

Multimodal Interface Based on Novel HMI UI/UX for In-Vehicle Infotainment System

Jinwoo Kim, Jae Hong Ryu, and Tae Man Han

We propose a novel HMI UI/UX for an in-vehicle infotainment system. Our proposed HMI UI comprises multimodal interfaces that allow a driver to safely and intuitively manipulate an infotainment system while driving. Our analysis of a touchscreen interface-based HMI UI/UX reveals that a driver's use of such an interface while driving can cause the driver to be seriously distracted. Our proposed HMI UI/UX is a novel manipulation mechanism for a vehicle infotainment service. It consists of several interfaces that incorporate a variety of modalities, such as speech recognition, a manipulating device, and hand gesture recognition. In addition, we provide an HMI UI framework designed to be manipulated using a simple method based on four directions and one selection motion. Extensive quantitative and qualitative in-vehicle experiments demonstrate that the proposed HMI UI/UX is an efficient mechanism through which to manipulate an infotainment system while driving.

Keywords: Multimodal interface, human-machine interface, user interface, user experience, gesture recognition, speech recognition, multitasking, in-vehicle infotainment, IVI, GENIVI, MMI, HMI, UI, UX.

Manuscript received Aug. 10, 2014; revised Apr. 27, 2015; accepted May 14, 2015.

This work was supported by Institute for Information & Communications Technology Promotion (IITP) grant funded by the Korea government (MSIP) (B0101-15-0134, Development of Decision Making/Control Technology of Vehicle/Driver Cooperative Autonomous Driving System (Co-Pilot) Based on ICT).

Jinwoo Kim (corresponding author, jwkim81@etri.re.kr), Jae Hong Ryu (jhyu@etri.re.kr), and Tae Man Han (tmhan@etri.re.kr) are with the IT Convergence Technology Research Laboratory, ETRI, Daejeon, Rep. of Korea.

I. Introduction

The rapidly increasing demand for more in-vehicle infotainment (IVI) has led to a need for driver convenience and safety when manipulating a car application while driving. Until now, popular car makers have focused on commercializing a central display based on an HMI and telematics services. In addition, they have limited the functionality of the infotainment systems they sell in an effort to reduce the risks drivers face when manipulating such systems.

Currently, however, most car makers, graphics companies, and other tier 1 companies are researching the convergence between a multimodal interface and HMI UI framework. Additionally, they are trying to enhance the user experience (UX) using recognition technology, in which voice, speech, gesture, and posture recognition are used to track a user's attention [1]–[2]. American and Chinese users in particular prefer a natural language enabled system over a command and control system [3].

These days, many drivers use a smart device or navigation system while driving, which can cause a dangerous situation or accident because of the driver's inattention. The impact of driver inattention on crash risk was researched by the National Highway Traffic Safety Administration. They introduced the idea that naturalistic data can help complete gaps in the transportation research between epidemiology and empirical methods by collecting sufficient data to conduct epidemiological analyses while still collecting detailed driver behavior and driving performance data [4].

To solve the problem of driver inattention, researches on

HMI UI and UX have been conducted (for example, on an evaluation system that uses an eye tracker [5]), and diverse methods for measuring and reducing driver inattention have been developed; in-car task components, in particular, are visually distracting to drivers [6]–[7].

Until now, most user interfaces for IVI have used a touchscreen, which frequently distracts the driver. Recently, a method was introduced to understand how drivers naturally make swiping gestures in a vehicle as compared with a stationary setting [8]; gesture methods differ according to their application and purpose. To find the density space for gesturing, a new standard on permitted gesture properties (time and space) in a car was proposed in [9].

BMW has already developed its *i-drive* for safe driving and expanded its UX to simple gesture recognition. Additionally, the company proposed a new in-vehicle augmented reality (AR) concept connecting an adaptive cruise control (ACC), advanced driver assistance systems, and a navigation system through a head-up display (HUD). Audi has created Multi Media Interface (MMI) for infotainment content using gesture and voice recognition. Benz has also demonstrated an in-vehicle AR using speech and pointing gestures. GM has researched driver gaze tracking while driving to provide AR information to drivers [10]. Finally, Ford has introduced its “MyFord Sync” using the convergence of wheel keys and voice recognition.

Several types of interfaces have been proposed to accommodate the dynamic settings of in-vehicle content delivery [11]. Drivers in near collisions while using an infotainment system exhibit distinct glancing behaviors [12].

An in-vehicle HUD system combined with AR provides recognition, unlike existing HUD systems. In-vehicle HUD systems that are combined with AR technology display information registered to the driver’s view and are developed for the robust recognition of obstacles under bad weather conditions [13]–[14].

As mentioned earlier, multimodal-based HMI UI/UX products designed for IVI systems are aimed at reducing driver inattention. Additionally, to improve driver attention, recognition technologies are becoming involved in infotainment systems. The current trend in IVI user interfaces is toward non-touch-based interfaces in an attempt to provide drivers with a more intuitive HMI UX.

Additionally, a driver’s inattention can be reduced by using a multimodal interface-based HMI UI and UX in a way whereby they are able to complement one another. In particular, voice recognition and knob-based interfaces require a novel IVI system. Hand gesture interfaces — ones that uses point, tap, swipe, and touch capabilities — are becoming more and more important to AR in-vehicle HUD systems. Such systems are

dependent on an HMI UI/UX to come up with an interface for each gesture.

In this paper, we propose a new HMI UI framework concept that aims to provide a driver with a safe and efficient UX while driving. To do so, we consider a multimodal interface-based HMI UX that uses speech; a knob, which is rotary device; and gesture recognition for manipulation without a touchscreen. In addition, we provide touch-based gestures for convenience. Furthermore, we modified a standard infotainment platform (GENIVI) in such a way so that it could support our HMI UI/UX. This means that our HMI UI/UX can be commercialized more easily. We also provide a web-based application based on our proposed HMI UI framework to verify the proposed multimodal-based UI/UX.

II. Proposed Multimodal Interface-Based HMI UI/UX

1. Overall Procedure

The proposed HMI UI/UX is characterized by the safety it provides its users and the fact that it does not act as a distraction to a driver as they manipulate an IVI system (that is utilizing the proposed HMI UI/UX) while driving. The HMI UI/UX provides a driver with intuitive manipulation methods from which to navigate an HMI UI menu; confirmation of any reactions given by the driver; a UI framework to connect with SW platforms and user interfaces; and a design structure for a GUI.

The amount of time a driver spends looking at the HMI screen of the proposed HMI UI/UX can be reduced through non-touchscreen-based methods of operation. Until now, drivers have had to use their infotainment system’s touchscreen to select something, such as a desired icon. However, this is very dangerous while driving; hence, some vehicle manufacturers, such as BMW and Audi, have substituted touchscreens with other alternative methods of manipulation, such as a knob or touch pad.

Our proposed HMI UI/UX allows a driver to manipulate HMI menus via a natural hand gesture interface. Additionally, voice, knob, and touch interfaces can also be used to manipulate our proposed HMI UI/UX — such interfaces provide the user with the same UX as given by the natural hand gesture interface. UIs have been adapted for other applications and can be accessed from the main menu (see Fig. 1). Additionally, we developed HMI functions, such as a task manager function, diagnostics function, and multitasking function, in our infotainment platform.

We used the open-source GENIVI platform [15] to create web-based applications (that is, map and browser services) that could be used with our proposed HMI UI. In this paper, we

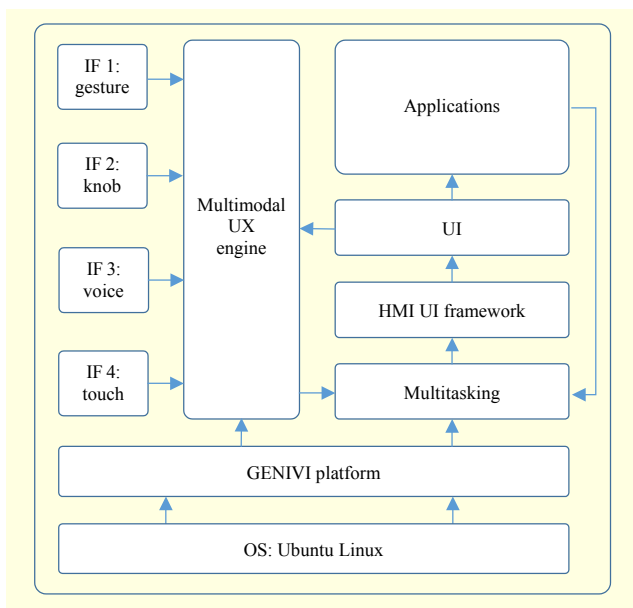


Fig. 1. Architecture of proposed HMI UI/UX.

provide a detailed account of the proposed HMI UI framework in relation to the proposed HMI UI/UX.

Figure 1 shows the architecture of our proposed HMI UI/UX. We built a multimodal UX engine, multitasking capability, and an HMI UI framework on the GENIVI platform. First, we provided four user interfaces to use the HMI UI effectively. These are connected to the multimodal UX engine, which acts as a bridge between the four interfaces — gesture, knob, voice, and touch — and the UI and multitasking functions. Second, the main function of the multimodal UX engine is to manage the four interfaces in an activated state in real time. When a driver uses one of the interfaces, the state information of the interface is transferred to the multitasking block. Third, the multitasking block notifies each other block connected to it of the current layer, input interface, and HMI UI status. Because our HMI UI/UX architecture has multiple layers (see Section II-2 below), the states of the layers must be checked and the layers themselves arranged, all in real time. Finally, this information is provided to the HMI UI framework block. In this block, applications can use the HMI UI framework and HMI UI/UX information.

2. HMI UI/UX

In the overall HMI UI/UX composition, there are four layers — in which the background, main menu icons, submenu icons, and pop-up menus lie. The UI framework for the proposed HMI main screen consists of two arrays. The bottom array represents the main menu icons and is the more significant of the two. The upper array consists of submenu icons associated

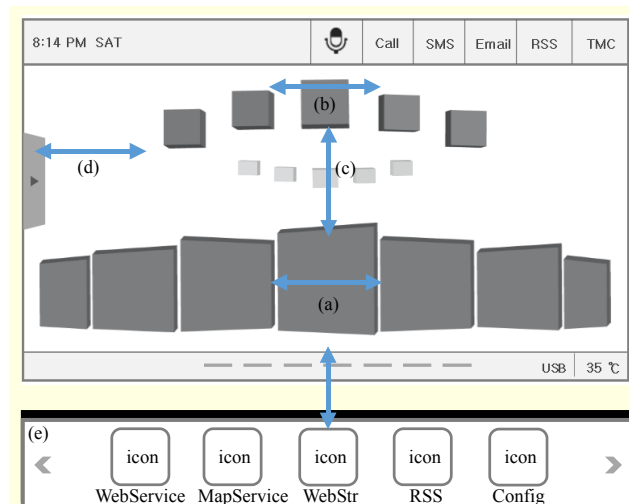


Fig. 2. Construction of UI for main screen of proposed HMI UI/UX.

with the currently selected main menu icon. In addition, the proposed HMI UI/UX consists of a main screen, GUI for USB connection, subscreens, and multitasking UI framework.

Figure 2 shows the proposed HMI UI framework and possible movements required to navigate menus. Figure 2(a) shows the main menus, and Fig. 2(b) shows the associated submenus. One can scroll through the menu using an appropriate manipulation method. As the main menus are rotated, the corresponding submenus are also changed. The main menus consist of several icons, and it is possible to change the number of such icons. Figure 2(c) shows the change in position between the main menu and submenus using the gesture and knob interfaces. Figure 2(d) shows the pop-up action layer. When connecting a USB memory stick or manipulating the center of the main menu using a tap gesture, the pop-up slide moves to the right on the main screen. Figure 2(e) shows the multitasking pop-up layer, which appears from the bottom of the HMI main screen whenever a multitasking event begins.

We use two 3D curve patterns to express an HMI UI movement, such as a rotation, translation, or zigzag effect. The first is a quadratic curve, and the other is a cubic curve. These curves can be combined in such a way so as to make a *complex* curve. Our 3D HMI UI positioning mechanism utilizes such information.

Figure 3 shows the aforementioned positioning mechanism at work. In the figure, p_a and p_b represent the centers of two objects (menu icons) from the HMI main screen. The complex curve (red solid line) passing through them illustrates the path along which they will move. Point p_i is a center position between two neighboring objects along the given complex curve. In the equation for “positioning translation,” d , a positive

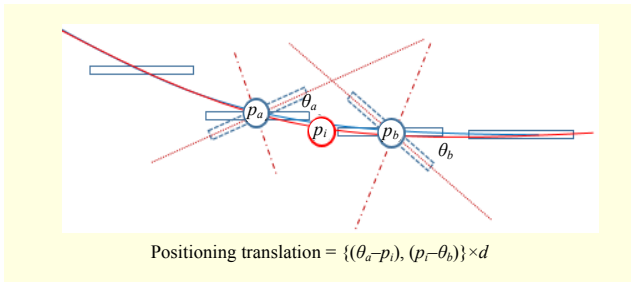


Fig. 3. Positioning translation for movement of main screen menus.



Fig. 4. UX based on four-directional movement: HMI UI manipulation is performed using knob or touch interface.

or negative number, indicates the direction of an icon. In addition, θ is the positioning angle of a given point. The movement for menus is formulated as the sum of the discrepancy between an object position and neighbor point p_i . The values of the translation positioning and direction factor d are then multiplexed.

Figure 4 illustrates two of the available manipulation methods for the main screen — the use of a knob and touch interface. A driver can control the search menus on the main screen using two direction pairs — up and down, and left and right. When controlling the knob, the indicator box (white box) appearing at the center of the main menu or submenu informs the driver as to which icon is currently selected. When navigating the icons of the main menu, the user can simply move the knob in an upwards direction and the indicator (white

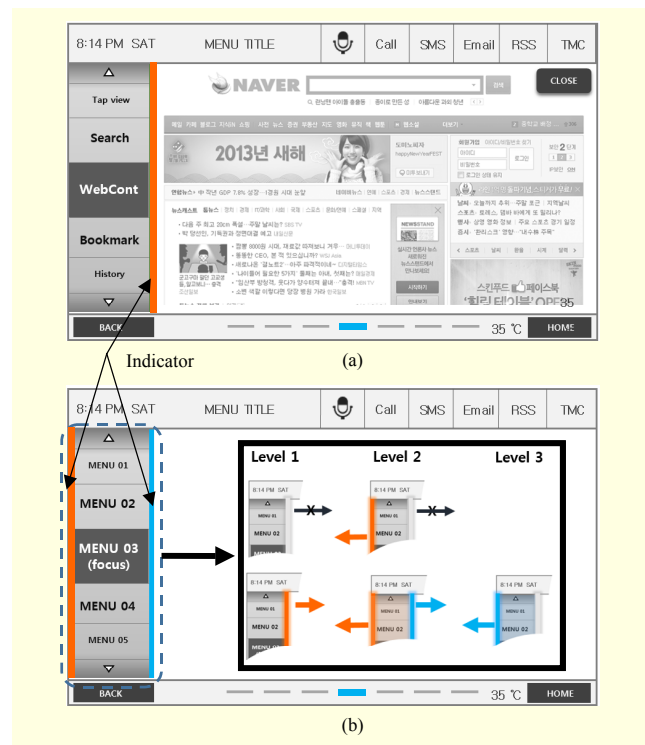


Fig. 5. Subscreen UI framework: (a) first-level menus and (b) second-level menus and example of level structure.

box) will then move up to the center of the submenu. Alternatively, the user can scroll through the icons in the main menu (or submenu) by swiping the screen; the corresponding submenus will accordingly appear automatically.

When selecting an icon from the submenu, the user will be taken to a subscreen. Each subscreen has a sidebar containing menu buttons. Through these menu buttons, a further two menu sublevels can be accessed, giving a maximum of three menu levels in total. Furthermore, the user can recognize the main menu category (color coded) currently selected from an indicator that is located to the bottom right of the sidebar, at the bottom of the screen. For voice recognition, the driver simply gives a command that includes a search word at the current stage after pressing the start key (which is located at the top-right corner of every screen).

Figure 5 illustrates a subscreen of the UI framework. Figure 5(a) shows the first subscreen upon starting a specific application. Immediately to the left and right of the sidebar, menu-level indicators express the status of the current menu level. Three colors are used — red, blue, and grey. Red indicates that a driver can enter the second menu sublevel; blue indicates that the driver can enter the third menu sublevel; and grey indicates that no further menu sublevels exist (thus the driver would know that they have reached the final menu sublevel; that is, the third menu sublevel).

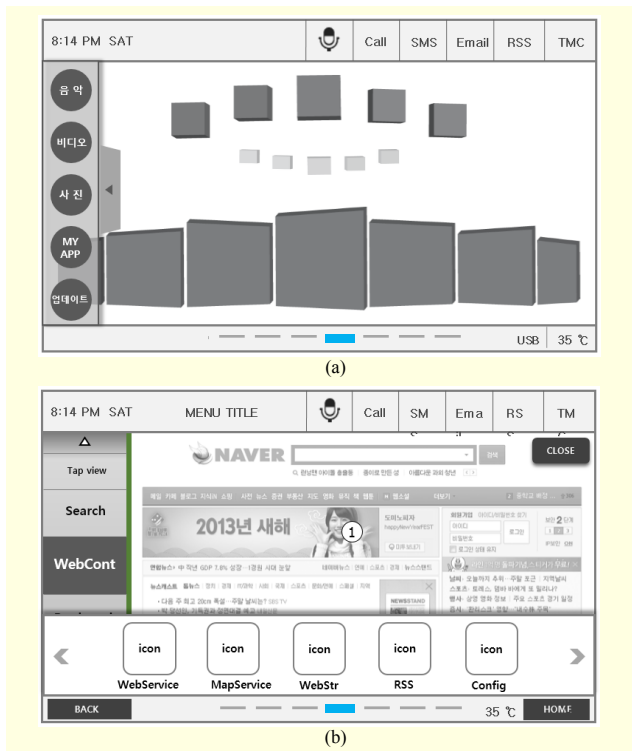


Fig. 6. (a) USB connection and (b) multitasking UI framework.

Up to this point, we have described the UI framework design for both a subscreen and the main screen. Although the two HMI UI frames are different from each other, the UX for each HMI UI is the same. All menus are simply manipulated using four directions — up, down, left, and right. The pop-up layer-based UI has three types: a USB connection, multitasking, and favorite function. As with the UX, these UI frameworks can also be manipulated using the aforementioned four directions.

Figure 6(a) shows the UI action design available for a USB connection. For a USB-connected application, five menu buttons are expressed on a sliding sidebar menu located on the left of the screen. This sliding sidebar menu moves from left to right when the USB device is connected. In addition, the user can select the menu buttons using multimodal interfaces. Figure 6(b) shows a multitasking UI framework. This UI framework is also included in the pop-up application, and moves from down to up at the bottom of the screen. It contains icons of recently used applications; drivers can select or delete an icon linking to an application. We illustrated the design of our HMI UI framework and UX, and in the next section will describe the applications based on the proposed UI framework.

To provide the menu management freely to the driver, we also designed a favorite function. Figure 7 shows the UI framework used to register favorite icons. Drivers can arrange their wanted icons using the following main steps: tap and hold any submenu icon on the screen, and then drag to the “+”

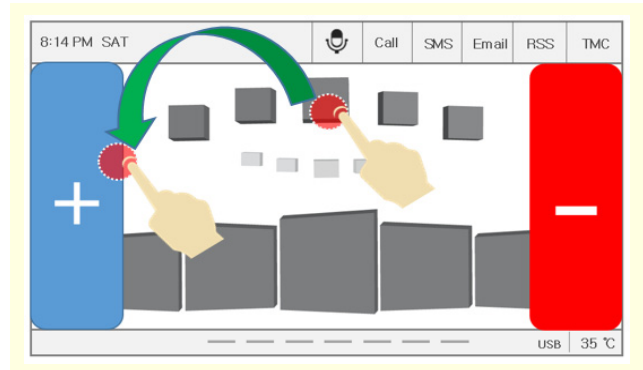


Fig. 7. Favorite function based on UI framework.

region on the left side of the main screen and then release.

The dragged icons are then registered on the favorites menu. Thus, whenever the driver starts the HMI UI/UX or reboots the favorites menu located at the middle part of the main screen, the registered icons on the menu are updated. On the other hand, the driver can remove registered icons by dragging a submenu icon to the “-” region, which is located on the right side of the main screen.

3. Gesture Recognition

We illustrated the UI framework and its functions using multimodal interfaces, which are a knob and touch capability. In this section, we describe gesture recognition using the UI framework. Our proposed HMI UI/UX can be manipulated in only four directions. Thus, we made four types of hand gesture recognition. In detail, we obtain 3D information from a Leap Motion device sensor. We used two gesture groups — swipe and tap — from Leap Motion SDK [16], but we needed to modify the provided gesture samples to fit our HMI UI/UX. Because the gestures must be strongly separated into two classes, we calculated the average positions of detected fingers and palms (six positions in total) in a frame-to-frame manner. Then, we normalized the average value of each of the six positions to “1” and found a threshold to determine each gesture regarding each direction.

Gesture recognitions consist of four directions and one tap gesture to start an application at the main screen. In this paper, we do not define the use of other functions including multitasking; a USB connection; and favorite and specific application functions because the purpose of gesture recognition is to prove that our HMI UI framework optimizes multimodal interfaces including gestures. Figure 8 shows the gesture UX used to manipulate our proposed UI framework. Figures 8(a) through 8(d) indicate four swipe motion directions; that is, left, right, up, and down. Figures 8(e) and 8(f) show the tap motion using the z-axis.

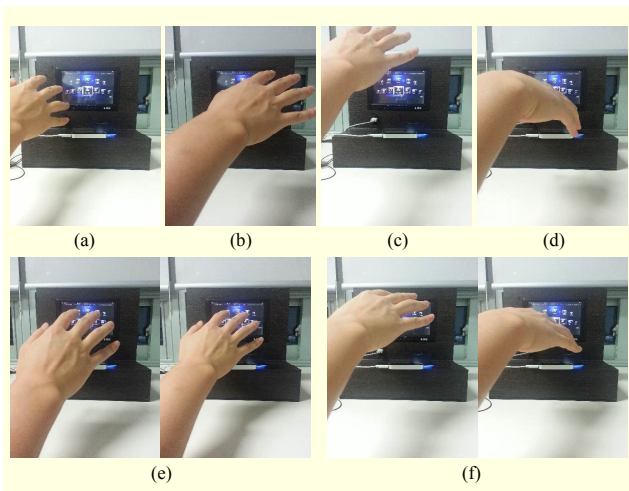


Fig. 8. Gesture recognition used to manipulate our proposed HMI UI/UX. Four directions of swipe motion: (a) left, (b) right, (c) up, (d) down, (e) tap in direction of z -axis to select motion, and (f) tap in direction of y -axis to select motion.

4. Web-Based Applications

We designed a multimodal interface-based UI framework for an IVI system. In this section, we discuss our self-made web-based application map and browser services that are operated without a touch-based interface. These applications can be manipulated using either a knob, speech recognition, or gestures. In the case of the knob and gesture capability, the UI framework uses four directions — right, left, up, and down — and one selected motion.

The purpose of our web-based map service is such that a driver can manipulate the application in an effective and safe manner without a touch-based interface to obtain map information while driving. We used the map information and basic functions from Naver OpenAPI [17]. We then made a connection between our UI framework and the functions that express traffic status, distance measurement, current position, and map resizing.

Figure 9 shows the application-based UI framework used to verify our proposed HMI UI/UX. All of the menus can be controlled using the multimodal UI, including the speech, gesture, and knob interfaces. Figure 9(a) shows the real-time traffic function on the map. In addition, we can find the current position using GPS, and the current traffic status can be displayed, as shown in Fig. 9(d). Figure 9(c) shows that a driver can register their favorite destination using a knob interface. Figure 9(e) shows the menu used to resize the map. After using the right command or selecting the appropriate multimodal UI, a driver can search for diverse map sizes, as defined in Fig. 9(f). At this point, the sidebar indicators notify the driver as to whether more depth layers are available. All



Fig. 9. Web-based map service based on UI framework: (a) real-time traffic status, (b) registered location, (c) measurement of distance, (d) setting current position, and (e) and (f) map resizing.



Fig. 10. Web-based browser service based on UI framework: (a) full screen of web pages, (b) expressed side menus, (c) web search, (d) and (e) web-page control menus, and (f) making new tab.

menus can be manipulated using two direction pairs, and the indicator box is always focused at the center of a menu array.

A driver may want to find information from a web page using a smartphone or infotainment system while driving. To do so, drivers often use time-waiting for a signal or voice recognition service through a connected smartphone service. In the case of controlling a web page in detail, voice recognition is sometimes inconvenient because a driver has to memorize all

kinds of commands to manipulate different functions.

To compensate for this problem, we suggest the use of a UX-based web browser. Figure 10(a) shows a full-screen view of a web page. When a driver wishes to use sidebar menus to control a web page, the sidebar can be displayed on the screen through the driver making a designated motion to the right, as shown in Fig. 10(b). Figure 10(c) shows the function of searching through web information by entering a specific address. Figure 10(d) illustrates the opening and closing of a new web page in detail without the use of a touch control. In addition, drivers can control a web page using back, forward, and page reload, as shown in detail in Fig. 10(f).

III. Experimental Results

1. Test Environment

The software for the prototype system was developed using Qt SDK, OpenGL, Ubuntu 12.04, Leap SDK, and the GENIVI platform. We use the GENIVI component based on HTML5, DLT demon, Webkit, and a layer manager to make the HMI UI/UX and applications. The hardware running the software is automotive, consisting of an Intel Atom D525 1.8 GHz CPU with 2 GB of main memory. Our HMI UI/UX system is based on Linux, which is a standard OS, and includes the GENIVI compliance platform. To verify the real-time manipulating performance of the system, we carried out system and performance tests to confirm how the frame rate changes according to the HMI UI control screen. The average frame rate for manipulating the UI framework is 15 fps, and the maximum CPU share is under 43%.

Figure 11(a) shows the outlook of our proposed system including multimodal interfaces. The system consists of a mono mic, knob, touch-based display expressed UI, 3Dconnexion knob [18], and Leap Motion sensor for gesture recognition. For the speech recognition, we used 45 key words, which could be used by a driver for the purposes of searching and implementing menus and menu functions. The display has a resolution of 800×480 and is a piezoelectric touchscreen. The knob is a 3Dconnexion to control the UI using the five aforementioned UXs. For gesture recognition, we use the 3D information of the five finger tips and palm from the Leap Motion sensor shown in Fig. 11(b).

We tested our multimodal-based HMI UI/UX to verify its capability over a touch-based HMI UI/UX. As shown in Fig. 12, we set up the driving conditions and tested two types of HMI systems to compare the UI/UX performance. Figure 12(a) shows the driving simulation, where we connected a knob with our system (see Fig. 12(b)). We recorded a driver's attentiveness as they manipulated firstly a

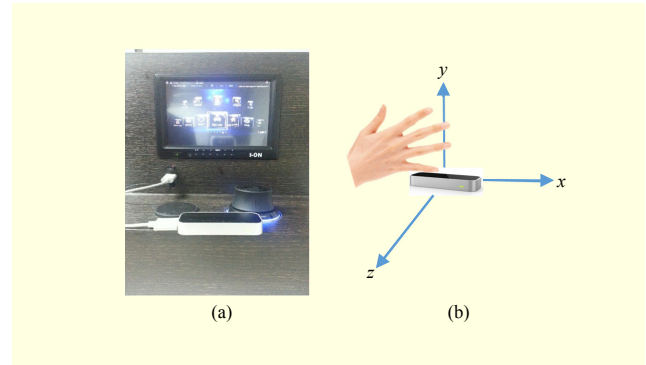


Fig. 11. System outlook: (a) multimodal-based HMI UI/UX and (b) region of gesture recognition using Leap Motion.



Fig. 12. Performance demonstration of multimodal interface-based HMI UI/UX during driving simulation.

standard touch-based HMI navigation system (see Fig. 12(d)) and then our multimodal-based HMI (see Fig. 12(c)). We then compared the results of these driver attentiveness experiments. More specifically, we used a commercial navigator (Thinkware inc.), i-navi UX, which can only be controlled via a touch-based UI [19]. We tested our multimodal-based HMI to verify its capability over the touch-based HMI. The simulator that we used included instances of straight and curved sections of road (see Figs. 12(e) and 12(f)). More specifically, we used the 3D driving game "Euro Truck," which served as the provider of driving information, as opposed to the driving simulator, so as to provide a greater level of reality to the tester. Since the 3D driving game realized real driving conditions, including curve, intersection, access road, and driving with other cars, we were

left only to consider an integrated cognitive model that incorporated the driving path, car, and driving with use of either our HMI UI/UX or a commercial navigator.

We set up a scenario that required a tester to operate a navigation application while driving, and we applied this same scenario to all testers under the same conditions. The scenario is as follows: (1) testers should find the “navigation” icon and select it; (2) zoom in on the map; (3) zoom out until on the map; (4) find the main screen; (5) start the function of “multitasking”; (6) find the “navigation” icon and select again; (7) find the submenu icons “web browser” and “DMB”; and (8) find the “navigation” icon again.

Testers had to control the HMI using either a touch-based UI (T-Navi) or our proposed multimodal interface-based UI (M-HMI) without touch capability. We tested our multimodal-based UI to verify its capability over T-Navi. The testers manipulated M-HMI using its range of diverse interfaces — gesture, voice, and knob — without regulation. Since T-Navi has only a touch-based interface, we could not evaluate cognitive and audiometry elements. In short, we conducted an evaluation of other elements instead (see Fig. 2).

2. Results

Several projects about HMI methods have been conducted, such as AIDE, HUMANIST, SafeTE, HASTE, CAMP, and so on [20]. To evaluate our proposed HMI UI/UX system, we considered general elements such as driver response time (the average of the total measurements from the start of the listening command to the action using our UX or a touchscreen), average driving speed, duration of distraction from the HMI UI screen (distractions were separated into those lasting less than 1 s, between 1 s and 2 s, and more than 2 s. We defined “more than 2 s” as a dangerous distraction), number of crashes (includes crashes with cars to the front and side of the vehicle or obstacles), and lane keeping [21]. To obtain the above measurements, we tried to use an eye tracker. However, we had to manually check each captured video for each tester because of an error detection problem with the eye tracker.

Additionally, we divided the testers into two groups according to their driving career. The experienced group members had more than five years of driving experience and were skilled in both types of HMI UI/UXs (according to our survey results). The non-experienced group members had less than five years of driving experience. The criterion of division by five years was chosen in accordance with our survey results, which revealed that this was the average amount of time necessary for a tester to become accustomed to manipulating their own in-car HMI UI/UX or navigator without endangering themselves while driving.

Table 1. Result of evaluations regarding two types of HMI UI/UXs for groups of drivers (all items are mean of results from each group of testers).

Items (mean)	Experienced (30)		Nonexperienced (30)	
	Proposed HMI	Commercial navigator	Proposed HMI	Commercial navigator
Response (s)	1.5	2.3	3.4	3.8
Speed (km/h)	75	80	67.6	75.6
Distraction ($D < 1$ s)	21	11	15	16.7
Distraction ($1 \text{ s} \leq D \leq 2$ s)	11	30	26.3	16.7
Distraction ($D > 2$ s)	1	2	4.3	6
Crash (count)	0	0	1	1.3
Derailment (count)	3	7	7	9.3

We created a scenario to control each UI framework and manually measured testers’ response times and distraction factors while they drove in the simulator and operated the two aforementioned HMIs (each on a separate occasion). A total of 60 testers were used, and the results of the simulations are shown in Table 1.

Drivers with five or more years of driving experience showed less distractions and quick reaction times when manipulating our proposed HMI UI/UX while driving. However, the result of the touch-based HMI showed a long response time and more distractions when controlling and searching the screen. Moreover, the drivers looked at the screen several times to find the touch position. On the other hand, inexperienced drivers showed more distractions of less than 2 s when using our HMI compared with the touch-based version. The reason for this is that they were unfamiliar with our HMI. However, there were fewer distractions of more than 2 s, crashes, and derailments compared with the results of using the touch-based HMI.

In the case of our proposed HMI UI/UX, the number of recorded distractions of under 1 s in the experienced drivers group was greater than that recorded for between 1 s and 2 s (experienced drivers group). On the other hand, the results for T-Navi show a stark contrast; that is, experienced drivers kept glancing at the HMI screen to check their actions when using the multimodal interface-based UX for a short time period. In the case of the inexperienced drivers group, while operating our proposed HMI UI/UX, the results show a difference with the experienced drivers group. The number of recorded distractions for between 1 s and 2 s was greater than that recorded for under 1 s. This was due to the fact that the inexperienced drivers were not familiar with multitasking tasks while driving. Based on these results, we evaluated our proposed

Table 2. Analysis of psychomotor workload for proposed HMI UI/UX.

Elements	1 < Year ≤ 5		5 < Year ≤ 10		10 < Year ≤ 20		20 < Year		Avg.	
	Pro HMI	Com Nav	Pro HMI	Com Nav	Pro HMI	Com Nav	Pro HMI	Com Nav	Pro HMI	Com Nav
Esthetic	91.4	62.9	91.4	64.8	88.6	55.7	88.0	62.2	90	61
Intuitive	74.3	77.1	83.8	75.2	80.0	78.6	84.0	78.7	81	77
Usability	85.7	77.1	84.8	64.8	92.9	65.7	92.0	67.9	89	69
Structure	88.6	82.9	83.8	64.8	81.4	62.9	92.0	60.2	86	68
Trends	77.1	62.9	89.5	51.4	84.3	51.4	96.0	47.0	87	53
Design	94.3	62.9	83.8	62.9	78.6	55.7	88.0	57.2	86	60
Delivery	80.0	74.3	72.4	72.4	81.4	61.4	84.0	63.0	79	68
UI usability	77.1	71.4	90.5	50.5	90.0	52.9	93.1	55.8	88	58
Purchase capability	80.0	48.6	87.6	59.0	92.9	52.9	92.1	54.6	88	54
Safety	85.7	48.6	91	50.5	92.9	44.3	94.6	41.5	91	46
Keep eyes forward	77.1	51.4	82	47.6	87.1	50.0	87.9	44.4	84	48

* Pro HMI: propose HMI and Com Nav: commercial navigator

method as being better than that of the touch-based HMI UI/UX method in terms of the *distractions* evaluation criterion.

Evidently, the touch-based HMI UI method demanded drivers' attentions while they were driving more so than the proposed method. In addition, we conjecture that if drivers have more experience using our proposed UX, then they will find that it will become easier to operate it in a safer and more efficient manner while driving. In detail, a driver can focus on driving while at the same time performing a search of either the main menu, a submenu, or a sidebar menu with the aid of different sound effects. The sound effects are generated whenever the driver uses either a gesture motion, voice recognition, or the knob.

To provide a cognitive element to our proposed HMI UI/UX, several types of feedback effects were considered, such as sound, haptic, and visualization notices. In short, a driver need not necessarily look at the HMI screen to successfully operate the proposed HMI UI/UX, once they become accustomed to such types of feedback; the same cannot be said for traditional touch-based HMI UIs

We additionally conducted an analysis of the psychomotor workload experienced by our testers as they operated M-HMI and T-Navi. Table 2 shows the results. The test group was divided by driving career. Generally, the results show that testers preferred M-HMI over T-Navi for almost all of the evaluation criteria in Table 2.

As shown in Fig. 13, we tested our proposed HMI UI/UX in a car while driving. The testers conducted multitasking,



Fig. 13. Experiment with multimodal-based HMI UX in car.

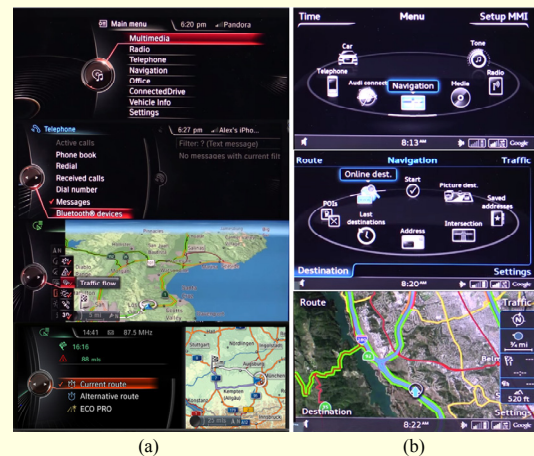


Fig. 14. Comparison of framework structures: (a) i-drive from BMW in X5, (b) MMI from 2015 Audi A3 sedan, and (c) our proposed HMI UI/UX.

searched for apps, and connected a USB to install apps or media content, by using either speaker recognition, gestures, or a knob.

Currently, state-of-the-art HMI UI/UXs lie in the hands of car manufacturers such as BMW or Audi. Although we do not have quantitative criteria from which to compare our own HMI UI/UX with those that belong to the likes of BMW and Audi, we were able to conduct a qualitative evaluation from a different point of view — the HMI UI/UX's framework. Figure 14 shows a comparison of framework structures. The UI frameworks of the aforementioned HMIs have some points

in common. One common point is that they all make use of only four directions and a rotation to manipulate the UI; moreover, most of the menus consist of a hierarchical UI framework. The difference between the framework structures of BMW and Audi and our proposed method, as shown in Fig. 14(c), is as follows. In the case of i-drive and MMI, the menus are fixed on the menu screen and it is the indicator that has to move within a selected list of menus. In contrast, because our UI indicator is fixed, a driver of a car that contains our proposed HMI UI/UX is more able to maintain their concentration on the road ahead while operating the HMI. Additionally, our UI framework maintains symmetry and reduces the staring time.

IV. Conclusion

This paper presented a multimodal interface-based HMI UI/UX that provides a more convenient, efficient, and safe manipulation than previous touch-based interface products while driving. Our proposed system makes it possible to manipulate while looking at the side of the screen. In addition, the UX mechanism can be learned in a short time period, and the controls can be used easily to manipulate the multimodal interface while keeping a front gaze. Our proposed UX mechanism is as easy to learn and manipulate as a typical HMI-based device. Because we built our HMI based on a standard infotainment platform, it is ready for commercial use. In addition, we verified that our HMI system provides an advanced vehicle UX for manipulating in-vehicle infotainment better than in previous products. Our multimodal in-vehicle interface can be combined with diverse gesture recognition, including posture, pointing, and body, hand, and head poses, and can be extended to augmented reality and instrument cluster interaction with the driver. As future work, we have a plan to combine our proposed UX and gesture recognition to interact with a driver statement recognition system, which is connected to our ongoing co-pilot system project.

References

- [1] H.M. Kim et al., "Dual Autostereoscopic Display Platform for Multi-user Collaboration with Natural Interaction," *ETRI J.*, vol. 34, no. 3, June 2012, pp. 466–469.
- [2] D. Lee and S. Lee, "Vision-Based Finger Action Recognition by Angle Detection and Contour Analysis," *ETRI J.*, vol. 33, no. 3, June 2011, pp. 415–422.
- [3] L. Hackenberg et al., "International Evaluation of NLU Benefits in the Domain of In-Vehicle Speech Dialog Systems," *Proc. Int. Conf. Automotive User Interfaces Interactive Veh. Appl.*, Eindhoven, Netherlands, Oct. 28–30, 2013, pp. 114–120.
- [4] NHTSA, "The Impact of Driver Inattention on Near-Crash/Crash Risk: An Analysis Using the 100-Car Naturalistic Driving Study Data," National Highway Safety Administration, Washington, DC, USA, Rep. DOT HS 810 594, 2006.
- [5] T.M. Gable et al., "Advanced Auditory Cues on Mobile Phones Help Keep Drivers' Eyes on the Road," *Proc. Int. Conf. Automotive User Interfaces Interactive Veh. Appl.*, Eindhoven, Netherlands, Oct. 28–30, 2013, pp. 66–73.
- [6] T. Kujala, J. Silvennoinen, and A. Lasch, "Visual-Manual In-Car Tasks Decomposed - Text Entry and Kinetic Scrolling as the Main Sources of Visual Distraction," *Proc. Int. Conf. Automotive User Interfaces Interactive Veh. Appl.*, Eindhoven, Netherlands, Oct. 28–30, 2013, pp. 82–89.
- [7] I. Politis, S. Brewster, and F. Pollick, "Evaluating Multimodal Driver Displays of Varying Urgency," *Proc. Int. Conf. Automotive User Interfaces Interactive Veh. Appl.*, Eindhoven, Netherlands, Oct. 28–30, 2013, pp. 92–99.
- [8] G. Burnett et al., "A Study of Unidirectional Swipe Gestures on In-Vehicle Touch Screens," *Proc. Int. Conf. Automotive User Interfaces Interactive Veh. Appl.*, Eindhoven, Netherlands, Oct. 28–30, 2013, pp. 22–29.
- [9] A. Riener et al., "Standardization of the In-Car Gesture Interaction Space," *Proc. Int. Conf. Automotive User Interfaces Interactive Veh. Appl.*, Eindhoven, Netherlands, Oct. 28–30, 2013, pp. 14–21.
- [10] C. Jablonski, "An Augmented Reality Windshield from GM," Mar. 19, 2010. Accessed Jan. 20, 2015. <http://www.zdnet.com/article/an-augmented-reality-windshield-from-gm/>
- [11] R. Ecker et al., "Visual Cues Supporting Direct Touch Gesture Interaction with In-Vehicle Information Systems," *Proc. Int. Conf. Automotive User Interfaces Interactive Veh. Appl.*, Pittsburgh, PA, USA, Nov. 11–12, 2010, pp. 80–87.
- [12] M.A. Perez, "Safety Implications of Infotainment System Use in Naturalistic Driving," *J. Prevention, Assessment, Rehabil.*, vol. 41, Feb. 2012, pp. 4200–4204.
- [13] H.S. Park et al., "In-Vehicle AR-HUD System to Provide Driving-Safety Information," *ETRI J.*, vol. 35, no. 6, Dec. 2013, pp. 1038–1047.
- [14] N. Parimal et al., "Application of Sensors in Augmented Reality Based Interactive Learning Environments," *IEEE Int. Conf. Sens. Technol.*, Kolkata, India, Dec. 18–21, 2012, pp. 173–178.
- [15] S. Khan, "SVP Infotainment and Connected Drive," *BMW Case Study*, BMW Group Corp., 2008. Accessed May. 10, 2014. http://www.genivi.org/sites/default/files/BMW_Case_Study_Download_040914.pdf
- [16] Developer Center, *GestureList*, Leapmotion Corp., 2014. Accessed Dec. 14, 2014. <https://developer.leapmotion.com/documentation/skeletal/cpp/api/Leap.Gesture.html>
- [17] Naver Developer Center, Map API, Naver Corp., 2014. Accessed Dec. 14, 2014. <http://developer.naver.com/wiki/pages/JavaScript>
- [18] 3D Connexion Support team, *Space Navigator for Notebooks*,

3D Connexion Corp., 2014. Accessed Nov. 14, 2014. <http://www.3dconnexion.eu/products/spacenavigator-for-notebooks.html>

[19] SCS Software, Euro Truck Simulator 2, *SCS Software Corp.*, Accessed Mar. 17, 2015. <http://www.eurotrucksimulator2.com/world.php>

[20] S. Franzen and M. Babapour, "HMI Certification - a Critical Review of Methods for Safety and Deficiency Evaluation of HMI Solutions for IVIS," *Chalmers University of Technology, Dept. Des. & Human Factors*, July 2011.

[21] P. Seewald et al., "D13.2: Results of HMI and Feedback Solutions Evaluations," *EcoDriver, SPI: Supporting Drivers in Eco-driving*, Tech. Rep. D13.2, Ver. 10, Nov. 2013.



Jinwoo Kim received his BS degree in electronic engineering from Yeungnam University, Kyungsan, Rep. of Korea, in 2007 and his MS degree in computer and electrical engineering from Hanyang University, Seoul, Rep. of Korea, in 2009. Since 2009, he has worked for the Electronics and Telecommunications Research Institute, Daejeon, Rep. of Korea, where he is now a senior researcher. His research interests include automotive and infotainment platforms; HMI; UI; UX; driver status recognition for autonomous systems; and interaction between humans and vehicles.



Jae Hong Ryu received his BS degree in electronic engineering from Pusan National University, Rep. of Korea, in 1991 and his MS degree in electronic engineering from Chungbuk National University, Cheongju, Rep. of Korea, in 2005. Since 1991, he has worked for LGIC and the Electronics and Telecommunications Research Institute, Daejeon, Rep. of Korea (where he is now a principle member of staff). His main research interests include embedded systems and IoT.



Tae Man Han received his BS degree in electronic engineering from Kyungbook National University, Daegu, Rep. of Korea, in 1985 and his MS degree in computer and electrical engineering from Chungnam National University, Daejeon, Rep. of Korea, in 2008. Since 1995, he has worked for the Electronics and Telecommunications Research Institute, Daejeon, Rep. of Korea, where he is now a principle member of staff. His main research interests include automotive and infotainment platforms; AUTOSAR; Infotainment platform-based smart connectivity technology software; ubiquitous sensor networks; and telematics.