of l (pixels) based on the result of detected eye positions and the compensation procedure. Then, our method uses a redefinition method to estimate the face ROI based on the l value as shown in Figure 3(a), where k_1, k_2, and k_3 are the ratio values considering face geometry and defined by experiment [13]. In detail, the ratio values (k_1, k_2, and k_3) were experimentally determined so as not to include background and hair regions for age estimation. The determined values of k_1, k_2, and k_3 are 0.5, 0.75, and 1.5, respectively.

2.3. The Proposed Method for Human Age Estimation

2.3.1. Age Estimation Based on MLBP, Gabor Filter, and SVR. In Figure 4, we show the proposed age estimation method based on the combination of MLBP, Gabor filter, and SVR. There have been many previous studies of local texture analysis [19]. In previous researches, the LBP has been successfully used for facial expression recognition, age estimation, and pattern recognition. However, those studies used only the single level for LBP instead of multilevel for LBP [7, 10]. Therefore, in our proposed method, the MLBP is used to create a stronger descriptor for age estimation.

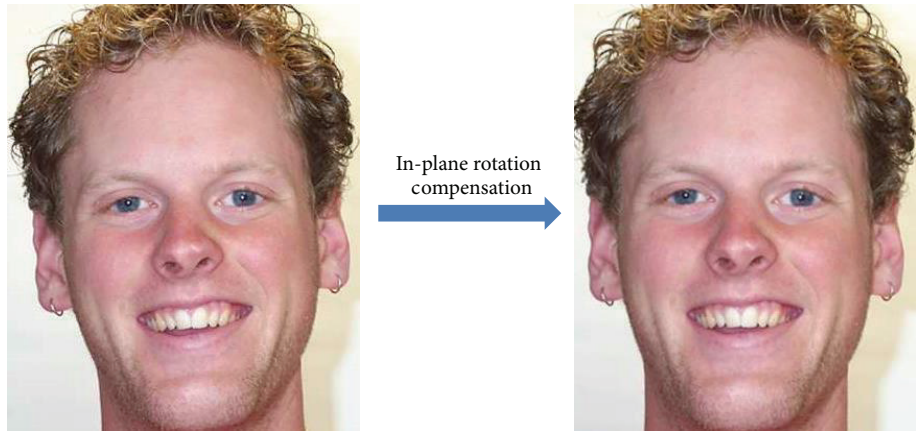
Along with global features extracted using MLBP, the proposed method also extracts local features for age estimation. The wrinkle feature is a very important feature appearing locally on the human face [7, 9]. Therefore, in our method, the wrinkle feature is used as a local feature for enhancing the age estimation result. For this purpose, we use Gabor filtering [7]. Finally, the feature vector formed by combining the global and local features is used for estimating age using SVR.

2.3.2. Global Feature Extraction Using MLBP. LBP is a powerful method for describing image texture by thresholding the surrounding pixels with a center pixel [7, 10, 13]. The LBP method has been widely used in many researches such as age estimation [7, 10, 13], gender recognition, finger-vein recognition, facial expression recognition, and face recognition. The main advantage of LBP method is that it offers the texture descriptor robustness to the variations of illumination and rotation. Besides, the fast processing can be done by LBP method. The LBP operator is defined as [13]

$$\mathrm{LBP}_{R,P} = \sum_{i=0}^{P-1} s\left(g_i - g_c\right) 2^i, \quad \text{where } s(x) = \begin{cases} 1, & \text{if } x \geq 0 \\ 0, & \text{if } x < 0, \end{cases} \quad (2)$$

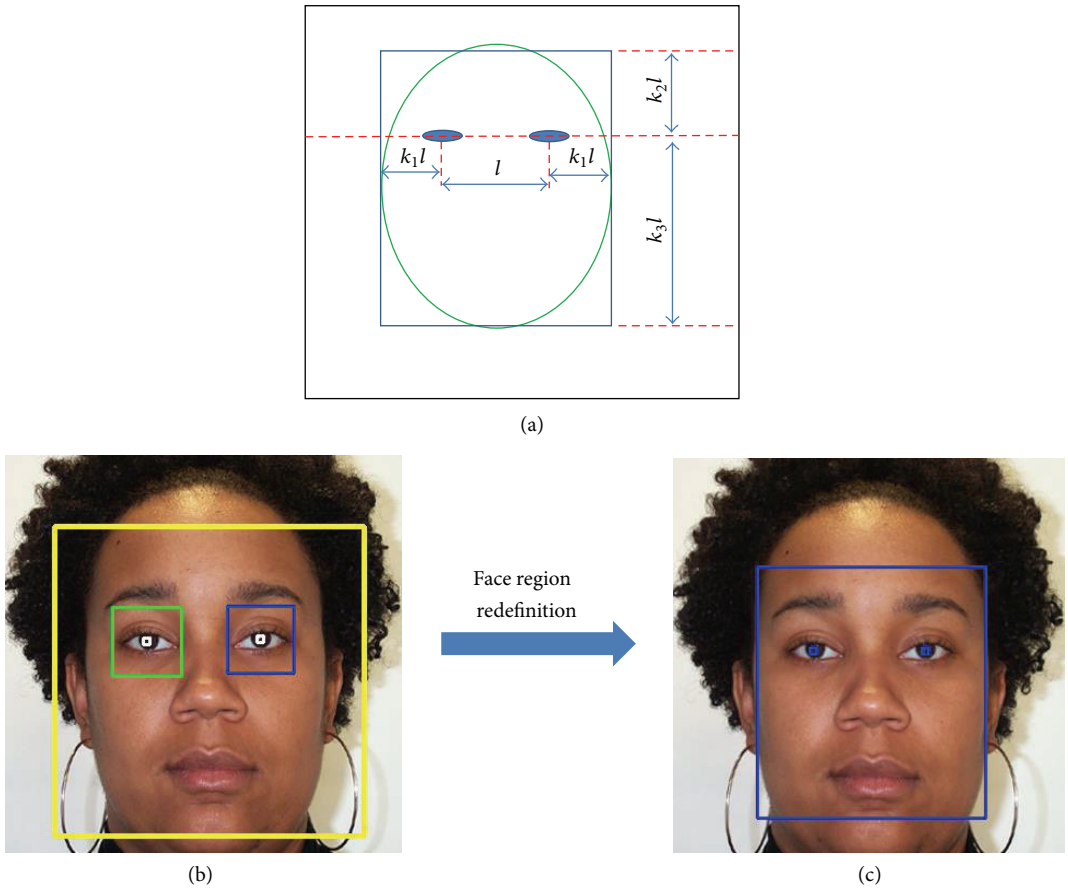
where P is the number of neighboring pixels, R is the radius of the LBP circle (the distance from the center pixel to the neighboring pixels), g_i and g_c are the gray level of neighboring pixels and center pixel, respectively, and $s(x)$ is the threshold function. Varying the values of R and P, we extract image texture features at different scales and resolutions [13]. Originally, the LBP operator makes the texture descriptor by using a 3×3 mask. In this case, the values of R and P are 1 and 8, respectively. According to the human's age, the features for age estimation appear in different size (bigger/smaller spot, wrinkle, etc.). In addition, the resolution and size of facial image also causes the size difference of the features. Consequently, varying the values of R and P makes our method obtain the sufficient information of features at different scales and resolution. Figure 5 shows an example of texture feature of a facial image at different values of R and P. As shown in Figure 5, while the LBP operator with smaller R and P values extracts the fine textures of narrow thickness (Figure 5(b)), that with the bigger values helps our method extract the coarse textures of wide thickness.

The extracted LBP binary codes from (2) are classified into uniform or nonuniform patterns, as shown in Figure 6 [13]. As the human becomes older, several facial texture features such as wrinkle and spot appear [7, 10, 13]. Based on this characteristic, our research uses the LBP operator to describe



(a) (b)

Figure 2: Example of in-plane rotation compensation: (a) an example rotated image and (b) in-plane rotation compensation result of the image in (a).



(a)

(b) (c)

Figure 3: The example of face region redefinition: (a) methodology for face region definition, (b) face region detected using AdaBoost and in-plane rotation compensation, and (c) result of face region of interest (ROI) redefinition.

the image texture and extract the feature for age estimation. A uniform pattern contains at most two bitwise transitions from 0 to 1 (or 1 to 0), as shown in Figure 6(a). Otherwise, a pattern is classified as nonuniform. Uniform patterns are useful for describing image textures, such as spots, corners, and lines, whereas nonuniform patterns contain insufficient information for describing image texture. Figure 7 shows the examples of representing the textures of spot, corner, and edge into the uniform patterns of 0, 3, and 4 of Figure 6(a), respectively.

Then, our system forms the image descriptor by accumulating the LBP code histogram (uniform or nonuniform

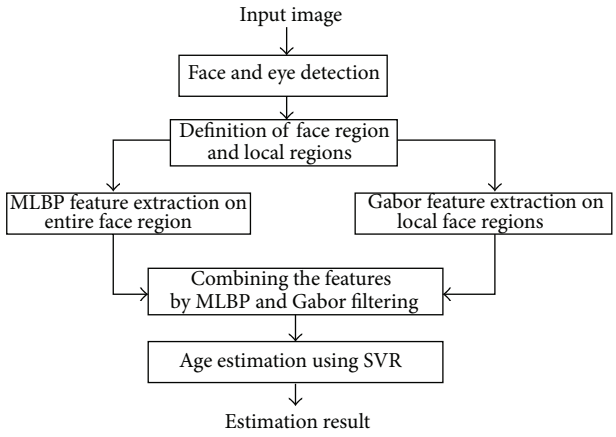


Figure 4: Proposed age estimation method based on MLBP, Gabor, and SVR.

code) over the texture. Figure 6 shows an example of uniform and nonuniform codes as well as the assigned codes in the case that R and P are one and eight, respectively. Uniform codes are assigned decimals from zero to eight, whereas all of the nonuniform codes are assigned the decimal nine. By accumulating the histogram of assigned decimal codes of uniform and nonuniform codes in an image, we form a histogram of texture appearance and use it as the image descriptor in age estimation [7, 10, 13].

The LBP code histogram represents the distribution of image textures in an image. The LBP features of binary code can represent the more local texture compared to the histogram features. However, the LBP binary code can be much affected by image misalignment, and we use the histogram features in our research. To extract the various features of image texture, the proposed system divides the input image into local subblocks and forms the descriptor (feature vector) for each subblock. Consequently, the final descriptor of the image is formed by concatenating the descriptor of each subblock. Figure 8 depicts the method for constructing the LBP descriptor of an image. The feature vectors of each subblock are concatenated together as shown in Figure 8. In our research, the order of concatenating the feature vectors of subblocks is from left to right and up to down. That is, the histogram of the upper-left subblock is included first and that of lower-right one is included last in the final LBP feature vector.

Based on the single-level LBP method of Figure 8, our method constructs the MLBP features by concatenating the several feature vectors of the single-level LBP, as shown in Figure 9. The accuracy of age estimation by LBP is affected by the size of subblock. With the larger subblock, the global features are extracted by LBP whereas with the smaller one, the local features can be obtained. In order to extract the various features globally and locally, we use the MLBP of Figure 9 instead of single-level LBP of Figure 8. The optimal level (with which the smallest error of age estimation is obtained) is determined experimentally to be three for MLBP [13].

2.3.3. Local Feature Extraction by Gabor Filtering. In previous work by Choi et al. [7], the wrinkle feature is extracted locally using Gabor filtering at nine specific face regions. The definition of wrinkle areas is based on the 68 facial feature points defined manually or automatically as an AAM fitting result. Although the definition of wrinkle areas based on 68 facial feature points is very accurate, it has the problem that it takes a long processing time to determine the 68 points using the AAM fitting process. In addition, the performance of AAM is affected by face movement, face area illumination, and background texture. Thus, we roughly define five wrinkle areas based on eyes and facial box position, as shown in Figure 10, rather than nine areas, as in [7]. Compared to the nine areas in [7], the four regions (those between the eye corner and left (or right) face boundary and those between the lip corner and left (or right) face boundary) are not used in our system because accurate face boundary detection is not guaranteed without an AAM.

For each local region, the features are extracted using Gabor filtering. The two-dimensional (2D) Gabor filter in the spatial domain is defined as follows [7, 20]:

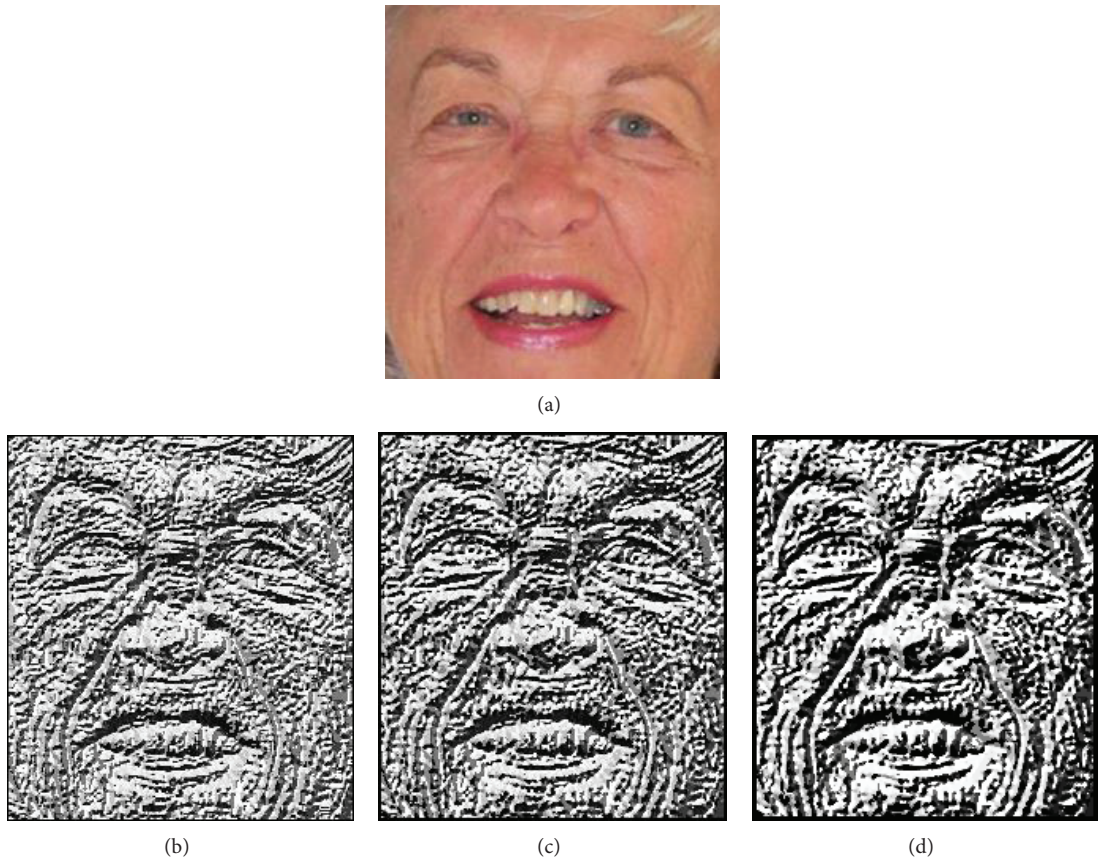
$$g(x, y) = \left(\frac{1}{2\pi\sigma_x\sigma_y}\right)\exp\left[-\frac{1}{2}\left(\frac{x^2}{{\sigma_x}^2} + \frac{y^2}{{\sigma_y}^2}\right) + 2\pi jWx\right], \tag{3}$$

where σ_x and σ_y are the standard deviations of $g(x, y)$ on the x- and y-axes, respectively, and W is the radial frequency of a sinusoid. In our method, only the real part of Gabor filter is used for fast processing. We obtained the filter coefficients experimentally considering the previous research [7]. According to the human age, the appearance (direction and thickness) of wrinkle feature in five selected local areas is different. With a young person, the thickness of wrinkle is narrow whereas it is wider when the person becomes older. Therefore, in order to extract the wrinkle feature of various thickness and directions, we used the Gabor filter with four scales and six directions, as shown in Figure 11. The mean and standard deviation of the Gabor filtering response are used as the wrinkle features. Consequently, the local feature using Gabor filtering is a vector in 240-dimensional space (5 regions × 4 scales × 6 directions × 2 features).

2.3.4. Feature Fusion and Age Estimation Using SVR. With the global features obtained using MLBP and the local features obtained using Gabor filtering, a final feature vector is constructed by concatenating the two normalized features, using z-score normalization [7] as shown in

$$f_i^{\text{norm}} = \frac{f_i - \mu_i}{\sigma_i}, \tag{4}$$

where f_i^{norm} stands for the ith normalized feature vector of the original feature f_i. μ_i and σ_i are the mean and standard deviation of the distribution of f_i, respectively. In our research, the values of i are 1 and 2 in case of MLBP and Gabor filtering features, respectively. After normalization, the



(a) (b) (c) (d)

Figure 5: Example of texture extraction by LBP method at different values of R and P: (a) an example facial image, texture extraction by LBP operator with (b) $R$and P of 1 and 8, respectively, (c) R and P of 2 and 8, respectively, and (d) R and P of 3 and 12, respectively.

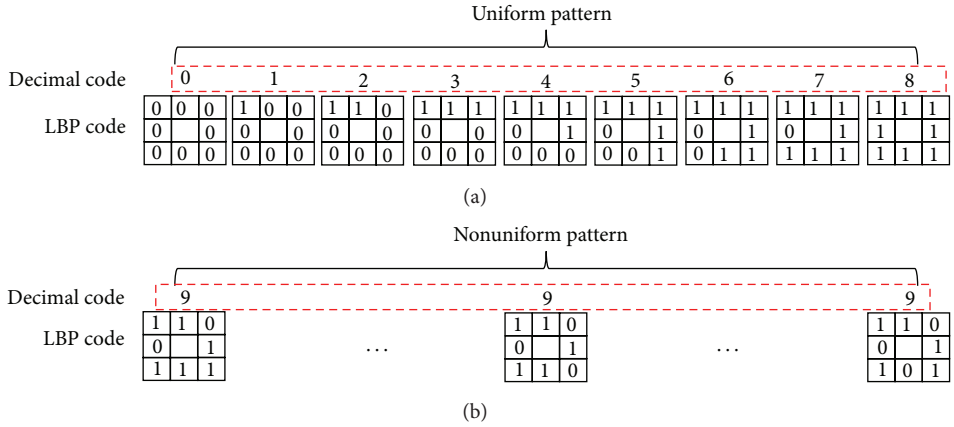


Figure 6: Example of uniform and nonuniform patterns and their assigned decimal codes in the case in which R and P are one and eight, respectively: (a) uniform patterns and (b) nonuniform patterns.

combined feature vector f is easily obtained by concatenating these two normalized vectors together:

$$f = \left[f_1^{\text{norm}}, f_2^{\text{norm}}\right]. \quad (5)$$

This feature vector is inputted to the SVR machine implemented using LibSVM software [21]. The optimal SVR kernel with its parameters is determined experimentally using training data, from which we can obtain the best relation between the extracted feature vector f and the ground-truth age of human in input image.

3. Experimental Result

3.1. Database. For experiments, a lifespan aging database (PAL database) that contains both gender and facial expression was used [22, 23]. The PAL database was obtained

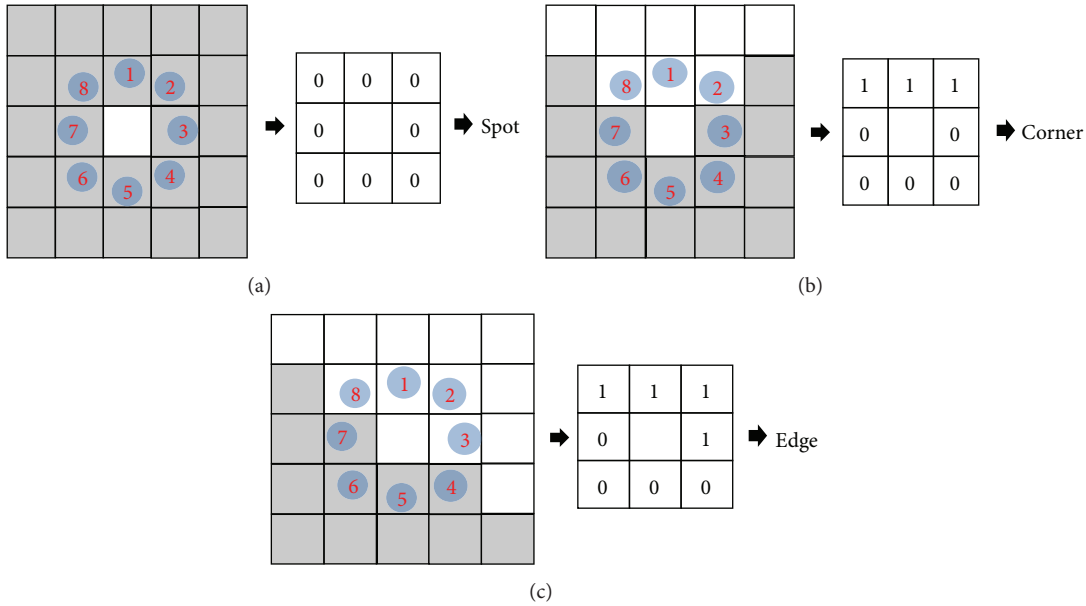


FIGURE 7: Examples of uniform patterns in describing the texture features: (a) spot feature, (b) corner feature, and (c) edge feature.

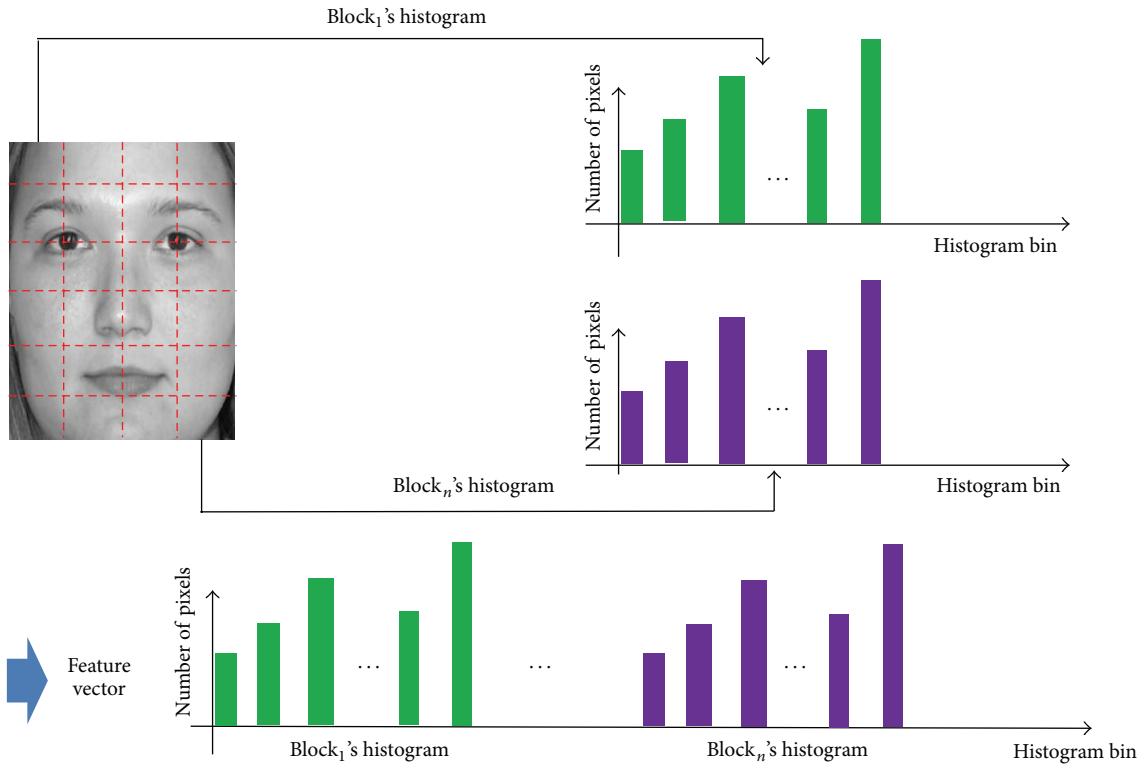


FIGURE 8: Method for constructing the LBP feature vector from multiple subblocks.

from 576 people of 18 to 93 years old, including Caucasians, African-Americans, and others. We used 1,045 of these images, excluding those for which face detection failed. These images include 429 males and 616 females, distributed into eight general facial expressions such as angry, annoyed, disgusted, grumpy, happy, neutral, sad, and surprised. Figure 12 shows some example PAL database images with varied genders and facial expressions.

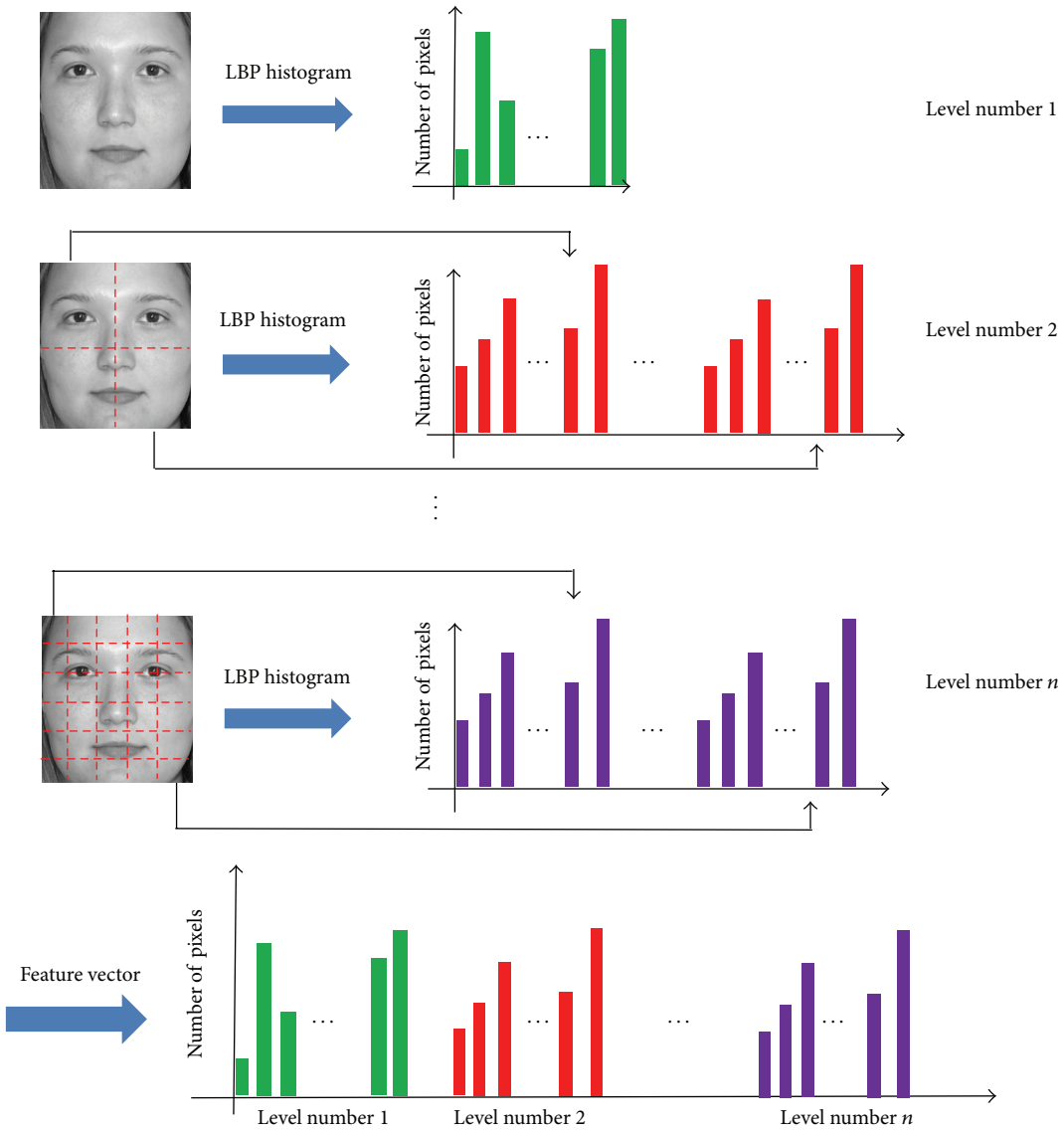


Figure 9: Method for constructing MLBP feature vector from multiple single-level LBP feature vectors.

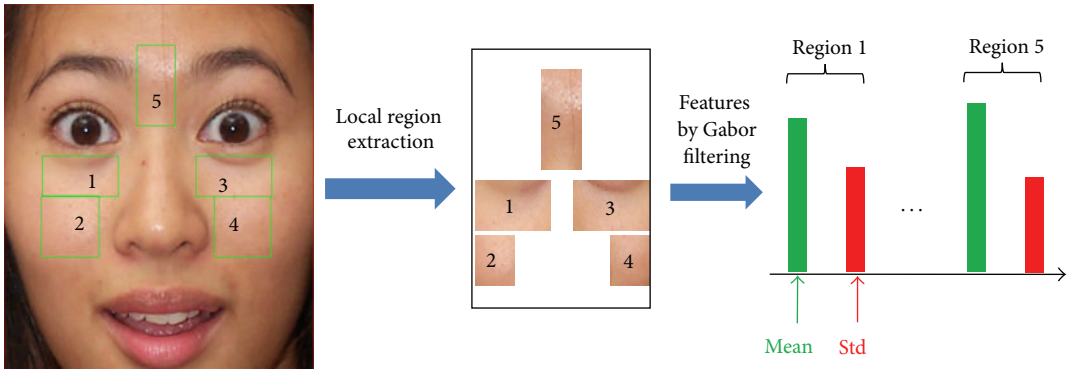


Figure 10: The 5 selected areas for extracting wrinkle features.

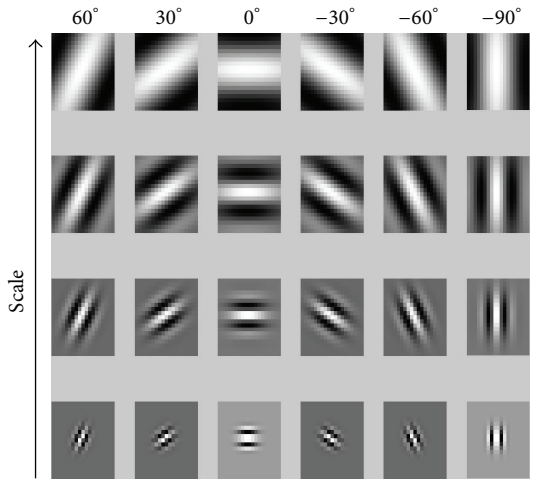


Figure 11: Example of Gabor filters with four scales and six directions.

3.2. Experimental Result

3.2.1. Performance Evaluation of Our Age Estimation System. To measure the accuracy of our age estimation system, we chose the mean absolute error (MAE), widely used in previous research and shown in [7, 9, 13]

$$\text{MAE} = \frac{1}{N}\sum_{k=1}^{N}\left|a_k' - a_k\right|, \tag{6}$$

where N is the total number of images in the testing data set, a_k is the ground-truth age of the kth image, and a_k' is its estimated one. It can be inferred from (6) that the smaller MAE value indicates better age estimation performance.

To measure the performance of our age estimation system, we performed 2-fold cross-validation. In each experiment we randomly divide the entire database into two parts: learning and testing databases. The PAL aging database has been widely used for age estimation in previous researches. However, this database does not provide the identity information. Instead, only the information of race, gender, age, and facial expression is associated with the name of image. Without the identity information, it is very difficult to separate the dataset into two parts containing different individuals. Therefore, we randomly divided the database into learning and testing twice in order to perform the experiments based on 2-fold cross-validation. All the parameters with kernels are trained with the learning database, and the MAE is measured with the testing database. In the second trial, the learning and testing databases are again determined randomly, and the procedure is repeated. From this procedure, two MAEs are obtained, with the average value of the two MAEs calculated as the final MAE.

For the first experiment, we compared the accuracies of our age estimation system based on single-level LBP and MLBP. For the experiments, the LBP accuracies with various R values (in the range of 1 to 5) and P values (8, 12, and 16) for (2) were compared. As shown in Figure 5, the extracted texture features by LBP operator are different according to various values of R and P. In order to obtain the optimal values of R and P (with which, the MAE of age estimation is minimized), we performed the experiments with the various values of R and P. In addition, we compared the accuracies of using a rectangular or square shape for each subblock of Figures 8 and 9, according to the various numbers of subblocks. As shown in Table 1, the MLBP-based method outperforms the LBP-based method, and we used the MLBP-based method for our age estimation system.

Table 1: Comparisons of MAEs of MLBP method to those of single-level LBP method (unit: years old).

Database and shape of subblock	Single-level LBP	MLBP
Testing database 1		
Rectangular block division	6.981	6.351
Squared block division	7.125	6.536
Testing database 2		
Rectangular block division	7.531	7.322
Squared block division	7.345	6.625
Average MAE		
Rectangular block division	7.256	6.837
Squared block division [13]	7.235	**6.581**

Table 2: The comparisons of MAEs using only MLBP, only Gabor filtering, and the proposed method combining the two methods (MLBP and Gabor filtering) (unit: years old).

Whole PAL database	Using only MLBP	Using only Gabor filtering	Proposed method
Testing database 1			
Rectangular block division	6.351	11.774	6.247
Squared block division	6.536		6.513
Testing database 2			
Rectangular block division	7.322	12.042	7.176
Squared block division	6.625		6.542
Average MAE			
Rectangular block division	6.837	11.908	6.712
Squared block division	6.581		**6.528**
Previous research [9]		8.44	
Previous research [13]		6.581	

In the next experiment, we compared the MAEs using only MLBP, only Gabor filtering, and the proposed method combining the two. In addition, we compared the MAE using the proposed method to that obtained in previous researches [9, 13]. In Table 2, the method using only MLBP with square block division is from [13]. The extracted LBP feature vectors can be different according to the shapes of each subblock, which can affect the accuracy of age estimation. So, we measured the MAEs according to the shapes of each subblock. The rectangular block means that the height and width of the subblock are different whereas the squared one indicates that the height and width of the subblock are the same. As shown in Table 2, the accuracy using the proposed

Figure 12: Example images in PAL database with varied genders and facial expressions: (a) female-neutral expression, (b) female-happy expression, (c) male-sad expression, (d) male-surprised expression, and (e) female-annoyed expression.

method is higher than that in the other cases and in previous research.

In our method, Gabor filtering is used to extract the local wrinkle feature. As the human becomes older, the wrinkle feature appears, and the strength (depth) of wrinkle feature can be used for the discrimination of different ages [7]. Although the accuracy of age estimation by the Gabor filtering is worse than that by MLBP method, the MAE by combining these two methods is lower than that by using only MLBP method or Gabor filtering as shown in Table 2.

We measured the difference between the MAEs by our method and that by previous one [13] of Table 2 based on effect size in descriptive statistics. In statistics, the power of a measured phenomenon can be evaluated based on effect size, and as a descriptive statistic, the effect size has been widely used [24, 25]. Based on the previous research [26], the values of 0.2, 0.5, and 0.8 can be defined as small, medium, and large for Cohen's d value, respectively.

In Table 3, Cohen's d show the difference between two means divided by a standard deviation of the data. The effect size of 0.2 to 0.3, around 0.5, and 0.8 to infinity (based on Cohen's d value) can be "small," "medium," and "large" effect, respectively [24, 25]. Since all the Cohen's d values of Table 3 are less than 0.2, we can find that there exists a difference between the MAEs by our method and previous one [13] as a small effect size.

In Figure 13, we show some examples of estimated ages using the proposed method and ground-truth.

In addition, we include the examples (where the errors of age estimation are large) as shown in Figure 14. The errors of Figures 14(a)–14(d) are caused by the facial expression. Those of Figures 14(e) and 14(f) are due to the facial makeup and mustache, respectively.

Table 3: The measured Cohen's d between the MAEs by our method and previous one [13].

	Testing database 1	Testing database 2	Average
Cohen's d	0.004183	0.014779	0.009481
Effect size	Small	Small	Small

3.2.2. Study of Effects of Facial Expression and Gender on Age Estimation. In the next experiments, we study the effects of facial expression and gender on the estimation system using our proposed age estimation method of Section 2.3 and Table 2. Although the number of images used in the above experiments is 1,045, as explained in Section 3.1, the images of angry, grumpy, and disgusted are not included for the next experiment because these images are too few. Consequently, the images of five expressions are used for the experiments, as shown in Table 4. As explained in Section 2.1, the facial expression databases are separated manually from the whole PAL database based on facial expression markers attached to an image's name.

As shown in Table 5, the average MAE (7.226) of age estimation with facial expression preclassification is greater than that (6.528) without preclassification. Only the neutral facial expression database produces a better estimation result compared to that of the whole PAL database. There are several

Figure 13: Examples of estimated and ground-truth ages: (a) ground-truth age of 19, estimated age of 19, (b) ground-truth age of 32, estimated age of 31, (c) ground-truth age of 41, estimated age of 40, (d) ground-truth age of 65, estimated age of 65, (e) ground-truth age of 72, estimated age of 72, and (f) ground-truth age of 81, estimated age of 80.

Table 4: The number of images used for the comparative experiments with/without the preclassification of facial expression.

Database	Number of learning images	Number of testing images
Facial expression database		
Annoyed	21	19
Happy	130	128
Neutral	291	289
Sad	32	32
Surprised	39	38
Total	**513**	**506**
Whole PAL database	523	522

reasons why the MAEs with facial expression preclassification are greater than those without preclassification, and these are explained in Section 4.

Next, we study the effects of gender (male and female) on the estimation system. All 1,045 images are used for the experiments, as shown in Table 6. As explained in Section 2.1, the gender databases are separated manually from the whole PAL database based on gender markers attached to an image's name. In Tables 5 and 6, the average MAEs were calculated by weighting each MAE with number of samples of each class.

As shown in Table 7, the average MAE (6.323) of age estimation with gender preclassification is less than that (6.528) without preclassification. In addition, the male database MAEs are lower than those of the female. These results are explained in Section 4.

As the next experiment, we compared the accuracy of our method to that in [27] which used octet-based MLBP method. With PAL database, we show the mean absolute errors (MAEs) of age estimation by the previous octet-based MLBP method [27] as shown in Table 8. In order to obtain the optimal sizes of image and LBP operator, we performed the experiments according to various sizes of facial image and LBP operator. As shown in Table 8, the lowest MAE

Figure 14: Examples of estimated and ground-truth ages with a large error: (a) ground-truth age of 19, estimated age of 30, (b) ground-truth age of 22, estimated age of 32, (c) ground-truth age of 29, estimated age of 37, (d) ground-truth age of 66, estimated age of 72, (e) ground-truth age of 75, estimated age of 68, and (f) ground-truth age of 64, estimated age of 74.

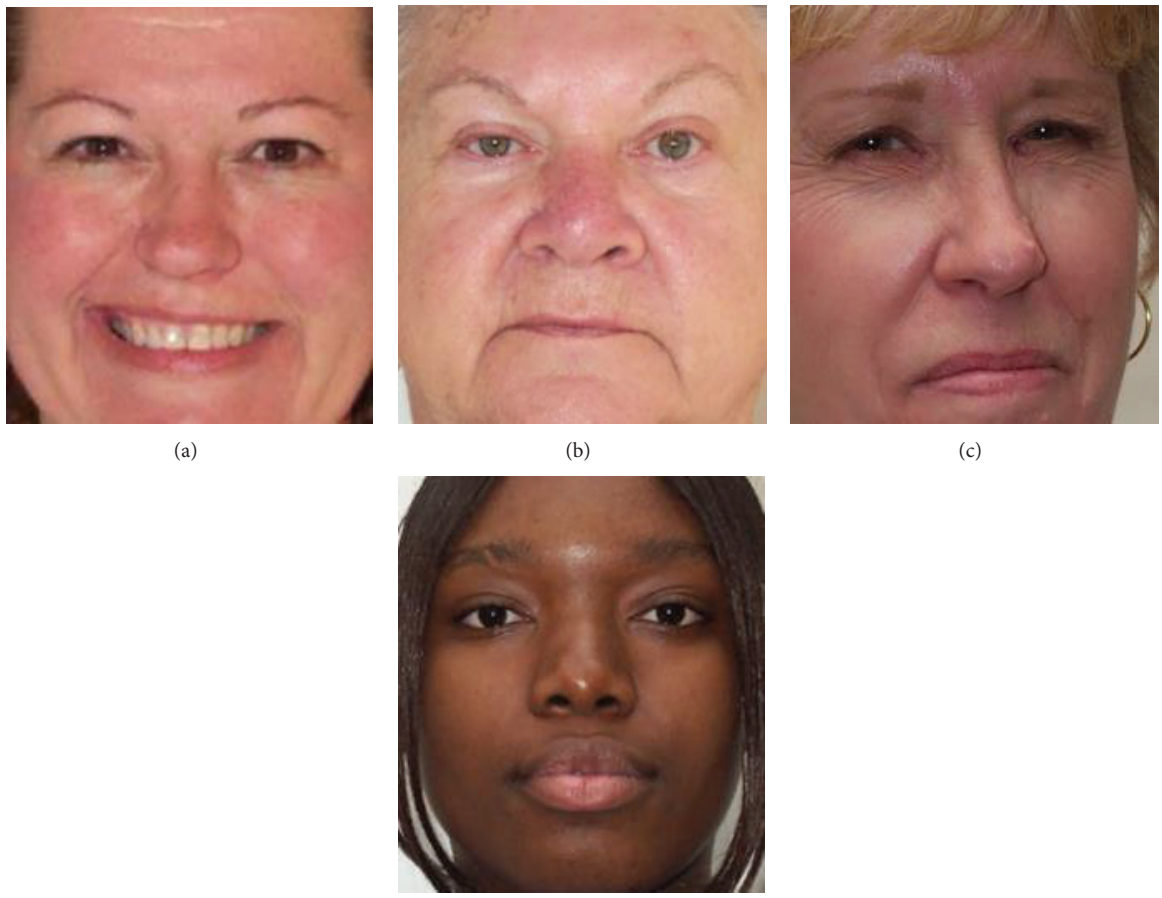
Table 5: Comparison of MAEs with and without the preclassification of facial expression (unit: years old).

Database	Using only MLBP	Using only Gabor filtering	Proposed method
Facial expression database			
Annoyed	6.00	10.526	6.605
Happy	8.121	12.168	8.102
Neutral	6.396	11.644	6.360
Sad	10.063	12.484	10.641
Surprised	8.645	10.671	8.290
Average MAE	7.218	11.715	**7.226**
Whole PAL database	6.581	11.908	6.528

is 12.693 years old which is much larger than that (6.581) by our method as shown in Table 1. Therefore, we find that our method outperforms the previous octet-based MLBP method.

4. Discussion

In Table 5, we showed the comparative estimation results of facial expression databases and the whole PAL database. The best performance was obtained for the neutral expression database with an MAE of 6.360 years old. The estimation results of other facial expression databases are unsatisfactory compared to those of the whole PAL database. Intuitively, we would expect that the MAEs of the whole PAL database would be greater than those of the facial expression database because the variation factors by facial expression in the whole PAL database also affect age estimation performance. However, by dividing the whole PAL database into the five facial expression databases, the number of images for learning in



(a) (b) (c) (d)

Figure 15: Example of estimation results of the female database: (a) ground-truth age of 42, estimated age of 43, (b) ground-truth age of 70, estimated age of 70, (c) ground-truth of 53, estimated age of 72, and (d) ground-truth of 20, estimated age of 36.

Table 6: The number of images used for the comparative experiments with/without the preclassification of gender.

Database	Number of learning images	Number of testing images
Gender database		
Male	215	214
Female	309	307
Total	**524**	**521**
Whole PAL database	523	522

Table 7: Comparison of MAEs with and without the preclassification of gender (unit: years old).

Database	Using only MLBP	Using only Gabor filtering	Proposed method
Gender database			
Male	5.796	11.086	5.783
Female	6.816	12.350	6.699
Average MAE	6.397	11.831	**6.323**
Whole PAL database	6.581	11.908	6.528

each expression database becomes so small that the learning database cannot reflect the characteristics of the testing database. However, the number of the neutral database is greater than that of the others, and the consequent learning of the neutral database can reflect testing database characteristics sufficiently. Consequently, the facial expression database MAEs (except for the neutral database) are greater than those of the whole PAL database.

The comparative estimation results of gender databases and the whole PAL database are presented in Table 7. Compared to those of the whole PAL database, the average gender database MAE is less than that of the whole PAL database, and we can confirm that gender preclassification can affect age estimation accuracy. However, the MAEs of the female database are greater than that of the whole PAL database, whereas that of the male database is much less than those of the other cases. The reason this occurs is as follows.

The female database includes wider hairstyle variation than the male, as shown in Figures 15(c) and 15(d). In addition, variations caused by cosmetics are wider in the female database. Thus, these variations of learning data in

Table 8: MAEs measured by the previous MLBP method (unit: years old).

Size of facial image	Size of LBP operator		
	3 × 3 operator	5 × 5 operator	7 × 7 operator
64 × 64	16.617	14.656	13.966
96 × 96	15.584	13.503	13.612
128 × 128	16.784	12.878	**12.693**
160 × 160	16.803	13.376	12.730
192 × 192	17.711	13.809	13.119
220 × 250	18.365	13.460	12.813

the female database cannot be trained adequately using only a female database with a small number of images; hence, the MAE of the whole PAL database, which was trained using more learning samples, can be smaller. Although the number of images in the male database is less than that in the female database, the variations in the male database are narrower; hence, the consequent learning images of the male database can reflect testing database characteristics adequately. Thus, the MAEs of the male database are less than those of the female and whole PAL databases.

There have been a lot of previous methods for automatic recognition of gender and facial expression. Therefore, the errors of preclassification are different according to the specific recognition method and the consequent effect of gender and facial expression on age estimation changes. Since we try to analyze only the effect of gender and facial expression on age estimation system without other factors, we used the method of manual preclassification of gender and facial expression in our experiment.

In our experiment, among a total of 1,046 facial images in PAL aging database, there was only 1 image where face detection failed using AdaBoost method. This error image was not used for age estimation because the main point of our research is not face detection but age estimation. In addition, in real applications, if the system fails in detecting the human's face from input image, the further step of age estimation is not performed, but the step of recapturing is repeated until the face detection is successful. Therefore, the 1 error image among 1,046 ones can be neglected in the real application of age estimation.

5. Conclusion

In this paper, we proposed a new age estimation method based on a combination of MLBP, Gabor filtering, and SVR. The experimental results showed that the proposed age estimation method outperforms previous methods by producing a better estimation result. Using the proposed age estimation method, we investigated the effects of gender and facial expression on estimation performance. We confirmed that gender and facial expression affect age estimation only if the system can be trained adequately with a large number of images. In future work, we plan to enhance the age estimation performance of our method using the scheme of assigning adaptive weights to MLBP features of each subblock based on neural networks or fuzzy systems. In addition, other effects of race, image resolution, and focusing condition on the performance of age estimation will be studied.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This study was supported by the MSIP (Ministry of Science, ICT and Future Planning), Korea, under the ITRC (Information Technology Research Center) Support Program (NIPA-2014-H0301-14-1021) supervised by the NIPA (National IT Industry Promotion Agency).

References

[1] Y.-S. Ho, "Challenging technical issues of 3D video processing," *Journal of Convergence*, vol. 4, no. 1, pp. 1–6, 2013.

[2] P. Verma, R. Singh, and A. K. Singh, "A framework to integrate speech based interface for blind web users on the websites of public interest," *Human-Centric Computing and Information Sciences*, vol. 3, article 21, 2013.

[3] M. Malkawi and O. Murad, "Artificial neuro fuzzy logic system for detecting human emotions," *Human-centric Computing and Information Sciences*, vol. 3, article 3, pp. 1–13, 2013.

[4] O. Chagnaadorj and J. Tanaka, "Gesture input as an out-of-band channel," *Journal of Information Processing Systems*, vol. 10, no. 1, pp. 92–102, 2014.

[5] C. K. Ng, G. K. Ee, N. K. Noordin, and J. G. Fam, "Finger triggered virtual musical instruments," *Journal of Convergence*, vol. 4, no. 1, pp. 39–46, 2013.

[6] Y. Fu, G. Guo, and T. S. Huang, "Age synthesis and estimation via faces: a survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 11, pp. 1955–1976, 2010.

[7] S. E. Choi, Y. J. Lee, S. J. Lee, K. R. Park, and J. Kim, "Age estimation using a hierarchical classifier based on global and local facial features," *Pattern Recognition*, vol. 44, no. 6, pp. 1262–1281, 2011.

[8] J. Txia and C. Huang, "Age estimation using AAM and local facial features," in *Proceedings of the 5th International Conference on Intelligent Information Hiding and Multimedia Signal Processing*, pp. 885–888, Kyoto, Japan, September 2009.

[9] S. E. Choi, Y. J. Lee, S. J. Lee, K. R. Park, and J. Kim, "A comparative study of local feature extraction for age estimation," in *Proceedings of the 11th International Conference on Control, Automation, Robotics and Vision (ICARCV '10)*, pp. 1280–1284, Singapore, December 2010.

[10] A. Günay and V. V. NabIyev, "Automatic age classification with LBP," in *Proceeding of the 23rd International Symposium on Computer and Information Sciences (ISCIS '08)*, pp. 1–4, Istanbul, Turkey, October 2008.

[11] H. Zhang, S. Lao, and T. Kurata, "Face recognition with consideration of aging," in *Proceedings of the IEEE International Conference on Automatic Face and Gesture Recognition and Workshops (FG '11)*, p. 343, Santa Barbara, Calif, USA, March 2011.

[12] G. Mahalingam and C. Kambhamettu, "Age invariant face recognition using graph matching," in *Proceedings of the 4th IEEE International Conference on Biometrics: Theory Applications and Systems (BTAS '10)*, pp. 1–7, Washington, DC, USA, September 2010.

[13] D. T. Nguyen, S. R. Cho, and K. R. Park, "Human age estimation based on multi-level local binary pattern and regression method," in *Future Information Technology*, vol. 309 of *Lecture Notes in Electrical Engineering*, pp. 433–438, Springer, 2014.

[14] P. Viola and M. J. Jones, "Robust Real-Time Face Detection," *International Journal of Computer Vision*, vol. 57, no. 2, pp. 137–154, 2004.

[15] X. Yang, G. Peng, Z. Cai, and K. Zeng, "Occluded and low resolution face detection with hierarchical deformable model," *Journal of Convergence*, vol. 4, no. 2, pp. 11–14, 2013.

[16] C. Shahabi, S. H. Kim, L. Nocera et al., "Janus—multi source event detection and collection system for effective surveillance of criminal activity," *Journal of Information Processing Systems*, vol. 10, no. 1, pp. 1–22, 2014.

[17] D. Ghimire and J. Lee, "A robust face detection method based on skin color and edges," *Journal of Information Processing Systems*, vol. 9, no. 1, pp. 141–156, 2013.

[18] H. L. Wang, J. Wang, W. Yau, X. L. Chua, and Y. P. Tan, "Effects of facial alignment for age estimation," in *Proceedings of the 11th International Conference on Control, Automation, Robotics and Vision (ICARCV '10)*, pp. 644–647, Singapore, December 2010.

[19] S. K. Vipparthi and S. K. Nagar, "Color directional local quinary patterns for content based indexing and retrieval," *Human-Centric Computing and Information Sciences*, vol. 4, no. 6, pp. 1–13, 2014.

[20] B. S. Manjunath and W. Y. Ma, "Texture features for browsing and retrieval of image data," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 18, no. 8, pp. 837–842, 1996.

[21] LIBSVM—A Library for Support Vector Machines, 2014, http://www.csie.ntu.edu.tw/~cjlin/libsvm/.

[22] M. Minear and D. C. Park, "A lifespan database of adult facial stimuli," *Behavior Research Methods, Instruments, and Computers*, vol. 36, no. 4, pp. 630–633, 2004.

[23] The PAL Face Database, 2014, http://agingmind.utdallas.edu/facedb.

[24] H. Heo, W. O. Lee, K. Y. Shin, and K. R. Park, "Quantitative measurement of eyestrain on 3D stereoscopic display considering the eye foveation model and edge information," *Sensors*, vol. 14, no. 4, pp. 8577–8604, 2014.

[25] "Effect size," http://en.wikipedia.org/wiki/Effect_size#Cohen.27s_d.

[26] J. Cohen, "A power primer," *Psychological Bulletin*, vol. 112, no. 1, pp. 155–159, 1992.

[27] R. Suguna and P. Anandhakumar, "Multi-level local binary pattern analysis for texture characterization," in *Proceedings of the 1st International Conference on Advances in Computing and Information Technology*, pp. 375–386, Chennai, India, July 2011.

Hindawi Publishing Corporation
The Scientific World Journal
Volume 2014, Article ID 683045, 9 pages
http://dx.doi.org/10.1155/2014/683045

Research Article

Method for User Interface of Large Displays Using Arm Pointing and Finger Counting Gesture Recognition

Hansol Kim, Yoonkyung Kim, and Eui Chul Lee

Department of Computer Science, Sangmyung University, Seoul 110-743, Republic of Korea

Correspondence should be addressed to Eui Chul Lee; eclee@smu.ac.kr

Received 27 June 2014; Accepted 15 August 2014; Published 1 September 2014

Academic Editor: Young-Sik Jeong

Although many three-dimensional pointing gesture recognition methods have been proposed, the problem of self-occlusion has not been considered. Furthermore, because almost all pointing gesture recognition methods use a wide-angle camera, additional sensors or cameras are required to concurrently perform finger gesture recognition. In this paper, we propose a method for performing both pointing gesture and finger gesture recognition for large display environments, using a single Kinect device and a skeleton tracking model. By considering self-occlusion, a compensation technique can be performed on the user's detected shoulder position when a hand occludes the shoulder. In addition, we propose a technique to facilitate finger counting gesture recognition, based on the depth image of the hand position. In this technique, the depth image is extracted from the end of the pointing vector. By using exception handling for self-occlusions, experimental results indicate that the pointing accuracy of a specific reference position was significantly improved. The average root mean square error was approximately 13 pixels for a 1920×1080 pixels screen resolution. Moreover, the finger counting gesture recognition accuracy was 98.3%.

1. Introduction

A significant amount of research has been conducted on hand gesture recognition. To perform interactive navigation and manipulation, pointing gesture and finger gesture recognition should be simultaneously executed.

Pointing gesture recognition methods can be categorized into two method types: two-dimensional (2D) image-based methods and three-dimensional (3D) methods. Although 2D image-based methods, dating back several decades, can be easily implemented today, their targeting accuracies are poor in comparison to more recent 3D methods. Therefore, 2D image-based methods are not considered in this paper.

Since the development of low cost, high depth perception 3D cameras, such as the Bumblebee and Kinect, 3D-based pointing gesture recognition methods have been widely researched. Yamamoto et al. proposed a real-time arm pointing gesture recognition method using multiple stereo cameras [1]. Because multiple stereo cameras cover a relatively wide area, the degree of freedom of a user's movement is relatively high. However, the calibration required to define epipolar geometric relations among multiple stereo cameras is considerably expensive. Other methods [2, 3] have considered head orientation to accurately estimate the hand pointing position. Head orientation typically changes as the hand targeting position changes. However, head orientation data cannot be reliably obtained, which can degrade the accuracy of the estimated hand targeting position. Another method [4] approached this problem by analyzing appearance, interactive context, and environment. However, the individual variations of these additional parameters can also lead to decreased targeting accuracy.

Recently, pointing gesture methods based on the skeleton model of the Kinect SDK (Software Development Kit) have been reported [5]. One particular pointing gesture method that was proposed utilized the skeleton model and a virtual screen [6]. The critical issue in this method, however, was defining a correspondence between the virtual screen and the physical display. In addition, this method did not consider self-occlusion;, it did not specifically address the issue of distinguishing both hand and shoulder points on a perspective line. Other 3D-based methods [7, 8] have also

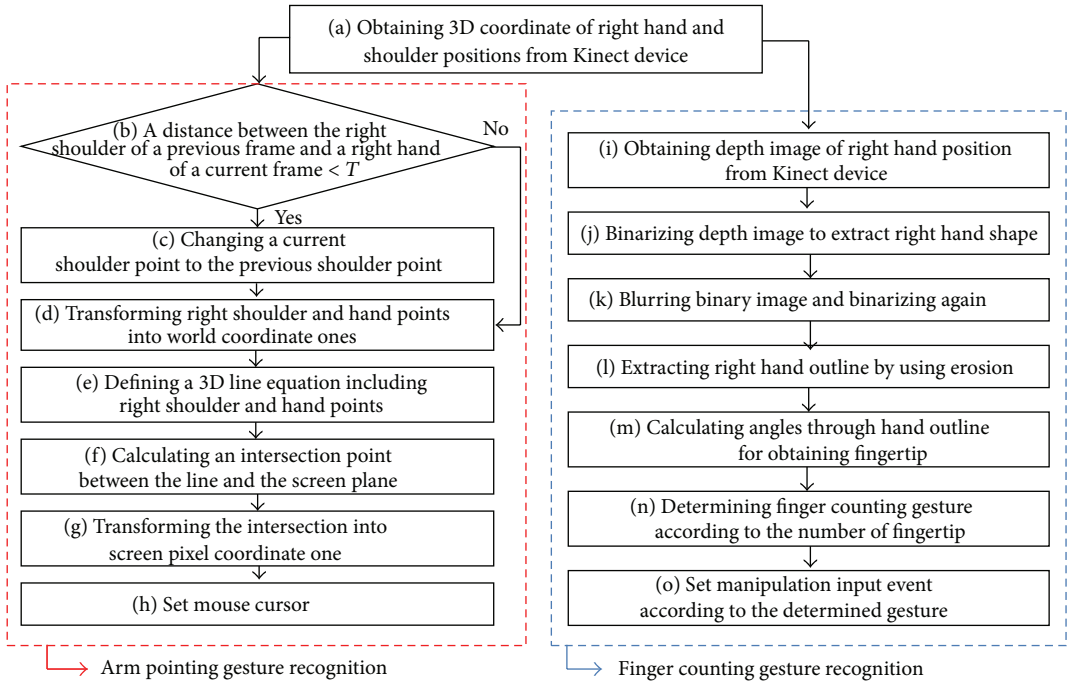


Figure 1: Flow diagram of the proposed method.

failed to address this issue. Although 3D-based methods are accurate in terms of defining a pointing vector for a fingertip, unstable dithering problems caused by low-resolution images can occur when a camera is positioned at a distance [9].

To facilitate interactive display manipulation, many finger gesture recognition methods have been studied. In a previous research effort [11], a fingertip detection method that combined depth images with color images was proposed. In this method, a finger outline tracking scheme was used, and its accuracy was relatively high. However, because the operational distance between the camera and hand was relatively short, the method cannot be considered in our large display and long distance environment. An appearance-based hand gesture recognition method using PCA (Principal Component Analysis) was described [12]. However, this method presents problems such as illuminative variation and hand orientation, which are similar to problems observed in PCA-based face recognition. In an alternative approach, a 3D template-based hand pose recognition method was proposed [13]. In this method, a 2D hand pose image was recognized by comparing 26 DOF (Degree of Freedom) 3D hand pose templates. However, the method is tightly coupled with a predefined 3D hand pose template. In addition, the computational complexity for estimating 3D hand poses from the captured 2D image stream was high. In a current research, a new hand posture recognition method was proposed based on the sparse representation of multiple features such as gray-level, texture, and shape [14]. However, this method is strongly dependent on a training database. Furthermore, the binary decision for each feature's sparsity presents a problem, because continuous values of sparse features must be considered.

To solve the problems related to previous pointing and hand gesture methods, a new arm pointing and finger counting gesture recognition method is proposed in this paper. Our proposed method is a user-dependent, calibration-free method based on the Kinect skeleton model. We resolve the self-occlusion problem in the arm pointing gesture recognition module. Moreover, finger counting gesture recognition is accurately performed using a low-resolution depth image. Both gesture recognition techniques are performed with a single Kinect device.

2. Proposed Method

Our proposed method is executed as per the steps shown in Figure 1. The method is organized into two parts, namely, arm pointing gesture recognition and finger counting gesture recognition.

2.1. Arm Pointing Gesture Recognition. Arm pointing gesture recognition is performed using the sequence shown in the red dotted box of Figure 1. First, 3D coordinates of the right-hand and shoulder positions are obtained using the skeleton model of the Kinect SDK. In the visible image captured from the Kinect device, the X and Y values of an arbitrary pixel's 3D coordinates are the same as their corresponding pixel coordinates in the visible image. The Z value, which is measured by the Kinect's depth camera, is multiplied by 10 mm. Next, we proceed to step (b), in which the Euclidean distance between the shoulder position in the previous frame and the hand position in the current frame is measured. When both the hand and shoulder positions lie on the same

Figure 2: Examples of shoulder point detection errors caused by the self-occlusion problem (red dot: shoulder point, yellow dot: hand point).

camera perspective line, the shoulder position cannot be accurately detected because of occlusion by the hand, as shown in Figure 2. We use exception handling to address such self-occlusion; if the distance measured in step (b) of Figure 1, specifically, is less than the empirically defined threshold (T = 10 pixels), the current shoulder position is set to that of the previous frame (step (c)). If the distance is greater than T, exception handling is not performed (i.e., step (c) is bypassed).

In the following step, the hand and potentially compensated shoulder coordinates (based on threshold T) are transformed into world coordinates. As shown in Figure 3, the principal point of the world coordinates is located in the top-left position of the screen. The transformation is performed according to the following equations [15]:

$$x_k = \left(i - \frac{w}{2}\right) \times (z_k + \text{minDist}) \times \text{SF} \times \frac{w}{h}, \quad (1)$$

$$y_k = \left(j - \frac{h}{2}\right) \times (z_k + \text{minDist}) \times \text{SF}, \quad (2)$$

$$z = z_k + 4000, \quad (3)$$

where minDist = −10 and SF =0.0021 are based on the calibration results of previous works [11] and i and j are the horizontal and vertical pixel positions of the captured image frame with a spatial resolution of 640 × 480 pixels. Because the default Z-distance value (z_k) can be as small as 400 mm, the Z-axis value in (3) must be compensated accordingly. Moreover, because the 3D coordinates (x_k, y_k, z) are measured from the principal point of the depth camera, the values of x_k and y_k should be adjusted by the offset ((D_x, D_y) in Figure 3) between the two principal world coordinate points and the depth camera coordinates. In our system configuration, D_x and D_y were 4450 mm and 950 mm, respectively, and were measured manually. The orientation variation between the Kinect and the screen is ignored. That is,

$$x = x_k + D_x, \qquad y = y_k - D_y. \quad (4)$$

The two world coordinate positions for the shoulder and hand are given by (X_s, Y_s, Z_s) and (X_e, Y_e, Z_e), respectively.

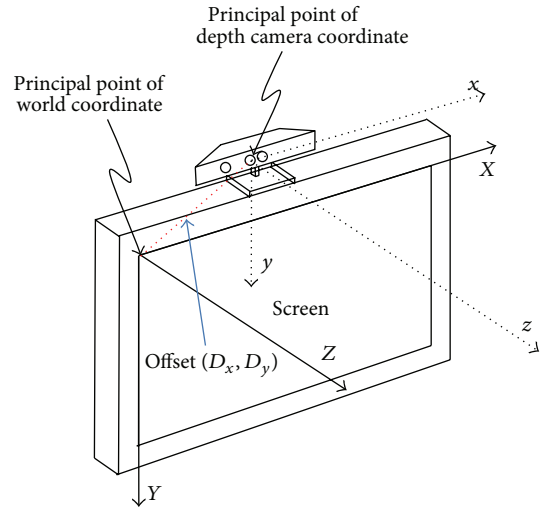


Figure 3: Geometric relation between the world coordinates and depth camera coordinates [10].

Next (step (e) in Figure 1), a 3D line equation is defined from these two 3D points using the following equation:

$$\frac{X_s - X}{X_s - X_e} = \frac{Y_s - Y}{Y_s - Y_e} = \frac{Z_s - Z}{Z_s - Z_e}. \quad (5)$$

Because the line equation is regarded as an arm-pointing vector and the planar equation of the screen is z = 0, the intersection point (X_i, Y_i) between the screen and the line equation is calculated in step (f) in Figure 1 as follows:

$$X_i = -\frac{Z_e(X_s - X_e)}{Z_s - Z_e} + X_e,$$
$$Y_i = -\frac{Z_e(Y_s - Y_e)}{Z_s - Z_e} + Y_e. \quad (6)$$

The intersection point is the physical targeting position shown in Figure 4. Because the physical targeting position (X_i, Y_i) is given in millimeters, its position must be transformed into logical pixel coordinates (x_p, y_p) in order

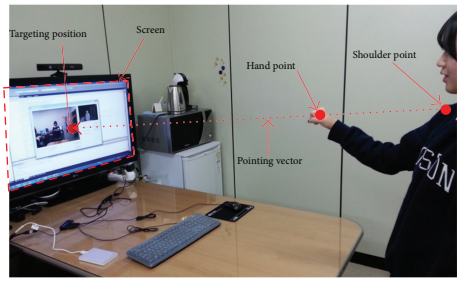


Figure 4: Conceptual diagram for representing the pointing vector and targeting position [10].

to control the system mouse cursor position (step (g) of Figure 1). These logical pixel coordinates are given by

$$x_p = \frac{X_i \cdot x_{\text{res}}}{W}, \qquad y_p = \frac{Y_i \cdot y_{\text{res}}}{H}, \tag{7}$$

where $(x_{\text{res}}, y_{\text{res}})$ is the spatial resolution of the screen and W and H are the actual width and height of the screen, respectively. For our system, $(x_{\text{res}}, y_{\text{res}}) = (1920, 1080)$, $W = 932$ mm, and $H = 525$ mm. Finally, the cursor position of the system mouse is moved to the calculated arm pointing position (x_p, y_p) using the WINAPI function SetCursorPos(int x, int y) [16].

2.2. Finger Counting Gesture Recognition. Finger counting gesture recognition is processed using the steps in the blue dotted box of Figure 1. In step (i), the right hand depth image is obtained based on the position of the right hand, which is acquired by using the Kinect SDK skeleton model. The spatial resolution of the image is 100×100. The gray levels of the depth image indicate the Z-distance between the Kinect depth camera lens and the corresponding object. Therefore, the higher the gray level, the shorter the distance between the camera lens and the object. In order to extract the right hand's shape, the right hand depth image is binarized by regarding the higher gray level in the depth image as the threshold (step (j) in Figure 1). However, an outline of right hand shape that has been binarized only once will be articulated, as shown in Figure 5(a). An extracted edge from a once-binarized right hand image will contain bifurcation, which may disturb fingertip detection that uses edge tracking. To solve this problem, a once-binarized right hand image is blurred by using a 7×7 average filter, as shown in Figure 5(b). Subsequently, a binarization is performed again using the median gray value (128 in a 0–255 gray scale) to obtain a right-hand shape (step (k) in Figure 1). A hand shape image with a flattened outline can be acquired, as shown in Figure 5(c).

Then, hand outline detection must be performed, to facilitate fingertip detection. Assuming that the twice-binarized image and the structural element for morphological erosion ($\odot$) are A and B, respectively, the hand outline image ($\beta(A)$) can be extracted by subtracting the erosion image from A (step (l) in Figure 1) using the following equation:

$$\beta(A) = A - (A \odot B). \tag{8}$$

As a result, the outline image of the right hand can be acquired as shown in Figure 6.

Subsequently, counterclockwise edge tracking is performed; the edge pixel that has the minimum Y-axis value is used as the starting point. If two points on the edge have the same minimum Y-axis value, the point with the lowest X-axis value is used as the starting point. The 8-neighbor pixels (Figure 7(a)) surrounding the starting point are assigned priorities 1 through 8, as shown in Figure 7(b).

According to the priority, the 8-neighbor pixels are analyzed to determine whether the pixel is an edge (gray level value = 255) and whether it is "nonvisited." If an edge pixel that is "nonvisited" is detected among the 8-neighbor pixels, the pixel is determined to be the new center position. Accordingly, the previous center position is marked as "visited." These steps are repeated until no pixels are found that satisfy the two conditions (edge and nonvisited) among the 8-neighbor pixels.

If an 8-neighbor pixel priority is not assigned, edge tracking will be performed abnormally. For example, in the right hand edge of Figure 8, the minimum Y-axis value is determined as the starting point and is labeled in the figure. Edge tracking is performed by using the starting point as a center position. Then, the $(x - 1, y + 1)$ pixel of the starting point's 8-neighbor pixels is changed to the next center point, according to the predefined priority order. If the $(x + 1, y + 1)$ pixel has a higher priority than the $(x - 1, y + 1)$ pixel, the priority is appropriate for clockwise edge tracking. Therefore, 8-neighbor pixels that have a value of $(x - 1)$ as their X-index are assigned a higher priority than pixels that have $(x + 1)$ as their X-index, to facilitate counterclockwise edge tracking. Edge tracking proceeds normally until arriving at position A. In position A, if the $(x - 1, y + 1)$ pixel of the center point has a higher priority than the $(x, y + 1)$ pixel, the $(x, y + 1)$ pixel will not be visited. Then, in case the priority of the $(x + 1, y)$ pixel is higher than the $(x - 1, y)$ pixel, edge tracking will terminate abnormally when the bottom of A becomes the center position. Likewise, in position B, if the $(x - 1, y - 1)$ pixel has a higher priority than the $(x - 1, y)$ pixel, the $(x - 1, y)$ pixel will not be visited and edge tracking will terminate abnormally. To prevent these abnormal cases, edge tracking should be performed according to a predefined priority.

While edge tracking is performed, three sequential points, at fifth-next-adjacent intervals, must be extracted as shown in Figure 8 (red points). Then, the angle between the three extracted points as Figure 9 must be calculated, using the following equation (step (m) in Figure 1):

$$\theta = \left(\tan^{-1}\left(\frac{y_1 - c_y}{x_1 - c_x}\right) - \tan^{-1}\left(\frac{y_2 - c_y}{x_2 - c_x}\right)\right) * \frac{180}{\pi}. \tag{9}$$

Here, the angle of the three points is calculated using the atan2 function included in math.h header of the C standard library [17]. However, the atan2 function's output ranges are $-\pi$ to π. Therefore, if the value of $\tan^{-1}((y_2 - c_y)/(x_2 - c_x))$ is negative and the value of $\tan^{-1}((y_1 - c_y)/(x_1 - c_x))$ is positive, the opposite angle of the three points will be calculated, as shown in Figure 10(b).

(a) (b) (c)

Figure 5: Binarized right hand images; (a) once binarized image, (b) blurred image with 7 * 7 average filter, and (c) twice binarized image.

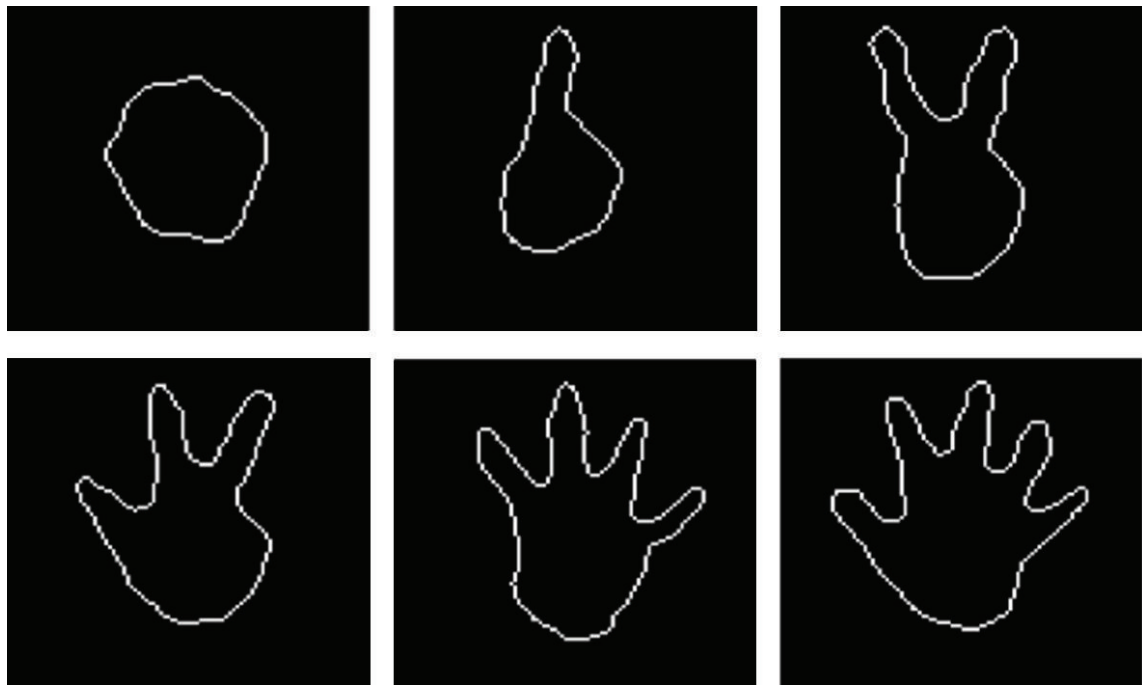


Figure 6: Examples of right hand outline images obtained by the proposed method.

To solve this problem, the angle of the three points is calculated using the following equation, as illustrated in Figure 10(c):

$$\theta = \left(\left(\tan^{-1}\left(\frac{y_2 - c_y}{x_2 - c_x}\right) - 2\pi\right) - \tan^{-1}\left(\frac{y_1 - c_y}{x_1 - c_x}\right)\right) * \frac{180}{\pi}. \quad (10)$$

Then, if θ is lower than the predefined threshold ($T = 110^{\circ}$), the center point of the three points is regarded as the fingertip (steps (n) and (o) in Figure 1). Finally, exception handling will be performed if one of the two noncenter points has already been identified as a fingertip, because if two of the three extracted points satisfy the condition, this indicates that the two points are on the same fingertip.

3. Experimental Result

To validate the proposed method, experiments were performed to measure the accuracy of the arm pointing and finger counting gesture recognition techniques. In the experiments, the distance between the subject's body and the screen was approximately 2.2 m. Software capable of recognizing upper body pointing gestures was implemented using C++, MFC (Microsoft Foundation Classes), and the Kinect SDK. The implemented software, as shown in Figure 11, could be operated in real time (approximately 18.5 frames/s) without frame delay or skipping on a PC with an Intel i7-3770 CPU, 8 GB RAM, and a 42-inch display.

In our first experiment, targeting accuracy for specific pointing positions was measured for eight subjects. Each subject pointed to five predefined reference positions (indicated

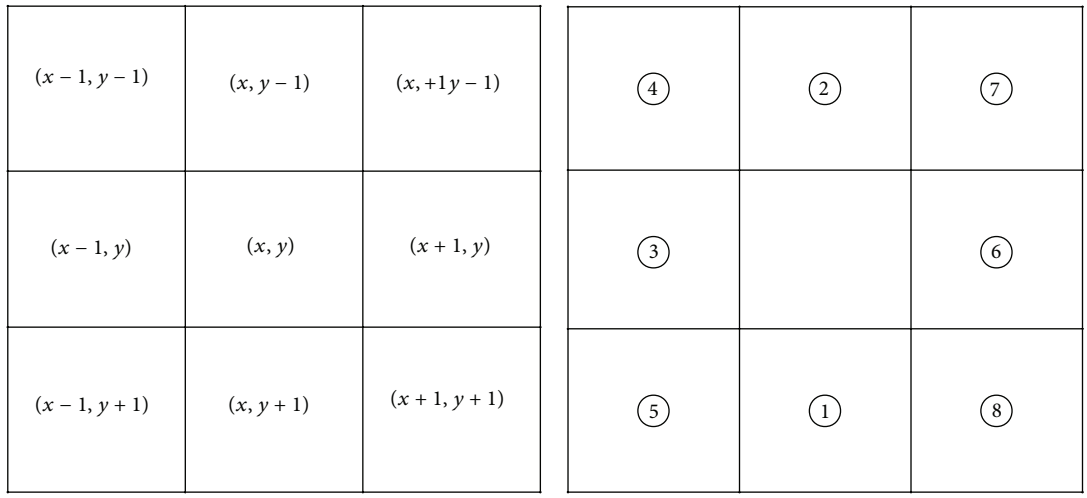


Figure 7: (a) 8-neighbor pixels and (b) assigned priority of the 8-neighbor pixels.

Table 1: Targeting error against reference positions [10].

Reference positions	Error without compensation			Error with compensation		
	X-axis	Y-axis	RMS	X-axis	Y-axis	RMS
1	62.90	52.19	81.73	16.95	22.04	27.81
2	4.54	5.95	7.49	4.29	3.41	5.48
3	6.33	6.95	9.40	0.54	5.87	5.89
4	2.4	0.54	2.51	14.04	7.25	15.80
5	7.79	3.25	8.44	10.12	0.91	10.16

Unit: pixel.

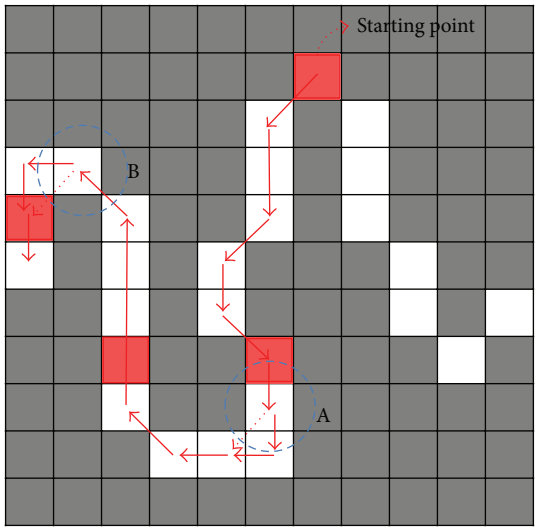


Figure 8: Edge tracking path example for explaining the necessity of the predefined priority (full arrow: normal edge tracking process, dotted arrow: abnormal edge tracking process).

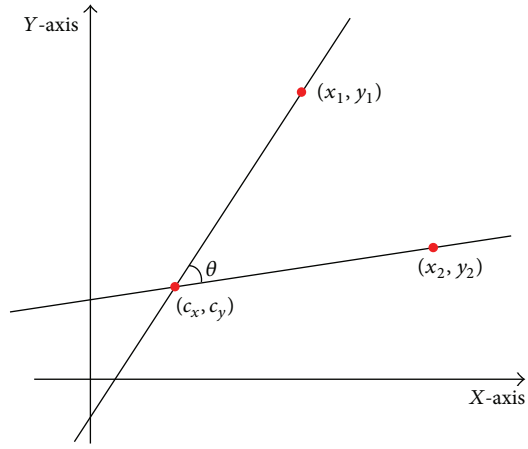


Figure 9: Example with three arbitrary points in a 2D coordinate.

by the "×" in Figure 12); this sequence was repeated three times. The indicated order was assigned randomly. Tests were performed with and without the self-occlusion compensation function in order to validate the performance of our proposed compensation method.

The measured accuracy results from the experiment are shown in Figure 12 and Table 1. Four outliers caused by detection errors of the hand or shoulder were not included. As shown in Figure 12 and Table 1, position 1 experienced

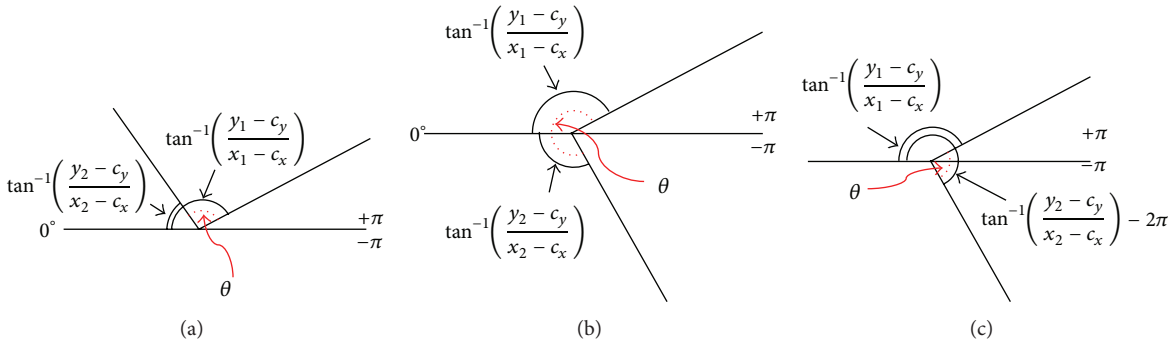


Figure 10: Angle calculation using three sequential points of the hand's outline edge. (a) Obtaining angle in a normal case. (b) Obtaining the opposite angle in error case. (c) Compensating error case (θ: calculated angle of three points at the intervals of five adjacent edge points).

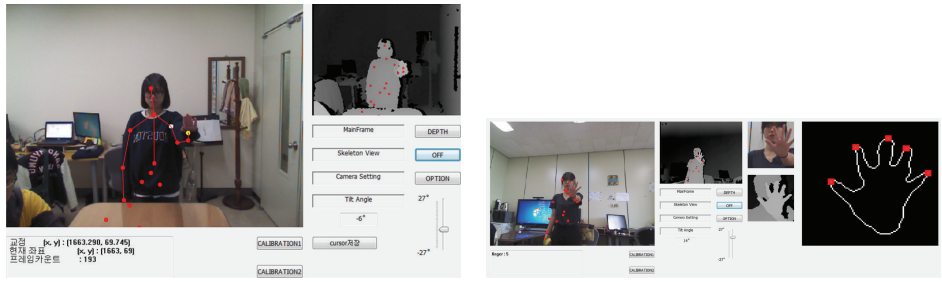


Figure 11: Lab-made gesture recognition software based on the upper body skeleton model in the Kinect SDK. (a) Pointing gesture recognition software (white dot: shoulder point, yellow dot: hand point) [10]. (b) Finger counting gesture recognition software.

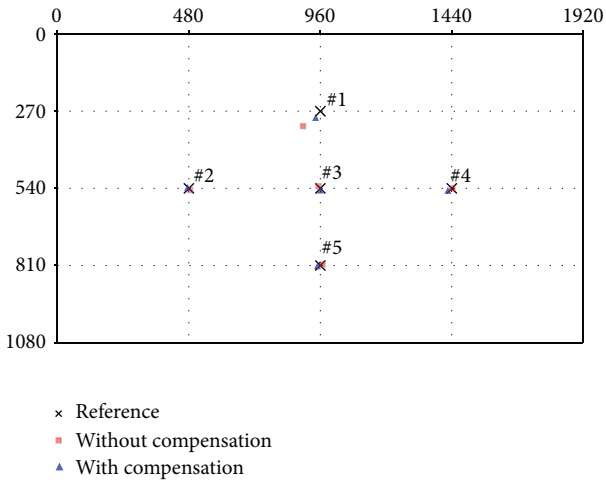


Figure 12: Experimental results of detecting target positions against reference positions [10].

a much higher error rate compared to the other reference positions. This can be attributed to self-occlusion occurring most frequently in position 1; specifically, both 3D shoulder and hand points are positioned on a single camera perspective line. After adopting the proposed compensation method, we confirmed improvements in targeting accuracy for position 1. In this case, the X-axis error received more compensation than that of the Y-axis, as shown in Table 1. The average RMS errors from tests with and without self-occlusion compensation were approximately 21.91 pixels and 13.03 pixels, respectively.

Table 2: Accuracy of fingertip recognition.

Number of fingertips	0	1	2	3	4	5	Average
Accuracy of recognition	98	99	98	97	100	98	98.3

Unit: %.

In our second experiment, the accuracy of the finger counting gesture recognition method was evaluated to validate the fingertip detection method. Five subjects participated in the experiment. Each subject performed six predefined finger-counting gestures, regardless of hand orientation, as shown in Figure 13. The order of the finger gestures was randomly announced. The accuracy was measured by comparing the number of fingers in the hand gesture to the number of fingertips that were detected.

Experimental results from the accuracy measurement are listed in Table 2. Here, the accuracy of the three-finger gesture was lower, compared to the other finger counting gestures. As shown in Figure 14, the shape of the folded ring and little fingers in the three-finger gesture is sharper than

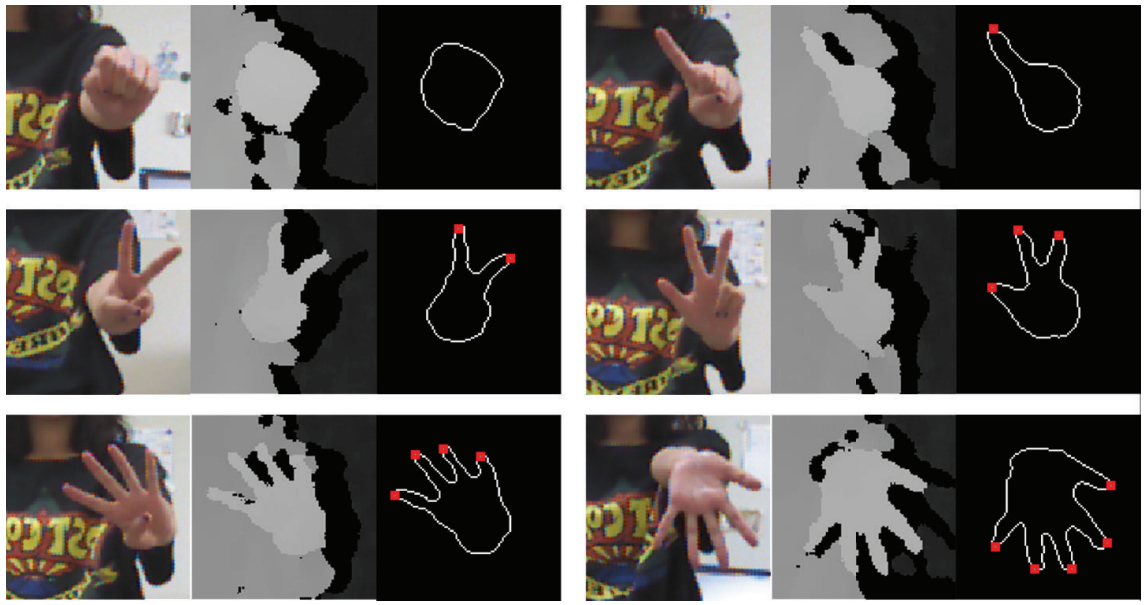


Figure 13: Examples of six different finger gestures used in the second experiment.

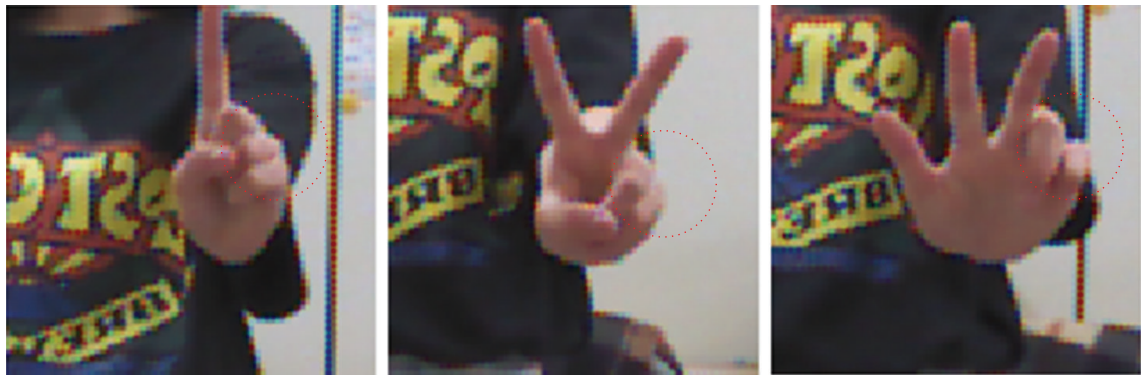


Figure 14: Shape comparison of folded ring and little fingers for one-, two-, and three-finger gestures.

Table 3: Average processing times for arm pointing and finger counting gesture recognition.

	Arm pointing gesture recognition	Finger counting gesture recognition
Average processing time	6.1	0.5

Unit: ms.

that in the one- and two-finger gestures. In one- and two-finger gestures, the thumb suppresses the folded ring and little fingers. Because the sharper shape of the ring and little finger in the three-finger gesture can be misinterpreted as fingertips, the three-finger gesture may have been interpreted as a four- or five-finger gesture. As a result, the average fingertip recognition accuracy for the six predefined finger gestures was 98.3%.

As shown in Table 3, the processing times for arm pointing and finger counting gesture recognition were considerably fast: 6.1 ms and 0.5 ms, respectively. The skeleton model detection time was not included in the calculated times. These experiments demonstrate that our proposed method can accurately recognize pointing and counting gestures in an efficient manner.

4. Conclusion

In this paper, we proposed a method for performing both pointing gesture and finger gesture recognition for large display environments, using a single Kinect device and a skeleton tracking model. To prevent self-occlusion, a compensation technique was designed to correct the shoulder position in cases of hand occlusion. In addition, finger counting gesture recognition was implemented based on the hand position depth image extracted from the end of the pointing vector. Experimental results showed that the pointing accuracy of a specific reference position significantly improved by adopting exception handling for self-occlusions. The average root mean square error was approximately 13 pixels for a 1920×1080 pixels screen resolution. Furthermore,

the accuracy of finger counting gesture recognition was 98.3%.

In future works, we will define effective manipulation commands for the detected finger counting gestures. Further, the proposed method will be applied to immersive virtual reality contents [18–20] as a natural user interface method for performing interactive navigation and manipulation.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgment

This research was supported by the MSIP (Ministry of Science, ICT and Future Planning), Korea, under the ITRC (Information Technology Research Center) support program (NIPA-2014-H0301-14-1021) supervised by the NIPA (National IT Industry Promotion Agency).

References

[1] Y. Yamamoto, I. Yoda, and K. Sakaue, "Arm-pointing gesture interface using surrounded stereo cameras system," in *Proceedings of the 17th International Conference on Pattern Recognition (ICPR '04)*, vol. 4, pp. 965–970, Cambridge, UK, August 2004.

[2] K. Nickel and R. Stiefelhagen, "Pointing gesture recognition based on 3D-tracking of face, hands and head orientation," in *Proceedings of the 5th International Conference on Multimodal Interfaces (ICMI '03)*, pp. 140–146, Vancouver, Canada, November 2003.

[3] K. Nickel and R. Stiefelhagen, "Real-time recognition of 3D-pointing gestures for human-machine-interaction," in *Proceedings of the 25th DAGM Symposium*, vol. 2781 of *Lecture Notes in Computer Science*, pp. 557–565, Springer, Magdeburg, Germany, 2003.

[4] M. Kolesnik and T. Kuleba, "Detecting, tracking, and interpretation of a pointing gesture by an overhead view camera," in *Pattern Recognition: 23rd DAGM Symposium Munich, Germany, September 12–14, 2001 Proceedings*, vol. 2191 of *Lecture Notes in Computer Science*, pp. 429–436, Springer, Heidelberg, Germany, 2001.

[5] Tracking Users with Kinect Skeletal Tracking, http://msdn.microsoft.com/en-us/library/jj131025.aspx.

[6] P. Jing and G. Yepeng, "Human-computer interaction using pointing gesture based on an adaptive virtual touch screen," *International Journal of Signal Processing, Image Processing*, vol. 6, no. 4, pp. 81–92, 2013.

[7] Y. Guan and M. Zheng, "Real-time 3D pointing gesture recognition for natural HCI," in *Proceedings of the 7th World Congress on Intelligent Control and Automation (WCICA '08)*, pp. 2433–2436, Chongqing, China, June 2008.

[8] S. Carbini, J. E. Viallet, and O. Bernier, "Pointing gesture visual recognition for large display," in *International Workshop on Visual Observation of Deictic Gestures*, pp. 27–32, 2004.

[9] R. Kehl and L. van Gool, "Real-time pointing gesture recognition for an immersive environment," in *Proceedings of the 6th IEEE International Conference on Automatic Face and Gesture Recognition (FGR '04)*, pp. 577–582, Seoul, Republic of Korea, May 2004.

[10] H. Kim, Y. Kim, D. Ko, J. Kim, and E. Lee, "Pointing gesture interface for large display environments based on the kinect skeleton model," in *Future Information Technology*, vol. 309 of *Lecture Notes in Electrical Engineering*, pp. 509–514, 2014.

[11] H. Park, J. Choi, J. Park, and K. Moon, "A study on hand region detection for kinect-based hand shape recognition," *The Korean Society of Broadcast Engineers*, vol. 18, no. 3, pp. 393–400, 2013.

[12] J. Choi, H. Park, and J.-I. Park, "Hand shape recognition using distance transform and shape decomposition," in *Proceedings of the 18th IEEE International Conference on Image Processing (ICIP '11)*, pp. 3605–3608, Brussels, Belgium, September 2011.

[13] I. Oikonomidis, N. Kyriazis, and A. Argyros, "Markerless and efficient 26-DOF hand pose recovery," in *Proceedings of the 10th Asian Conference on Computer Vision*, pp. 744–757, 2010.

[14] C. Cao, Y. Sun, R. Li, and L. Chen, "Hand posture recognition via joint feature sparse representation," *Optical Engineering*, vol. 50, no. 12, Article ID 127210, 10 pages, 2011.

[15] https://groups.google.com/forum/#!topic/openkinect/ihfBIY56Is.

[16] http://msdn.microsoft.com/en-us/library/windows/desktop/ms648394%28v=vs.85%29.aspx.

[17] http://msdn.microsoft.com/en-us/library/windows/desktop/bb509575%28v=vs.85%29.aspx.

[18] C. Ng, J. Fam, G. Ee, and N. Noordin, "Finger triggered virtual musical instruments," *Journal of Convergence*, vol. 4, no. 1, pp. 39–46, 2013.

[19] J. McNaull, J. Augusto, M. Mulvenna, and P. McCullagh, "Flexible context aware interface for ambient assisted living," *Human-Centric Computing and Information Sciences*, vol. 4, no. 1, pp. 1–41, 2014.

[20] A. Berena, S. Chunwijitra, H. Okada, and H. Ueno, "Shared virtual presentation board for e-Meeting in higher education on the WebELS platform," *Journal of Human-centric Computing and Information Sciences*, vol. 3, no. 6, pp. 1–17, 2013.

Hindawi Publishing Corporation
The Scientific World Journal
Volume 2014, Article ID 832871, 9 pages
http://dx.doi.org/10.1155/2014/832871

Research Article

Nonuniform Video Size Reduction for Moving Objects

Anh Vu Le,[1] Seung-Won Jung,[2] and Chee Sun Won[1]

[1] *Department of Electrical and Electronic Engineering, Dongguk University-Seoul, Seoul 100-715, Republic of Korea*
[2] *Department of Multimedia Engineering, Dongguk University-Seoul, Seoul 100-715, Republic of Korea*

Correspondence should be addressed to Chee Sun Won; cswon@dongguk.edu

Received 19 June 2014; Revised 4 August 2014; Accepted 15 August 2014; Published 31 August 2014

Academic Editor: Young-Sik Jeong

Moving objects of interest (MOOIs) in surveillance videos are detected and encapsulated by bounding boxes. Since moving objects are defined by temporal activities through the consecutive video frames, it is necessary to examine a group of frames (GoF) to detect the moving objects. To do that, the traces of moving objects in the GoF are quantified by forming a spatiotemporal gradient map (STGM) through the GoF. Each pixel value in the STGM corresponds to the maximum temporal gradient of the spatial gradients at the same pixel location for all frames in the GoF. Therefore, the STGM highlights boundaries of the MOOI in the GoF and the optimal bounding box encapsulating the MOOI can be determined as the local areas with the peak average STGM energy. Once an MOOI and its bounding box are identified, the inside and outside of it can be treated differently for object-aware size reduction. Our optimal encapsulation method for the MOOI in the surveillance videos makes it possible to recognize the moving objects even after the low bitrate video compressions.

1. Introduction

Surveillance cameras are ubiquitous and play an important role in our daily life. The recorded video data from the surveillance cameras provide rich information to many applications ranging from human and machine interactions [1–3] to content indexing and retrieval [4, 5]. For such applications of digital video surveillance and digital video recording (DVR) systems [6, 7], it is often required to examine moving objects for a long period of frames in recorded videos. This naturally demands highly efficient compressions for a huge amount of video data. Here, the conflicting requirement is how to maintain high visual quality, especially for the important information in the video such as the moving objects, in low bit-rate compressions.

A wide range of advanced techniques has been proposed to improve the conventional video compression framework. For example, an efficient block mode determination algorithm [8] was applied for an efficient scalable video compression, where video data can change their resolution to use the limited bandwidth efficiently. The scalable compression scheme is particularly useful for surveillance videos. Note that surveillance videos usually consist of alternating sequence of frames with static background and moving objects. Definitely, the moving objects are the important data to be preserved in the compressions. This requires the compression technique to distinguish the important moving objects of interest (MOOI) from the unimportant static background (non-MOOI) in the video and to treat them differently in the compression process. As a result, the natural user interface (NUI) via the face detection [9, 10] in the surveillance videos can be a feasible technique even for highly compressed videos. To differentiate the MOOI from the non-MOOI, the object segmentation and tracking processes can be applied [6, 7]. However, these methods need to identify accurate object boundaries, which often require expensive computations. Weng et al. [11] used Kalman filter for object detecting and tracking. This method can detect and track the object trajectory frame by frame accurately. However, the object boundary that differentiates the MOOI from the non-MOOI cannot be identified clearly. Goswami et al. [12] used a mesh-based technique to track moving objects in video sequence. Mesh-based motion estimation techniques

are more accurate than the block based method, but they are relatively slow due to the high computational complexity.

In this paper we differentiate the MOOI from the non-MOOI by detecting the bounding boxes surrounding the MOOIs for each group of frames (GoF). Then, the detected bounding boxes encapsulating the MOOIs are fixed throughout the GoF. To detect the bounding box we need to identify the pixels with spatiotemporal saliency. For this, we construct the spatiotemporal gradient map (STGM) of a GoF [13], where each pixel in the STGM represents the level of the temporal and spatial saliency. Then, the optimal size of the bounding box is determined to include the local pixels with the highest energy density of the STGM. Once the pixels including the MOOI are determined by the bounding boxes, we can apply linear transformations with different slopes to the inside and outside of the bounding boxes such that the MOOI is intact while those in the non-MOOI are the main target for size reduction. After this initial data reduction, the standard H.264/AVC compression is applied to the size-reduced frames for further compressions. At the receiver, the reverse processes including decompression by H.264/AVC and the size expansion by the inverse linear transformations are applied to restore the video data with the original size. The overall block diagram of our MOOI-based compression is shown in Figure 1.

As far as the image size reduction is concerned, various methods have been proposed for content-aware image and video retargeting context such as the seam carving methods [13–17]. These methods reduce the image size by removing the unimportant seam lines that have the low saliency. The output video has the reduced spatial resolution, where the rich texture areas are maintained but the homogenous areas are removed. These video retargeting methods are mainly for display purposes and it is not reversible to reconstruct the original image size from the retargeted videos unless the decoder knows exact locations of the discarded seam pixels. Therefore, the conventional video retargeting methods are not appropriate as the initial data reduction for video compression. Note that image pruning scheme with image downsampling as a preprocessing step of video compressions has been also used in Vo et al. [18], where one of the two consecutive image lines (i.e., even or odd lines) is to be discarded for image size reduction. Since the line dropping is limited for one of two consecutive lines and the criterion for line dropping is based on the least mean square errors (LMSE) of the interpolated image data, it is hard to differentiate the MOOI from the non-MOOI. A reversible nonuniform size reduction method was also proposed in Won and Shirani [19] without the bounding box. Finally, we note that this paper is the extended version of our previous single MOOI [20] to multiple MOOIs.

Our contributions of this paper can be summarized as follows: (i) we introduce a spatiotemporal gradient map (STGM) to trace the boundary of the MOOI within a GoF; (ii) based on the STGM, a cost function for determining the center and size of the bounding box encapsulating the MOOI is formulated; (iii) an optimization process for updating the center and size of the bounding box alternately is introduced; (iv) the subjective visual quality especially for the MOOI is enhanced by nonuniformly reducing the size of the video frames as a preprocessing for the H.264 video compressions.

This paper is organized as follows. In Section 2, the algorithm for detecting multiple moving objects in video is presented. Then, different linear transforms are applied to MOOI and non-MOOI for size reduction in Section 3. Section 4 shows the experimental results of proposed method. Section 5 concludes this paper.

2. Detection of Multiple Moving Objects of Interest

Moving objects in video can be detected by motion estimation, which is a computationally expensive process. Instead, in this paper, all MOOIs are detected by using spatial and temporal gradients in a GoF of H.264/AVC structure. Specifically, a spatial gradient map $S_g^t(i, j)$ at pixel (i, j) of a frame I^t with a size of $N_r \times N_c$ can be defined as an average of the magnitude of spatial gradients within a window $(2\psi + 1) \times (2\psi + 1)$ as follows:

$$S_g^t(i,j) = \frac{1}{(2\psi+1)^2} \sum_{u=-\psi}^{\psi} \sum_{v=-\psi}^{\psi} g^t(i+u, j+v), \quad (1)$$

where $g^t = |\partial I^t/\partial i| + |\partial I^t/\partial j|$ is the magnitude of the spatial gradient. Using S_g^t, we define the temporal saliency cost S_{tsc}^t by computing the temporal gradient of the spatial gradients between the two consecutive even numbered frames (even numbered frames give more temporal deviations with less computational load):

$$S_{\text{tsc}}^t = \left| S_g^t - S_g^{t+2} \right|. \quad (2)$$

As a result of the spatiotemporal gradients in (1) and (2), the boundary pixels of the moving objects are highlighted in the temporal saliency cost S_{tsc}^t. Then, we can construct a STGM R^k by mosaicking the maximum saliency cost S_{tsc}^t at each pixel location throughout the GoF with even numbered frames starting from the first even numbered frame m_0 to $m_0 + N_0 - 2$ for N_0 frames in the kth GoF of the H.264/AVC as follows:

$$R^k(i,j) = S_{\text{tsc}}^{t^*}(i,j), \quad (3)$$

where $t^* = \arg\max_{t \in [m_0, m_0+N_0-2]} S_{\text{tsc}}^t(i, j)$. Note that, according to the definition of $R^k(i, j)$ in (3), the value of $R^k(i, j)$ corresponds to the spatial or temporal boundary of the MOOI in the GoF. Therefore, we can define a bounding box (or window) with $(2w_r + 1) \times (2w_c + 1)$ to encompass the trace of each MOOI, where the weighted sum of $R^k(i, j)$ within the optimal window at the center $C = (C_r, C_c)$ yields the peak value. Therefore, our problem boils down to the determination of the optimal window size (w_r, w_c) and the center of the bounding box $C = (C_r, C_c)$. In this paper, we propose a novel method to find bounding boxes and their centers for multiple MOOIs in a GoF. Our approach to determine (w_r, w_c) and $C = (C_r, C_c)$ is to take an alternate optimization process between (4) and (5). That is, starting

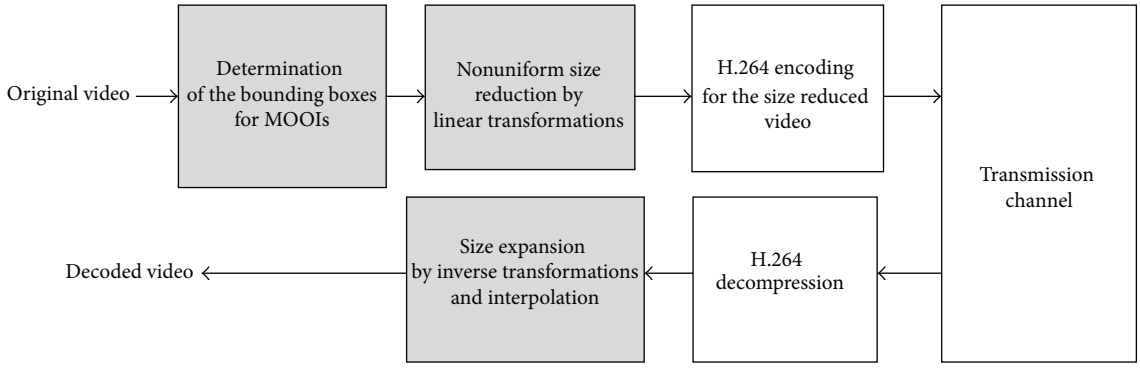


Figure 1: Flowchart of the proposed video encoding and decoding system.

from an initial value of $(w_r^{(0)}, w_c^{(0)})$ we apply (4) to have $(C_r^{(1)}, C_c^{(1)})$. Then $(C_r^{(1)}, C_c^{(1)})$ is used to update $(w_r^{(1)}, w_c^{(1)})$ by (5). This alternate process continues until there is no more change from $(w_r^{(n-1)}, w_c^{(n-1)})$ to $(w_r^{(n)}, w_c^{(n)})$. Consider

$$\left(C_r^{(n)}, C_c^{(n)}\right) = \underset{1\le i\le N_r, 1\le j\le N_c}{\arg\max} \left\{ \sum_{u=-w_r^{(n-1)}}^{w_r^{(n-1)}} \sum_{v=-w_c^{(n-1)}}^{w_c^{(n-1)}} R^k(i+u, j+v) \right\}, \quad (4)$$

$$\left(w_r^{(n)}, w_c^{(n)}\right) = \underset{\substack{w_{\min}\le w_r\le w_{\max} \\ w_{\min}\le w_c\le w_{\max}}}{\arg\max} \left\{ B_{w_r,w_c}^{\text{edge}} \chi\left(B_{w_r,w_c}^{\text{non-edge}} < T_l\right) \right\}, \quad (5)$$

where

$$B_{w_r,w_c}^{\text{edge}} = \sum_{u=-w_r}^{w_r} \sum_{v=-w_c}^{w_c} \chi\left(R^k\left(C_r^{(n)}+u, C_c^{(n)}+v\right) > T_g\right),$$

$$B_{w_r,w_c}^{\text{non-edge}} = \sum_{u=-w_r}^{w_r} \sum_{v=-w_c}^{w_c} \chi\left(R^k\left(C_r^{(n)}+u, C_c^{(n)}+v\right) < T_g\right) \quad (6)$$

and χ is an indicator function such that

$$\chi(\varphi) = \begin{cases} 1, & \text{if } \varphi \text{ is True} \\ 0, & \text{otherwise.} \end{cases} \quad (7)$$

T_g and T_l are predetermined thresholds. Note that, given the current window size $(w_r^{(n-1)}, w_c^{(n-1)})$, we find the center of the bounding box $(C_r^{(n)}, C_c^{(n)})$ for all pixels in the image by (4). Then, given the current center $(C_r^{(n)}, C_c^{(n)})$ of the bounding box, we examine all possible window sizes to find the maximum number of strong edges within the window (i.e., $B_{w^r,w^c}^{\text{edge}}$ in (5)) under the condition that the number of weak edges is less than a threshold T_l (i.e., $B_{w^r,w^c}^{\text{non-edge}}$ in (5)). Figure 2 shows the convergence of our alternate optimization process of (4) and (5) for $w_{\min} = 16$ and $w_{\max} = 128$, where our method converges after about the 5th iteration. Note that the center and the size of the bounding box improve after every iterative step. This tells that the center and size of the bounding box are updated cooperatively, which eventually leads to the final convergence. Because our method is based on a binary decision by the threshold, the computation for each iteration is very simple and the convergence is very fast.

For the case of multiple MOOIs in the GoF, after the first MOOI (denoted as MOOI-1) and its bounding box are defined, the bounding box for the next MOOI (i.e., MOOI-2) can be found by repeating the alternating optimization process of (4) and (5). This time, in order not to detect the already-found bounding box again we set the pixel values of the STGM under the predetermined bounding boxes to zeroes before we search for the next MOOI (i.e., MOOI-2). This search process for the next MOOIs and their optimal bounding boxes continues until the sum of all pixel values of the STGM under the bounding box is less than a threshold T_s. After all moving objects and their bounding boxes in the GOF are found, we have P multiple MOOIs from MOOI-1 to MOOI-P.

3. Linear Transformations for Nonuniform Size Reduction

After all MOOIs and their bounding boxes in each GOF are determined, we can reduce the sizes for MOOIs and non-MOOIs nonuniformly. In this paper, to treat the inside and outside regions of the bounding boxes differently and to speed up the size reduction process, linear transformations with different slopes for MOOIs and non-MOOIs are applied. So, to squeeze the original frame of $N_r \times N_c$ to $M_r \times M_c$ ($M_r < N_r$ and $M_c < N_c$) we first apply 1D linear transformations to reduce the number of rows from N_r to M_r. Subsequently, the number of columns is reduced from N_c to M_c to have the squeezed frame of $M_r \times M_c$. Since these sequential 1D reductions for the rows and columns are similar, we describe only the row reduction in this section.

For each row n_r ($1 \le n_r \le N_r$) we need a linear mapping function $S_{\text{row}}[n_r]$ to convert the original row index n_r to $S_{\text{row}}[n_r]$ ($1 \le S_{\text{row}}[n_r] \le M_r$). Depending on the existence of the MOOI at n_r or not, the slope of the linear function $S_{\text{row}}[n_r]$ takes either α_r for the MOOI or β_r for the non-MOOI (see Figure 3). So, the slopes control the amount of the size reduction between the MOOI and

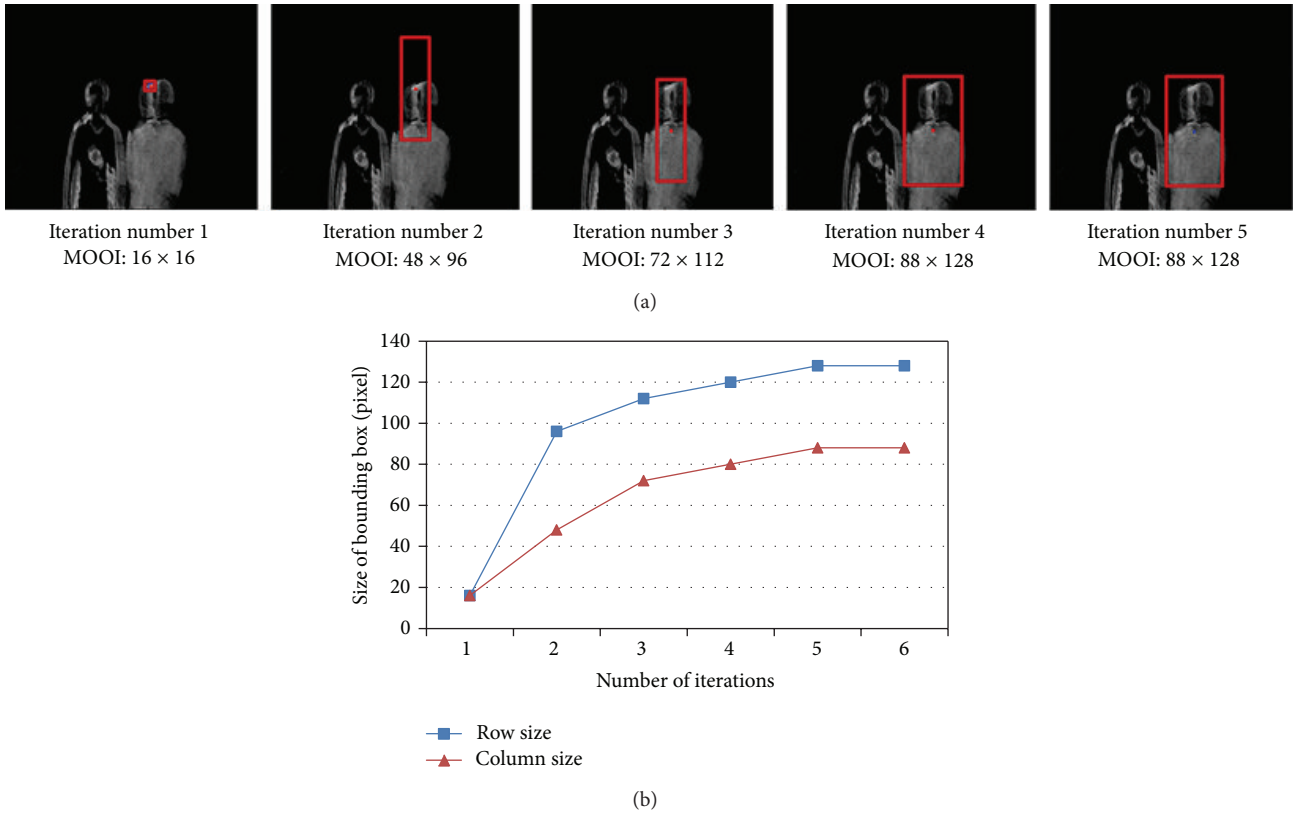


FIGURE 2: Convergence of the alternating optimization process: (a) detected bounding boxes for different iteration numbers and (b) convergence of the alternating optimization method with respect to the size and the number of iterations.

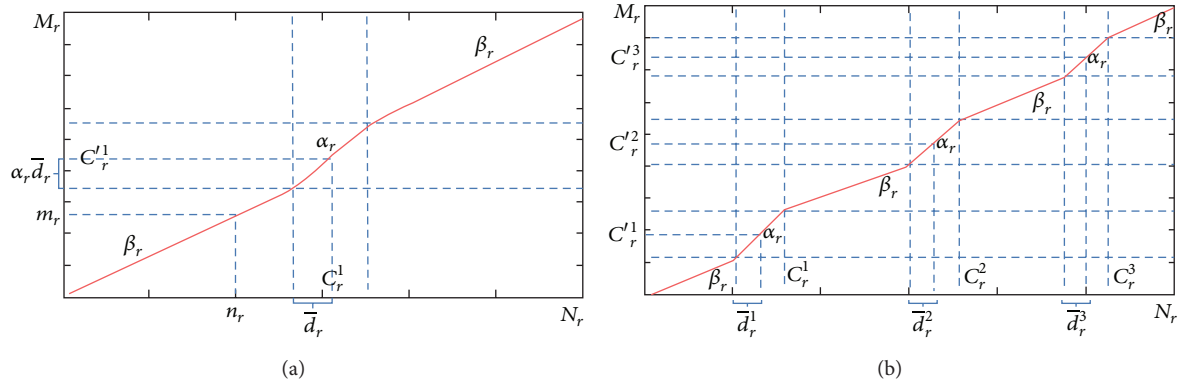


FIGURE 3: Linear transformation for (a) single MOOI and (b) three MOOIs.

the non-MOOI and we have $\beta_r < \alpha_r \leq 1$. Specifically, for the pth MOOI (i.e., MOOI-p), $1 \leq p \leq P$, we denote $d^p(n_r) = |C_r^p - n_r|$ as the absolute distance from the center of MOOI-p, C_r^p, to the row index n_r in the original frame and $\overline{d}_r^p$ represents one-half of the vertical size of the MOOI-p. Also, $C_r'^p$ denotes the row index of the center of the MOOI-p at the reduced frame. Note that the index p of the MOOI-p is assigned sequentially from the left to the right of the image space and we start the linear mapping with MOOI-1 by the following linear transformation $S_{\text{row}}[n_r]$ for each row n_r in $1 \leq n_r < C_r^1 + \overline{d}_r^1$:

$$S_{\text{row}}[n_r] = \begin{cases} C_r'^1 - \alpha_r \overline{d}_r^1 - \beta_r \left(d^1(n_r) - \overline{d}_r^1\right) & \text{if } 0 < n_r < C_r^1 - \overline{d}_r^1, \\ C_r'^1 - \alpha_r d^1(n_r) & \text{if } C_r^1 - \overline{d}_r^1 \leq n_r < C_r^1, \\ C_r'^1 + \alpha_r d^1(n_r) & \text{if } C_r^1 \leq n_r < C_r^1 + \overline{d}_r^1. \end{cases} \tag{8}$$

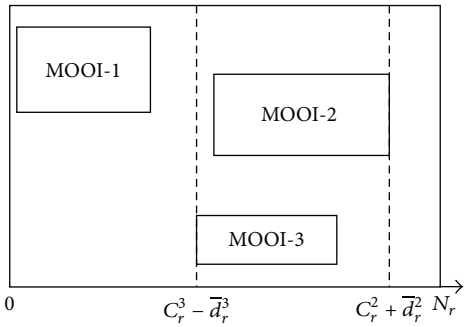


Figure 4: Illustration of the case when the two bounding boxes are overlapped horizontally.

Then, for the next rows in $C_r^p + \overline{d}_r^p \le n_r \le C_r^{p+1} - \overline{d}_r^{p+1}$ with $p \ge 1$ we have the following mapping function $S_{\text{row}}[n_r]$:

$$S_{\text{row}}[n_r] = \begin{cases} C_r'^p - \alpha_r \overline{d}_r^p + C_r'^{p-1} + \alpha_r \overline{d}_r^{p-1} + \beta_r \left(d^p(n_r) - \overline{d}_r^p \right) \\ \quad \text{if } C_r^{p-1} + \overline{d}_r^{p-1} \le n_r < C_r^p - \overline{d}_r^p, \\ C_r'^p - \alpha_r d^p(n_r) \\ \quad \text{if } C_r^p - \overline{d}_r^p \le n_r < C_r^p, \\ C_r'^p + \alpha_r d^p(n_r) \\ \quad \text{if } C_r^p \le n_r < C_r^p + \overline{d}_r^p, \\ C_r'^p + \alpha_r \overline{d}_r^p + \beta_r \left(d^p(n_r) - \overline{d}_r^p \right) \\ \quad \text{if } C_r^p + \overline{d}_r^p \le n_r < C_r^{p+1} - \overline{d}_r^{p+1}. \end{cases} \tag{9}$$

Finally, for the rows in the last MOOI-$PC_r^P - \overline{d}_r^P \le n_r \le N - 1$ we have the mapping function $S_{\text{row}}[n_r]$ as follows:

$$S_{\text{row}}[n_r] = \begin{cases} C_r'^P - \alpha_r d^P(n_r) \\ \quad \text{if } C_r^P - \overline{d}_r^P \le n_r < C_r^P, \\ C_r'^P + \alpha_r d^P(n_r) \\ \quad \text{if } C_r^P \le n_r < C_r^P + \overline{d}_r^P, \\ C_r'^P + \alpha_r \overline{d}_r^P + \beta_r \left(d^P(n_r) - \overline{d}_r^P \right) \\ \quad \text{if } C_r^P + \overline{d}_r^P \le n_r \le N_r - 1. \end{cases} \tag{10}$$

Given the reduction rate α_r for the MOOIs, the reduction rate for the non-MOOI β_r should be determined by considering the overall reduction ratio from N_r to M_r as well as α_r. So, for a single MOOI, β_r is given by

$$\beta_r = \frac{M_r - 1 - 2\overline{d}_r^1 \alpha_r}{N_r - 1 - 2\overline{d}_r^1}. \tag{11}$$

For multiple MOOIs β_r is calculated as follows:

$$\beta_r = \frac{M_r - 1 - 2\overline{d}_r^1 \alpha_r - \sum_{l=2}^{p} 2\overline{d}_r^l \alpha_r}{C_r^1 - \overline{d}_r^1 + N_r - 1 - C_r^p - \overline{d}_r^p + \sum_{l=2}^{p} C_r^l - \overline{d}_r^l - C_r^{l-1} - \overline{d}_r^{l-1}}. \tag{12}$$

The index of the center of the MOOI is also changed from C_r^p to $C_r'^p$ after the reduction. Specifically, the indices for the first MOOI and the next MOOIs are given as the following equations, respectively:

$$C_r'^1 = \beta_r \left(C_r^1 - \overline{d}_r^1 \right) + \alpha_r \overline{d}_r^1, \tag{13}$$

$$C_r'^p = C_r'^{p-1} + \alpha_r \overline{d}_r^1 + \beta_r \left(C_r^p - \overline{d}_r^p - C_r^{p-1} - \overline{d}_r^{p-1} \right) + \alpha_r \overline{d}_r^p. \tag{14}$$

In practice, bounding boxes of MOOIs can be overlapped horizontally and/or vertically. Specifically, we define that bounding boxes are overlapped when their 1D projections are overlapped. This is because 1D linear transformations are applied to rows and columns separately. Figure 4 shows the example when the bounding boxes are overlapped horizontally. To deal with the overlapped bounding boxes, the boundaries of MOOI-2 and MOOI-3 are merged; that is, new left and right boundaries and center are determined as $C_r^3 - \overline{d}_r^3$, $C_r^2 + \overline{d}_r^2$, and $(C_r^2 + C_r^3 + \overline{d}_r^2 - \overline{d}_r^3)/2$, respectively. The row reduction is then performed using (8)–(14) for MOOI-1 and the merged MOOI. The column reduction is performed in a similar manner.

Our goal of the linear transformations is to keep the original image data in the MOOIs as much as possible after the size reduction, while achieving the major size reduction in the non-MOOI. Therefore, we first set $\alpha_r \approx 1$ and adjust β_r ($\beta_r < \alpha_r$) to meet the requirement of the size reduction. After the transformations of (8)–(10) the integer valued indices at the reduced rows are determined by the interpolation from the actual mapped indices. After the row reduction, the transformation-interpolation process is applied to the columns to complete the size reduction.

After the size reduction, the conventional H.264/AVC is used to further compress the size-reduced frames and the compressed bit stream is sent to the receiver. At the receiver, after decoding compressed bit-stream, the decompressed frames are expanded to the original size by the inverse transformation-interpolation for the columns and the rows sequentially. The sizes of bounding boxes, their centers, and the size reduction rate of MOOIs are sent to the decoder as side information for the size expansion. Note that we can use the same GoF boundaries as those from the H.264/AVC.

4. Experimental Results

Our experiments have been conducted to demonstrate two aspects: (i) accuracy of the proposed MOOI detection with

(a) (b)

Figure 5: Comparison for the detection of boundary box: (a) Kalman filtering object detection and tracking [11] (b) the proposed method.

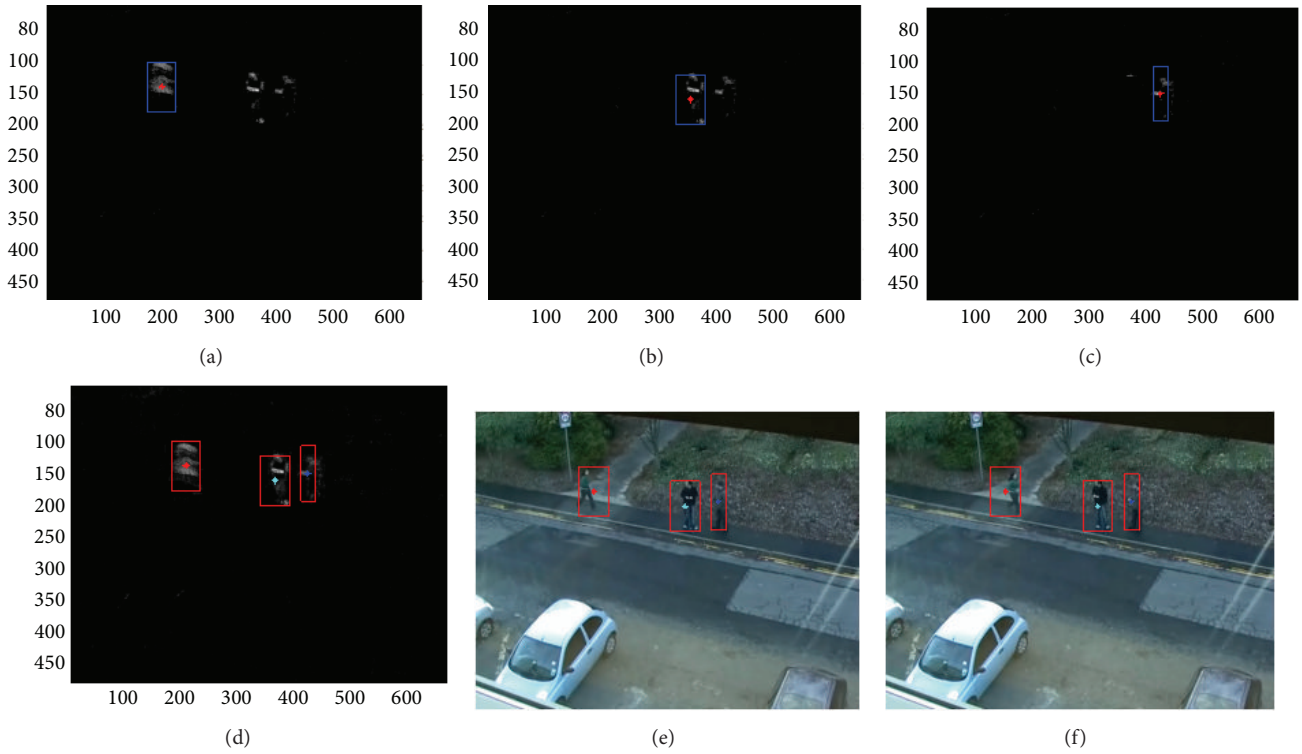


Figure 6: Proposed bounding box detection for multiple moving objects in a GOF: (a) first MOOI (MOOI-1), (b) second MOOI (MOOI-2), (c) third MOOI (MOOI-3), (d) bounding boxes for all MOOIs in the STGM, (e) bounding boxes for the first frame of the GoF, and (f) bounding boxes for the last frame of the GoF.

the bounding box and (ii) usefulness of the proposed MOOI detection. The accuracy of the proposed MOOI detection with the bounding box is judged by visual comparisons with the previous inter-frame based Kalman filtering approach [11]. The usefulness of the proposed MOOI detections is demonstrated by applying the detected MOOI to the content-aware image resizing with the comparison of the LMSE method [18] and to the image size reduction as a preprocessing for the H.264/AVC compressions.

The surveillance video sequences [21, 22] were used to evaluate the performance of the proposed method. In all our experiments, the parameters are predetermined and fixed as follows: $\psi = 3$, $w_{\min} = 16$, $w_{\max} = 128$, $T_l = 0.8(2w_{\max} + 1)(2w_{\max} + 1)$, $T_g = 0.09$, $T_s = (2w_{\min} + 1)(2w_{\min} + 1)/2$. The threshold parameters affect the accuracy of the bounding box detection. The users can interact with the system by adjusting these parameter values. For comparisons, the proposed method and the LMSE method in Vo et al. [18] were applied to reduce the size of the video frames before we apply H.264/AVC. Then, the visual qualities after the H.264/AVC decompression and the size expansion are compared.

(a) (b) (c)

Figure 7: Size reduction by 30% with a single MOOI: (a) original, (b) LMSE [18], and (c) the proposed method.

(a) (b) (c)

Figure 8: Size reduction by 30% with three MOOIs: (a) original, (b) reduced frame by LMSE [18], and (c) the proposed method.

The proposed bounding box detection method is also compared to the moving object detecting and tracking method [11]. As shown in Figure 5, the Kalman filtering method [11] tends to detect only the moving part between the consecutive frames not the whole body of the moving object, which demonstrates the power of our STGM-based formulation of the cost function for a GoF. For the case of multiple MOOIs, Figure 6 shows the order of MOOI detection from the first MOOI (i.e., MOOI-1 in Figure 6(a)) at the leftmost side of the image to the last MOOI (i.e., MOOI-3 in Figure 6(c)) at the rightmost side of the image in a GoF. This demonstrates the extension of our previous work [20] to the problem of multiple MOOIs. Since our bounding box is determined on the basis of the GoF, the first bounding box of the walking person includes all pixels along the motion trajectories from the first frame (Figure 6(e)) to the last one (Figure 6(f)) of the GoF.

Once the MOOIs are detected with the bounding boxes, we can differentiate the image regions of the moving objects from the rest of the image regions of nonmoving objects. This allows us to treat MOOIs and non-MOOI separately for image size reduction. That is, we can nonuniformly reduce the size of the image frames in the video before compressions such that the non-MOOIs are the major target for the size reduction. Frame reductions by 30% and bitrates of 50–200 kbps were tested for the visual comparisons of the MOOIs after the decompressions and size expansions. Figure 7 and Figure 8 show the results of the size reduction by the LMSE in Vo et al. [18] and our method for a single MOOI and three MOOIs, respectively. As shown in the figures, the moving objects are almost intact after the size reduction by the proposed method. Figure 9 for the Stair sequence in the database [21] demonstrates the differences more clearly. As one can see, the proposed method outperforms the LMSE inside regions of the MOOI in terms of PSNR and visual quality. Figure 10 compares the numerical results by the rate-distortion graphs. Although our method yields PSNR slightly lower than the LMSE for the whole image, inside the MOOI regions, it achieves 3~4 dB higher PSNRs than the LMSE and the H.264/AVC compressions without the size-reduction at bitrates lower than critical bitrate of 150 kbps.

5. Conclusions

Optimal bounding box detection method for the moving object of interest (MOOI) has been proposed. Multiple MOOIs as well as a single one can be automatically detected by the proposed method. Once the bounding boxes are identified, one can treat the MOOI and the non-MOOI differently to preserve the visual quality of the important MOOIs. Linear transformations with different slopes are used to nonuniformly reduce the sizes of the MOOIs and non-MOOI. Our size reduction method can be applied as an initial compression for the H.264/AVC video compression standard. Experimental results show that the decompressed videos of the H.264/AVC using the proposed method yield better PSNRs for the MOOI about 3 dB higher than the LMSE.

(a) (b) (c) (d)

Figure 9: 300th frame of Stair sequence: size reduction by 30% and bitrate 100 kbps (MOOI inside the bounding box): (a) original, (b) H.264/AVC without the size reduction, (c) the LMSE [18], and (d) the proposed method.

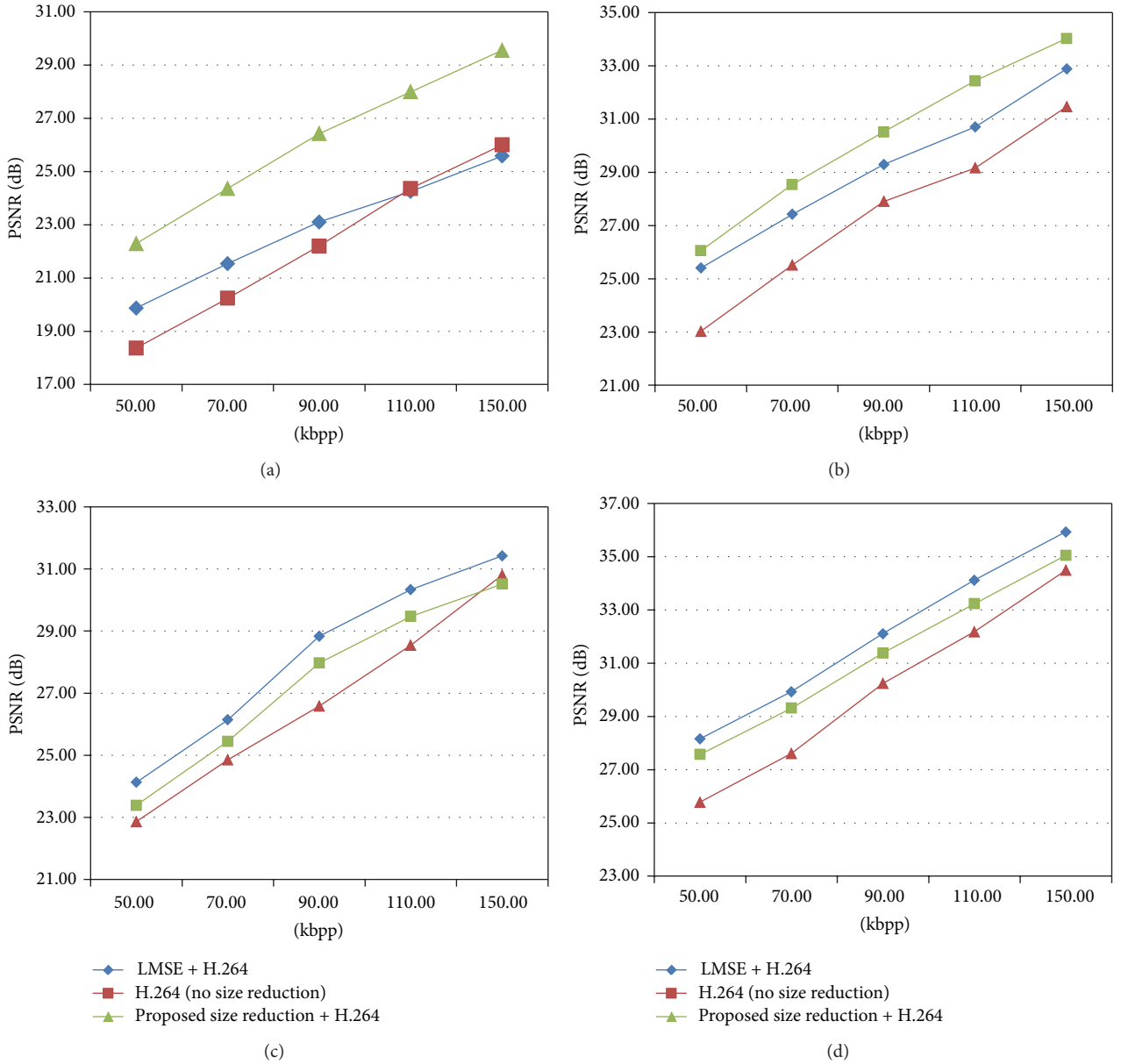


Figure 10: Rate and distortion graphs: (a) Stair sequence for the MOOI only, (b) Hallway sequence for the MOOI only, (c) Stair sequence for the whole image, and (d) Hallway for the whole image.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgment

This research was supported by the MSIP (Ministry of Science, ICT and Future Planning), Republic of Korea, under the ITRC (Information Technology Research Center) support program (NIPA-2014-H0301-14-4007) supervised by the NIPA (National IT Industry Promotion Agency).

References

[1] S.-M. Chang, H. H. Chang, S. H. Yen, and T. K. Shih, "Panoramic human structure maintenance based on invariant features of video frames," *Human-Centric Computing and Information Sciences*, vol. 3, no. 1, pp. 1–18, 2013.

[2] N. Howard and E. Cambria, "Intention awareness: improving upon situation awareness in human-centric environments," *Human-centric Computing and Information Sciences*, vol. 3, no. 1, pp. 1–17, 2013.

[3] C. Shahabi, S. H. Kim, L. Nocera et al., "Janus—multi source event detection and collection system for effective surveillance of criminal activity," *Journal of Information Processing Systems*, vol. 10, no. 1, pp. 1–22, 2014.

[4] S. K. Vipparthi and S. K. Nagar, "Color directional local quinary patterns for content based indexing and retrieval," *Human-Centric Computing and Information Sciences*, vol. 4, no. 1, article 6, 2014.

[5] P. B. Patil and M. B. Kokare, "Interactive semantic image retrieval," *Journal of Information Processing Systems*, vol. 9, no. 3, pp. 349–364, 2013.

[6] D. Venkatraman and A. Makur, "A compressive sensing approach to object-based surveillance video coding," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '09)*, pp. 3513–3516, April 2009.

[7] S. Kim, B.-J. Lee, J.-W. Jeong, and M.-J. Lee, "Multi-object tracking coprocessor for multi-channel embedded DVR systems," *IEEE Transactions on Consumer Electronics*, vol. 58, no. 4, pp. 1366–1374, 2012.

[8] T. J. Kim, B. G. Kim, C. S. Park, and K. S. Jang, "Efficient block mode determination algorithm using adaptive search direction information for scalable video coding (SVC)," *Journal of Convergence*, vol. 5, no. 1, 2014.

[9] X. Yang, G. Peng, Z. Cai, and K. Zeng, "Occluded and low resolution face detection with hierarchical deformable model," *Journal of Convergence*, vol. 4, no. 1, pp. 11–14, 2013.

[10] D. Ghimire and J. Lee, "A robust face detection method based on skin color and edges," *Journal of Information Processing Systems*, vol. 9, no. 1, pp. 141–156, 2013.

[11] S.-K. Weng, C.-M. Kuo, and S.-K. Tu, "Video object tracking using adaptive Kalman filter," *Journal of Visual Communication and Image Representation*, vol. 17, no. 6, pp. 1190–1208, 2006.

[12] K. Goswami, G. S. Hong, and B. Kim, "A novel mesh-based moving object detection technique in video sequence," *Journal of Convergence*, vol. 4, no. 1, pp. 20–24, 2013.

[13] H. T. Nguyen and C. S. Won, "Video retargeting based on group of frames," *Journal of Electronic Imaging*, vol. 22, no. 2, Article ID 023023, 2013.

[14] S. Avidan and A. Shamir, "Seam carving for content-aware image resizing," in *Proceedings of the ACM SIGGRAPH*, 2007.

[15] M. Rubinstein, A. Shamir, and S. Avidan, "Improved seam carving for video retargeting," in *Proceedings of the 2008 ACM SIGGRAPH*, 2008.

[16] M. Grundmann, V. Kwatra, M. Han, and I. Essa, "Discontinuous seam-carving for video retargeting," in *Proceeding of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '10)*, pp. 569–576, San Francisco, Calif, USA, June 2010.

[17] J. Kiess, B. Guthier, S. Kopf, and W. Effelsberg, "SeamCrop: changing the size and aspect ratio of videos," in *Proceedings of the ACM 4th Workshop on Mobile Video (MoVid '12)*, pp. 13–18, February 2012.

[18] D. T. Vo, J. Sole, P. Yin, C. Gomilaan, and T. Q. Nguyen, "Selective data pruning-based compression using high-order edge-directed interpolation," *IEEE Transactions on Image Processing*, vol. 19, no. 2, pp. 399–409, 2010.

[19] C. S. Won and S. Shirani, "Size-controllable region-of-interest in scalable image representation," *IEEE Transactions on Image Processing*, vol. 20, no. 5, pp. 1273–1280, 2011.

[20] A. V. Le and C. S. Won, "Bounding box and frame resizing for moving object of interest," in *Future Information Technology*, vol. 309 of *Lecture Notes in Electrical Engineering*, pp. 445–450, Springer, Berlin, Germany, 2014.

[21] H. Sohn, W. de Neve, and Y. M. Ro, "Privacy protection in video surveillance systems: analysis of Subband-adaptive scrambling in JPEG XR," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 21, no. 2, pp. 170–177, 2011.

[22] S. J. Blunsden and R. B. Fisher, "The BEHAVE video dataset: ground truthed video for multi-person behavior classification," *Annals of the BMVA*, vol. 4, pp. 1–12, 2010.

Hindawi Publishing Corporation
The Scientific World Journal
Volume 2014, Article ID 781234, 12 pages
http://dx.doi.org/10.1155/2014/781234

Research Article

A User Authentication Scheme Using Physiological and Behavioral Biometrics for Multitouch Devices

Chorng-Shiuh Koong,[1] Tzu-I Yang,[2] and Chien-Chao Tseng[2]

[1] *Department of Computer Science, National Taichung University of Education, Taichung 40306, Taiwan*
[2] *Department of Computer Science, National Chiao Tung University, Hsinchu 30010, Taiwan*

Correspondence should be addressed to Tzu-I Yang; tiyang@cs.nctu.edu.tw

Received 5 April 2014; Revised 22 June 2014; Accepted 29 June 2014; Published 24 July 2014

Academic Editor: Young-Sik Jeong

With the rapid growth of mobile network, tablets and smart phones have become sorts of keys to access personal secured services in our daily life. People use these devices to manage personal finances, shop on the Internet, and even pay at vending machines. Besides, it also helps us get connected with friends and business partners through social network applications, which were widely used as personal identifications in both real and virtual societies. However, these devices use inherently weak authentication mechanism, based upon passwords and PINs that is not changed all the time. Although forcing users to change password periodically can enhance the security level, it may also be considered annoyances for users. Biometric technologies are straightforward because of the simple authentication process. However, most of the traditional biometrics methodologies require diverse equipment to acquire biometric information, which may be expensive and not portable. This paper proposes a multibiometric user authentication scheme with both physiological and behavioral biometrics. Only simple rotations with fingers on multitouch devices are required to enhance the security level without annoyances for users. In addition, the user credential is replaceable to prevent from the privacy leakage.

1. Introduction

Owing to the rapid growth of mobile device computation power, personal digital assistants, smart phones, and tablets have become sort of keys controlling our daily life. Most of them provide user friendly interfaces that can be easily operated through fingers and multitouch display. Mobile devices are not only used to make calls, receive messages, take photos, and play games, but also give all kinds of help for both personal business and financial services. Users can transfer money, manage bank accounts, pay for products and game credits using digital money, sell stocks, and even pay vending machines using mobile devices online almost anytime, anywhere. As a consequence, user authentication for mobile devices has become an important issue [1]. User authentication is the act of confirming a person using personal identities, which often involves verifying at least one form of identification. There are three major factors to authenticate users, based on something the user knows (password and challenge response), something the user has (ID, security token, device, and equipment), and something the user is (fingerprint, DNA, and other biometric identifiers). Each authentication factor covers a range of elements used to authenticate a person's identity, which can be used to grant the access authorization, approve a transaction request, and sign documents.

The authentication on mobile devices can currently be classified into three major approaches. PIN (personal identification number) or passwords, the secret-knowledge approach, are the most popular authentications with the features of quick operation and low cost. Financial PINs are often four-digit numbers in the range 0000–9999, resulting in 10,000 possible numbers. However, some banks do not give out numbers where all digits are identical, consecutive, numbers that start with one or more zeroes, or the last four digits of your social security number. Although a more complicated approach named two-factor authentication [2] uses SMS combined with the one time password (OTP) user authentication scheme that is widely used by leading commercial companies, it still may suffer from the phishing

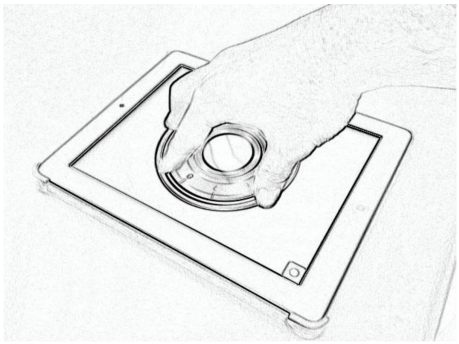


Figure 1: An example of pbLogon.

attacks [3]. On the other hand, passwords seem alternatively more secure because of the more possible combinations by using all symbols and alphabets. Unfortunately, people always use the same password everywhere and rarely change it. Although the security level can be enhanced through forcing users to change password periodically, it may also add annoyances for users. On the other hand, sharing passwords and phishing attacks are serious problems that happen frequently in our daily life [3, 4]. Phishing attack is the act of attempting to steal sensitive information, such as passwords and credit card details (knowledge factor), by masquerading as a trustworthy entity in an electronic communication [4]. Phishing attack is typically performed through email spoofing [5], instant messaging [6], and SMS services [7]. It often leads users to enter personal information on a fraudulent website, which makes the user look and feel the same as the legitimate one. Although several antiphishing technologies were revealed against these malicious behaviors, it still needs user training and public awareness to make it work.

The second approach is the SIM (subscriber identification module), so-called token-based system, which is an integrated circuit that securely stores the IMSI (international mobile subscriber identity) and the related keys used to identify and authenticate subscribers on mobile telephony devices. A SIM is embedded into a removable SIM card, which can be transferred between different mobile devices. It is usually used for small payments, such as vending machines and game point cards. However, removing the SIM is not recommended because it would cause the loss of signal and other inconvenient manners.

The last one is authentication through biometric characteristics, which are unique enough to distinguish each person. The development of the biometric authentication technology has the trend of replacing the traditional verification method and can solve the traditional security problems. Biometric approaches are typically divided into two categories: physiological and behavioral biometrics. Physiologic biometrics refer to physical measurements of the human body, including face, fingerprint, hand geometry, retina, and iris (please refer to Figure 1). The recognition system based on physiological characteristics has a relatively high accuracy [8, 9]. However, the fingerprint of those people working in chemical industries is often affected. On the other hand, people affected with diabetes, the eyes also get affected resulting in differences. In addition, the use of physiologic biometrics introduces privacy concern since the body characteristics are irreplaceable [10].

Behavioral biometrics relate to a specific behavior of a human while performing some tasks, such as handwriting, speaking, and typing [11, 12]. Usually, handwriting recognition used signature as identity, which means it is not suitable for general-purpose authentications [13]. Voice biometric authentication uses the voice pattern to verify the identity of the individual. However, automatic speaker authentication systems may be affected by the extreme emotional states, sickness, and aging of the speaker and noise [14]. Keystroke dynamics [15] is considered of the most successful behavioral biometrics with the benefit of almost free as the only hardware required is the physical keyboard. Users' keystroke rhythms are measured to develop a unique biometric template for future authentication. However, a person's hands can also get tired or sweated after prolonged periods of typing which resulted in major pattern differences [16]. In addition, typing patterns may vary based on the keyboard layout, the person's posture and language dependency. On the whole, both physiological and behavioral biometrics approaches require different equipment for extracting the characteristics for verification, which may not be portable and can be expensive.

This study proposes a novel scheme pbLogon (physiological and behavioral user authentications), which combines both physiological and behavioral biometric characteristics. The multitouch panel is the only equipment required, which is built in almost every modern mobile device. It aimed to provide a strong user authentication environment, which uses at least two authentication factors, also suggested and grounded by different research [1, 17].

2. Related Works

2.1. Physiological Biometrics. Physiology is the characteristic of the body and thus it varies from person to person, including fingerprint, hand geometry, face, and iris and retina recognition. The fingerprint [18] is using patterns which are aggregate characteristics of ridges and minutia points. It provides an over 99% recognition accuracy that is widely used by governments and leading industries [19]. The palm print technology [20] can be considered the same despite of the scale size being different. A face recognition technique [21] is applications that identify or verify a person automatically from a digital image or a video frame from a video source. It is the most natural mean of biometric identification. Facial metric technology relies on the manufacture of the specific face recognition feature, such as the position of eyes, nose and mouth, and distances between these features. Face recognition may suffer from the rise of wrong identifications owing to the surrounding environment and lighting affecting the quality of images acquired [22]. As for the iris technology [23], it uses the colored area that surrounds the pupil. Iris patterns are unique, which can be a combination of specific

characteristics known as the corona, crypts, filaments, freckles, pits, furrows, striations, and rings. As for retina geometry technology [24], it is based on the pattern of blood vessel in the retina that has unique patterns from person to person. Hand geometry technology [25] is based on the fact that nearly every person's hand is shaped differently and that the shape of human hands does not change after a certain age. These techniques include the estimation of length, width, thickness, and surface area of the hand. Essentially, hand identification approaches can be classified into two categories based upon the nature of image acquisition: contact-based and contact-free. With the contact-based approach, users are often asked to place their hands on a flat surface or a digital scanner. Recently, the contact-free approaches are increasingly being considered because of their characteristics in user acceptability, hand distortion avoidance, and hygienic concerns [26]. Besides, more information can be obtained since contact-free approaches can obtain both 2D and 3D hand geometry information [27]. Both of them need extra equipment, which is not portable. In our proposed scheme, we introduce a novel hand geometry technique, which is the contact-based but measures the relative position of each finger instead, since the researchers [8, 25, 26, 28] stand guarantee for hand geometry. In addition, more unique features such as the natural pose, rotation angle, and polygon area of different fingers, which come by using the touch panel in advance.

2.2. Behavioral Biometrics. Behavioral biometrics are the behavioral characteristics that related to the pattern of people doing something, such as signature, typing rhythm [29], gait [11], and mouse movement [30]. The signature recognition is based on the dynamics of making the signature, rather than a direct comparison of the signature itself afterwards. The dynamic is measured as a mean of the pressure, direction, acceleration and the length of the strikes, and dynamic number of strokes and their duration. A keystroke dynamic is based on the assumption that different people have unique habitual rhythm patterns in the ways they typed and analyzed using statistical technique traditionally. By introducing the pattern recognition technique, such as z-test, Bayesian classifiers, and neural network, they have brought the recognition rate to a new level [15, 29]. Hence, the analysis of keystroke becomes one of the most useful authentication schemes because it is based on the user's experience and individual skills. However, people who use different input methods such as phonetic may suffer from the language problem which also causes the embarrassments of detecting biometrics. Recently, a few studies have considered the keystroke dynamics of mobile devices, which have investigated the keystroke recognition using the virtual keyboard [31]. However, Clarke and Furnell [32] investigated the feasibility of authenticating users based on their typing habits using the neural network showing that only partial participants' characteristics can be discriminated. Furthermore, most of the biometric approaches require additional equipment to verify these biometric characteristics, which may also increase the manufacturing cost.

2.3. Gesture-Based User Identification. Gesture-based user identification uses human body gestures and gaits to recognize the user. Researchers use different equipment, such as accelerometer [33], video [34], and Kinect [35], to track the patterns while human walking or performing poses in different ways. The benefits can be the easy operation while performing user authentication. However, extra equipment may be required to perform user login, which means it may not be portable and not suitable for daily use.

Sae-Bae et al. [36] advocated that using only hand gestures acquired by multitouch panels can reach a 90% accuracy rate with a single gesture. They use the hand operational features, which include parallel, closed, opened, and circular hand movements. However, how to normalize these operations can be relatively difficult and hard to detect if the hand moving area can be varied case by case while using different Apps. Therefore, this study introduces pbLogon, which provided the carrier, virtual wheel lock, to limit the operation area of users and further increase the usability and raise the accuracy while acquiring users' biometric information. Besides, more analysis can be carried out through recording the operational force that is provided by the built-in gyroscope. In addition, resizing the virtual wheel lock can bring the benefit of revocable biometric templates.

2.4. Biometric Privacy Concerns. One disadvantage of biometrics is that they cannot be easily revoked. Physiological biometrics is generally irreplaceable which means it may suffer from the privacy issue [37, 38]. Although some research provides a more advanced protection to prevent from the privacy leakage of user template, it may also increase the complexity and power consumption of mobile devices [39, 40].

Another serious problem is the irreplaceable of biometric characteristics. Traditionally, while the user account has been compromised, the passive way is to ask the user to change password, and the more active way is to change the layout of keyboards that prevent from further remote stolen. However, the extraction of new biometrics can be limited because of the quantity limitation, such as ten fingerprints. As a consequence, it is also important to provide both revocable and replaceable biometric authentication schemes. This study proposes a novel biometric authentication scheme, which includes the features of both physiological and behavioral biometrics so-called pbLogon to solve the problems mentioned above. It aims to build a multifactor user authentication system, which is a strong user authentication, biometric-based, and replaceable as a privacy concern.

3. pbLogon Scheme

Figure 1 brings the example of pbLogon, which uses an Apple iPad2 as the equipment for gathering personal biometrics. The iPad2 has a multitouch panel, which can track up to 5 fingers simultaneously. The pbLogon system will only react while user performs rotations on the virtual wheel lock area. Users can input their credentials by rotating the wheels either clockwise or counterclockwise. Then both physiological and

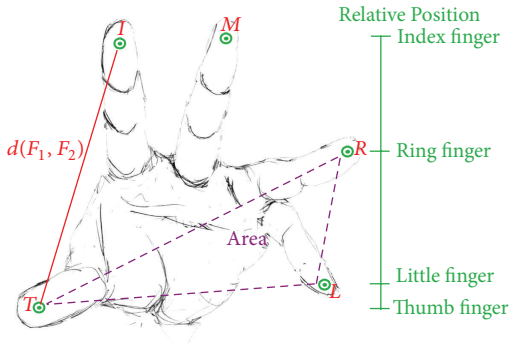


Figure 2: Example of hand's physiological biometric.

behavioral biometrics can be obtained through these operations.

3.1. Extraction of Physiological and Behavioral Biometrics. It is relatively easy to gather users' physiological biometric information using touch panels rather than traditional optical devices. The physiological phase extracts the finger information, which may include the relative position, distance of different fingertips, and area of each three fingers (please refer to Figure 2). To correctly compare any two biometric templates, they need to be acquired and stored in a consistent order.

Hence the first step is to reorder the touch sequences into a canonical form. The standard order employed was the touches generated by thumb, index, middle, ring, and little fingers. It is hard to determine the right sequences because the acquisition process may capture points in an arbitrary order depending on which fingertips made contact with the touch panel first. To correctly match touch sequences with fingers we use known natural characteristics of human hand geometry. First, we sort the acquired data with x-axis in ascending order. Then we check the y-axis for the thumb, which is located at the lowest position in a natural pose. The corresponding order then can be determined by comparing the thumb to index and little fingers. More detailed information will be provided in Algorithm section. By examining the information the user provided, it can easily collect the user's sketch and the composition of his fingers. In addition, whether index finger is longer or the ring finger is decided by personal DNAs which can also be used to identify the user.

In behavioral phases, analyzing the rotational dynamics can reveal more behavioral information. Figure 3(b) brings the example of a left-handed user with pbLogon operations. A left-handed user can rotate more in the clockwise direction rather than the counterclockwise (Figure 3(c)). On the other hand, a left-handed user may also rotate much faster for clockwise than counterclockwise direction. In addition, tracking the moving speed and path may also help identify the user. Both phases provide rich information to decide whether the user with corresponding ID and password is the compromised one or not. On the other hand, a virtual wheel lock is adapted to limit the operational area while using pbLogon. The main purpose is to help the biometric extraction engine for gathering biometric characteristics more accurately. If the user touches outside of the virtual wheel lock, then pbLogon will not start the extraction process and will prompt the user to put fingers on the wheel lock to input passwords.

In our proposed system, the identity of the user is given to the system along with a proof of the biometric; that is, only the biometric information is used for user authentication. Correctness of the identity is then evaluated by the system; passwords are not involved in the authentication process. After that, either accepting or rejecting the user is given based on the evaluation result. In order to verify the proof, the system needs to have a prior knowledge, for example, the user profile. Generally, there exist two stages in a user authentication system: enrollment and verification stages. The purpose of enrollment stage is to register the users' data in the system by acquiring, extracting and storing biometric templates corresponding to the user. In the verification stage, the input biometric instance is compared with the stored biometric templates for the claimed identity in order to authenticate a user.

3.2. Notation. The notations used throughout this paper are listed as Notations section. *Password* is composed of n-digits numeric password, for example, PIN, which is given by rotating the wheel lock displayed on screen. N is the total number of fingers that you put on the touch screen. G is the function calculating the number of corresponding fingers F_i acquired by touch panel, which means $N = G(F_i)$. It also detects the hand pose. D is the set of the relative distance of each two fingers, for example, $D = \{d_{i,j} \mid d_{i,j} = d(F_i, F_j), \forall 0 < i \neq j < 6\}$. A is the set of areas that formed by any three fingers, for example, $A = \{a_{i,j,k} \mid a_{i,j,k} = \text{area}(F_i, F_j, F_k), \forall 0 < i \neq j \neq k < 6\}$. L indicates the relative length of index and ring fingers; if an index finger is longer than ring fingers, L will be one, and otherwise it will be zero. R records whether the user is left-handed or right-handed. V is the set of velocity of the user performing wheel lock for digits, which also contain the information about the rotational direction, for example, record the counterclockwise rotations with negative values.

3.3. Assumptions. There are a few assumptions to proceed with the experiments and the usage of pbLogon. First the user is willing and wants to log into the system. Second, only natural hand poses and normal operations are accepted. Natural hand poses can be detected through the horizontal angle formed by the thumb and little fingers. Third, pbLogon only accepts and verifies the input by handheld poses. The main reason is that with handheld poses, more behavioral biometric information can be obtained through accelerometer and gyroscope. Fourth, pbLogon only allows the hand, the user registered, to login pbLogon. Uncooperated operations are prohibited and considered misuses and attacks.

3.4. Algorithms. Several algorithms were proposed to restore the hand pose and extract the physiological and behavioral biometrics.

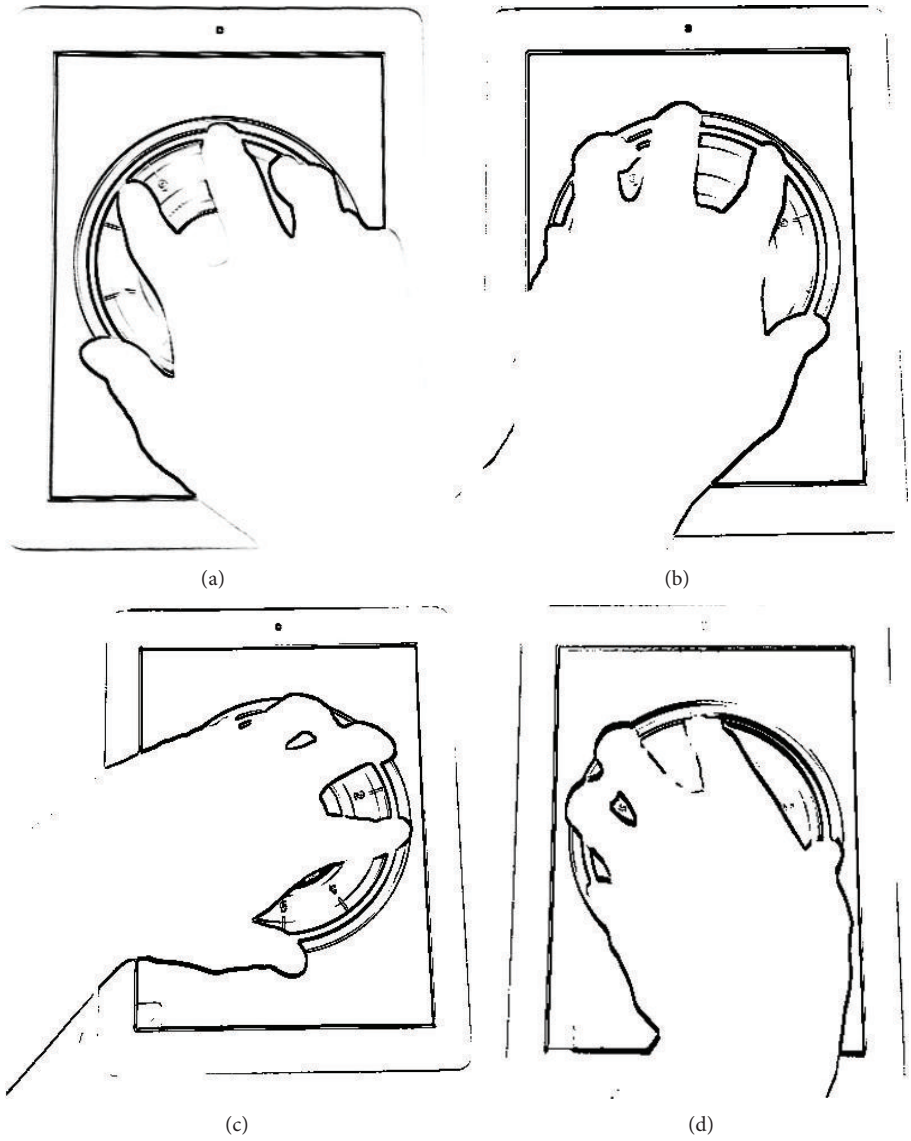


Figure 3: An example of behavioral biometrics. (a) Presents a right-handed user. (b) Presents a left-handed user. (c) Presents a left-handed user with clockwise rotation. (d) Presents a left-handed used with counterclockwise rotation.

3.4.1. Tablet Orientation Detection. In order to obtain the dynamic orientation while user performing the virtual wheel lock, the tablet orientation detection is required to check if the user inputs their credential by holding the tablet. The gyroscope is a modern piece of equipment that can report the device orientation and is widely built in most of the handset devices. The DeviceOrientationEvent provided by HTML5 and JavaScript is adapted to obtain the orientation information of users' tablet (Algorithm 1 and Figure 4). pbLogon will check the beta factor, which can be used to detect whether the user holds the tablet or puts it on the table.

3.4.2. Hand Natural Poses Restoration. Hand nature poses are defined with fingers in the sequence of thumb, index, middle, ring, and little finger. During the experiment, several different poses were shown in Figure 5. Figure 5(a) demonstrates the idea of natural hand pose in the sequence of thumb, index, middle, ring, and little finger. However, for the experimental participants of this dissertation, most of them operate pbLogon using Figure 5(b) pose, that is, with the sequence of index, thumb, middle, ring, and little finger. Therefore, we proposed the hand pose restoration algorithm to handle different types of hand acquired (Algorithm 2).

3.4.3. Check the User Is Left-Handed or Right-Handed. Since it is required to calculate the relative distance of each two fingers, the left-handed or right-handed user must be separated to obtain the correct finger orders. Algorithm 3 gives the check right-hand algorithm. It starts with the sorting of F_i by x_i in ascending order. Then it is required to detect whether the thumb is in ideal position by finding $\max(y_i)$. Let A be the array of F_i; if the leftmost of the sorted A is the thumb, then it is the ideal natural hand pose. Otherwise, it is required to shift the thumb to leftmost or rightmost for

```
Input: DeviceOrientationEvent  //By JavaScript
Output: alert()
(1) assign beta = event.beta; //event is provided by JavaScript
(2) if Math.round(beta * 10)/10) < 10 then
        alert ("please hold the tablet to start input procedure");
(3) else
(4)     start physiological biometrics extraction;
```

Algorithm 1: Check orientation algorithm.

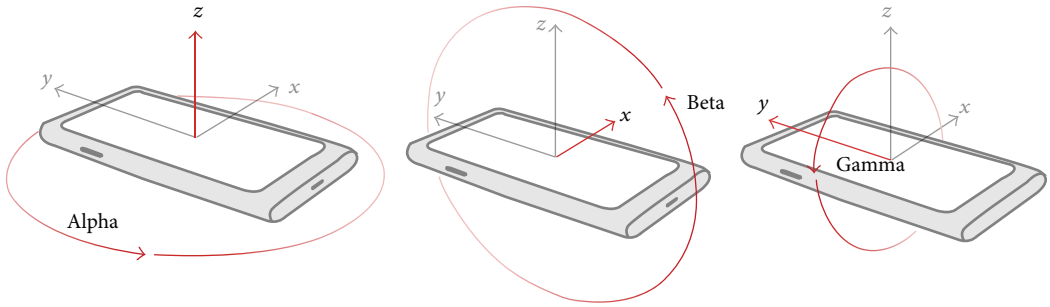


Figure 4: The device orientation definition.

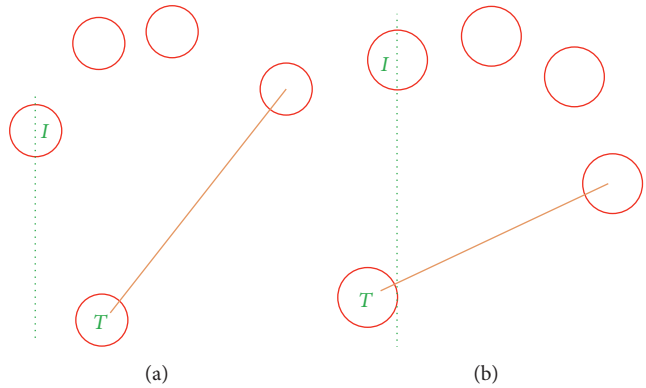


Figure 5: (a) Natural pose with the fingers in order, (b) natural pose with different thumb position, and natural pose with left-handed side user.

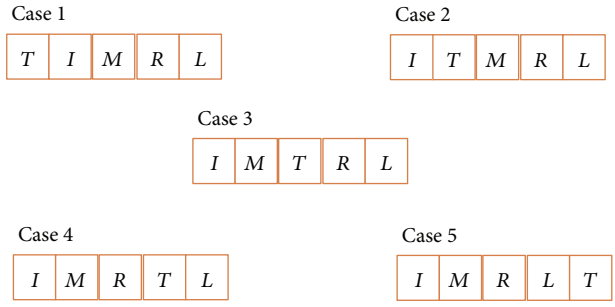


Figure 6: The combinations of thumb in different positions.

checking hand-side. There will be five cases with the thumb in different positions; please refer to Figure 6. After determining the fingers relative position, the relative distances of thumb to index and little finger are used to estimate the hand-side. It will be a little bit longer while comparing the distance of thumb to little finger than thumb to index finger if the user poses his hand in natural hand pose. Participants of this dissertation all tally with the phenomenon, and it stands until the hand operates over 90 degrees. By comparing the sorted A, if thumb to leftmost finger is longer than thumb to rightmost finger, then the system will return 0, which means that the input F_i is a left-handed user. Otherwise, it will return 1, which means that the input F_i is a right-handed user.

3.4.4. Build the User Profile. When MU wants to use P, MU must perform the enrollment for building U_i. At the enrollment stage, our system starts with checking the number of MU's fingers.

Step 1 (MU $\rightarrow$ P : F_i). Then $G(\cdot)$ will compute the F_i and return N whether MU poses hand in natural or not. It

Input: $F_i = \{(x_i, y_i) \mid 0 < i < 6)\}$ //F_i is the user input fingers
Output: array A //array with fingers in order
(1) **sort**(F_i) by x_i in ascending order
(2) assing $t = k$, which **max**(y_k) of F_i //Detect thumb by finding the maximum position of y-axis
(3) assing r = **CheckRight**(F_i); //check user is left-hand
(4) **if** r = "right-hand" **then** shift $A[t]$ to $A[1]$;
(5) **else** shift $A[t]$ to $A[5]$;

Algorithm 2: Hand natural pose restoration.

Input: array $R = \{r_j \mid r_j = (x_j, y_j), 0 \le j \le 4\}$ //R is non-order finger coordinate of the user input
Output: RHS //RHS is the right-hand side of the input user
(1) **sort**(F_i) by x_i in ascending order
(2) assing $t = k$, which **max**(y_k) of F_i //Detect thumb by finding the maximum position of y-axis
(3) **switch** t:
(4) **Case 0:** //if the $R[0]$ is thumb (Figure 6, Case 1)
(5) assign $F_i = 1$; //$R[1]$ is the coordinate of index finger
(6) assign $F_l = 4$; //$R[4]$ is the coordinate of little finger
(7) **break;**
(8) **Case 4:** //if the $R[4]$ is thumb (Figure 6, Case 5)
(9) assign $F_i = 3$; //$R[3]$ is the coordinate of index finger
(10) assign $F_l = 0$; //$R[0]$ is the coordinate of little finger
(11) **break;**
(12) **Case 1, 2, 3:** //if $R[1]$, $R[2]$, or $R[3]$ is thumb (Figure 6, Case 2, 3, and 4)
(13) assign $F_i = 0$; //$R[0]$ is the coordinate of index finger
(14) assign $F_l = 4$; //$R[4]$ is the coordinate of little finger
(15) **break;**
(16) **if** $d(F_t, F_i) < d(F_t, F_l)$ **then** output the user is left-hand
(17) **else** output the user is right-hand

Algorithm 3: Check right-hand algorithm.

is important that more fingers will bring higher reliability and sturdy biometric information. During the enrollment stage, the natural position of the user's hand is also another important issue. By analyzing the relative position of the thumb and little finger, we can identify whether the user's hand poses in nature or not. It is natural and comfortable if you put your hands touch panels with little finger higher than the thumb in horizontal (please refer to Figures 3(a) and 3(b)). If both requirements are met, pbLogon will start to extract the physiological biometrics including D, A, L, and R and then ask MU start to input pass as demands.

Step 2 ($G(F_i) \rightarrow U_i = \{D, A, I, R\}$). D of each two fingers can be calculated through the Euclidean distance formula as follows:

$$D = \left\{d_{i,j} \mid d_{i,j} = d\left(F_i, F_j\right), \forall 0 < i \neq j < 6\right\},$$
$$d\left(F_i, F_j\right) = \sqrt{\left(X_i - X_j\right)^2 + \left(Y_i - Y_j\right)^2}. \quad (1)$$

A of each three fingers can be calculated through the area formula as follows:

$$A = \left\{A_{i,j,k} \mid A_{i,j,k} = \text{area}\left(F_i, F_j, F_k\right), \forall 1 \le i \neq j \le 5\right\}$$
$$S = \frac{d\left(F_i, F_j\right) + d\left(F_j, F_k\right) + d\left(F_k, F_i\right)}{2}$$
$$a\left(F_i, F_j, F_k\right) = \sqrt{S\left(S - d\left(F_i, F_j\right)\right)\left(S - d\left(F_j, F_k\right)\right)\left(S - d\left(F_k, F_i\right)\right)}. \quad (2)$$

For each time MU enters a single digit, P will obtain the behavioral biometrics and calculate the U_i of MU.

Step 3 (MU $\rightarrow$ P : n-digit password, F_i). Consider the following:

$$G\left(F_i\right) \longrightarrow U_i \cup \{V, B\}. \quad (3)$$

By rotating all the n-digits into pbLogon, U_i will be established with the procedure Steps 1 to 3 repeatedly. And finally we will have $U_i = \{F_i, D, A, L, R, V, B\}$.

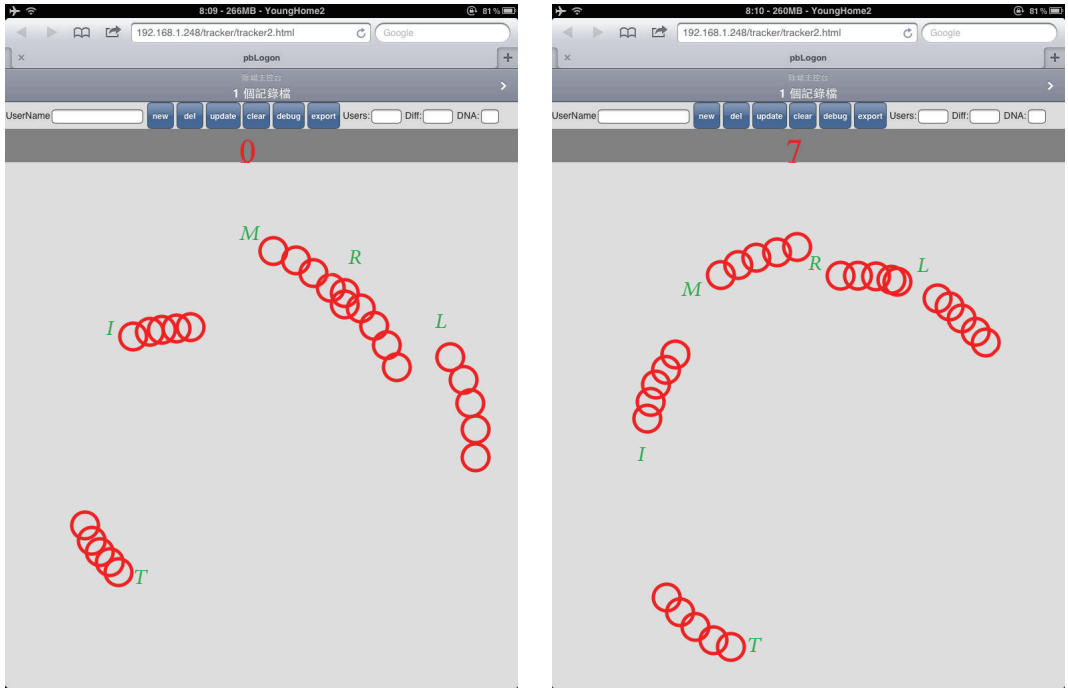


Figure 7: The rotational dynamics.

During the user entering *password*, we can explore even more behavioral biometrics, such as the user is right-handed or left-handed and the rotational dynamics (please refer to Figure 7). A left-handed user can rotate more angles in the direction of counterclockwise. The velocity and rotation dynamics can easily be analyzed from; one may prefer to enter his entire *password* by using the same direction by fingers leaving P while the others may use bidirection to finish the job. We believed that there exist more patterns that we can analyze in the future works.

3.5. Dissimilarity Score. The decision of users' biometrics depends on the similarity of the input biometrics provided by MU and the stored template U_i. In other words, if the dissimilarity score of the input biometric compared to the template is lower than predefined thresholds, the input biometric is verified. Otherwise, the system will reject the user. To calculate the dissimilarity score ΔS between the registered user's templates and the input, all distances between the coming gesture and templates are used to calculate the dissimilarity score along with the distances between all the stored templates themselves. For each feature FEA_i, there will be low-bound LB_{i1} and up-bound UB_{ik} to determine how similar the user is, and the dissimilarity score is calculated by

$$\Delta S = \sum \left| D'_{i,j} - D_{i,j} \right|, \quad \forall 0 < i \neq j < 6$$

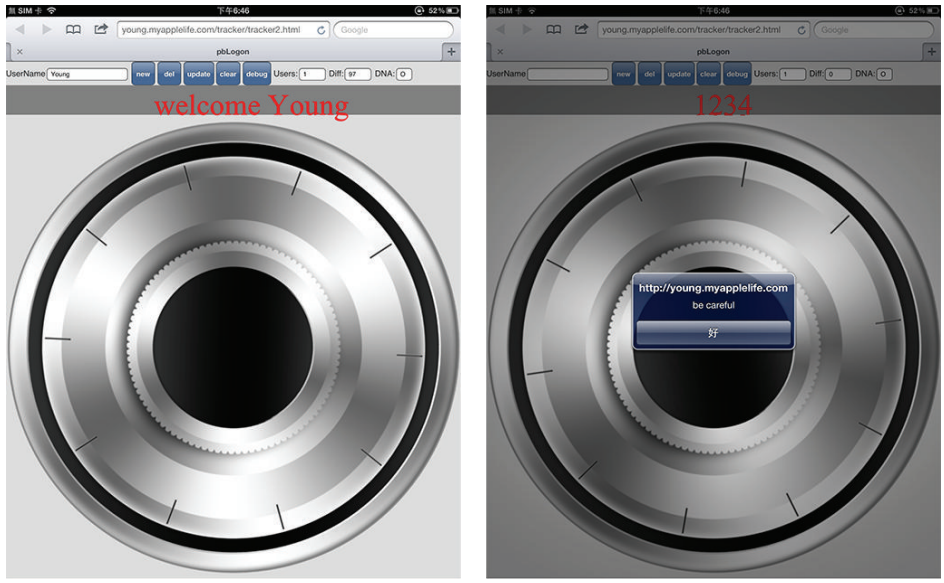
$$D_{i,j} = \left(H_i, C_k, \mathrm{FEA}_i\right) = \begin{cases} H_i \quad c_1 \quad \langle \mathrm{LB}_{i1}, \mathrm{UB}_{ik} \rangle \\ \qquad \vdots \qquad\qquad \vdots \\ \qquad c_k \quad \langle \mathrm{LB}_{ik}, \mathrm{UB}_{ik} \rangle, \\ \text{for } 0 < i \neq j < 6. \end{cases} \tag{4}$$

4. Experiments and Discussions

4.1. Experiment. Forty-three participants are involved in the experiment; they are university students from of centeral Taiwan. An HTML5 web application is developed for experiment; iPad2 is adapted which has the ability to track up to 5 fingers simultaneously. As a visualization aid, the application provides simple visual traces of the user's rotations and fingertip movements (Figures 7 and 8). In each session, we first ask the user to input their student ID by rotating the wheel lock in Figure 8 twice. The experiment lasted for one month, and every participant was asked to input ten times (20 records in total).

4.2. Analysis of Biometric Data. Figure 9 brings the experimental results. If more than four fingers were adopted, most of the classify scheme can reach a near 90% success rate. One of the reasons may be the control difficulties of using only three fingers that lead to the overvariation of fingers' distance and areas. Therefore, it is suggested to use as many fingers as possible to bring a higher recognition and success rate. The experimental results show that with five fingers combined with area characteristic can reach a 95% successful login rate (Figure 9). Another finding is that the rotational velocity may change according to how familiar the user is with pbLogon, so it may not be useful as expected to help recognize users.

4.3. FAR and FRR Analyses. The performance of user authentication system may involve different criteria that sometimes it is more important to consider the false reject rate (FRR) and false acceptance rate (FAR). Figure 10 shows the FAR and EER trends. The x-axis represents the different variance



(a) (b)

Figure 8: Example of pbAuth. (a) Presents the user logon successfully with a welcome message. (b) Presents the user does not make it due to the threshold we made.

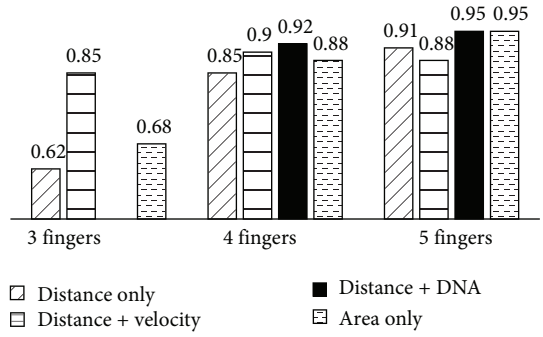


Figure 9: The percentage of successful login rate.

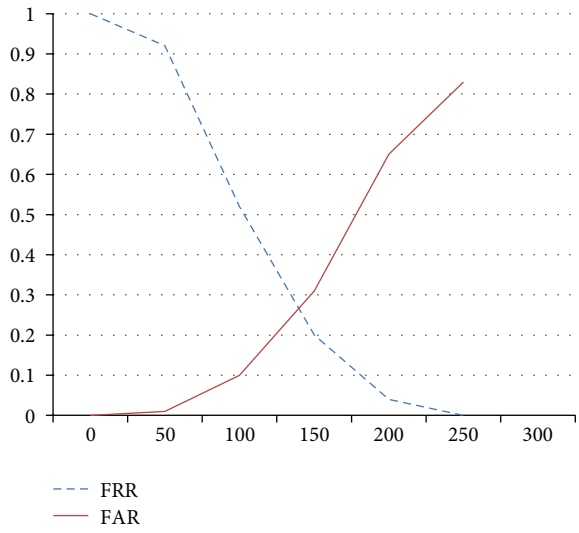


Figure 10: The FRR and FAR diagrams.

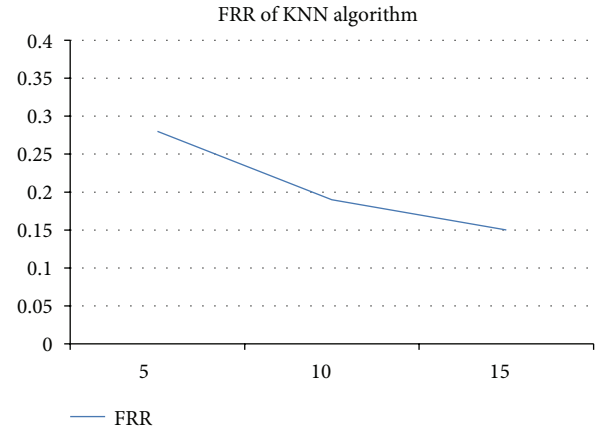
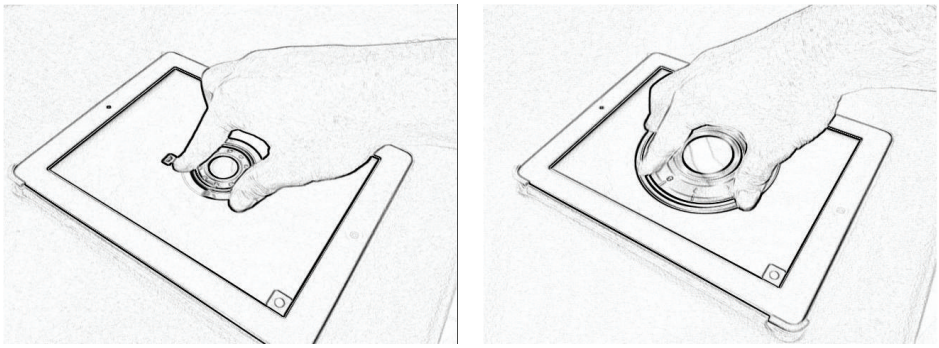


Figure 11: The experimental analysis using KNN algorithm.

as thresholds, and y-axis is the corresponding rates. While the threshold setup is over 100 pixels, the false acceptance rate will increase up to 10%, which may be the barrier if the supervisor needs relatively high security environment. The FRR may also reveal the potential problems that users may need to perform more times to log into the system if the improper threshold is chosen. The equal error rate (EER) is about 26.8%, which means the performance may not be good as expected. However, it depends on usage of different scenarios and limitations.

4.4. Analysis Using k-Nearest Neighbors Algorithm. The k-nearest neighbors algorithm is also adopted to evaluate the physiological biometrics. By the suggestion of SPSS, 80% of the user logs were used for training and 20% for prediction. Figure 11 shows the analytic results using KNN



(a) (b)

Figure 12: Example of different identities of the same user.

module provided by SPSS. The x-axis is the number of records adopted for KNN analysis, and y-axis is the FRR. The result shows that the more the user operations are, the lower the FRR will be. And ten records can reduce the FRR to 20%, which is also similar to the experimental results of pbLogon. However, it is more complex and requires more computation power to perform KNN analysis. pbLogon has the benefit of quick response and portable to handheld devices for power saving propose.

4.5. Other Factors May Influence the Accuracy of pbLogon. Several additional factors influence the effectiveness of pbLogon biometrics: (1) the human hand is flexible object and the projection of its finger may suffer nonlinear deformations when multiple finger positions are acquired from the same person. That is especially true when users are untrained or noncooperative or are fooling the system. Improper thumb placement and little fingers that would not straighten were found by [41] to generate statically significant differences in matching scores. Mobile devices of different touch screen size may also suffer the similar problem. To handle this kind of problem, in the proposed scheme, we can slightly resize the images of virtual wheel lock to intimate the user to expand or narrow their fingers since we train and recognize by the relative position of the fingers. The natural pose restoration algorithm is also provided to extract the reliable hand pose for both training and verification purposes. (2) The sweating hand and emotional states will not affect the verification of pbLogon since it provides both physiological and behavioral biometric extractions. pbLogon also controls the user behavior by providing a fixed touching area and rotational speed. Therefore, if the user is not willing to login with the provided touching will be considered misuse and attacks. (3) The environment requirement of pbLogon system is only a multitouch device. Since only the touch coordinates and fingers' distance are required to perform user authentication, pbLogon has the ability to prevent other environmental facts, such as lights, temperatures, and other biases.

Chen et al. [42] showed that hand shape systems are vulnerable to spoof attacks. They build fake hands out of silhouette images captured by a HandKey II hand geometry reader and hand them to be accepted by the system. In the proposed scheme, we introduce the behavioral biometrics, which were implemented by using rotation dynamics which is alternatively hard to imitate.

4.6. Privacy and Replacement Issue. It is commonly known that the biometric trait of a person cannot be easily replaced. Once a biometrics is ever compromised, it would mean the loss of a user's identity forever. Therefore, protecting the biometric templates is a major concern and also a challenging task [43]. Cancellable biometrics is a way in which the biometric template is secured by incorporating the protection and replacement features into biometrics. A good cancellable biometrics formulation must fulfill four requirements: (1) diversity: the same cancellable template cannot be employed in two different applications; (2) reusability: straightforward revocation and reissue in the occurrence of compromise; (3) one-way permutation: implement nonreversible template calculation to avoid recovery of biometric data; (4) performance: the recognition performance should not be deteriorated by the formula.

In the proposed scheme, the relative distance of different fingers was adopted to prevent privacy leakage. The replacement of users' identities can be easily achieved by changing the size and the type of carrier for verification. Figure 12 demonstrates an example for changing the biometrics by resizing the wheel lock. The other ways to replace the existing users' biometric traits is to change the rotation speed of lock wheel. Dynamically adjusting the rotation speed of wheel lock can also affect the biometrics significantly, including the angles and other rotational dynamics. In addition, we can use Bayesian classifiers and neural network as the learning method; the requirements of one-way permutation can also be realized.

5. Conclusion

This study proposes a novel authentication approach consisting of both physiological and behavioral biometrics. The proposed scheme derives the possibility of performing complicated biometrics without extra equipment, but only multitouch panel integrated in most mobile devices.

The experiments showed that it can be used to handle the general user authentication scenario and provide a relatively secure environment to prevent attacks. We also demonstrate how the biometric privacy can be obtained through the biometric identity replacement. The future works can be divided into experiment and implementation. The experiments will be used to verify and evaluate the feasibility of the proposed scheme with large-scale participants. With regard to implementation, multiple mobile devices, which have touch panel as interface, should be applicable or portable with the corresponding pbLogon in all respects.

Notations

MU:	Mobile user MU = $\{U_i \mid i > 0\}$
U_i:	User profile of ith user
P:	pbLogon system
Password:	n-digits numeric password
N:	N presents the total number of fingers on touch panel
$G(\cdot)$:	$G(\cdot)$ returns the number of fingers, hand pose, and other biometric characteristics
H_i:	The ith hand of U_i
F_i:	ith finger; ex: thumb presents as F_1
D:	$D = \{d_{ij} \mid 0 < i \neq j < 6\}$, where d_{ij} is the distance between fingers
A:	$A = \{a_{ijk} \mid 0 < i \neq j \neq k < 6\}$, where a_{ijk} is the area of any three fingers
L:	$L = \{0, 1 \mid L = 1$ if F_2 (index finger) is longer than F_4 (ring finger), otherwise $L = 0\}$
R:	$R = \{0, 1 \mid R = 0$ if the user is left-handed, otherwise $R = 1\}$
V:	The set of the rotation velocity of the user; clockwise presents with positive value and counterclockwise presents in negative value
B_i:	Behavioral biometrics B_i of U_i
$C_k(\cdot)$:	Classifiers with kth feature
LB_{ik}:	The low-bounds value of kth feature
UB_{ik}:	The up-bounds value of kth feature
FEA_i:	$\mathrm{FEA}_i = (\mathrm{LB}_{ik}, \mathrm{UB}_{ik})$ of every feature can be directly obtained from the low-bounds and up-bounds of training patterns
Auth(·):	The user authentication function
ΔS:	The dissimilarity score
$Q \rightarrow T : M$:	Entity T receives a message M from Q.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This research was supported by Grant MOST-103-2511-S-142-016 from the Ministry of Science and Technology, Taiwan. The researchers also appreciate grant from the National Taichung University of Education and Ministry of Education, Taiwan, for financially supporting this research under grants of the web service curriculum development project for information software talent nurturing - web services and applications of Internet of Things (100C079).

References

[1] D. A. Ortiz-Yepes, R. J. Hermann, H. Steinauer, and P. Buhler, "Bringing strong authentication and transaction security to the realm of mobile devices," *IBM Journal of Research and Development*, vol. 58, no. 1, pp. 1–11, 2014.

[2] M. H. Eldefrawy, M. K. Khan, K. Alghathbar, T. Kim, and H. Elkamchouchi, "Mobile one-time passwords: two-factor authentication using mobile phones," *Security and Communication Networks*, vol. 5, no. 5, pp. 508–516, 2012.

[3] J. Hong, "The state of phishing attacks," *Communications of the ACM*, vol. 55, no. 1, pp. 74–81, 2012.

[4] Z. Ramzan, "Phishing attacks and countermeasures," in *Handbook of Information and Communication Security*, no. 23, pp. 433–448, Springer, Berlin, Germany, 2010.

[5] T. N. Jagatic, N. A. Johnson, M. Jakobsson, and F. Menczer, "Social phishing," *Communications of the ACM*, vol. 50, no. 10, pp. 94–100, 2007.

[6] N. Leavitt, "Instant messaging: a new target for hackers," *Computer*, vol. 38, no. 7, pp. 20–23, 2005.

[7] A. van der Merwe, R. Seker, and A. Gerber, "Phishing in the system of systems settings: mobile technology," in *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, vol. 1, pp. 492–498, October 2005.

[8] A. El-Sallam, F. Sohel, and M. Bennamoun, "Robust pose invariant shape-based hand recognition," in *Proceeding of the 6th IEEE Conference on Industrial Electronics and Applications (ICIEA '11)*, pp. 281–286, Beijing, China, June 2011.

[9] A. Heydarzadegan, M. Moradi, and A. Toorani, "Biometric recognition systems: a survey," *International Research Journal of Applied and Basic Science*, vol. 6, no. 11, pp. 1609–1618, 2013.

[10] K. Simoens, P. Tuyls, and B. Preneel, "Privacy weaknesses in biometric sketches," in *Proceedings of the 30th IEEE Symposium on Security and Privacy*, pp. 188–203, May 2009.

[11] T. Hoang, T. Nguyen, C. Luong, S. Do, and D. Choi, "Adaptive cross-device gait recognition using a mobile accelerometer," *Journal of Information Processing Systems*, vol. 9, no. 2, pp. 333–348, 2013.

[12] S. Trewin, C. Swart, L. Koved, J. Martino, K. Singh, and S. Ben-David, "Biometric authentication on a mobile device: a study of user effort, error and task disruption," in *Proceedings of the 28th Annual Computer Security Applications Conference (ACSAC '12)*, pp. 159–168, December 2012.

[13] E. Sesa-Nogueras and M. Faundez-Zanuy, "Biometric recognition using online uppercase handwritten text," *Pattern Recognition*, vol. 45, no. 1, pp. 128–144, 2012.

[14] N. Yamasaki and T. Shimamura, "Accuracy improvement of speaker authentication in noisy environments using bone-conducted speech," in *Proceedings of the 53rd IEEE International Midwest Symposium on Circuits and Systems (MWSCAS '10)*, pp. 197–200, Seattle, Wash, USA, August 2010.

[15] S. P. Banerjee and D. Woodard, "Biometric authentication and identification using keystroke dynamics: a survey," *Journal of Pattern Recognition Research*, vol. 7, no. 1, pp. 116–139, 2012.

[16] D. Shanmugapriya and G. Padmavathi, "A survey of biometric keystroke dynamics: approaches, security and challenges," *International Journal of Computer Science and Information Security*, vol. 5, no. 1, pp. 115–119, 2009.

[17] C.-L. Tsai, C.-J. Chen, and D.-J. Zhuang, "Trusted M-banking verification scheme based on a combination of OTP and Biometrics," *Journal of Convergence*, vol. 3, no. 3, pp. 23–30, 2012.

[18] C.-I. Fan and Y.-H. Lin, "Full privacy minutiae-based fingerprint verification for low-computation devices," *Journal of Convergence*, vol. 3, no. 2, pp. 21–24, 2012.

[19] B. T. Ulery, R. A. Hicklin, J. Buscaglia, and M. A. Roberts, "Accuracy and reliability of forensic latent fingerprint decisions," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 108, no. 19, pp. 7733–7738, 2011.

[20] A. Kong, D. Zhang, and M. Kamel, "A survey of palmprint recognition," *Pattern Recognition*, vol. 42, no. 7, pp. 1408–1418, 2009.

[21] M. P. Satone and G. K. Kharate, "Face recognition based on PCA on wavelet subband of average-half-face," *Journal of Information Processing Systems*, vol. 8, no. 3, pp. 483–494, 2012.

[22] X. Yang, G. Peng, Z. Cai, and K. Zeng, "Occluded and low resolution face detection with hierarchical deformable model," *Journal of Convergence*, vol. 4, no. 2, pp. 11–14, 2013.

[23] L. Birgale and M. Kokare, "Iris recognition using ridgelets," *Journal of Information Processing Systems*, vol. 8, no. 3, pp. 445–458, 2012.

[24] A. Hussain, A. Bhuiyan, and A. Mian, "Biometric security application for person authentication using retinal vessel feature," in *Proceedings of the International Conference on Digital Image Computing: Techniques and Application (DICTA '13)*, pp. 1–8, Tasmania, Australia, November 2013.

[25] N. Duta, "A survey of biometric technology based on hand shape," *Pattern Recognition*, vol. 42, no. 11, pp. 2797–2806, 2009.

[26] A. de-Santos-Sierra, C. Sánchez-Ávila, G. B. del Pozo, and J. Guerra-Casanova, "Unconstrained and contactless hand geometry biometrics," *Sensors*, vol. 11, no. 11, pp. 10143–10164, 2011.

[27] V. Kanhangad, A. Kumar, and D. Zhang, "Combining 2D and 3D hand geometry features for biometrie verification," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '09)*, pp. 39–44, June 2009.

[28] D. Schmidt, M. K. Chong, and H. Gellersen, "HandsDown: hand-contour-based user identification for interactive surfaces," in *Proceedings of the 6th Nordic Conference on Human-Computer Interaction: Extending Boundaries (NordiCHI '10)*, pp. 432–441, October 2010.

[29] A. Serwadda and V. V. Phoha, "Examining a large keystroke biometrics dataset for statistical-attack openings," *ACM Transactions on Information and System Security (TISSEC)*, vol. 16, no. 2, article 8, 2013.

[30] B. Sayed, I. Traore, I. Woungang, and M. S. Obaidat, "Biometric authentication using mouse gesture dynamics," *IEEE Systems Journal*, vol. 7, no. 2, pp. 262–274, 2013.

[31] F. Li, N. Clarke, M. Papadaki, and P. Dowland, "Active authentication for mobile devices utilising behaviour profiling," *International Journal of Information Security*, vol. 13, no. 3, pp. 229–244, 2014.

[32] N. L. Clarke and S. M. Furnell, "Authenticating mobile phone users using keystroke analysis," *International Journal of Information Security*, vol. 6, no. 1, pp. 1–14, 2007.

[33] S. Choi, I.-H. Youn, R. LeMay, S. Burns, and J.-H. Youn, "Biometric gait recognition based on wireless acceleration sensor using k-nearest neighbor classification," in *Proceedings of the International Conference on Computing, Networking and Communications*, pp. 1091–1095, 2014.

[34] X. Zhou and B. Bhanu, "Feature fusion of side face and gait for video-based human identification," *Pattern Recognition*, vol. 41, no. 3, pp. 778–795, 2008.

[35] J. Wu, J. Konrad, and P. Ishwar, "Dynamic time warping for gesture-based user identification and authentication with kinect," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 2371–2375, 2013.

[36] N. Sae-Bae, K. Ahmed, K. Isbister, and N. Memon, "Biometric-rich gestures: a novel approach to authentication on multi-touch devices," in *Proceedings of the 30th ACM Conference on Human Factors in Computing Systems (CHI '12)*, pp. 977–986, May 2012.

[37] K. Simoens, P. Tuyls, and B. Preneel, "Privacy weaknesses in biometric sketches," in *Proceedings of the 30th IEEE Symposium on Security and Privacy*, pp. 188–203, Berkeley, Calif, USA, May 2009.

[38] S. Prabhakar, S. Pankanti, and A. K. Jain, "Biometric recognition: security and privacy concerns," *IEEE Security and Privacy*, vol. 1, no. 2, pp. 33–42, 2003.

[39] J. Yuan and S. Yu, "Efficient privacy-preserving biometric identification in cloud computing," in *Proceeding of the 32nd IEEE Conference on Computer Communications (INFOCOM '13)*, pp. 2652–2660, Turin, Italy, April 2013.

[40] Y. Sui, X. Zou, E. Y. Du, and F. Li, "Design and analysis of a highly user-friendly, secure, privacy-preserving, and revocable authentication method," *IEEE Transactions on Computers*, vol. 63, no. 4, pp. 902–916, 2014.

[41] E. Kukula and S. Elliott, "Implementation of hand geometry an analysis of user perspectives and system performance," *IEEE Aerospace and Electronic Systems Magazine*, vol. 21, no. 3, pp. 3–9, 2006.

[42] H. Chen, H. Valizadegan, and C. Jackson, "Fake hands: spoofing hand geometry systems," in *Biometrics Consortium Conference*, Arlington, Va, USA, 2005.

[43] Y. Sutcu, Q. Li, and N. Memon, "Protecting biometric templates with sketch: theory and practice," *IEEE Transactions on Information Forensics and Security*, vol. 2, no. 3, pp. 503–512, 2007.

Hindawi Publishing Corporation
The Scientific World Journal
Volume 2014, Article ID 542824, 9 pages
http://dx.doi.org/10.1155/2014/542824

Research Article

A Novel Method for Functional Annotation Prediction Based on Combination of Classification Methods

Jaehee Jung,[1] Heung Ki Lee,[1] and Gangman Yi[2]

[1] *Samsung Electronics, Suwon, Republic of Korea*
[2] *Department of Computer Science & Engineering, Gangneung-Wonju National University, Gangwon, Republic of Korea*

Correspondence should be addressed to Gangman Yi; gangman@cs.gwnu.ac.kr

Received 14 April 2014; Accepted 29 June 2014; Published 16 July 2014

Academic Editor: Young-Sik Jeong

Automated protein function prediction defines the designation of functions of unknown protein functions by using computational methods. This technique is useful to automatically assign gene functional annotations for undefined sequences in next generation genome analysis (NGS). NGS is a popular research method since high-throughput technologies such as DNA sequencing and microarrays have created large sets of genes. These huge sequences have greatly increased the need for analysis. Previous research has been based on the similarities of sequences as this is strongly related to the functional homology. However, this study aimed to designate protein functions by automatically predicting the function of the genome by utilizing InterPro (IPR), which can represent the properties of the protein family and groups of the protein function. Moreover, we used gene ontology (GO), which is the controlled vocabulary used to comprehensively describe the protein function. To define the relationship between IPR and GO terms, three pattern recognition techniques have been employed under different conditions, such as feature selection and weighted value, instead of a binary one.

1. Introduction

Conventionally biologists have deduced protein functions through manual experimentation, which is time consuming and involves high expenditure [1]. As increasing wealth of genome data such as DNA sequencing and microarrays, it is clear that manual functional annotation cannot be executed, but the needs for automatic annotation of protein function are increased [1]. In particular, as a boom of sequence analysis research areas such as next generation sequence, functional annotation for unknown sequences, and so forth, became one of the very active research areas. As for new proteins that have not yet been revealed experimentally, the protein functions could be automatically annotated by processing [2] if the model was created using the known protein function. If some features are highly related with some biological functions and specific pattern methods can be used for defining the function, the suggested model can be established to automatically assign the function. Therefore, it would be possible to predict the function in much shorter time than that required by the existing method based on experimentation. One of the popular methods for the automatic assignment is to identify a relationship between features and GO terms [3], where GO provides a controlled vocabulary of terms for annotating proteins. The simplest example is GOA [4], which is manually mapping InterPro terms to GO terms by the InterPro team at EBI. However, this approach is based on the manual mapping; thus recently researchers investigate the relationship using various features such as protein domain, microarray, and protein-protein interaction.

Automated gene annotation research often uses functional databases such as protein functional site, protein family, or gene expression. These databases are usually used for the patterns of base sequence and sequence similarity since sequence similarity is usually related to the functional homology. InterPro (IPR) [5] combined several protein family databases such as Prosite, Prints, Pfam, Prodom, SMART, TIGRFams, and PIR SuperFamily, in order to provide the functional analysis of protein. In addition, the InterPro Consortium provides the InterProScan package [6], which gives the InterPro by simply putting sequences. It would therefore be appropriate to use it as a feature to define

the functions of an unknown protein. Moreover, unlike the traditional free text description, controlled vocabularies of various types have been employed. Gene ontology (GO) provides a controlled vocabulary of terms for annotating proteins. Every GO term has a unique numerical identifier that represents the gene function. Each GO term is assigned to one of the three categories of molecular functions, biological processes, or cellular components. These terms are organized into a directed acyclic graph (DAG), which provides a rich framework for describing the function of proteins. Each GO term has a more specific GO term (child) and more than one less-specific term (parent). The database is still under development by the GO Consortium and aims to describe the comprehensive features of the genome.

This paper aims to analyze automatic annotation processing methods by comparing the relationship between IPR and GO-utilizing known data and a range of methods. We started by examining the characteristics of the treated dataset as that was very sparse and skewed. We then described the methods used to reproduce the dataset. The suggested method uses the combinatorial analysis of feature selection and three different pattern recognition approaches so that we would be able to analyze the performance and find the optimized options.

2. Related Works

In the case of the functional annotation that identifies the function of genome automatically, various studies related to the database and automatic annotation are in progress in order to define the function of genome of various species from human beings to small microorganisms. The studies are in progress to allow for the automatic prediction of the protein function by an easy access through web or automatic installation, and mostly the database to manage this systematically is also in progress as it has continued to be updated. However, the method of defining the protein function automatically is still at the initial phase; thus, the accuracy is not very high. In the case of using the interpro2go that was manually mapped as in GOA [4], mapping is relied for defining the function; thus, the accuracy is not high. Most studies are conducted by small species to increase the accuracy, and the prediction methods for the protein function based on the calculation that has been researched make a judgment mostly by utilizing the similarities of sequence.

As for the most frequently used tools, they include Gotcha [7], OntoBLAST [8], Blast2GO [9], AutoFACT [10], and so forth; Gotcha [7] can have the similarity of sequence and the directed acyclic graph (DAG) of gene ontology; in other words, the parent node can have several offspring nodes. Thus, it is the method of utilizing the property in which the parent node means the functions of more comprehensive meaning. It is the methods of automatically naming GO by assigning a score to GO owned by the genome that is determined to be similar by judging the similarity of sequence. Blast2GO [9] is the method of annotating new protein functions that cannot be known by gene ontology that is owned by a similar sequence after judging the similarity of sequence by utilizing BLAST [11]. It is the prediction model for the accuracy by assigning weights in accordance with the evidence code that is the annotation code of GO at this point. The evidence code means the code of GO to indicate whether it is automatically named (IEA) and it is determined by the similarity (ISS). OntoBLAST [8] is the method of finding possible protein functions from GO, which are obtained also from BLAST search. AutoFACT [10] proposed a fast annotation method by utilizing BLAST with the relevant database.

3. Methods

3.1. Features of Data. The data to be used is *Saccharomyces cerevisiae*; it is one of yeast fungi; thus, it belongs to the fungus class and it is the most well-known data by the experiments. Since it forms a relatively small dataset as compared with the other species and it already brings out its related function; thus, it would be an appropriate data for establishing a model for the automatic annotation processing. For the extraction of this data, 4,370 proteins could be obtained as a result of searching and extracting *Saccharomyces cerevisiae* only from SWISS-PROT.

The property to be used as a feature to create a model of data is IPR. IPR has the appropriate features for the reference data that include the protein family binding the protein functions in a similar way and the functions of Prosite, Prints, Pfam, Prodom, SMART, TIGRFams, and PIR SuperFamily that play the central role to refer to the functional domain database. GO is utilized as the reference data for defining the function automatically. GO forms a hierarchical structure and divided into the three big classes—cellular component, molecular function, and biological process.

When counting the total number of IPR and GO terms possessed by the 4,370 extracted proteins of *Saccharomyces cerevisiae*, it was found to have 2,624 IPRs and 2,438 GO terms. When this data had one of the properties of IPR or GO term for each protein, it was represented in a binary form. It is represented in a large matrix (4370 $*$ 2438) of GO in a binary form by representing "1" when the proteins have one term of particular GO terms and "0" when they do not have one as parsing gene ontology at ontology in the data section of SWISS-PROT. Also as for IPR, the IPR data was configured in a matrix form of 4,370 $*$ 2,624 by a matrix of binary form as representing whether each protein has it through listing IPRs possessed by *Saccharomyces cerevisiae* proteins after extracting InterPro in the family and domain database section with the same method as described above. A diagram for representing GO of IPR for each protein in a matrix and lining up the quantity of GOs that can be represented by "1" and, in other words, the quantity owned by the proteins would be the same as shown in Figure 1. As shown in Figure 1, it has the problem that it does not have a sufficient quantity for each GO to conduct the learning. When viewed from the perspective of one single GO, the number of cases in which it has only one single GO is 414. This means that only one protein owns the relevant GO; thus, it would not be appropriate to utilize it as the learning data. In addition, the validity was tested through the 10-fold cross validation; thus, GO that has less quantity than a certain level would not be appropriate for the use as the learning data.

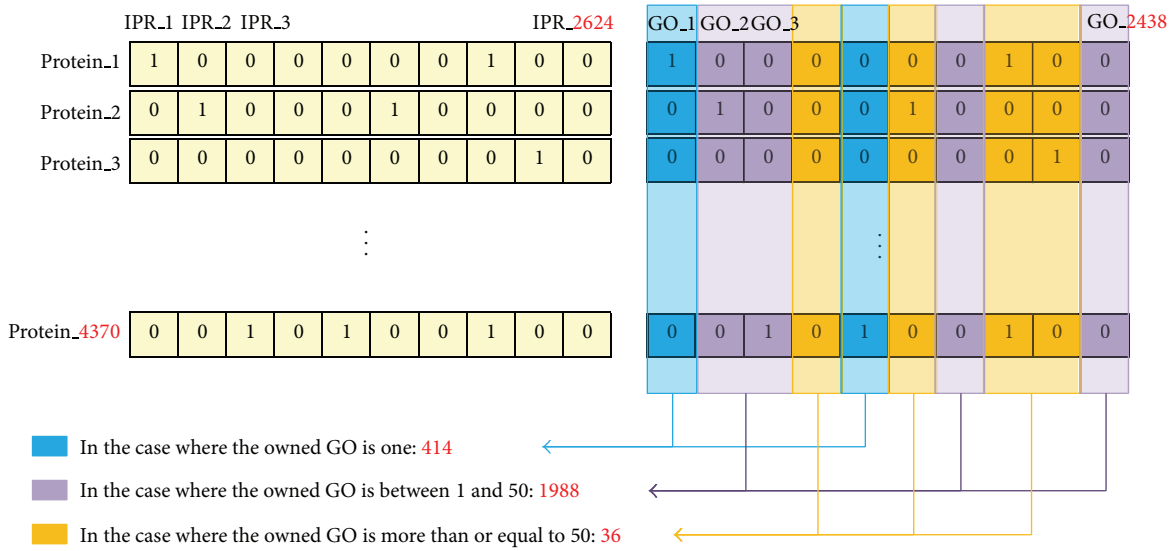


Figure 1: Features of data.

Table 1: Learning methods.

Method	Data			
	Original IPR		Weighted IPR	
Adaboosting	W/O feature selection	W feature selection	W/O feature selection	W feature selection
SVM	W/O feature selection	W feature selection	W/O feature selection	W feature selection
SMO	W/O feature selection	W feature selection	W/O feature selection	W feature selection

However, the biggest problem of the data is that the data exists sparse even though it has a relatively sufficient quantity to be utilized as the learning data. And the fact that the protein data does not have the relevant IPR or GO are inclined to one side than the data having the relevant IPR or GO when viewed from a particular IPR or GO term is also a problem. For instance, only 50 proteins have a particular GO out of 4,370 proteins when viewed from the perspective of a particular single GO; therefore, they are represented by "1" and the remaining 4,320 proteins are represented by "0" since they do not have it. In the case of conducting the learning and experiment with such data, it is quite often predicted that most do not have it since the learning is conducted as being excessively inclined to "0" that is not owned by the learning result; thus, it cannot become an effective model for the automatic function prediction and command processing that has to assign new functions. There are many cases represented by "0" representing "not having" in the case of IPR in addition to GO. This cannot be utilized as an effective feature. Due to such properties of these two features, this paper aims to apply the method as to the feature selection and balanced dataset. Moreover, it aims to analyze the results by converting the binary form of the data into a nonbinary form (weighted IPR) by utilizing the correlation coefficients since the data to be processed is not a binary form.

3.2. Prediction Method of Protein Function. This paper aims to compare and analyze the prediction method of protein function by utilizing the data having a sufficient quantity of data as the data of learning and experiment of this paper among the data described in Section 3.1. Before the analysis, it would be essentially required to have a process of reconfiguring it as a balanced dataset due to the feature of not being balanced with the sparseness of data. It aims to compare the case of applying the feature selection by the three mutually different learning methods and the case of conducting the weighted IPR that adds weights to the data, respectively [12–14].

This paper shows comparison and analysis of the prediction methods by the methods presented in Table 1. First as for the learning methods, adaboosting [15] is the method of creating an optimal classification through several times of learning by assigning weights as to the instances wrongly classified by the method of weak learner. SVM is the method of seeking a boundary that makes the error of margin that can differentiate the class to be classified at the hyperplane; thus, it is one of the learning methods of machine learning. SMO [16] is the most well-known tool of libsvm [17]; thus, it can be regarded as the method that has simplified the complexity of SVM by the sequential minimal optimization. As for the methods to be presented in Section 3.2.2, the case of using the feature selection method and the case of not using it were compared and analyzed as W/O in Table 1 meant Without and W meant With. Furthermore, it compared the case of using the method called weighted IPR to be stated in Section 3.2.3. with the case of using the original data as it was.

3.2.1. Dataset Reconfiguration to Adjust Balance. As shown in Figure 2, there are more proteins not having the relevant GO

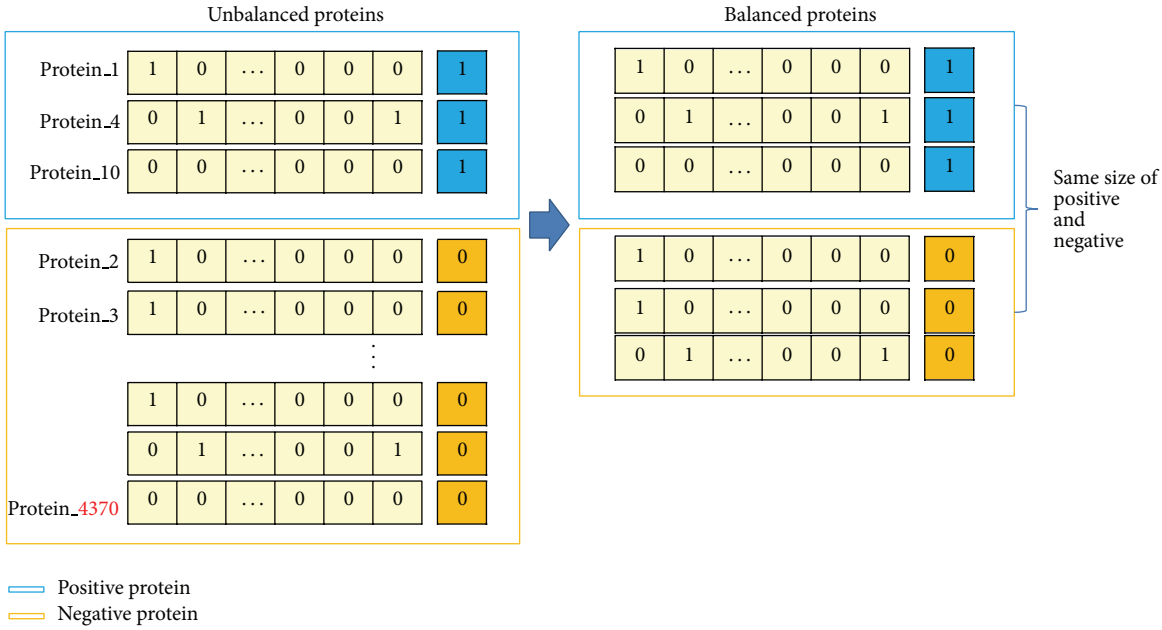


Figure 2: Data reconfiguration using undersampling.

than those having it when viewed based on a particular GO. However, there are more proteins not having a particular GO and, in other words, negative proteins, when learning with such data; therefore, there would be a high degree of probability for the modeling that most of learning results turned out to be not having it. However, it is only possible to find it out by creating a model having the relevant GO rather than a model not having GO. When experimenting with proteins that are not able to perform the function, it is impossible to obtain the desired result. Thus, balanced sampling approach is employed to overcome this handicapped data property [18].

There are the undersampling method and the oversampling method in terms of reconfiguring the data that consists of balanced proteins; the oversampling [19] is the method of making the number equal by generating the data that become the major in terms of quantity as many as the quantities at which the relatively fewer data becomes the major in a random way. The undersampling [20] is the method of meeting the ratio by selecting more data randomly based on the data whose quantity is few. In this experiment, the data that is relatively few in quantity is more important information; therefore, this paper reduces that quantity by utilizing the undersampling. As shown in Figure 2, the data indicated by "1" is to be named as positive protein, whereas the data indicated by "0" is to be named as negative protein. And it is supposed to be trained with proteins that are fewer than 4,370 in terms of the quantity of protein by reconfiguring the data for the learning model at each GO through selecting the negative proteins just as many as the quantities of positive proteins.

3.2.2. Feature Selection. As shown above Figure 1, IPR has the matrix of many binary features of 2,624 when viewed based on one GO. It is the well-known fact that learning and experimenting by selecting only meaningful features would reduce the time to be taken and have a better result as compared with learning and experimenting the method presented above by these many matrices [16, 21–23]. When representing the case in which "1" represents that each IPR has protein by positive data and the case of not having it by "0," the positive negative data is to be counted for each protein. The positive data is represented as "IPos," "GPos" and the negative data is represented as "INeg", "GNeg" at IPR and GO, respectively, and it is possible to classify the state of IPR and GO for each protein. They can become 4 states as shown in Table 2.

Table 2: Number of cases in accordance with the status of IPR and GO.

IPR	GO	Express
0	0	$N_{\text{GNeg\_INeg}}$
0	1	$N_{\text{GPos\_INeg}}$
1	0	$N_{\text{GNeg\_IPos}}$
1	1	$N_{\text{GPos\_IPos}}$

It is possible to calculate the four probabilities ($N_{\text{GPos\_IPos}}/N_{\text{Pos}}$, $N_{\text{GNeg\_IPos}}/N_{\text{Posv}}$, $N_{\text{GPos\_INeg}}/N_{\text{Neg}}$, and $N_{\text{GNeg\_INeg}}/N_{\text{Neg}}$) by utilizing the 4 data, where N_{Pos} stands for the total number of positive proteins and N_{Neg} means the total number of negative proteins. These probabilities represent a conditional probability that the gene ontology term may possess depending on the conditions of each IPR. When viewing the property by adding these conditional probabilities as an example of GO:0000329, the diagram as shown below could be viewed. The x-axis means several IPRs that are being experimented and the y-axis is the value of adding the conditional probabilities. It is possible to see the phenomenon

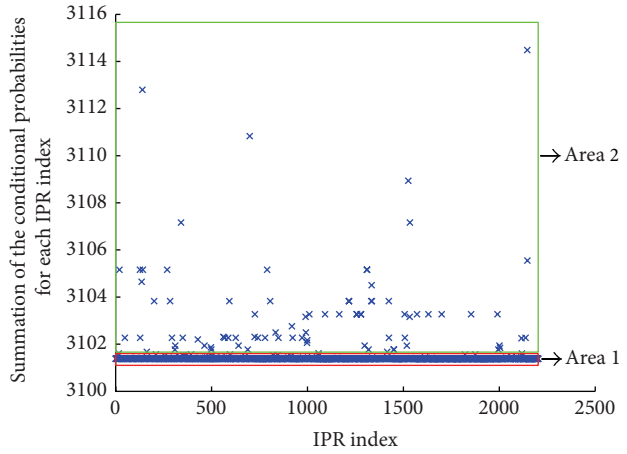


Figure 3: Plot the summation of probability of IPR in terms of GO:0000329.

Table 3: The example of Original IPR.

	IPR1	IPR2	IPR3	IPR4	IPR5	IPR6
Protein 1	0	1	1	1	0	0
Protein 2	1	1	0	0	1	0
Protein 3	0	0	0	0	0	1
Protein 4	1	1	1	0	0	1
Protein 5	0	1	0	1	0	0

of which most are concentrated in Area 1. On that account, 99 percent of them are those IPRs having negative IPR term and also negative gene ontology (Figure 3). This paper selected the features based on the IPRs that are concentrated in Area 2 as excluding these IPRs. In other words, this is the learning method of utilizing only the selected index as a feature by selecting only the index of IPRs in Area 2 among the 2204 IPRs by calculating the conditional probabilities above for each GO.

3.2.3. Weighted IPR. IPR that is utilized as the feature is the binary data that consists of 0 and 1. When converting this data into a continuous form rather than binary form by utilizing a correlation coefficient, IPR feature data would be expected to select a feature without partiality [24]. This paper aims to analyze the performance between the two methods by the differences between the feature extraction using the binary data that consists of 0 and 1 and the weighted IPR of a continuous form as naming this data as the weighted IPR.

For instance, as shown in Table 3, the table that is composed of 0 and 1 would be modified into a table that utilizes a correlation coefficient (Table 4). A correlation coefficient becomes a value closer to 1 with a higher degree of correlation, whereas it is represented by a value close to 0 when there is no correlation. In addition, it becomes a negative value when there is a mutually contradicting correlation.

This paper aims to change to weight coefficients as proposed by Formula (1) based on this correlation coefficient. First, each protein P possesses IPR from 1 to n. All the proteins

Table 4: Correlation Coefficient among the IPRs.

	IPR1	IPR2	IPR3	IPR4	IPR5	IPR6
IPR1	1.0000	0.4082	0.1667	−0.6667	0.6124	0.1667
IPR2	0.4082	1.0000	0.4082	0.4082	0.2500	−0.6124
IPR3	0.1667	0.4082	1.0000	0.1667	−0.4082	0.1667
IPR4	−0.6667	0.4082	0.1667	1.0000	−0.4082	−0.6667
IPR5	0.6124	0.2500	−0.4082	−0.4082	1.0000	−0.4082
IPR6	0.1667	−0.6124	0.1667	−0.6667	−0.4082	1.0000

Table 5: Weighted IPR.

	IPR1	IPR2	IPR3	IPR4	IPR5	IPR6
Protein 1	0.5251	0.2076	0.1462	0.1462	0.1376	−0.1628
Protein 2	0.2008	0.1295	−0.2501	0.3750	0.1697	0.3750
Protein 3	0.1389	0.3934	0.0889	−0.1334	0.0122	0.5000
Protein 4	0.2633	0.0725	0.2633	0.2500	0.2500	−0.0991
Protein 5	0.7990	0.2500	−0.0633	0.2500	−0.1724	−0.0633

to be experimented are represented by IPR of n units. A particular protein having IPR would be represented by 1, whereas those not having IPR would be represented by 0.

For instance, Protein 1 in Table 3 is represented as not having IPR1, IPR5, and IPR6, which are 0, whereas IPR2, IPR3, and IPR4 are represented by IPR possessed by the relevant protein. At this point, there is a relationship between IPR5 and IPR6 since IPR1 is not a property that is not owned when viewed by each IPR of Protein 1. In reference with Table 4, the weight (IPR1) value of IPR 1 of corr (IPR 1, IPR5) = 0.6124 and corr (IPR1, IPR6) = 0.1667 Protein 1 is 0.6124 + 0.1667 = 0.7791. Moreover, the value of weight sum (IPR1) is represented by IPR1 = 0; therefore, the value of adding all the correlation coefficients of IPR5 and IPR6 becomes 0.7418. Essentially the value was the binary form of 0 and 1 in order to calculate the weighted sum (IPR1) as to IPR1 of Protein 1 of the calculated value; therefore, this finally generates the value of $0.5 * 0.7791/0.7418 = 0.5251$ by giving the weighted value 0.5. A new data defined in the new weighted IPR would be generated by such method. Table 5 can be regarded as one of such cases.

Converting weighted IPR by the correlations and weights:

$$P = \{\mathrm{IPR}_1, \ldots, \mathrm{IPR}_n\},$$

$$\mathrm{Weight}\left(\mathrm{IPR}_i\right) = \sum_{j=1}^{|P|} \mathrm{corrcoeff}\left(\mathrm{IPR}_i, \mathrm{IPR}_j\right), \quad \text{where } i \neq j,$$

$$\mathrm{Weight}_{\mathrm{sum}}\left(\mathrm{IPR}_i\right) = 0.5 \times \frac{\mathrm{Weight}\left(\mathrm{IPR}_i\right)}{\sum_{j=1}^{|P|} \mathrm{Weight}\left(\mathrm{IPR}_j\right), \text{where } \mathrm{IPR}_i = \mathrm{IPR}_j}. \quad (1)$$

Figure 4 shows the error rate from applying the 12 methods to GO:0003700, GO:003677, GO:005515, and GO:006412. The GO samples were randomly selected. The three mutually

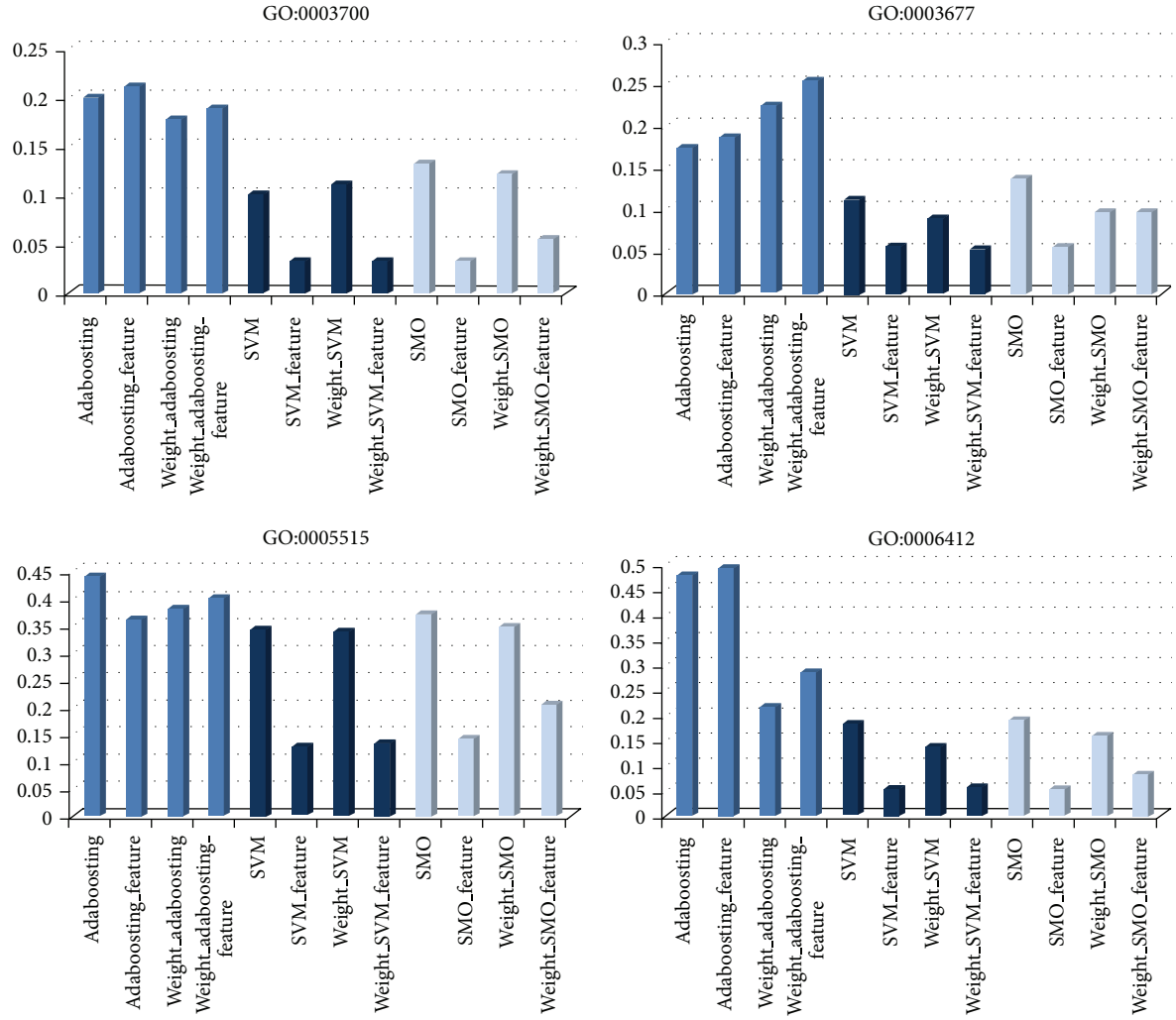


Figure 4: Error rate applying several methods for GO:0003700, GO:0003677, GO:0005515, and GO:0006412.

different methods presented in Table 1 are shown in different colors. Better performance is indicated by a smaller error. Blue shows the error rate from adaboosting, where dark blue represents SVM p performance and light blue represents SMO. As shown, the error rate for each method varies depending on the GO selection. For example, among the three different methods, SVM has the smallest error rate at GO:0003700 and GO:00367, but SMO has the best performance at GO:006412. Selecting features suggested by SVM or SMO generally resulted in similar or smaller error rates. For the four methods utilizing the original, the feature selection, weighted method, and two combinations, the results for each GO term are as shown in Table 6. This finding shows that they have a high prediction rate ranging from 97 to 99.

As an extension to Figure 4, Figure 5 shows the error rates dependent on the pattern recognition technique for each GO term. This enables a comparison of the performance under the suggested method, such as feature selection and weighted IPR. The error rate after applying feature selection or weighted IPR is slightly lesser or more than the original dataset, as shown in Figure 5(a). However, for SMO and SVM the errors were dramatically reduced under feature selection (Figures 5(b) and 5(c)).

4. Conclusion

This paper compared and evaluated the performance that could define the protein function by applying the classification algorithm by utilizing the feature selection and data transformation. As for the data to be processed, the data having GO term has been composed in much less quantity than the protein not having GO term when viewed by individual GO term. In addition, IPR that is set as the feature point is sparsely distributed; thus, it becomes difficult to learn all the protein data through the general classification algorithm. Due to such limitations, the performance as to the automatic annotation was compared by various classification methods through extracting only the GO term having the standard level or more as the learning subject. Moreover, the performance with the original data was also analyzed by the method of using the binary data as the correlation coefficient through converting it into a newly weighted coefficient.

Table 6: Error rate for each GO term.

GO	Error rate using SVM with feature selection	Error rate using SMO with feature selection	Error rate using weighted IPR SVM with feature selection	Error rate using weighted IPR SMO with feature selection
GO:0000324	0.030303	0.045455	0.030303	0.166667
GO:0000329	0.090909	0.136364	0.075758	0.257576
GO:0000398	0.025641	0.025641	0.025641	0.128205
GO:0003677	0.055556	0.055556	0.051852	0.096296
GO:0003700	0.033333	0.033333	0.033333	0.055556
GO:0003723	0.026316	0.026316	0.035088	0.096491
GO:0003735	0.033333	0.038889	0.053333	0.06
GO:0005515	0.128571	0.142857	0.144444	0.266667
GO:0005524	0.166667	0.177778	0.144444	0.266667
GO:0005730	0.116667	0.116667	0.108333	0.166667
GO:0005732	0.060606	0.060606	0.060606	0.106061
GO:0005743	0.208333	0.263889	0.208333	0.416667
GO:0005783	0.22069	0.234483	0.224138	0.275862
GO:0005789	0.075758	0.075758	0.106061	0.242424
GO:0005829	0.090909	0.090909	0.1	0.254545
GO:0005886	0.123077	0.123077	0.115385	0.182692
GO:0005935	0.111111	0.111111	0.041667	0.152778
GO:0006281	0.151515	0.181818	0.151515	0.227273
GO:0006355	0.133333	0.15	0.133333	0.166667
GO:0006365	0.075758	0.090909	0.075758	0.166667
GO:0006412	0.05303	0.05303	0.056818	0.079545
GO:0006457	0.030303	0.030303	0.045455	0.166667
GO:0006468	0.009804	0.039216	0.009804	0.058824
GO:0006511	0.030303	0.030303	0.030303	0.121212
GO:0006888	0	0	0	0.083333
GO:0006897	0.013889	0.013889	0.013889	0.152778
GO:0006950	0.090909	0.106061	0.060606	0.212121
GO:0007047	0.083333	0.125	0.092593	0.148148
GO:0009060	0.066667	0.066667	0.066667	0.316667
GO:0009277	0	0	0	0.016667
GO:0016020	0.05	0.05	0.05	0.166667
GO:0016021	0.029412	0.029412	0.019608	0.078431

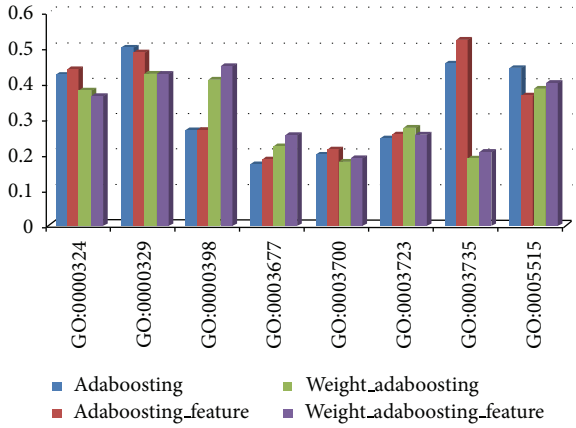
However, as for the data sampling and feature selection processed in this paper, the GO term is trained primarily the data of protein having a certain amount or more for the learning at *Saccharomyces cerevisiae*, thus, there is the limitation that the quantity of learned data of GO term is small. If it is to be trained by utilizing the data that includes a variety of species such as SWISS PROT in order to overcome this limitation, it will be possible to expect to utilize the automatic function prediction by learning more GO terms with the use of large quantity of data. Thus, this paper aims to study a learning method that is appropriate for this. In addition, it aims to prepare a base to allow for the automatic annotation by seeking for different features that can be utilized as a keyword in addition to IPR when trying to find out unknown protein functions by identifying the correlation with GO.

Conflict of Interests

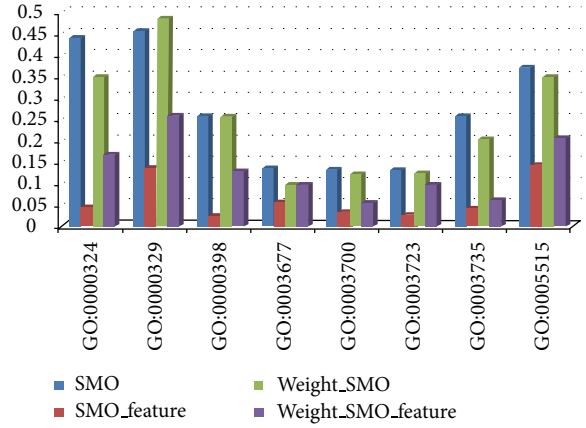
The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

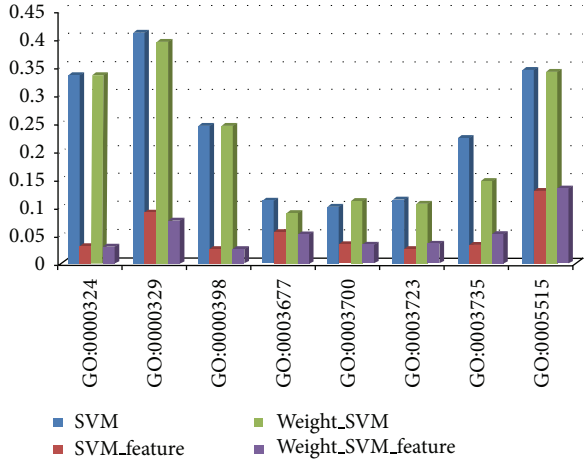
This paper is a revised and extended version of a paper that was originally presented at the 2014 FTRA International Symposium on Frontier and Innovation in Future Computing and Communications. This research was supported by Basic



(a) The error rate of employing adaboosting under different conditions at randomly selected GO terms



(b) The error rate of employing SMO under different conditions at randomly selected GO terms



(c) The error rate of employing SVM under different conditions at randomly selected GO terms

Figure 5: The error rate of three different techniques.

Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (2013R1A1A2063006).

References

[1] J. H. Jung, *Automatic Assignment of Protein Function with Supervised Classifier*, 2008.

[2] E. Elsayed, K. Eldahshan, and S. Tawfeek, "Automatic evaluation technique for certain types of open questions in semantic learning systems," *Human-Centric Computing and Information Sciences*, vol. 3, article 19, 2013.

[3] M. Ashburner, C. A. Ball, J. A. Blake et al., "Gene ontology: tool for the unification of biology. The Gene Ontology Consortium," *Nature Genetics*, vol. 25, pp. 25–29, 2000.

[4] E. Camon, M. Magrane, D. Barrell et al., "The gene ontology annotation (GOA) database: sharing knowledge in uniprot with gene oncology," *Nucleic Acids Research*, vol. 32, pp. D262–D266, 2004.

[5] S. Hunter, P. Jones, A. Mitchell et al., "InterPro in 2011: new developments in the family and domain prediction database," *Nucleic Acids Research*, vol. 40, no. 1, pp. D306–D312, 2012.

[6] E. Quevillon, V. Silventoinen, S. Pillai et al., "InterProScan: protein domains identifier," *Nucleic Acids Research*, vol. 33, no. 2, pp. W116–W120, 2005.

[7] D. M. A. Martin, M. Berriman, and G. J. Barton, "GOtcha: a new method for prediction of protein function assessed by the annotation of seven genomes," *BMC Bioinformatics*, vol. 18, no. 5, article 178, 2004.

[8] G. Zehetner, "OntoBlast function: from sequence similarities directly to potential functional annotations by ontology terms," *Nucleic Acids Research*, vol. 31, no. 13, pp. 3799–3803, 2003.

[9] A. Conesa, S. Götz, J. M. García-Gómez, J. Terol, M. Talón, and M. Robles, "Blast2GO: a universal tool for annotation, visualization and analysis in functional genomics research," *Bioinformatics*, vol. 21, no. 18, pp. 3674–3676, 2005.

[10] L. B. Koski, M. W. Gray, B. F. Lang, and G. Burger, "AutoFACT: an automatic functional annotation and classification tool," *BMC Bioinformatics*, vol. 6, article 151, 2005.

[11] S. F. Altschul, T. L. Madden, A. A. Schäffer et al., "Gapped BLAST and PSI-BLAST: a new generation of protein database search programs," *Nucleic Acids Research*, vol. 25, no. 17, pp. 3389–3402, 1997.

[12] M. Malkawi and O. Murad, "Artificial neuro fuzzy logic system for detecting human emotions," *Human-Centric Computing and Information Sciences*, vol. 3, article 3, 2013.
[13] K. Salim, B. Hafida, and R. S. Ahmed, "Probabilistic models for local patterns analysis," *Journal of Information Processing Systems*, vol. 10, no. 1, pp. 145–161, 2014.
[14] A. Al-Shahib, R. Breitling, and D. Gilbert, "Feature selection and the class imbalance problem in predicting protein function from sequence," *Applied Bioinformatics*, vol. 4, no. 3, pp. 195–203, 2005.
[15] Y. Freund and R. Schapire, "A short introduction to boosting," *Journal of Japanese Society for Artificial Intelligence*, vol. 14, no. 5, pp. 771–780, 1996.
[16] C. P. John, "Sequential minimal optimization: a fast algorithm for training support vector machines," Tech. Rep. MSR-TR-98-14, 1998.
[17] C. Chang and C. Lin, "LIBSVM: a Library for support vector machines," *ACM Transactions on Intelligent Systems and Technology*, vol. 2, no. 3, article 27, 2011.
[18] E. B. Jo, J. H. Lee, S. Y. Park, and S. M. Kim, "Predicting osteoporosis and osteoporotic fractures by analyzing the fracture patterns and trabecular microarchitectures of the proximal femur," *Journal of Convergence*, vol. 5, no. 1, pp. 1–8, 2014.
[19] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: synthetic minority over-sampling technique," *Journal of Artificial Intelligence Research*, vol. 16, pp. 321–357, 2002.
[20] M. Kubat and S. Matwin, "Addressing the curse of imbalanced training sets: one-sided selection," in *Proceedings of the 14th International Conference on Machine Learning*, pp. 179–186, 1997.
[21] I. Guyon and A. Elisseeff, "An introduction to variable and feature selection," *Journal of Machine Learning Research*, vol. 3, pp. 1157–1182, 2003.
[22] S. Hong and J. Chang, "A new k-NN query processing algorithm based on multicasting-based cell expansion in location-based services," *Journal of Convergence*, vol. 4, no. 4, pp. 1–6, 2013.
[23] A. James, B. Mathews, S. Sugathan, and D. Raveendran, "Discriminative histogram taxonomy features for snake species identification," *Human-Centric Computing and Information Sciences*, vol. 4, article 3, 2005.
[24] W. H. Kao, B. S. Liou, W. H. Shen, and Y.L. Tsou, "Applying Boolean logic algorithm for photomask pattern design," *Journal of Convergence*, vol. 4, no. 3, pp. 25–30, 2013.

Hindawi Publishing Corporation
The Scientific World Journal
Volume 2014, Article ID 781014, 9 pages
http://dx.doi.org/10.1155/2014/781014

Research Article

A Rhythm-Based Authentication Scheme for Smart Media Devices

Jae Dong Lee,[1] Young-Sik Jeong,[2] and Jong Hyuk Park[1]

[1] *Department of Computer Science and Engineering, Seoul National University of Science and Technology, Seoul 139-743, Republic of Korea*
[2] *Department of Multimedia Engineering, Dongguk University, Seoul 100-715, Republic of Korea*

Correspondence should be addressed to Jong Hyuk Park; parkjonghyuk1@hotmail.com

Received 12 March 2014; Accepted 19 April 2014; Published 7 July 2014

Academic Editor: Jason C. Hung

In recent years, ubiquitous computing has been rapidly emerged in our lives and extensive studies have been conducted in a variety of areas related to smart devices, such as tablets, smartphones, smart TVs, smart refrigerators, and smart media devices, as a measure for realizing the ubiquitous computing. In particular, smartphones have significantly evolved from the traditional feature phones. Increasingly higher-end smartphone models that can perform a range of functions are now available. Smart devices have become widely popular since they provide high efficiency and great convenience for not only private daily activities but also business endeavors. Rapid advancements have been achieved in smart device technologies to improve the end users' convenience. Consequently, many people increasingly rely on smart devices to store their valuable and important data. With this increasing dependence, an important aspect that must be addressed is security issues. Leaking of private information or sensitive business data due to loss or theft of smart devices could result in exorbitant damage. To mitigate these security threats, basic embedded locking features are provided in smart devices. However, these locking features are vulnerable. In this paper, an original security-locking scheme using a rhythm-based locking system (RLS) is proposed to overcome the existing security problems of smart devices. RLS is a user-authenticated system that addresses vulnerability issues in the existing locking features and provides secure confidentiality in addition to convenience.

1. Introduction

Recently, extensive studies have been conducted on smart devices with touch screens in various fields. Some of the examples of smart devices with touch screens include tablets, smartphones, smart TVs, smart refrigerators, and smart media devices. In particular, a smartphone is a representative example of the capability of smart devices to provide a range of functionality despite device miniaturization. This has happened because smartphones continuously evolve as more smartphones with advanced performance capabilities are introduced in the market. Smart devices provide not only several basic functions such as a telephone, alarm clock, notes, schedule, and health management but also additional entertainment features such as books, movies, music, and shopping. They also provide various business functions such as mobile office, real-time SNS, and payment manager to improve business efficiency, and, in particular, big data processing based on smart devices with mobile cloud computing infrastructure. Although miniaturization and the lightweight feature of smart devices can provide users with the convenience of portability, smart device has potential risks of being lost or stolen [1–9]. Accordingly, a countermeasure to mitigate risks on smart devices loss or theft is required now more than ever. Smart devices also have critical data. Hence, they expose users to potential losses due to data leakage and malicious attacks. To protect confidential data, smart devices provide many forms of locking features such as drag, motion, pattern, password, personal identification number (PIN), and face, fingerprint, or a combination of face and voice recognition. However, they are less secure and highly vulnerable to shoulder surfing or smudge attacks [10–16].

In this paper, a novel locking scheme called rhythm locking system (RLS) is proposed to provide a convenient locking

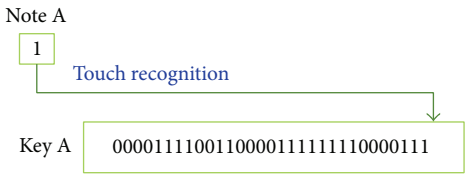


Figure 1: Generation of a single key.

activity using rhythm while overcoming the vulnerability of basic locking functions. RLS is a user authentication system that provides secured confidentiality and convenience using unique rhythms set by the user. It also provides a simple interface, thereby enabling easy locking and fast unlocking.

The remainder of this paper is organized as follows. In Section 2, security authentication systems and basic embedded locking features are discussed. In Section 3, the locking scheme of RLS, proposed in this paper, is explained. Next, in Section 4, the design of RLS is detailed and its implementation is described in Section 5, followed by its performance evaluation in Section 6. Finally, the conclusion and future research activities are described in Section 7.

2. Related Works

In this section, we discuss the basic embedded locking features used in smart devices and various secure authentication systems. Security authentication systems and their descriptions are summarized in Table 1. Basic locking features embedded in smart devices are summarized in Table 2.

3. Locking Scheme of RLS

3.1. Key Generation. In this paper, we propose a RLS which uses touch rhythm as a secret pattern in a smart media device, which is dependent on auditory and behavior memory. The RLS receives a touch rhythm via a touch screen from a user and defines a track to record this rhythm. At the same time, unit time is measured for tracks. The measurement period ranges from first touching time to the configured time. Figure 1 shows a single key created for a touch button called A.

Figure 2 shows the generation of a union key using multiple tracks to increase the complexity of the secret pattern stored internally. A union key for tracks is generated through the touch recognition of four buttons, A, B, C, and D, using the matching table summarized in Table 3.

This method is distinctively different from the existing button pressing password setup. The available number of rhythm patterns for key generation increases exponentially depending on the time precision setup. Thus, even if malicious users know the positions of the buttons, it is very difficult to infer a user's unique rhythm pattern.

3.2. Authentication Process. The RLS authentication consists of four steps conducted using four modules. Figure 3 shows the RLS authentication process. In Step 1, a single key is generated through user input value from the interface. In Step 2, a single union key is generated from the keys generated in Step 1. In Step 3, after the key generation process, improper noise is filtered for authentication. In Step 4, authentication is performed by comparing the stored rhythm pattern with a noise-filtered union key from Step 3.

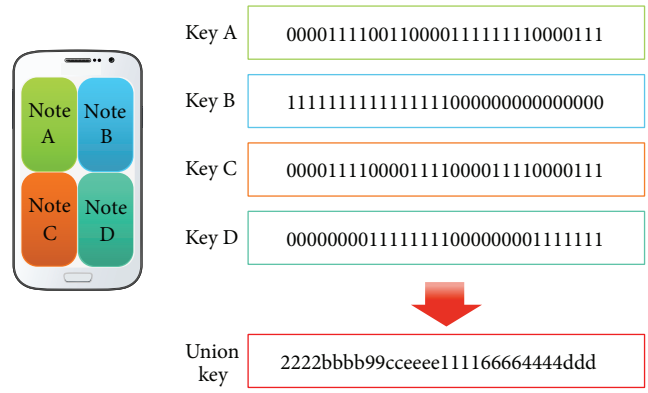


Figure 2: Generation of a union key using a single track for each key.

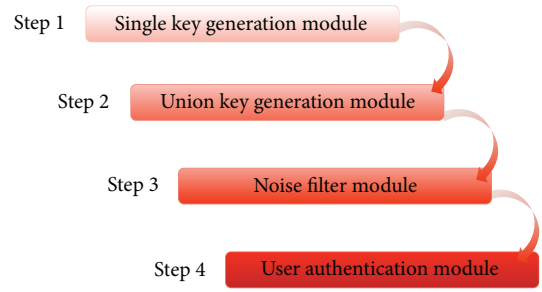


Figure 3: RLS authentication process.

4. Design of RLS

The RLS consists of six main components. The first component is largely in terms of functionality and the second is the user interface. The third component is the time resolution inspector (TRI) that measures time precision of the RLS. The fourth component is a key manager (K-manager) that manages the entered rhythm patterns. The fifth component is a lock service (L-service) that maintains and manages the locking service of the RLS, and the final component is a handler that delivers information for the visualization of activities. Figure 4 shows the overall architecture of the RLS.

The *user interface* component is divided into two modules, that is, rhythm and setting. Rhythm consists of four interfaces, Note A, Note B, Note C, and Note D, to set up rhythm patterns from the user. Settings are configured to select one of the three levels, high (H), medium (M), and low (L), depending on the acceptable error range and precision that are set for rhythm pattern input.

TRI measures the time interval for which the input is detected, according to the time precision set in the user interface. These measured values consist of a pair of note

Table 1: Security authentication systems and their descriptions.

Security authentication system	Description
Something you know	This authentication mode relies on an end user's memory. That is, this mode depends on an individual's memory. In general, users refer to personal information when setting a secret key. Although it has the easy-to-remember advantage, it is vulnerable because malicious attackers can easily take advantage. It can incur additional damage due to leakage of the key as the authentication process can be exposed because of user carelessness. Using this method, a user must memorize the key. If a user forgets the secret key, even a rightful user cannot access the system or services.
Something you have	This authentication mode uses object(s) that a user owns. For example, objects such as barcodes, QR codes, magnetics, and RFIDs are used. That is, this mode depends on the object(s) that a user possesses. If the object is possessed always, this mode provides convenience of authentication and relatively less leakage risk than something you know. However, this method is somewhat inconvenient as the user must always possess the object. If the object is lost or stolen, additional damage can be incurred from malicious attackers. Further, if the object is damaged, a rightful user cannot access the system or services.
Something you are	This authentication mode uses biometric information. This mode uses two different types of techniques: (a) recognition of physiological information and (b) recognition of behavior patterns. A scheme of recognizing physiological information uses individual characteristics of the user. For example, fingerprint recognition, iris recognition, vein recognition, face recognition, voice recognition, and palm print recognition are used. That is, this mode depends on a user's unique biological characteristics. Identity theft by malicious attackers is nearly impossible, and a risk of loss or change is extremely low, which is important for security. A prerequisite for such a method is to have very high recognition precision. If recognition rate is low, authentication of malicious attackers, who have similar or mimicked personal characteristics, can successfully gain access to the system.

Table 2: Basic locking features embedded in smart devices.

Locking system	Description
Pattern lock	This authentication system uses end user's visual memory. Using nine points in a three-by-three grid, a user creates a drag pattern. This method belongs not only to the something you know category, which is based on memory, but also to the behavior pattern recognition category, since it utilizes finger motion memory. The number of available secret patterns in this system is 388,912, which is relatively small due to the limited and fixed arrangement of the nine possible points. This method can be vulnerable to a brute force attack if a user creates a drag pattern using a fewer than suggested number of points to unlock the screen faster. It is also vulnerable to a shoulder surfing attack by malicious attackers due to the visual aspects of a drag pattern. Finally, it is vulnerable to the smudge attack as well, which uses the characteristics of touch screen, in case of theft.
Face recognition	This authentication mode uses biometric information. This method depends on the camera in smart devices and has the advantage of requiring additional memory or management of the locking key due to unique characteristics. However, unlocking the locked screen could be difficult not only due to low performance of the embedded camera but also due to environmental factors (e.g., face recognition range can be limited because of the amount of ambient light). Further, it is vulnerable to the application of similar faces or recognition using photos and videos, which is why this method, in general, is rarely used.
Password	This authentication system uses visual memory and is familiar to most users. Passwords are used by offering a virtual keypad where a combination of alphabetic, numeric, and special characters can be used. Security is dependent on the strength of the password, which depends on the combination of chosen characters. If a password is considerably short in length, for quicker screen unlocking, then it could be vulnerable to a smudge or shoulder surfing attack. If the length of password is considerably long, the user may experience password memory loss possibly due to confusion. Password is also vulnerable to the dictionary attack if the attacker has access to the user's personal information.
PIN	This authentication system has yet to overcome confusion due to the complicated combination of characters used for a password. PIN uses only a numerical value from 0 to 9. Further, it uses combinations of only four numbers so that the available PIN count is less than 10,000, which is considerably small. Thus, it is vulnerable to a brute force attack. It is also vulnerable to shoulder surfing and smudge attacks due to generated traces and the visual nature of pressing four numbers.

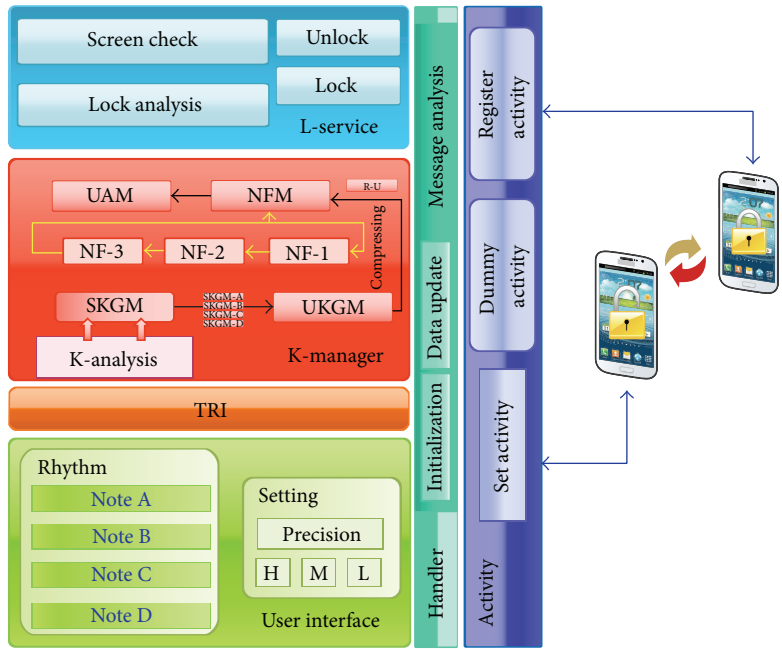


Figure 4: Architecture of RLS.

Table 3: Matching table for generation of union key.

Key combination	Union note
Non	0
A	1
B	2
C	3
D	4
A + B	5
A + C	6
A + D	7
B + C	8
B + D	9
C + D	a
A + B + C	b
A + B + D	c
A + C + D	d
B + C + D	e
A + B + C + D	f

types and times followed by the inputs of Note A, Note B, Note C, and Note D, which are transferred to a K-manager.

K-manager consists of four modules, that are, key analysis (K-analysis), single key generation module (SKGM), union key generation module (UKGM), noise filter module (NFM), and user authentication module (UAM). K-analysis analyzes a pair of data received from the TRI and classifies them according to the note type. SKGM converts a classified note from K-analysis into a single key. UKGM transforms the converted single key from notes to a single complex union key. NFM performs filtering in three steps, NF-1, NF-2, and NF-3, with respect to the raw union key (R-U), converted in the UKGM, according to the acceptable error range set in the user interface. UAM performs either confirmation, when rhythm pattern authentication is set up, or comparison with the existing rhythm patterns to unlock the system when the RLS is executed on a smart device.

L-service consists of a screen check that provides a screen according to execution of the RLS operation, a lock analysis that analyzes a locking status, a lock that starts the RLS, and an unlock that stops the RLS.

Handler is responsible for delivering data synchronization and control messages between activity and user interface and between activity and L-service. Message analysis, in handler, analyzes received data and delivers visual information about the activity.

Activity consists of the following modules: register activity for running the RLS by receiving the rhythm pattern values from a user; dummy activity for visualization while the RLS is running; and set activity for input, confirmation of time precision of the RLS, and other setup activities of a user.

5. Implementation of the RLS

The initial screen of the RLS proposed in this paper is shown in Figure 5. Pressing ①, as shown in Figure 5, deletes a previously set rhythm pattern, while pressing ② moves to activity, by which a user can set the noise and precision of rhythm patterns. Pressing ③ moves to activity, by which the user's unique rhythm pattern can be registered.

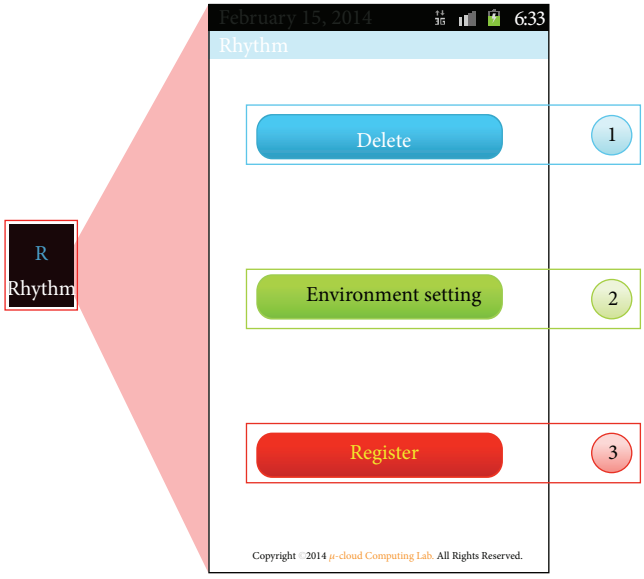


FIGURE 5: Initial screen of the RLS.

Figure 6 shows the setup of activity, which sets an acceptable error of noise and rhythm of the RLS. For noise sensitivity, a level of H, M, and L can be selected according to the filtering level of noise. For rhythm sensitivity, a level of H, M, and L can be selected according to the acceptable error range, which is recognized when a user enters a rhythm. Setup values selected in noise and rhythm sensitivity are applied to the input sensitivity for setting up rhythm patterns and unlocking the screen when the RLS is running. The default setup value is M for both noise and rhythm sensitivity.

Figure 7 shows a screen for input of rhythm patterns to execute the RLS. In Figure 7, ① is the register activity, which consists of Note A, Note B, Note C, and Note D. ② displays the success or failure of recognition in the system when a user enters a rhythm on Note A, Note B, Note C, or Note D. Here, a blue-colored timer progress bar is shown.

Figure 8 shows rhythm inputs of C, D, C, A, and B in order as entered by a user. In Figure 8, ① shows the status that initial input has not been detected, while ② shows that a timer progress bar is displayed as input C is recognized, and ③ shows the status that input D is recognized while a timer progress bar is running. As such, the timer progress bar runs independently while the initial input is recognized. While the timer progress bar is running, each single key for A, B, C, and D is internally generated. Once the timer progress bar terminates, single keys entered up to now are composed into a union key. Since the RLS adds not only the physical Interfaces A, B, C, and D entered by a user but also the logical time, it provides an enhanced security functionality.

6. Performance Evaluation

6.1. Evaluation of Security Strength. We conducted an experiment using a prototype of our proposed scheme. The prototype was developed on Android 4.3 Jelly Bean. In the experiment, we used a smart media device with a Qualcomm Snapdragon 800 2.3 GHz CPU and DDR3 3 GB RAM.

We performed experiments to determine the false acceptance rate (FAR) and false rejection rate (FRR) of the RLS to evaluate its security strength. The standard keys for FAR and FRR are shown in Table 4. For FAR, an arbitrary control key with the same length as the original key was created to perform the comparison. For FRR, an arbitrary control key was created by considering falsely rejected circumstances to perform the comparison. In Key 1, 85/1:10/0:5/2:10/0:5/3:10/0:5/2:10/0:5/3:10/0:5/4:10, in Table 4, "85" refers to a key length. Next, 1:10 indicates that Interface A, in Figure 7, was entered 10 s after input began, while 1 refers to the converted value obtained from the matching table. That is, the first number, 1, indicates the entered interface, while the following number, 10, after colon, refers to the time which elapsed while the input is received.

Figure 9 shows a graph of FAR and FRR with Key 1 in Table 4. As the value length tolerance in the lower left area becomes larger, the allowable error range becomes less when the length of each interface is examined. As the noise recognition range in the lower right area becomes larger, false recognition due to noise becomes more frequent. As numbers with respect to these two increase sequentially, an error rate is calculated by comparing the control key, which is created dynamically, and the standard key. As shown in Figure 9, a 0% error rate was obtained irrespective of the effect of the value length tolerance and the noise recognition range.

Figure 10 shows a graph of FAR and FRR with Key 2 in Table 4. It is configured in the same manner as in Figure 9, reaffirming that error rate decreases as the allowable range of the value length tolerance becomes smaller with respect to Key 2. It also shows that, as the noise recognition range becomes larger, error rate becomes smaller.

Figure 11 shows a graph of FAR and FRR with Key 3 in Table 4. As with Figure 10, as the value length tolerance and noise recognition range become larger, the error rate becomes smaller. Thus, if a user sets a rhythm pattern of the RLS to one more than a specific threshold value, strong security can be ensured.

6.2. Comparison with Existing Locking Schemes. In this section, existing locking schemes such as pattern lock, PIN, and password are compared with the RLS proposed in this paper, with respect to various attacking techniques.

Against a brute force attack, the number of patterns that can be set for locking determines security strength. PIN provides relatively weak security compared to other locking schemes, because it has a limited input length as well as the restriction that only a number can be used. Pattern lock has an advantage in terms of input of various patterns; it provides security stronger than PIN but weaker than password and the proposed RLS. Password and the RLS have similar security strength against brute force attacks.

The shoulder surfing attack begins when a user enters a pattern to unlock the screen. Pattern lock, which uses various patterns but is vulnerable to visual memory, and PIN, which uses fixed arrangement of numbers, both, therefore, provide

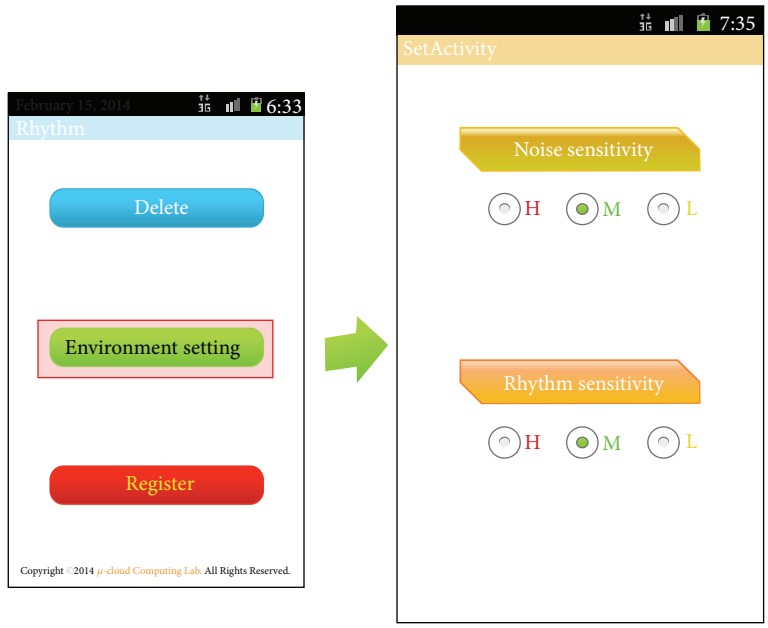


Figure 6: Screen of noise and rhythm setup for the RLS.

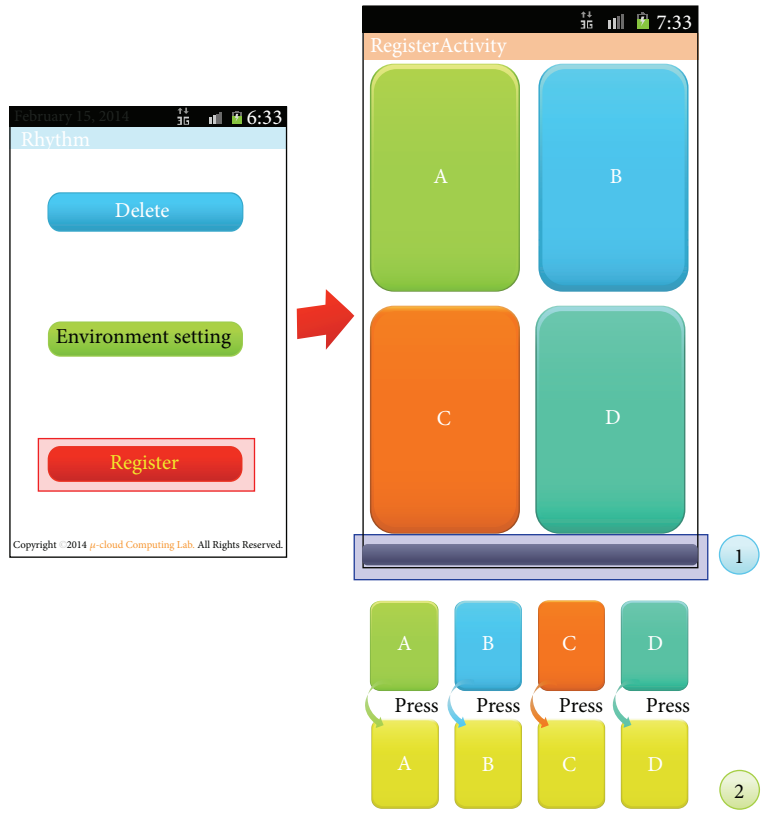


Figure 7: Register activity of the RLS.

weak security. Password is robust against the shoulder surfing attack owing to the large number of possible patterns. The RLS also has robust security, with a rhythm-based locking scheme using a logical time.

The dictionary attack is a method that employs all meaningful words or sentences in a dictionary. The pattern lock and the RLS provide robust security against dictionary attack because they use an entirely different method to set the

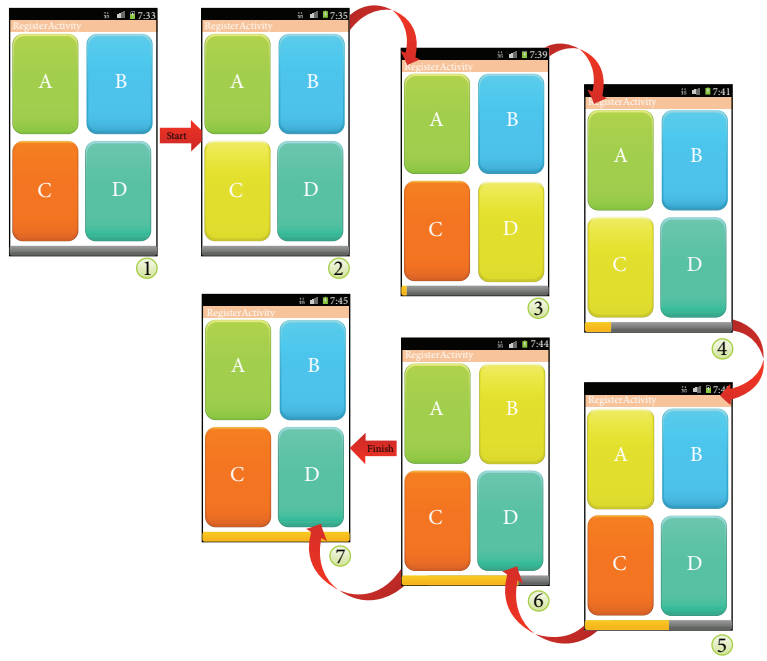


Figure 8: Rhythm pattern setup process in the RLS.

Table 4: Three types of the standard key based on key length.

Type	Rhythm locking pattern	Locking pattern characteristics
Key 1	85/1:10/0:5/2:10/0:5/3:10/0:5/2:10/0:5/3:10/0:5/4:10	(i) Short length (ii) Six single notes (iii) Simple rhythm
Key 2	158/3:20/0:3/2:20/0:3/5:20/0:3/2:20/0:3/3:20/0:3/5:20/0:3/2:20	(i) Medium length (ii) Two composite notes (iii) Six single notes (iv) Simple rhythm
Key 3	220/1:50/0:5/5:10/0:5/2:40/0:15/3:30/0:5/a:60	(i) Long length (ii) Two composite notes (iii) Three single notes (iv) Irregular rhythm

Table 5: Comparison of the existing locking systems and the RLS against various attacks.

	Pattern lock	PIN	Password	RLS
Brute force attack	△	X	O	O
Shoulder surfing attack	X	X	O	O
Dictionary attack	△	X	△	O
Smudge attack	X	X	△	O

O: strong, △: medium, and X: weak.

locking pattern. However, PIN and password are moderately vulnerable because a user may employ meaningful numbers, symbols, or words.

The smudge attack uses a simple trace to discern a locking pattern. A trace is deployed to infer a locking pattern while the user's input is entered to unlock the screen. Pattern lock and PIN show weak security because of easy collection of trace owing to the fixed arrangement on the screen. If a password is set with a long and complicated pattern, it can provide robust security; however, if it is set with a short and simple pattern, it provides weak security. The RLS provides robust security against smudge attacks because it combines physical and logical schemes.

Table 5 shows relative security of the existing locking systems and the RLS against various attack methods. The proposed RLS displays the strongest security against the brute force attack, shoulder surfing attack, dictionary attack, and smudge attack, which are some of the widely used attacks for touch-screen-based smart devices.

7. Conclusion

Smart media devices provide users with a variety of functions leading to their wide use. Most vendors have developed a variety of functions to provide better services to the end users.

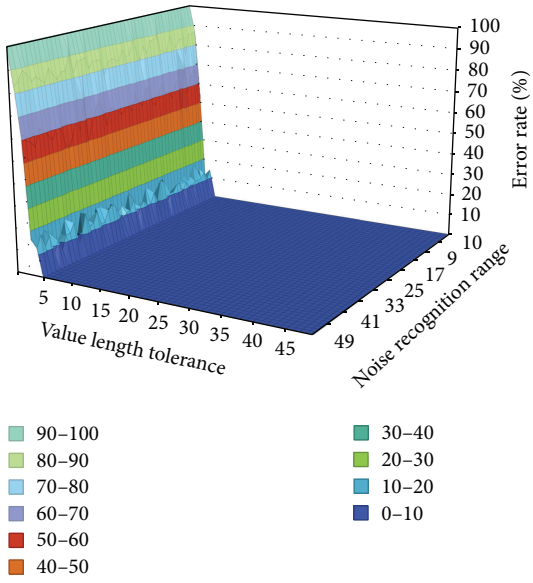


Figure 9: FAR and FRR performance of Key 1.

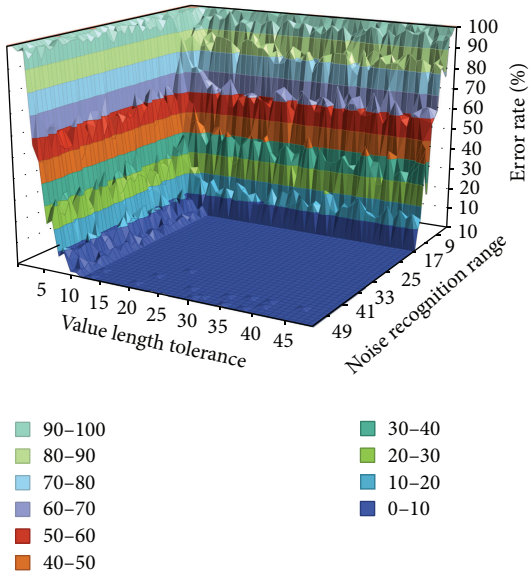


Figure 10: FAR and FRR performance of Key 2.

In particular, smart media devices have made significant advancement in terms of weight reduction, miniaturization, and various functions offered, but the most basic security issues have been ignored. As a result, although many basic locking features are embedded, smart media devices are vulnerable to a number of attacks.

In this paper, we proposed a rhythm-based locking system which considers rhythm as logical behavior. The proposed system provides not only strong security against malicious attackers but also convenience of memory to users. In addition, it is composed of a simple interface structure so that all ages can use it conveniently. Even if pressing positions

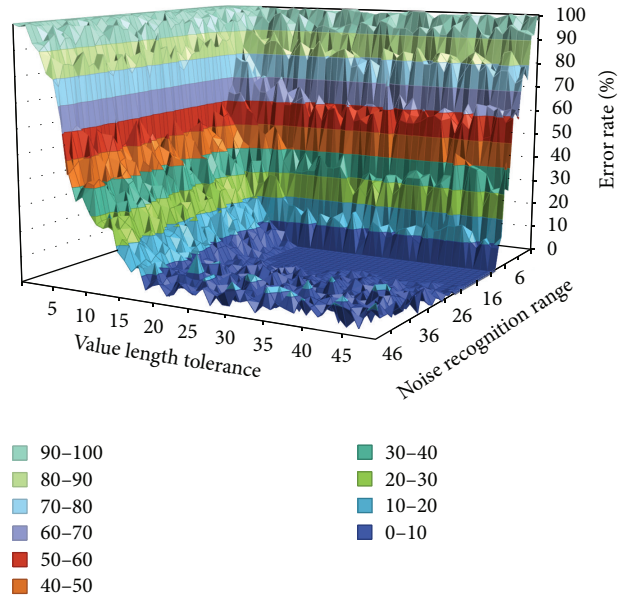


Figure 11: FAR and FRR performance of Key 3.

are exposed, it is extremely difficult to predict precise timings, thereby ensuring high security strength.

In the future, a user authentication structure utilizing various sensors embedded in smart media devices will be studied. Stronger locking functions will be provided by considering the angle or the number of slopes. A unique type of user authentication system structured through unique recognition media will also be researched.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgment

This research was supported by the Basic Science Research Program through the National Research Foundation of Korea (NRF), funded by the Ministry of Education, Science and Technology (2011-0024052).

References

[1] H.-W. Kim, J.-H. Kim, D. Ko, E.-H. Song, and Y.-S. Jeong, "8-Way lock for personal privacy of smart devices based on human-centric," in *Proceedings of the 40th Conference of the KIPS*, vol. 20, pp. 735–737, KIPS, November 2013.

[2] K. Peng, "A secure network for mobile wireless service," *Journal of Information Processing Systems*, vol. 9, no. 2, pp. 247–258, 2013.

[3] J. Ahn and R. Han, "An indoor augmented-reality evacuation system for the smartphone using personalized pedometry," *Human-Centric Computing and Information Sciences*, vol. 2, article 18, 2012.

[4] C.-L. Tsai, C.-J. Chen, and D.-J. Zhuang, "Trusted M-banking Verification Scheme based on a combination of OTP and Biometrics," *Journal of Convergence*, vol. 3, no. 3, pp. 23–29, 2012.

[5] G. Wang, W. Zhou, and L. T. Yang, "Trust, security and privacy for pervasive applications," *Journal of Supercomputing*, vol. 64, no. 3, pp. 661–663, 2013.

[6] Y. Gong, *Implications and Agreement of Smartphone*, vol. 22, no. 4, Korea Information Society Development Institute, Seoul, Republic of Korea, 2010.

[7] ITU-T, "Security aspects of mobile phones," T09 SG17 100407 TD PLEN 1012, April 2010.

[8] C. Mulliner, G. Vigna, D. Dagon, and W. Lee, "Using labeling to prevent cross-service attacks against smart phones," in *Detection of Intrusions and Malware & Vulnerability Assessment: Proceedings of the 3rd International Conference, DIMVA 2006, Berlin, Germany, July 13-14, 2006*, vol. 4064 of *Lecture Notes in Computer Science*, pp. 91–108, Springer, Berlin, Germany, 2006.

[9] M. Park, *The evolution of the mobile phones with touchscreen and the prospect of future: focused on the SRI-Tech [M.S. thesis]*, Incheon University, 2011.

[10] E. Chin, A. P. Felt, V. Sekar, and D. Wagner, "Measuring user confidence in smartphone security and privacy," in *Proceedings of the 8th Symposium on Usable Privacy and Security (SOUPS '12)*, 16, p. 1, Washington, DC, USA, July 2012.

[11] A. J. Aviv, K. Gibson, E. Mossop, M. Blaze, and J. M. Smith, "Smudge attacks on smartphone touch screens," in *Proceedings of the 4th USENIX Conference on Offensive Technologies*, pp. 1–10, August 2010.

[12] B. Chojar, D. Lal, K. Gandhi, and K. Salariya, "Study of smartphone attacks and defenses," *International Journal of Engineering and Computer Science*, vol. 2, no. 4, pp. 1018–1022, 2013.

[13] A. Mylonas, S. Dritsas, B. Tsoumas, and D. Gritzalis, "Smartphone security evaluation—the malware attack case," in *Proceedings of the 8th International Conference on Security and Cryptography (SECRYPT '11)*, pp. 25–36, Seville, Spain, July 2011.

[14] B. Kim and Y. Kim, "A study on emotional interface design based on each Smart-phone application category," *Korea Design Knowledge Society*, vol. 20, pp. 181–192, 2011.

[15] D.-R. Kim and K.-H. Han, "A study on multi-media contents security using smart phone," *The Journal of Digital Policy and Management*, vol. 11, no. 11, pp. 675–682, 2013.

[16] G. Kim and S. Cho, "Security vulnerability trends in smartphones," in *Proceedings of the Korea Computer Congress (KCC '11)*, vol. 37, no. 2, pp. 90–94, November 2011.

Hindawi Publishing Corporation
The Scientific World Journal
Volume 2014, Article ID 769149, 12 pages
http://dx.doi.org/10.1155/2014/769149

Research Article

Real-Time Terrain Storage Generation from Multiple Sensors towards Mobile Robot Operation Interface

Wei Song,[1] Seoungjae Cho,[2] Yulong Xi,[2] Kyungeun Cho,[2] and Kyhyun Um[2]

[1] *College of Information Engineering, North China University of Technology, Beijing 100144, China*
[2] *Department of Multimedia Engineering, Dongguk University, Seoul 100-715, Republic of Korea*

Correspondence should be addressed to Kyungeun Cho; cke@dongguk.edu

Received 5 April 2014; Accepted 8 June 2014; Published 2 July 2014

Academic Editor: Young-Sik Jeong

A mobile robot mounted with multiple sensors is used to rapidly collect 3D point clouds and video images so as to allow accurate terrain modeling. In this study, we develop a real-time terrain storage generation and representation system including a nonground point database (PDB), ground mesh database (MDB), and texture database (TDB). A voxel-based flag map is proposed for incrementally registering large-scale point clouds in a terrain model in real time. We quantize the 3D point clouds into 3D grids of the flag map as a comparative table in order to remove the redundant points. We integrate the large-scale 3D point clouds into a nonground PDB and a node-based terrain mesh using the CPU. Subsequently, we program a graphics processing unit (GPU) to generate the TDB by mapping the triangles in the terrain mesh onto the captured video images. Finally, we produce a nonground voxel map and a ground textured mesh as a terrain reconstruction result. Our proposed methods were tested in an outdoor environment. Our results show that the proposed system was able to rapidly generate terrain storage and provide high resolution terrain representation for mobile mapping services and a graphical user interface between remote operators and mobile robots.

1. Introduction

In recent times, technologies for dynamic terrain reconstruction and modeling using multiple sensors have been extensively researched in order to provide mobile vehicles with the ability to conduct free-space detection and support collision-free navigation [1]. In such applications, datasets received from multiple sensors, including 3D point clouds, video images, global positioning system (GPS) data, and rotation states, are integrated to produce accurate and reliable terrain information.

Light detection and ranging (LiDAR) sensors are widely used to measure distances and capture 3D surfaces using lasers, such as Velodyne [2] and Sick. These sensors collect 3D point clouds that do not contain any color or texture information owing to the characteristics of the laser. As a result, synchronized video images are required to map the color information to the sensed 3D point cloud, thus realizing intuitive terrain visualization towards mobile mapping services.

Conventional real-time visualization systems mostly apply a voxel map or a color mesh to represent a terrain model. A voxel map is generated by integrating the sensed 3D point clouds into regular grids. From the voxel map, a terrain mesh is generated by integrating the top points in the x-z cells into a regular triangular mesh [3]. These methods allocate one color per voxel or vertex so that the resolution of the terrain model is low.

In order to improve terrain visualization so as to rapidly obtain an intuitive representation, real-time terrain modeling and photorealistic visualization systems have been developed [4]. A photorealistic visualization attempts to realistically represent a 3D terrain and object models in the virtual world [5]. To improve the visualization speed, real-time terrain visualization methods have been studied [6, 7]. In recent times, with the rapid development of graphics cards, a graphics processing unit (GPU) with a highly parallel many-core architecture has been widely used in 2D image processing, 3D data analysis, visualization, and other fields [8, 9]. GPU programming can be used to realize a high-speed

and high-quality large-scale terrain reconstruction system [10].

Our study aims to reconstruct an intuitive terrain model from large-scale datasets using limited memory for providing mobile robot operator with a graphical user interface (GUI) of surrounding environment. Typically, the captured video images and sensed point clouds are registered into the terrain model. However, when the sensed datasets are registered incrementally, they become very large in size and exceed the computer memory capacity. Furthermore, the large computational cost incurred for processing large-scale datasets makes terrain modeling and visualization slow. Therefore, it is necessary to develop an effective redundancy removal method for reducing the size of the 3D point cloud in order to realize real-time large-scale terrain reconstruction.

In this paper, we describe a real-time terrain storage generation and intuitive representation system using multiple sensors. The first step for real-time terrain modeling is a redundancy removal method for the large-scale 3D point cloud. To register the sensed datasets into the terrain model having limited memory, we develop a voxel-based flag map as a comparative table for removing redundant points. The compressed point clouds are registered into a nonground point database (PDB) and ground texture database (MDB) using a height histogram method [11]. To visualize the reconstructed terrain model, the nonground PDB is represented using a voxel map, generated by integrating the nonground 3D points into regular grids. The ground MDB is implemented as a node-based terrain mesh. Each node in the mesh contains a certain number of ground surface vertices and a node texture. The node textures are integrated to form a texture database (TDB). To realize real-time TDB generation, the GPU is used to map the triangles of the node texture in parallel. Finally, we represent the reconstructed terrain model by rendering the points in the nonground PDB and overlaying the MDB with the TDB.

The proposed real-time terrain storage generation technique allows a mobile robot to survey, navigate through, and interact with its environment by providing quickly accessible and accurate information regarding the surrounding terrain [12].

The remainder of this paper is organized as follows. In Section 2, we survey related works on terrain modeling and representation methods. In Section 3, we explain the terrain storage generation and representation system. In Section 4, we analyze and evaluate the performance of the proposed multithread-based terrain storage generation system. Finally, in Section 5, we present our conclusions.

2. Related Work

During navigation and interactive tasks, rapid feedback on intuitive representations of a robot's surrounding terrain is required for real-time operation. Conventionally, a voxel map and textured mesh have been applied for this type of terrain modeling. Rovira-Más et al. [13] applied a voxel map to represent a reconstructed terrain model. However, they allocated only one color per voxel, which caused distortions. Sukumar et al. [14] integrated sensed datasets into a texture mesh for terrain reconstruction. However, it is difficult for these systems to process large-scale datasets of the kind obtained in outdoor environments in real time.

To realize real-time transmission and registration of large-scale point clouds with limited memory, researchers investigated data redundancy removal methods to reduce the buffer size of the terrain model [15, 16]. Kammerl et al. [17] applied a voxel map to quantize points into regular grids in order to compress 3D point clouds. The generated voxel map is stored on a hard disk by using an octree data structure. Although this method employs lossy compression, it efficiently removes redundant data using a traversing process to compare whether or not this point is stored. In a large-scale environment with a low-density point cloud, the depth of an octree is large, which causes high data searching complexity. Thus, the data size of each node in an octree is large, which leads to a low compression ratio. For a large-scale terrain model, this traversing process requires substantial computational power. Therefore, an effective and rapid data compression method is necessary for real-time data transmission.

Gingras et al. [18] reconstructed an unstructured surface from a 360° point cloud scan and represented the traversable areas using a compressed irregular triangular mesh. They applied a mesh simplification algorithm to reduce the number of triangles on the large-scale terrain surface. Zhuang et al. [19] proposed an edge-feature-based iterative closest point (ICP) algorithm to extract edge points, which were registered on a 3D roadmap integrated with planar features and elevation information. By using this method, a large number of redundant points of pseudoedges could be removed to realize rapid large-scale point cloud registration. However, the visualization achieved in these studies was not sufficiently intuitive for unmanned vehicle operations. To overcome this limitation, it is necessary to overlay the 3D terrain model with captured video images.

Meanwhile, the image buffers of the reconstructed terrain model become increasingly full when a mobile robot navigates in a large-scale environment. Even when using compression algorithms, such as Huffman coding, JPEG, wavelet filter, and MPEG for 2D images [20, 21], the buffers of the captured video images are registered incrementally and they ultimately become so large that they exceed the memory capacity of the mobile robot. An effective and rapid image registration method is therefore necessary to realize intuitive terrain reconstruction with limited memory consumption.

In this study, we use a voxel map and texture mesh to represent nonground and ground terrain information separately. To improve the performance of terrain storage on the hard disk, we propose a terrain storage generation and updating method that does not need traverse elements stored on the hard disk, thus realizing improved speed. To compress video buffers, we present a TDB generation method by integrating several captured images into a node texture.

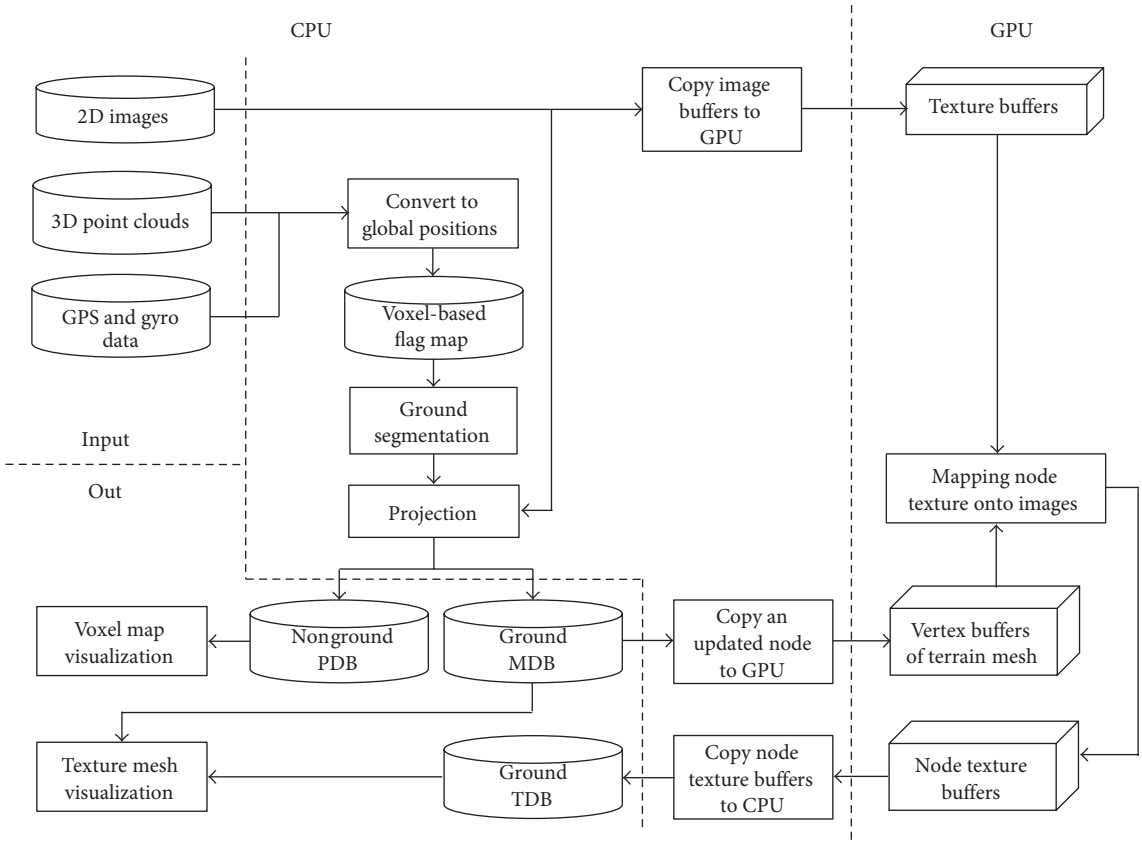


Figure 1: Framework of terrain storage generation system.

3. Terrain Storage Generation from Multiple Sensors

In this section, we develop a terrain storage generation system in which we register large-scale datasets into a terrain model with limited memory in real time. In this system, we employ a 3D point cloud compression method that uses a voxel-based flag map. Then, we register nonground points into a voxel map and ground points into a texture mesh. Next, we employ a GPU-based mapping algorithm to convert the 3D mesh triangle into a 2D image. Finally, the intuitive visualization process is implemented by rendering the generated terrain storage.

3.1. Terrain Storage Generation and Representation System. In this section, we describe the multithread-based terrain storage generation system for mobile robots, as shown in Figure 1. This system involves several processes such as data collection, dataset compression, nonground PDB generation, ground MDB, and TDB generation.

A mobile robot collects 3D point clouds, 2D images, GPS, and rotation states as a real-world interface. The received 3D points are converted into global positions on the basis of the GPS and rotation states. By quantizing the 3D point clouds into regular voxels, we create a voxel-based flag map to remove redundancies.

Some nonground objects have overhanging parts such as roofs and leaves. It is difficult to represent these objects using a height map. In this study, we propose the use of a voxel map to represent nonground objects and a terrain mesh to represent the ground surface. Therefore, we classify these points into ground and nonground using the height histogram method [11] before terrain modeling.

The color data of the nonground voxels in the PDB are computed by the projection from the 3D voxels to the sensed 2D image. From the nonground points, we select a vertex as the top point in an x-z cell and insert it into a terrain mesh. To realize intuitive visualization, we create a texture triangle of several pixels for each triangle in the MDB by mapping the captured images onto the mesh. We apply GPU programming to map these texture triangles that are combined into a TDB. By mapping the generated TDB onto the mesh, a textured mesh is represented.

3.2. Voxel-Based Flag Map. We developed a voxel-based flag map to register 3D points into the terrain model without reduplication. To realize real-time terrain modeling, we register a point into the flag map based on the spatial relation without a traversal process.

The coordinate system of a sensed 3D point $p(x, y, z)$ is centered at the robot position L. Before inserting this point into the flag map, we convert it to a relative position based

on the coordinate system at the center of the flag map. The converted coordinate is formulated as follows:

$$p' = R(p + L_c) + L - L_0, \quad (1)$$

where L_c is the 3D vector from the 3D sensor location to the GPS sensor location, L_0 is the center of the flag map, and R is the mobile rotation matrix.

Subsequently, the converted positions are quantized into a space of regular voxels, as shown in Figure 2(a). The constants w, h, and d are the maximum measurements along the x-, y-, and z-axes of the flag map, respectively. The size of the voxel is defined as μ. In this manner, a flag map has $8hwd$ grids, which represents a space of $8hwd\mu^3$ m^3. Subsequently, we specify a bit-stream to define a voxel-based flag map, as shown in Figure 2(b). If the coordinates of $p'(x', y', z')$ satisfy $|x'| < w\mu$, $|y'| < h\mu$, $|z'| < d\mu$, the voxel index mapped from p' is formulated as follows:

$$v = 4wh\left\lfloor \frac{z'}{\mu} + d \right\rfloor + 2h\left\lfloor \frac{x'}{\mu} + w \right\rfloor + \left\lfloor \frac{y'}{\mu} + h \right\rfloor, \quad (2)$$

where the function $\lfloor N \rfloor$ returns the largest integer that is less than or equal to N.

We allocate a 1-bit variable $f(v)$ for each voxel v. We initialize the flag map by specifying $f(v \in V) = 0$, where the set $V = \{v \in [0, 2h \times 2w \times 2d)\}$. When at least one point exists in the covered area of the voxel v, $f(v) = 1$; otherwise, $f(v) = 0$. After the robot collects several consecutive frames of 3D point clouds, some points are inserted into the same voxel. This causes wasteful duplication of memory if we register these points into the terrain model. To remove redundant points, we register voxels when they are sensed for the first time. Hence, when a new point is converted to a voxel v and $f(v) = 1$, the robot does not register this point in the terrain model.

We register voxels when they are sensed for the first time in order to avoid redundancy. After generating a flag map from several consecutive frames of 3D point clouds, we find that covered voxels exist between the current and the previous sensed frames.

For long-term navigation in an environment, the range of the sensed points exceeds the defined range of the flag map. To solve this problem, we shift the center of the flag map to the position of the vehicle when the vehicle moves to a certain distance. It is necessary to drop the passing information from the memory of the flag map and store new sensed points. In this manner, we utilize a flag map with limited range to represent the information about the dynamic environment surrounding the robot.

3.3. Nonground and Ground Terrain Modeling. It is impossible to sense the points below the ground surface using 3D sensors such as LiDAR sensors. The sensors provide the top points of the ground surface and the surface points of nonground objects. To represent an intuitive ground surface, we apply a texture mesh by mapping the texture onto a digital elevation model (DEM). The road and grass group in Figure 3(a) are represented using a texture mesh.

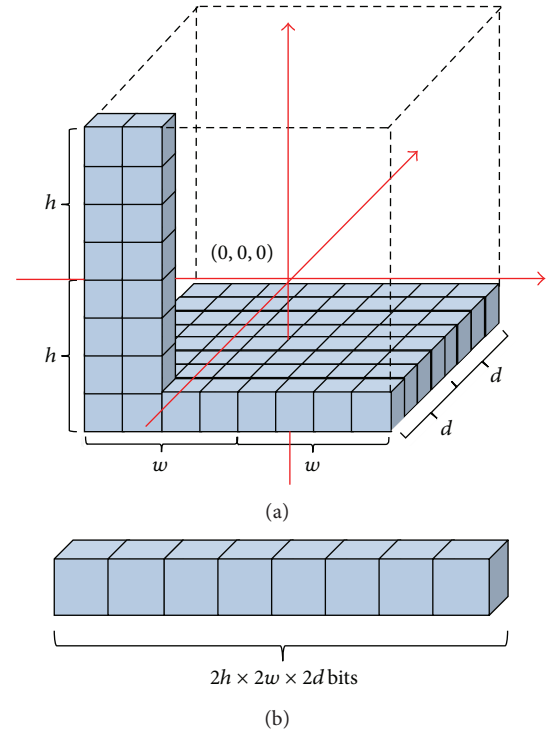


Figure 2: Definition of voxel-based flag map. (a) A space of regular voxels. (b) A buffer stream of $2h \times 2w \times 2d$ bits.

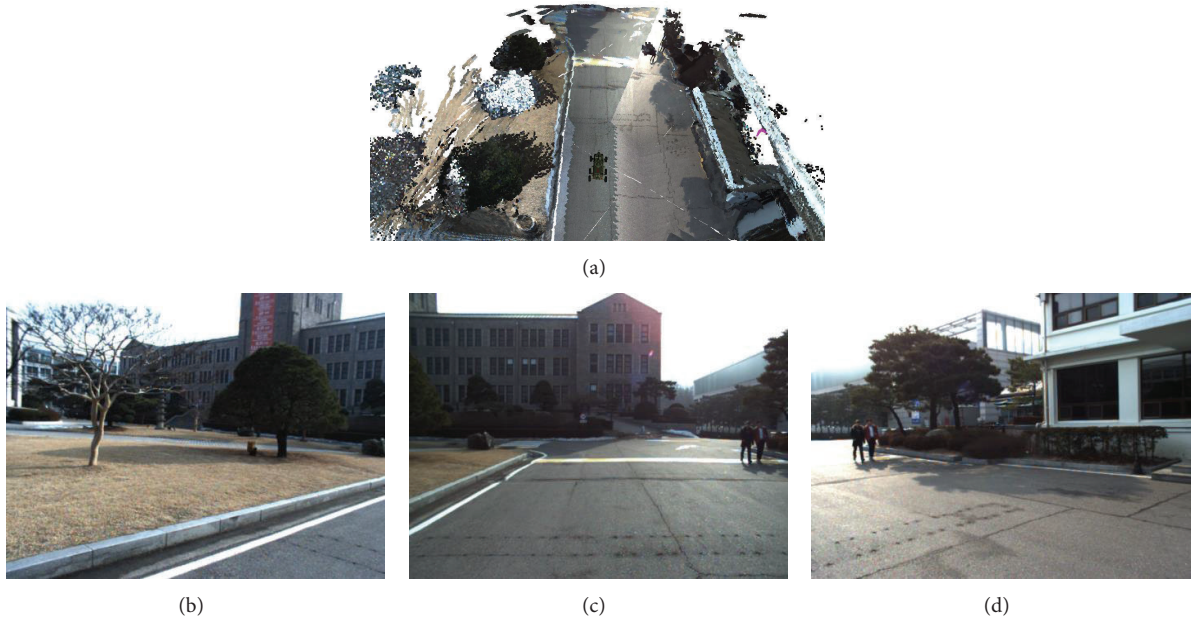
This method provides large-scale terrain with low memory consumption. However, it is difficult to represent objects that have overhanging parts, such as roofs and leaves. To realize real-time terrain modeling, we apply a color voxel map consisting of a list of voxels to represent nonground objects. The trees and building in Figure 3(a) are represented using a color voxel map.

Before terrain modeling, it is necessary to segment the registered points into ground surface and nonground objects using the voxel-based flag map method. Based on the spatial distribution of ground surface and nonground objects, we segment the ground surface using the height histogram method described in [11], which is a fast and dynamic ground segmentation method. The classified nonground and ground voxels are registered into the PDB and the MDB, respectively.

Owing to the characteristics of a LiDAR sensor, the sensed points contain no color information. This makes remote operation inconvenient and less perceptual. To represent the terrain model world with its real appearance, we project the 3D voxels of PDB and the vertices of MDB to the captured 2D images, as shown in Figures 3(b)–3(d).

3.4. TDB Generation. A terrain mesh is always generated by integrating the top points in the x-z cells into a regular triangular mesh. By overlaying the 3D terrain mesh with captured video images, the visualization system provides perceptual imagery of the 3D terrain geometrical model, as shown in Figure 4.

We represent the nonground MDB using a node-based texture mesh. The mesh is generated using several nodes. In



(a) (b) (c) (d)

Figure 3: A terrain reconstruction model. (a) Nonground objects and ground surface representation. (b)–(d) Captured images.

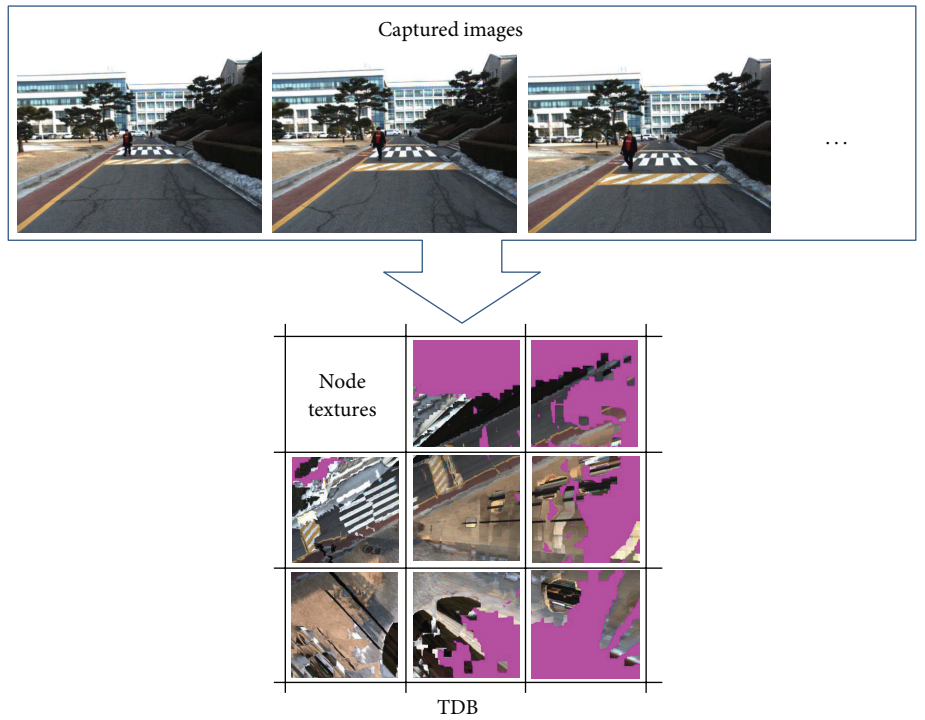
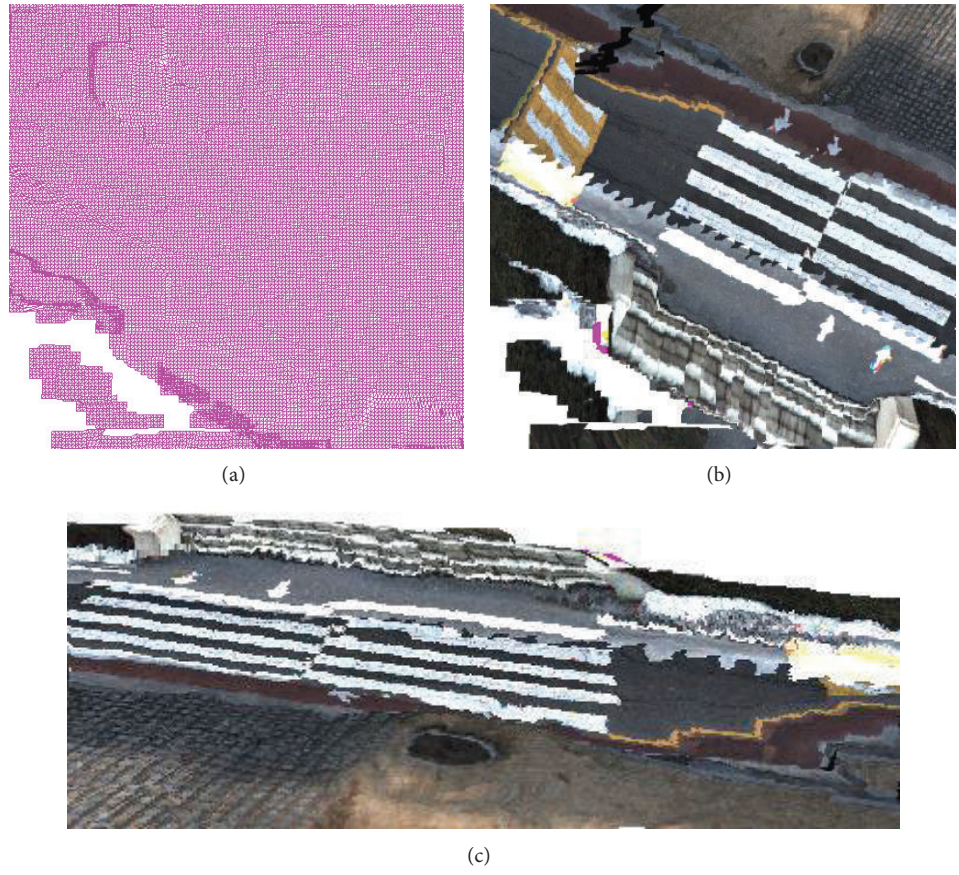


Figure 4: TDB generation from captured images.

our application, each node has 128×128 textured vertices, and the cell size is $0.1 \times 0.1\,m^2$. The height value of each vertex in the mesh is updated with the registered ground 3D voxels. If a new 3D voxel is to be inserted into the reconstructed terrain mesh but is outside the existing nodes, we create a new node to register this voxel.

To represent an intuitive ground surface, we traditionally map the captured images onto the terrain mesh. However, the sensed images are registered incrementally, which become so large that they exceed the memory capacity. To reconstruct a texture terrain mesh using limited memory, we propose a TDB generation method, registering several captured images into node textures without redundant pixel buffers.

We project each triangle in a node mesh, as shown in Figure 5(a), onto the captured 2D images and store the pixel buffers of the mapped triangles in the images. We store these



(a) (b) (c)

Figure 5: Texture mapping for MDB. (a) A node of the terrain mesh. (b) The node texture of (a). (c) Mapping the node texture onto the node mesh.

pixel buffers into a node texture, as shown in Figure 5(b), which is combined with the TDB. By mapping the mesh node with its node texture, a texture mesh is generated, as shown in Figure 5(c).

Figure 6(a) shows the process of node texture generation. For a 3D triangle (p_1', p_2', p_3') in the mesh of that node, we create a 2D triangle $(\theta_1, \theta_2, \theta_3)$ in a node texture, which has a set of triangle pixels. Subsequently, a 2D triangle $(\theta_1', \theta_2', \theta_3')$ in a captured image is projected from the 3D triangle (p_1', p_2', p_3'). We then duplicate triangle (p_1', p_2', p_3') from triangle $(\theta_1', \theta_2', \theta_3')$. After all of the triangles in a node mesh are mapped from the captured images, the node texture is updated and combined with the TDB system.

The pseudocode for triangle duplication is as follows:

$$\begin{aligned}
&\text{for (int } m = 0; m < r; m++) \\
&\quad \text{for (int } n = 0; n < r - m; n++) \\
&\quad\quad \text{dest_pixel}\left(\theta_1 + \frac{m\overrightarrow{\theta_1\theta_2}}{r} + \frac{n\overrightarrow{\theta_1\theta_3}}{r}\right) \\
&\quad\quad = \text{sourc_pixel}\left(\theta_1' + \frac{m\overrightarrow{\theta_1'\theta_2'}}{r} + \frac{n\overrightarrow{\theta_1'\theta_3'}}{r}\right),
\end{aligned} \tag{3}$$

where r is the resolution of the destination triangle and the symbol $\overrightarrow{\theta_1\theta_2}$ stands for a vector from θ_1 to θ_2. As shown in Figure 6(b), each source pixel in a captured image is mapped onto its destination pixel in a node texture.

When the robot navigates a large-scale environment, a large number of mesh nodes are generated and a large number of triangles of these nodes are mapped from the captured video images. To realize real-time TDB generation, we apply GPU programming to implement the mapping process in parallel. The TDB generation process using the GPU is shown in Figure 1. After the current captured images are registered into the GPU memory, we copy the mesh and texture of an updated node and the current captured image to GPU memory. Next, we project each triangle of the node mesh onto the captured images in order to acquire the mapped triangles within the images. Then, we duplicate the mapped triangle to the node texture buffers. Next, we load the updated node texture in the GPU memory to the TDB of the terrain model in the CPU memory. Finally, we render the terrain model by overlaying the MDB with the TDB.

4. Experiments and Analysis

In this section, we analyzed the performance of the proposed real-time terrain storage generation and intuitive representation system.

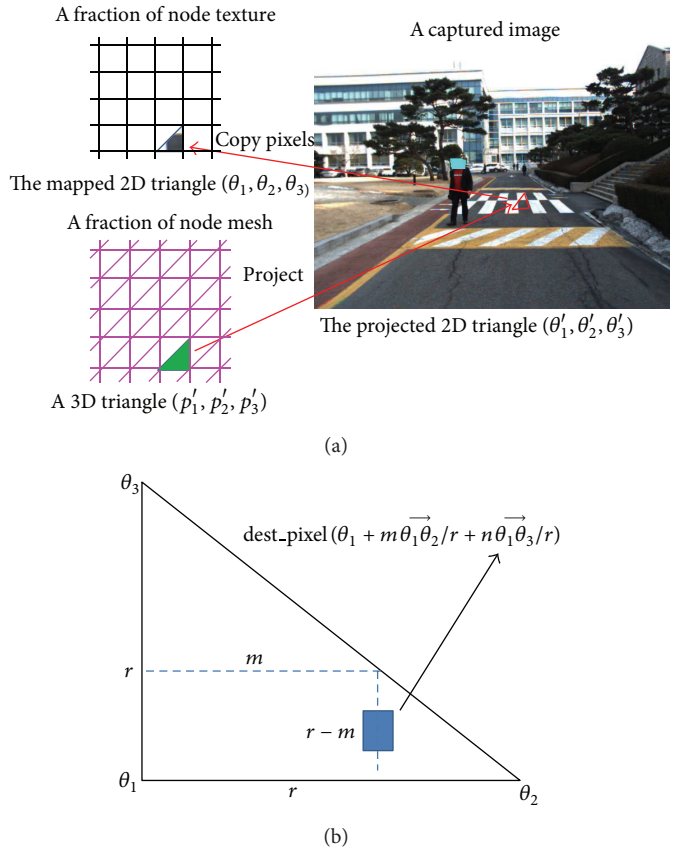


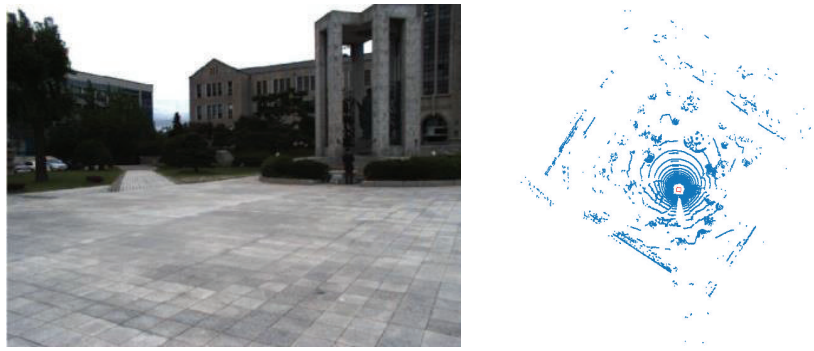
FIGURE 6: Node texture generation method. (a) Projection from the triangles of a node mesh onto a captured image. (b) Texture triangle duplication.

4.1. Experimental Setup. We carried out the experiments using a mobile robot with integrated sensors, as shown in Figure 7, including a GPS, a gyroscope, three video cameras, and an HDL-32E Velodyne LiDAR sensor. The proposed algorithms were implemented on a computer with a 2.82 GHz Intel Core 2 Quad CPU, a GeForce GTX 275 graphics card, and 4 GB RAM. We drove the robot around a 100 × 100 m^2 outdoor environment in Dongguk University campus, including vehicles, buildings, and vegetation.

We used three GC655 VGA CCD cameras to capture images mounted in front of the robot. We captured RGB-color images with a resolution of 659 × 493 RGB pixels, as shown in Figure 8(a). The HDL-32E provided 32 × 12 points in a packet, for a total packet time of 552.96 μs. It gives approximately 1,808 packets of 3D point clouds per second, which contain 694,292 points. The measurement range is 5–100 m. The valid data range was approximately 70 m from the robot. The field of view is +10.67° to −30.67° in vertical and 360° in horizontal. The angular resolution is 1.33° in vertical. Using the HDL-32E, the received datasets from 180 packets of point clouds are represented in Figure 8(b). The 3D data collection duration is 0.1 s. To realize real-time terrain modeling, the duration of each proposed modeling algorithm needs to be less than 0.1 s for 180 packets.

To integrate measurement sections of LiDAR scans with accurate transformation, we applied a GPS receiver and an inertial measurement unit (IMU) to detect the absolute position and orientation of the LiDAR sensor. In our project, we utilized a GPS-aided, IMU-enhanced MTi-G-700 sensor that offers high-quality orientation and position data. The MTi-G-700 is mounted on the vehicle to report the rotational states, including yaw, pitch, and roll values, and the position, including east, north, and elevation values.

FIGURE 7: Experimental mobile vehicle.



(a) (b)

Figure 8: Datasets received from multiple sensors. (a) A captured 2D image. (b) 180 packets of point cloud received from HDL-32E.

4.2. Point Registration Performance Using Voxel-Based Flag Map. By registering the sensed 3D point clouds into the voxel-based flag map, we remove the redundancy for real-time terrain reconstruction with low memory. In our project, we defined a voxel as a $10 \times 10 \times 10\,cm^3$ cube. We drove the robot around an outdoor area at an average speed of 3.87 m/s. To demonstrate the performance of the flag map, we recorded the counts of the sensed points and the registered voxels, as shown in Figure 9(a).

At the beginning of testing, the flag map was empty. Many points were registered into some voxels that were sensed for the first time. Therefore, the counts of new registered voxels at the beginning are more than those at other navigation times. Figure 9(b) shows the total sensed points and registered voxels during vehicle navigation. We can see that the flag map removed ~90.06% redundant points from the sensed datasets after the robot navigated for 10 s. Meanwhile, we can see that the faster the vehicle moved, as shown in Figure 9(c), the more voxels in front were sensed and registered. Figure 9(d) shows the point cloud registration durations during robot navigation. The registration duration for 180 packets was 0.023 s, less than 0.1 s, which satisfied the real-time requirement. When the robot speeded up at the time 12 seconds, there were many new voxels registered in the voxel map. The new registered voxels caused a surge of the terrain database registration duration, as shown in Figure 9(d).

We applied the position of the first registered point to represent a voxel in the terrain model. By using the flag map, we created a voxel map as shown in Figure 10. In Figure 10(a), the sensed validate point count is 600,952 and the number of voxels in the flag map is 91,546. In Figure 10(b), the point count is 6,644,033 and the voxel count is 660,460.

4.3. Performances of MDB and TDB Generation Using GPU. After the voxels were registered in the terrain model, we applied the height histogram method to estimate the height range of the ground surface. Subsequently, we segmented the points into the ground dataset and nonground dataset using the estimated height range as a threshold. In our project, we implemented the ground segmentation procedure once for 180 packets of registered voxels. The ground segmentation duration was 0.5271 ms on average, as shown in Figure 11, which is much faster than 0.1 s and satisfies the real-time requirement.

We used a voxel map to represent nonground data and a texture mesh to represent ground data. By projection from the vertices of the terrain model onto the captured images, we computed the color information for the nonground PDB and ground MDB, as shown in Figure 12. After we classified the ground data from the voxels in Figure 10(b), the TDB is generated using the node texture mapping method described in Section 3.4. By overlaying the node textures in the TDB onto the MDB, we represented an intuitive terrain model as shown in Figure 12(a). Figure 12(b) provides a top view of the terrain model reconstructed at another environment. The regions in the MDB which could not be mapped onto the captured images are represented as the pink regions in Figure 12.

We captured and registered three images of 659×493 pixels every 0.1 s into the memory. If we store the captured images into the memory incrementally, they exceed the computer memory capacity. To solve this problem, Huber et al. [6] applied a color mesh, as shown in Figure 13(a), where a vertex contains RGB color information. As a result, it is not necessary to store the captured images. In our application, a node of the color mesh has 128×128 vertices and 128×128 colors. Using the color mesh, the projected pixels in the images from the vertices of the mesh were only stored, as shown in Figure 13(b). However, many pixels were lost, causing distortion and a blurry scene. Using the TDB, we defined a texture of $128 \times 128 \times 4 \times 4$ XRGB pixels for a node. In this manner, an intuitive ground surface was represented, as shown in Figure 13(c). If the resolution of the node textures increased, the CPU-GPU memory copying speed for updating the node texture buffers became low. To balance the visualization quality and system processing speed, we defined 4×4 pixels for a grid in the node textures.

We compared the sizes of the TDB and the captured images, as shown in Figure 14. At the beginning of the testing, the robot sensed 28 nodes for the MDB and allocated 28 node textures for the TDB. After 20 s, 600 images were captured and 81 nodes were registered in the MDB. The TDB buffer size of these nodes was 81.0 MB, generated from 557.7 MB of

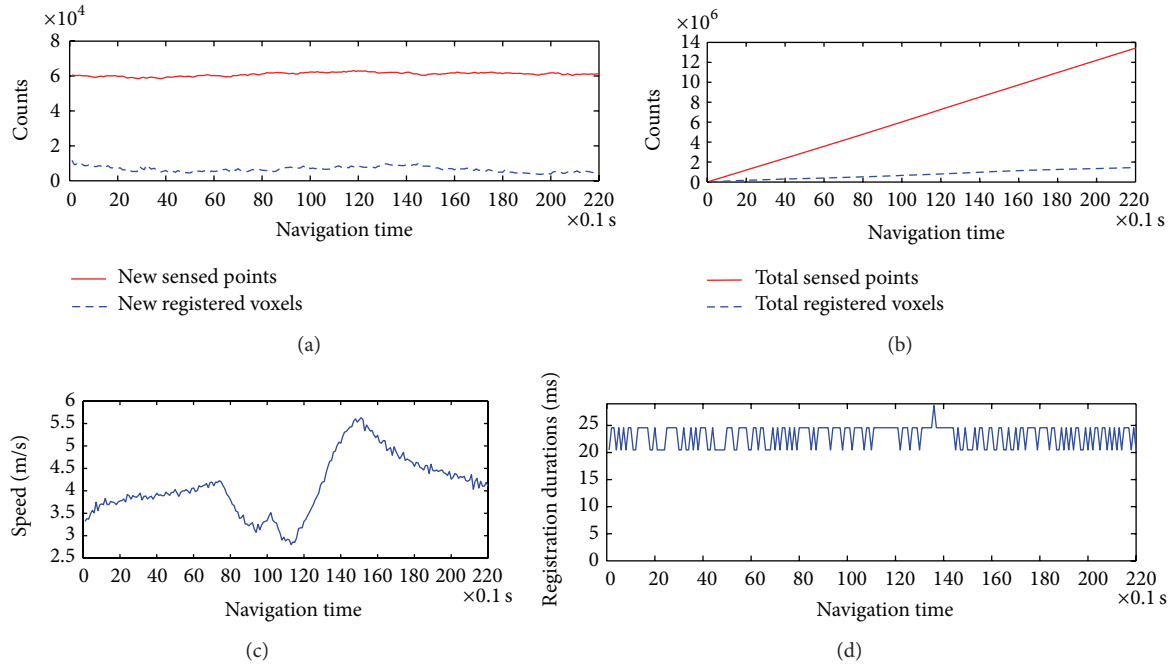


Figure 9: 3D point cloud registration using the voxel-based flag map. (a) Counts of new sensed points and registered voxels. (b) Counts of total sensed points and registered voxels. (c) Driving speed. (d) Registration durations of point clouds.

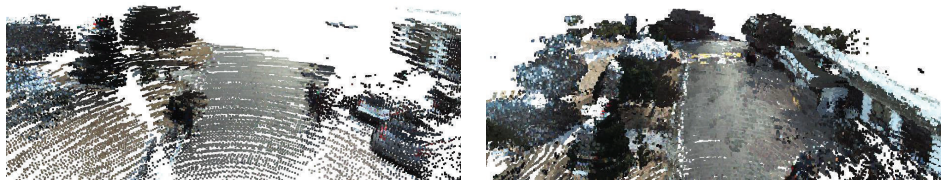


Figure 10: Fractions of generated voxel maps by using the flag map. (a) Registration from 1,808 packets of 3D point clouds. (b) Registration from 18,080 packets of 3D point clouds.

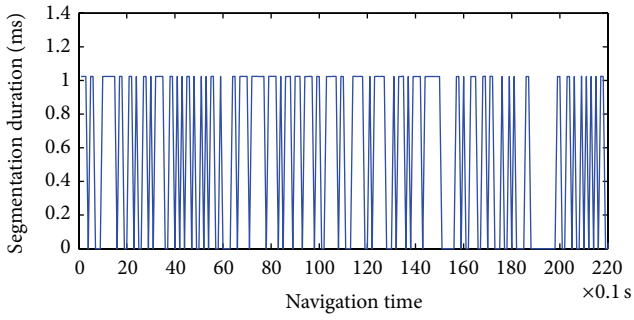


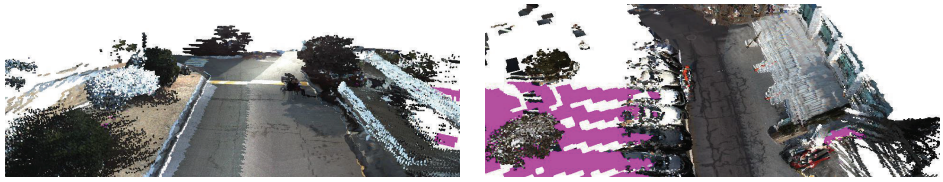
Figure 11: Segmentation duration using height histogram method.

video images. The results demonstrate that the video images captured by the GC655 cameras were registered to the TDB with low memory. In our projects, we rendered 9 nodes of the ground mesh surrounding the robot. The other nodes were stored on a hard disk. Therefore, only 9.31 MB of memory space was used to represent the surrounding ground mesh.

To speed up the computation of the TDB generation, we used GPU programming to implement the mapping process in parallel. The duration of the total mapping process was reduced to 17.29 ms on average for every 180 packets, as shown in Figure 15. Because the capability of the applied GTX 275 graphics card is not high, the memory copying process took around 11.83 ms; this was longer than the mapping process in the GPU, which took 5.46 ms on average. TDB generation took much lower than 0.1 s, satisfying the real-time requirement.

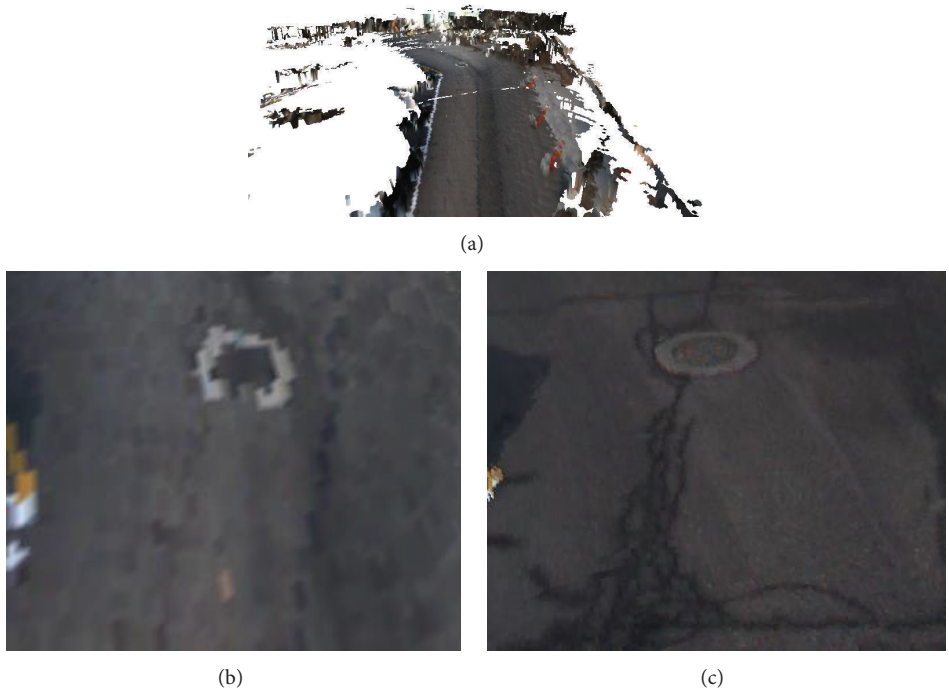
After updating the nonground PDB, ground MDB, and TDB, we utilized the DirectX SDK to render the terrain model as shown in Figure 16. The three images below the terrain representation results of (a) and (b) were captured by the three cameras.

The terrain reconstruction and visualization speed is shown in Figure 17. We compared the GPU-based texture mesh generation method with the CPU-based method and the color mesh method. Using the CPU, we reconstructed and rendered the terrain model at a speed of 7.12 fps on average. Using the GPU, the speed was improved to 18.85 fps.



(a) (b)

Figure 12: Terrain reconstruction results using ground segmentation method. (a) Nonground voxel map and ground texture mesh representation. (b) Reconstruction result of another environment.



(a)

(b) (c)

Figure 13: Ground surface representation using a color mesh. (a) A color mesh reconstructed from the segmented ground voxels. (b) A fraction of the color mesh of (a). (c) A fraction of the texture mesh of Figure 12(b).

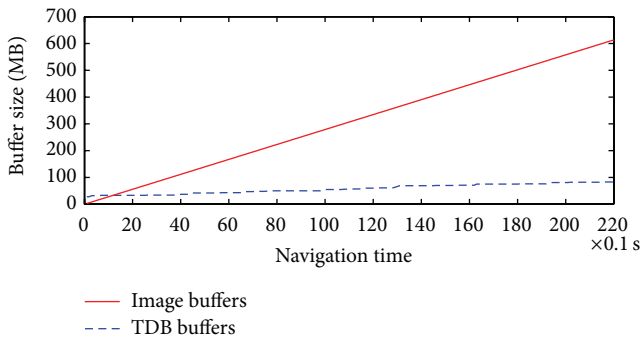


Figure 14: Buffer sizes of the TDB and the captured video images.

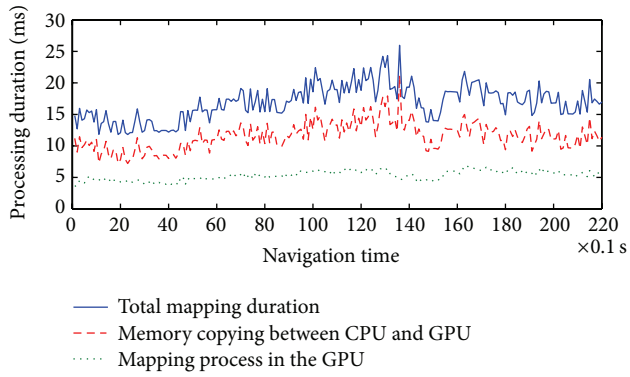
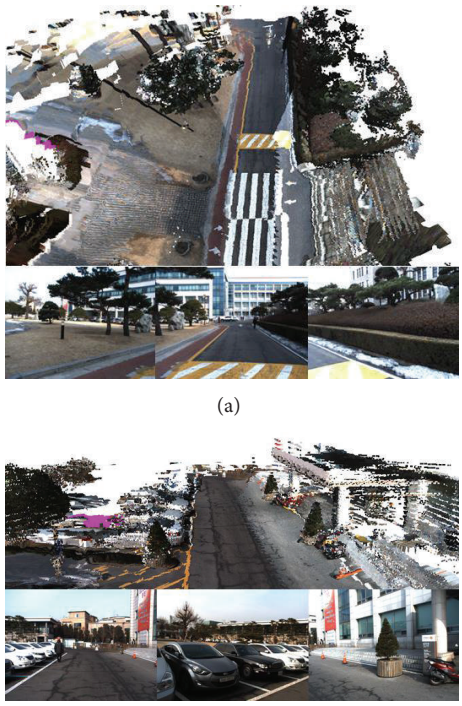


Figure 15: Durations of the mapping process for TDB generation.

Although the speed by using the color mesh is around 27.5 fps, the visualization quality is much lower than that by using the texture mesh.

5. Conclusions

In this study, we developed a real-time intuitive terrain reconstruction system using a nonground voxel map and a ground texture mesh for automated surveying and mobile mapping services. The mobile robot collects 3D point clouds, 2D images, GPS, and rotation states through multiple sensors. One of our objectives is to register the large-scale point clouds into the terrain model in real time. We proposed



(a)

(b)

Figure 16: Terrain reconstruction results from the nonground PDB, ground MDB, and TDB. (a) A top view of environment 1. (b) A front view of environment 2.

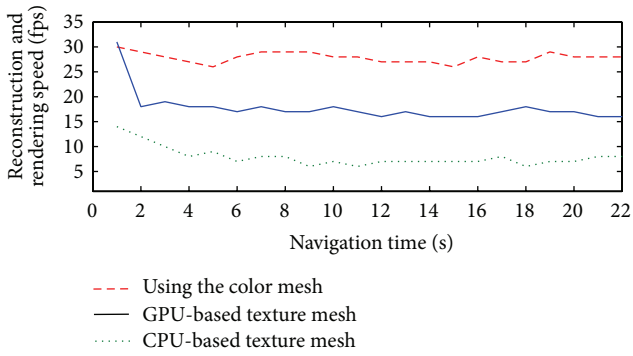


Figure 17: Speed performance comparison of terrain reconstruction and rendering processes using the color mesh, GPU-based, and CPU-based methods.

a voxel-based flag map to register the 3D points into the terrain model without redundancy. Then, we created a color voxel map to represent nonground PDB and a node-based texture mesh to represent the ground surface. By using the GPU, we realized real-time TDB generation from the large-scale captured images so as to reconstruct the texture mesh with low memory consumption. Finally, we rendered the reconstructed terrain model by overlaying the TDB onto the MDB to provide rapid and intuitive information about the surrounding environment, which provided a GUI between the operators and the mobile robots.

We tested our approach using a mobile robot mounted with integrated sensors in a large-scale environment. Through the flag map, we realized real-time large-scale 3D dataset registration. Using the texture mesh with TDB, we offered an intuitive representation of the reconstructed terrain model, which provides the operator with intuitive visualization support. The time required for terrain reconstruction was faster than that for dataset gathering, which satisfied the real-time requirement.

In our testing result, we found that there were some errors in the GPS and IMU when the robot was shaking. We dropped these packets to remove the errors. Therefore, there were some areas in the voxel map with low density. We need to study the calibration algorithms to adjust these errors in the future.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgment

This work was supported by the Agency for Defense Development, Republic of Korea.

References

[1] Y. Matsushita and J. Miura, "On-line road boundary modeling with multiple sensory features, flexible road model, and particle filter," *Robotics and Autonomous Systems*, vol. 59, no. 5, pp. 274–284, 2011.

[2] X. Gong, Y. Lin, and J. Liu, "Extrinsic calibration of a 3D LIDAR and a camera using a trihedron," *Optics and Lasers in Engineering*, vol. 51, no. 4, pp. 394–401, 2013.

[3] S. Yu, S. R. Sukumar, A. F. Koschan, D. L. Page, and M. A. Abidi, "3D reconstruction of road surfaces using an integrated multi-sensory approach," *Optics and Lasers in Engineering*, vol. 45, no. 7, pp. 808–818, 2007.

[4] A. Agrawal, R. C. Joshi, and M. Radhakrishna, "Real-time photorealistic visualisation of large-scale multiresolution terrain models," *Defence Science Journal*, vol. 57, no. 1, pp. 149–162, 2007.

[5] A. Dey, A. Cunningham, and C. Sandor, "Evaluating depth perception of photorealistic mixed reality visualizations for occluded objects in outdoor environments," in *Proceedings of the IEEE Symposium on 3D User Interfaces (3DUI '10)*, pp. 127–128, March 2010.

[6] D. Huber, H. Herman, A. Kelly, P. Rander, and J. Ziglar, "Real-time photo-realistic visualization of 3D environments for enhanced tele-operation of vehicles," in *Proceedings of the International Conference on 3D Digital Imaging and Modeling (3DIM '09)*, pp. 1518–1525, October 2009.

[7] Y. Zhao, X. Cui, and Y. Cheng, "High-performance and real-time volume rendering in CUDA," in *Proceedings of the 2nd International Conference on Biomedical Engineering and Informatics (BMEI '09)*, pp. 1–4, October 2009.

[8] D. Udayan, H. Kim, and J. Lee, "Fractal Based Method on Hardware Acceleration for Natural Environments," *Journal of Convergence*, vol. 4, pp. 6–12, 2013.

[9] S. Chang, H. Chang, S. Yen, and T. K. Shih, "Panoramic human structure maintenance based on invariant features of video frames," *Human-Centric Computing and Information Sciences*, vol. 3, article 14, 2013.

[10] C. Gong, J. Liu, H. Chen, J. Xie, and Z. Gong, "Accelerating the Sweep3D for a Graphic Processor Unit," *Journal of Information Processing Systems*, vol. 7, no. 1, pp. 63–74, 2011.

[11] W. Song, K. Cho, K. Um, C. S. Won, and S. Sim, "Intuitive terrain reconstruction using height observation-based ground segmentation and 3D object boundary estimation," *Sensors*, vol. 12, no. 12, pp. 17186–17207, 2012.

[12] A. Kelly, N. Chan, H. Herman et al., "Real-time photorealistic virtualized reality interface for remote mobile robot control," *International Journal of Robotics Research*, vol. 30, no. 3, pp. 384–404, 2011.

[13] F. Rovira-Más, Q. Zhang, and J. F. Reid, "Stereo vision three-dimensional terrain maps for precision agriculture," *Computers and Electronics in Agriculture*, vol. 60, no. 2, pp. 133–143, 2008.

[14] S. R. Sukumar, S. J. Yu, D. L. Page, A. F. Koschan, and M. A. Abidi, "Multi-sensor integration for unmanned terrain modeling," in *8th Unmanned Systems Technology*, vol. 6230 of *Proceedings of SPIE*, pp. 65–74, Orlando, Fla, USA, April 2006.

[15] J. M. Noguera, R. J. Segura, C. J. Ogáyar, and R. Joan-Arinyo, "Navigating large terrains using commodity mobile devices," *Computers and Geosciences*, vol. 37, no. 9, pp. 1218–1233, 2011.

[16] S. Lee and I. Lee, "A secure index management scheme for providing data sharing in cloud storage," *Journal of Information Processing Systems*, vol. 9, no. 2, pp. 287–300, 2013.

[17] J. Kammerl, N. Blodowy, R. B. Rusuz et al., "Real-time compression of point cloud streams," in *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 778–785, 2012.

[18] D. Gingras, T. Lamarche, É. Dupuis, and J. Bedwani, "Rough terrain reconstruction for rover motion planning," in *Proceedinsg of the 7th Canadian Conference on Computer and Robot Vision (CRV '10)*, pp. 191–198, June 2010.

[19] Y. Zhuang, W. Wang, H. Chen, and K. Zheng, "3D scene reconstruction and motion planning for an autonomous mobile robot in complex outdoor scenes," in *Proceedings of the International Conference on Modelling, Identification and Control (ICMIC '10)*, pp. 692–697, Okayama, Japan, July 2010.

[20] V. Kumar and S. K. Agarwal, "Image compression by using new wavelet bi-orthogonal filter coefficients," in *Proceedings of the IEEE International Conference on Signal Processing, Computing and Control (ISPCC '12)*, pp. 1–5, Waknaghat Solan, India, March 2012.

[21] M. Al-Zoubi and A. Al-Zoubi, "Querying relational MPEG-7 image database with MPEG query format," *International Journal of Multimedia and Ubiquitous Engineering*, vol. 6, no. 4, pp. 39–44, 2011.