Dynamic and Effect-Driven Output Service Selection for IoT Environments Using Deep Reinforcement Learning

KyeongDeok Baek and In-Young Ko, *Member, IEEE*

Abstract—**In the context of the recent emergence of the Internet of Things (IoT), human users and IoT-based services are interacting via physical effects, such as light and sound. Therefore, it is necessary to consider the quality of the delivery of physical effects to users by IoT devices for selecting services in IoT environments. However, traditional service-selection algorithms focus primarily on the network-level Quality of Service (QoS), such as latency and throughput. In this study, we improve on the visual-service effectiveness metric developed in our previous work to measure the effectiveness of the personalized delivery of physical effects of visual services to users by considering user- and application-specific factors. We evaluate the metric by conducting a user study, and the results show that the metric reflects users' perceived effectiveness with high accuracy. We also investigate the use of virtual reality (VR) to imitate physical environments for efficient evaluation of the metric. Based on this metric, we develop a dynamic effect-driven output-service selection agent (DEOSA) that selects output services dynamically by considering the effectiveness of service-effect delivery. By adopting a state-of-the-art reinforcement-learning algorithm, DEOSA can learn the optimal policy for selecting output services that can be generalized to various environments. We evaluate DEOSA in simulated IoT environments and show that it can learn the optimal policy successfully; it generally performs better than traditional greedy algorithms in terms of the visual service effectiveness metric and the replacement overhead in randomly generated test environments.**

Index Terms—**Effect-driven service selection, Internet of Things (IoT), reinforcement learning, virtual reality (VR), visual-service effectiveness.**

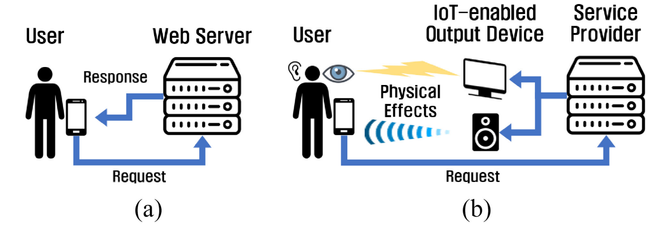


Fig. 1. Provision of (a) Traditional Web services and (b) IoT services.

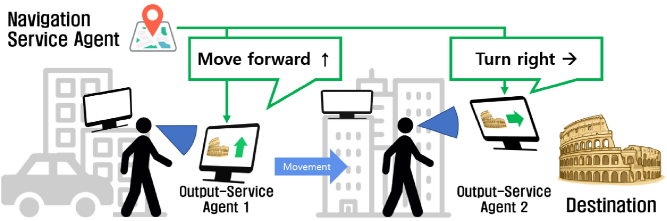


Fig. 2. Motivating scenario for the effect-driven selection of IoT services.

I. Introduction

WEB SERVICES have recently been extended to the Internet of Things (IoT) services, which utilize various interconnected IoT devices deployed in physical environments [1], [2]. IoT services interact with users by sensing and generating physical effects, such as light and sound, via IoT devices; in contrast, traditional Web services interact with users mostly by exchanging messages. Specifically, many IoT-enabled services utilize IoT devices as media to produce physical effects as outputs to their users, as illustrated in Fig. 1. Therefore, it is critical to correctly select IoT devices that can appropriately deliver the physical effects required by a given service to users. However, there is a lack of quantitative metrics to measure the quality of physical effects that users perceive, which makes the selection of services associated with IoT devices challenging [3], [4].

Fig. 2 illustrates a motivating scenario of selecting an appropriate IoT service by considering the delivery of its output physical effect. In the scenario, we consider an enhanced street navigation system that shows directions to users using public displays embedded in a physical space. By using public displays instead of smartphones, this system can enhance user experience (UX) and safety by reducing the necessity of users switching their attention between their smartphones and the surrounding environment [5], [6]. We assume that each public display is servicized as a visual output service and provides an application interface with its description. A visual output service can describe and register itself to one of the service

Manuscript received 7 March 2022; revised 16 June 2022 and 12 August 2022; accepted 21 September 2022. Date of publication 3 October 2022; date of current version 6 February 2023. This work was supported in part by the Ministry of Science and ICT (MSIT), South Korea, under the Information Technology Research Center (ITRC) Support Program supervised by the Institute of Information and Communications Technology Planning and Evaluation (IITP) under Grant IITP-2022-2020-0-01795, and in part by the National Research Foundation of Korea (NRF) Grant funded by the Korea Government (MSIT) under Grant 2019R1A2C1087430. *(Corresponding author: KyeongDeok Baek.)*

This work involved human subjects or animals in its research. Approval of all ethical and experimental procedures and protocols was granted by the KAIST Institutional Review Board under Application No. KH2020-020.

The authors are with the School of Computing, Korea Advanced Institute of Science and Technology, Daejeon 34141, Republic of Korea.

Digital Object Identifier 10.1109/JIOT.2022.3211270

registries for discovery; hence, the registries work as brokers between users and service providers, following the paradigm of service-oriented architecture (SOA) [7].

There may be numerous visual output services available, the same as the number of public displays installed in the city. If one of the visual output services is selected without considering the user's physical context, the user may not recognize the navigation required because the associated public display would not be in the user's sight. Therefore, visual output services should be selected by considering the personal and environmental contexts of the users, such as visual acuity [8] and viewing angle toward public displays [9] that may affect the recognition of visual content. However, it is challenging to understand and quantitatively measure how various human and environmental factors affect the effectiveness of delivering the physical effects from visual output services to a user. Moreover, it is necessary to enable automatic selection and replacement of output services in highly dynamic IoT environments to reduce users' intervention.

In Fig. 2, output service 1 is selected, which is associated with a public display located in front of the user and sufficiently close. The selected visual output service should be replaced by another when the user can no longer receive the physical effect from the selected service. In Fig. 2, as the user proceeds to the destination, output service 1 should be replaced with output service 2, as the user travels beyond the range of output service 1 and into that of output service 2.

As illustrated in the scenario, it is important to dynamically select and replace appropriate output services considering the delivery of the physical effects of IoT services. However, traditional Web service selection algorithms [10], [11], [12], [13] cannot be applied directly to the selection of output services because of the following two challenges.

First, metrics for evaluating the effectiveness [14] of output services in terms of the extent to which physical effects are delivered appropriately to the user have not been developed [4]. Such quantitative metrics are necessary to use optimization-based service-selection algorithms. Traditionally, Quality of Service (QoS) attributes, such as response time, cost, trust, and availability have been considered for the quantitative evaluation of candidate Web services [10], [13], [15], [16], [17], [18], [19], [20], [21], [22]. In contrast, measuring the effectiveness of output services is not trivial because of the complexity of human perception, owing to various environmental and human factors that are difficult to model and predict [4]. Thus, designing appropriate metrics to measure the quality of the delivery of physical effects by considering the complex dynamics of IoT environments is a challenging problem.

Second, the quality and availability of IoT services may fluctuate because of the highly dynamic nature of IoT environments. However, existing service-selection algorithms that dynamically select services at runtime [11], [16], [17], [18] do not support the replacement of services when the selected service becomes unavailable or its quality degrades. Performing the selection again is a naive solution to the replacement of output services, but most existing selection algorithms only consider the current snapshot and ignore the future states of the environment. Therefore, it is necessary to dynamically select and replace output services in a predictive manner to maintain the effectiveness of the service in the long term based on the physical dynamics of the environment [3].

In a previous study [3], we modeled a visual output service as a representative among various types of output services. We defined a metric named visual-service effectiveness that estimates users' perceptions of the effectiveness of visual output services in terms of their delivery of visual effects to the users. The metric was developed based on the physical characteristics of lights and the human vision system, but a user study was not conducted to evaluate the metric subjectively. We defined a dynamic and effect-driven output-service selection problem based on the metric by extending the traditional service-selection problem. We developed an algorithm that finds an optimal selection policy maximizing the visual-service effectiveness and minimizing the overhead of replacing the output services while supporting users' tasks. We adopted a deep reinforcement learning technique, which is considered effective in solving sequential decision-making problems to train the policy model. By applying a deep reinforcement learning technique to the selection problem, complex dynamics of the IoT environments that affect the user and services could be learned to optimize the effectiveness metric. Although we used the deep Q-network (DQN) method [23], which is a powerful and the most common reinforcement learning algorithm, the scope for improvement nonetheless remained in terms of the performance and training stability of the output-service selection.

In this study, we improve the visual-service effectiveness metric from our previous study by considering user-specific factors and characteristics of output devices and application domains to extend the selection of output services to be more personalized to users. Furthermore, we validated the practicality of the metric by conducting a subjective user study. The results show that the metric estimated the perceived effectiveness of output services with high accuracy, precision, and recall scores.

In addition, we investigate the use of virtual reality (VR) techniques to reproduce the user study in a virtual environment. We found that VR techniques can effectively simulate situations of delivering physical effects to users via output services in an IoT environment. We can improve and test the service effectiveness metric efficiently under various scenarios in virtual IoT environments by using VR techniques.

We further propose a dynamic effect-driven output-service selection agent (DEOSA) as an improved solution to the dynamic and effect-driven output-service selection problem. DEOSA can deal with variable numbers of candidate services, in contrast to other reinforcement learning agents designed to consider static numbers of actions. To improve the stability of the training process, DEOSA adopts Munchausen reinforcement learning [24], which is one of the latest reinforcement learning algorithms. The performance of the DEOSA was evaluated in a simulated IoT environment, which reflects the user-specific factors found in the user study. The simulation results show that DEOSA successfully learned an optimal policy of selecting output services that can be generalized to various environments, and achieved higher service effectiveness and lower replacement overhead on average than the baseline approaches in testing environments substantially different from that in which DEOSA was trained.

The main contributions of this study are summarized as follows.

1) We characterized and defined a new type of service called output services that deliver physical effects to environments via IoT-enabled devices.
2) We extended the service selection problem to consider the effectiveness of the delivery of physical effects by IoT devices that are associated with service capabilities.
3) We showed that IoT environments can be reproduced efficiently using VR techniques to reduce the cost involved in evaluating the service effectiveness metric under various usage scenarios while considering personal and environmental conditions.
4) The results show that a deep reinforcement learning technique can learn an optimal policy to select output services dynamically that is generalized to various user contexts and environmental conditions. The implementations of DEOSA and the simulation used for its evaluation are publicly available.

The remainder of this article is organized as follows. In Section II, we discuss related work. In Section III, we present our improvements on the visual-service effectiveness metric and the evaluation based on a user study. In Section IV, we formally define the dynamic effect-driven output-service selection problem in IoT environments, propose DEOSA as a solution, and present the results of evaluations performed in simulated IoT environments. Finally, we present some concluding remarks in Section V.

II. Related Work

The dynamic service selection problem has been studied extensively by the researchers of Web services. This problem involves the sequential selection of services during runtime based on a predefined workflow or data flow of services to accomplish a given task on behalf of a user [11], [12], [16], [18]. Typically, the service selection problem is formulated as an optimization problem, and traditional algorithms select services from among functionally equivalent candidates to optimize network-level QoS attributes, such as response time, availability, and cost. To the best of our knowledge, there have been no works reported in the relevant literature that consider the continuous delivery and ensure the quality of physical effects generated by IoT devices for the dynamic selection of output services.

There have been some attempts to apply reinforcement learning to the composition of Web services. A series of works utilized multiple reinforcement learning agents to search for optimal services in terms of QoS for a given Web service composition problem [25], [26]. At each state, reinforcement learning agents select one of the candidate services for the corresponding function and transit to one of the next states until it reaches a terminal state. The main challenge is to solve the scalability problem caused by the massive set of candidate services, which contain every Web service globally. However, these algorithms focused on finding an optimal instance of an abstract composition problem by selecting concrete services while considering simple QoS fluctuations. Therefore, their approaches are not appropriate for dynamic selection and adaptation of physical IoT services during runtime, where the effectiveness of services is directly affected by selection decisions.

Commonly, under the framework of the Web service composition, the main goal is to search for the optimal services with the best QoS values for a given workflow of abstract services [27], [28]. Searching for the best combination of the services is a challenging problem because of the large set of candidate services, correlated QoS of candidate services [13], and the uncertainties of reality [27]. Therefore, traditional approaches to service composition mainly focus on reducing the search space to improve the efficiency of algorithms [26], [29]. In contrast, the main focus of our approach is on the generalization to learn an optimal policy for selecting services that can be applied to various situations generally and can adapt to real-time variations in the environments. Therefore, the reinforcement learning agents used in the proposed approach were trained under various environmental conditions, in contrast to existing methods that focus only on specific environments.

Previously, we proposed an approach that extended the service selection problem by including the spatial properties of IoT devices [30]. The proposed algorithm was designed to select a set of services to be aware of the spatial locations of IoT devices and users that should operate in a spatially cohesive manner. In this regard, a new metric referred to as spatio-cohesiveness was defined to measure how the associated devices of selected services locate cohesively. However, high spatio-cohesiveness cannot guarantee the successful delivery of physical effects to the user; therefore, this measure is insufficient.

In a more recent study, we developed a dynamic service replacement method referred to as a service handover [31]. Under the concept of service handover, services that require some provisioning time can be replaced with alternative services dynamically during runtime, allowing services to be provided continuously under the dynamically changing conditions that prevail in IoT environments. In this study, we adopted the concept of service handover for our output-service selection. Thus, the selection process has two phases: 1) an initial selection and 2) a dynamic handover. We also trained a selection agent using reinforcement learning, but the optimization goal of the agent was limited to the spatio-cohesiveness metric, and we only considered the spatial properties.

In the domain of vehicle resource management, a recent study proposed an algorithm that dispatches requests of vehicle resources to the appropriate resource provider [32]. The proposed algorithm utilizes the double-DQN technique [33], which is an extension of DQN [23], to balance the utilization of vehicle resources and success rate. Although the algorithm is for the management of vehicle resources, it is comparable to DEOSA regarding the matchmaking structure between requesters and providers. Therefore, we compare the performance of DEOSA with the double-DQN-based algorithm as a state-of-the-art in Section IV-C.

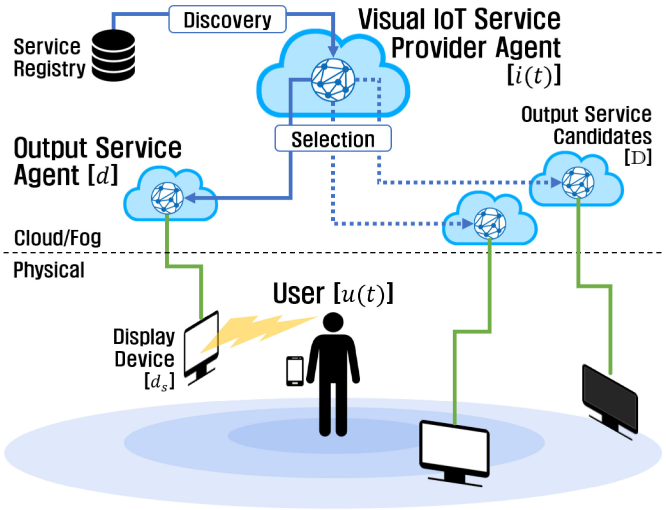

Discovery
Service Registry
Visual IoT Service Provider Agent $[i(t)]$
Selection
Output Service Candidates $[\mathbb{D}]$
Output Service Agent $[d]$
Cloud/Fog
Physical
User $[u(t)]$
Display Device $[d_s]$

Fig. 3. Service-oriented IoT environment.

III. Visual-Service Effectiveness Metric

In this section, we explain the improvements made to the visual-service effectiveness metric to select more personalized and customized output services for users. First, we formally define the basic models of IoT environments, users, and services considered. Based on these models, we define the improved visual-service effectiveness metric and evaluate how the metric estimated users' perceived effectiveness successfully by conducting a subjective user study. Finally, we explain the user study that we designed and ran in a VR environment to simulate an IoT environment and reproduce the results of a user study conducted in a physical environment.

A. Formal Models of Target Environments

1) Service-Oriented IoT Environment Model: Fig. 3 shows a service-oriented IoT environment [2] as considered in this study. In this environment, service providers run on either a cloud or local edge cloud (fog) and provide IoT services to users. In particular, output-service agents that require local IoT devices such as displays should be deployed and run on an edge cloud near the user. Available output-service agents near a user can be discovered using a service registry located in the cloud or edge clouds.

Formally, we define the status of an IoT environment at a given time t as follows: $\text{env}(t) = \{u(t), i(t), \mathbb{D}\}$, where $u(t)$ is a user, $i(t)$ is a visual IoT service provider, and $\mathbb{D}$ is a set of candidate output-service agents associated with display devices. When user $u(t)$ requests visual content from an IoT service provider $i(t)$, it initiates a service selection process to find the most appropriate output-service agent among the candidates $\mathbb{D}$ that can deliver the requested content to the user effectively. We assume that physical obstacles such as walls that may block the visual effects generated by output services are not present; however, we intended to consider such environmental constraints in future research. In this study, we focused on the characterization of users and visual output services. We define the physical locations of the user and the IoT devices in a 3-D Cartesian coordinate system consisting of x-, y-, and z-axes, $\vec{l} = (x, y, z)$, and define the distance $\delta(\vec{l}_a, \vec{l}_b)$ between two coordinates $\vec{l}_a$ and $\vec{l}_b$ as the Euclidean distance $\|\vec{l}_a - \vec{l}_b\|$.

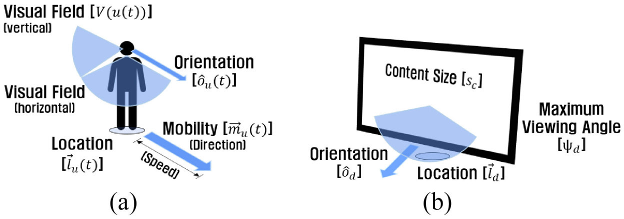

Visual Field $[\mathbb{V}(u(t))]$
(vertical)
Orientation $[\hat{o}_u(t)]$
Visual Field
(horizontal)
Mobility $[\vec{m}_u(t)]$
(Direction)
Location $[\vec{l}_u(t)]$
(Speed)
(a)
Content Size $[s_c]$
Maximum Viewing Angle $[\psi_d]$
Orientation $[\hat{o}_d]$
Location $[\vec{l}_d]$
(b)

Fig. 4. Models of physical entities in the environment. (a) User model. (b) Output service model.

2) User Model: Fig. 4(a) shows the user model adopted herein to represent the attributes of a user's condition and status that may affect their perception of the visual effects from the output services. A user $u(t) = \{\vec{l}_u(t), \hat{o}_u(t), \vec{m}_u(t)\}$ is defined as a person who performs the role of a service consumer utilizing the IoT service, where $\vec{l}_u(t)$ is a vector representing the location, $\hat{o}_u(t)$ is a unit vector representing the facing orientation, and $\vec{m}_u(t)$ is a vector representing the mobility of the user at time t. Based on the location and orientation of the user's eyes, we can calculate the visual field of the user $\mathbb{V}(u(t))$, in which the user's vision system perceives the light. The visual field of a user is defined as the total area that the user's eyes can see. Note that the range of the visual field may vary among users due to their eye conditions such as cataracts of elders, and also may vary in different circumstances, such as nervousness or panic.

3) Service Model: A visual content-delivery IoT service $i(t) = \{c_s(t), d_s(t)\}$ is defined as an activity that presents visual content $c_s(t)$ by utilizing an output service $d_s(t)$ at time t. The visual content $c_s(t)$ is defined as a set of visual objects such as text. An output service $d_s(t)$ may be replaced by an alternative $d_s(t+1)$ at the next time step. Fig. 4(b) shows the model of an output service. An output service $d = \{\vec{l}_d, \hat{o}_d, \psi_d, f_d\}$ is a service provided by a display device, where $\vec{l}_d$ is a vector representing the location, $\hat{o}_d$ is a unit vector representing the orientation, ψ_d is the maximum viewing angle of the associated display device, and f_d is the scaling factor for the content size. If the viewing angle exceeds its maximum value ψ_d, the content cannot be properly perceived. We assume that a display device has a rectangular shape, a static location, and a horizontal orientation, which is the most common type of display; the other types of displays as well as mobile units will be considered in future works.

B. Measuring Visual-Service Effectiveness

We define the visual-service effectiveness metric as a function that accepts three types of inputs, as shown in Table I, including: 1) user; 2) service; and 3) content-related factors. The output of the metric is a numerical value representing the visual-service effectiveness in the range [0, 1], where 0 and 1 represent the minimum and maximum effectiveness, respectively. The metric is designed with several conditions in a rule-based manner, given the user and service at a certain

TABLE I
INPUT FACTORS OF VISUAL-SERVICE EFFECTIVENESS METRIC

Type	Variability	Name (Notation)
User	Variable	Location of user ($\vec{l}_u(t)$)
	Variable	Orientation of user ($\hat{o}_u(t)$)
	Constant	Maximum visual field of user (θ_u)
	Constant	Visual acuity of user (a_u)
Service	Constant	Location of display device ($\vec{l}_d$)
	Constant	Orientation of display device ($\hat{o}_d$)
	Constant	Maximum viewing angle (ψ_d)
	Constant	Scaling constant (f_d)
Content	Variable	Size of objects (s_c)
	Constant	Scaling constant of the visual angle (k_c)

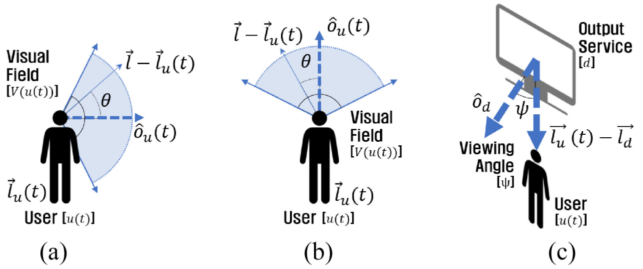


Fig. 5. Examinations of visibility conditions. (a) Vertical field. (b) Horizontal field. (c) Viewing angle.

time, and is expressed by the following equation:

$$e(u,i,t) = \begin{cases} 0, & \vec{l}_d \notin \mathbb{V}(u(t)) \\ 0, & \psi(u,i,t) > \psi_d \\ 0, & \dfrac{\phi(u,i,t)}{k_c} < 10^{a_u} \\ \rho(u,i,t), & \text{otherwise} \end{cases} \tag{1}$$

where the first three statements consider the visibility conditions of the user, and the last statement is the application-specific measurement of effectiveness. This metric was developed by extending the visual-service effectiveness metric defined in our previous work [3]. In particular, we improved the previous metric to use personalized values of visual acuity, maximum visual field, and maximum viewing angle for each visibility condition, instead of using average values. Additionally, we added application-specific customization of effectiveness in addition to visibility conditions. The details of the metric are provided in the following sections.

1) Visibility Conditions: To deliver visual effects to a user and enable them to recognize and perceive visual content from an output service, its associated display device should at least meet some specified visibility conditions. We consider three conditions to examine visibility from a user's perspective: 1) the user's visual field; 2) the orientation of the display device; and 3) the distance between the user and the display device.

First, if the associated display device is not in the user's visual field $\vec{l}_d \notin \mathbb{V}(u(t))$, the user cannot perceive the visual contents of the output service; in this case, the effectiveness should be measured as the minimum. As shown in Fig. 5(a) and (b), we determine the user's visual field at time t based on the angle between the vectors that represent the relative location of the associated display device to the user $\vec{l} - \vec{l}_u(t)$ and the orientation of the user $\hat{o}_u(t)$ as follows:

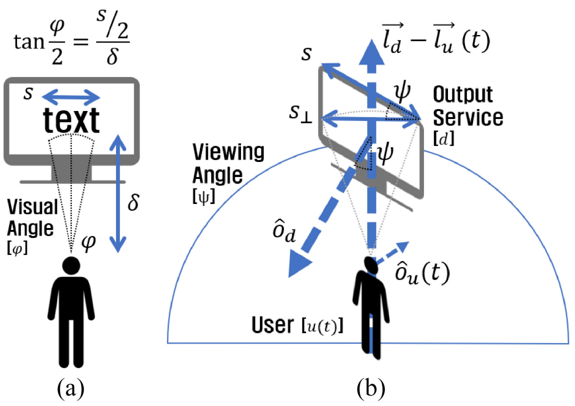


Fig. 6. Calculating perceived visual angle of a user. (a) Visual angle. (b) Perceived size of objects.

$$\mathbb{V}(u(t)) = \left\{\vec{l} \mid \theta\left(\hat{o}_u(t), \vec{l} - \vec{l}_u(t)\right) \le \theta_u\right\} \tag{2}$$

where the angle θ between two vectors can be calculated using the dot product of the vectors as follows:

$$\theta\left(\vec{i},\vec{j}\right) = \cos^{-1}\left(\frac{\vec{i}\cdot\vec{j}}{\|\vec{i}\| \times \|\vec{j}\|}\right). \tag{3}$$

Note that the maximum visual field of the user θ_u is the upper bound of the visual field that the user can perceive. This usually varies among different users. Therefore, to personalize the visual-service effectiveness metric, we consider the maximum visual field value θ_u in (2) for different users to calculate the visual field $\mathbb{V}(u(t))$. Furthermore, the visual field includes not only foveal vision but also peripheral vision, where acuity is low [34]. Instead of considering the low acuity of peripheral vision, we assume that users can adjust the orientation of their eyes and head toward the display device after noticing it to perceive its contents. Otherwise, with foveal vision, the effective range would be unnecessarily narrow.

Second, if the associated display device of the output service does not face the user, the user may be unable to perceive the service content even though the device is in the visual field. Whether the display device faces the user can be examined by measuring the viewing angle $\psi(u,i,t)$ between the orientation of the device $\hat{o}_d$ and the relative location of the user to the output service $\vec{l}_u(t) - \vec{l}_d$ as shown in Fig. 5(c), where

$$\psi(u,i,t) = \theta\left(\hat{o}_d, \vec{l}_u(t) - \vec{l}_d\right). \tag{4}$$

If the viewing angle $\psi(u,i,t)$ is larger than the maximum viewing angle ψ_d, which means that the display device does not face toward the user, in which case the effectiveness should be estimated as the minimum. The maximum viewing angle varies among different display devices based on their physical properties. Therefore, the visual-service effectiveness metric has been improved to consider device-specific maximum viewing angles ψ_d of the output devices when calculating the second visibility condition in (1).

Third, even when the visual effects are successfully perceived by the user, the user may be unable to recognize the content if it appears excessively small from the user's perspective, according to the user's visual acuity. As shown in Fig. 6(a), the visual angle of an object from the perspective of the user $\phi(u, i, t)$ can be calculated using the following formula:

$$\phi(u, i, t) = 2 \times \tan^{-1}\left(\frac{s_{\perp}(t)}{2 \times \|\vec{l}_u(t) - \vec{l}_d\|}\right) \tag{5}$$

where $s_{\perp}(t)$ is the perceived size of the object. As shown in Fig. 6(b), the perceived size of the object from the user's perspective may not be the same as the actual size when the user is not on the central axis of the display device, because the object may appear tilted. Considering this tilt distortion of the visual object, the perceived size of the object can be calculated using the following equation:

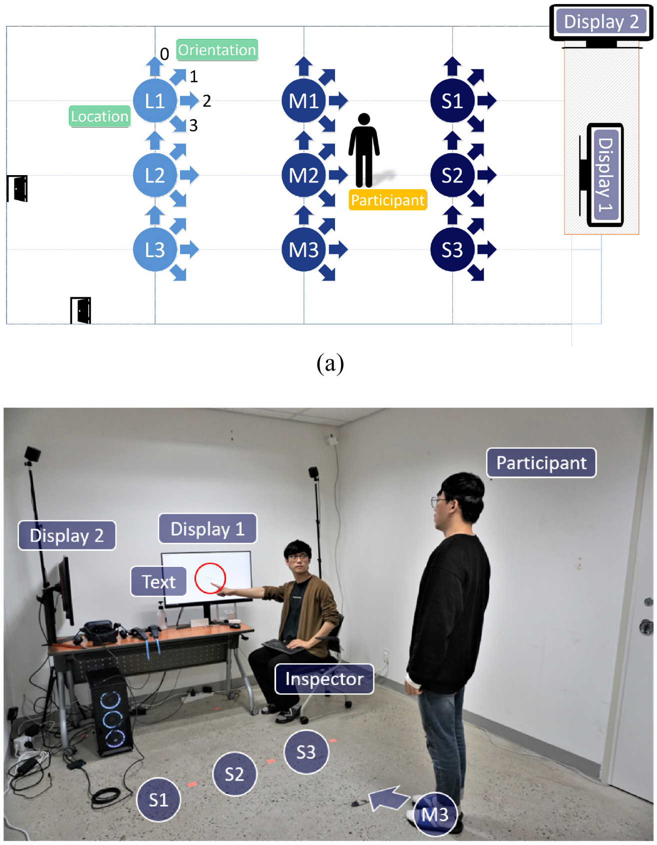
$$s_{\perp}(t) = s(t) \times \cos\psi(u, i, t) = s(t) \times \frac{\hat{o}_d \cdot \left(\vec{l}_u(t) - \vec{l}_d\right)}{\|\vec{l}_u(t) - \vec{l}_d\|} \tag{6}$$

where ψ is the viewing angle between the orientation vector of the associated display device and the relative location vector of the user and the display device. The scaled size of the object is calculated by multiplying the content-level object size and the scaling factor of the output service at time t, $s(t) = s_c(t) \times f_d$. For a visual object included in the content with a perceived size of $s_{\perp}$, we can examine whether the object is recognizable by the user by calculating the object's visual angle and comparing it with the user's visual acuity according to the following inequality:

$$\frac{1}{k_c} \times \left(2 \times \tan^{-1}\left(\frac{s_{\perp}(t)}{2 \times \|\vec{l}_u(t) - \vec{l}_d\|}\right)\right) < 10^{a_u} \tag{7}$$

where 10^{a_u} is the minimum visual angle that the user can resolve while the user's visual acuity a_u is on the logarithm of the minimum angle of resolution (logMAR) scale [35], and the user cannot recognize the object if the calculated visual angle is smaller than the minimum. The calculated visual angle is divided by the type-specific scaling constant k_c which varies for each content type, to compare the visual acuity and the actual sizes of the objects in the content based on the definition of visual acuity metrics. In the case of textual content, the value of the constant k_c is five according to the Snellen chart's scaling ratio between the visual acuity value and the size of the corresponding optotypes [36].

2) Application-Specific Effectiveness: When all visibility conditions are met, the application-specific visual-service effectiveness $\rho(u, i, t)$ can be measured quantitatively in various contexts. For instance, in the case of textual content, letter-level or word-level legibility can be examined to estimate whether a word is legible to the user. Furthermore, certain human cognitive factors could be considered to determine whether specific content could be consumed by the user according to the user's current cognitive status. However, we believe that no universal, the objective effectiveness metric can be developed, owing to the high heterogeneity of application-specific factors and contexts. Therefore, the engineering problem of efficiently developing an application-specific metric is left as a task for future work.



(a)

(b)

Fig. 7. Experiment in laboratory environment. (a) Configurations. (b) Snapshot.

C. Evaluation of Visual-Service Effectiveness Metric

In this section, we evaluate the estimation performance of the visual-service effectiveness metric to determine the extent to which the metric successfully estimated users' perceived effectiveness. We conducted a subjective user study in a laboratory environment for this purpose.

1) Subjective User Studies in Laboratory: We placed two display devices on either side of the laboratory for the experiment, as shown in Fig. 7(a), to simulate the situations in which the user is far from or side-by-side with the output service. A total of 216 conditions were considered for each user, including nine locations at three distances, four orientations, two display devices, and three text sizes. We placed markers to guide the locations and orientations of the users. We used two LG 32-inch QHD (2560 × 1440) monitors (model 32GK650F) using a VA panel with a maximum viewing angle of 89° as display devices in the experiments and set the brightness to 350 cd. Randomly selected texts, such as "MOVE FORWARD," "MOVE BACKWARD," "TURN RIGHT," and "TURN LEFT," were presented on the display devices for each condition to simulate the situation of providing directions.

Fig. 7(b) shows the experimental environment. First, we presented an instruction video to each participant to introduce the research work by providing participants with brief

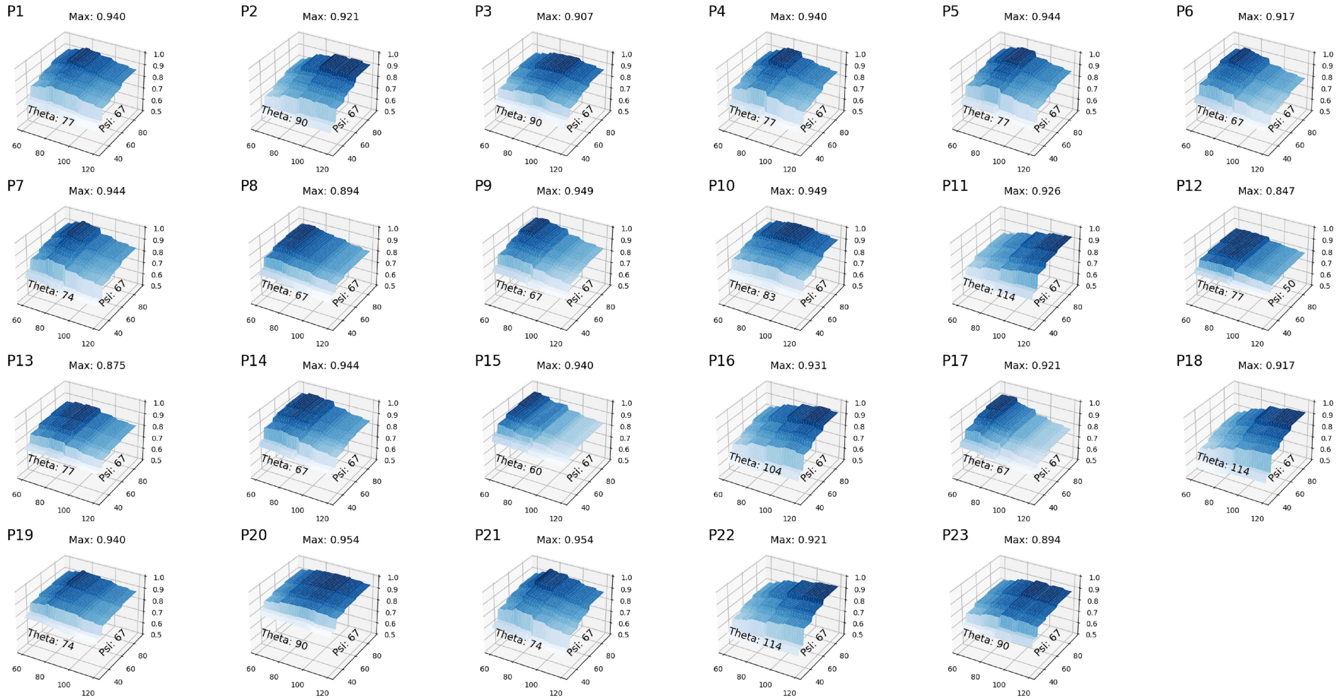


Fig. 8. Accuracy by the maximum visual field and maximum viewing angle for each participant.

background information and explaining the details of the experimental procedure. Then, the participants were asked to move and face the first location and orientation. We presented texts of different sizes on the display devices and asked the participant to report the perceived effectiveness of the output service by verbally indicating whether the shown texts are readable or not readable. Note that we asked the participants not to rotate their necks more than 30° left or right in the horizontal direction to restrict the orientation to the natural range. Afterward, the participants were asked to turn their body 45° right or move to the next location following the order. The participants repeated the process until all the conditions are covered. Additionally, the participants were asked to move to the center (M2) and face the monitor to approximate their visual acuity by identifying the minimum size of the text they could read. Finally, the participant were asked to complete a questionnaire of: 1) personal information, such as gender, age, height, visual acuity, and visual impairments; 2) information on their familiarity with IoT; and 3) general feedback related to the experiment.

A total of 23 participants were tested in the experiments. The visual acuity of each participant was recorded, although this was not possible for two participants because they were unaware of their visual acuity. Ten out of 23 participants (42%) were wearing glasses, and the heights ranged from 154 to 182 cm.

2) Results of the Estimating Performance of the Metric: As indicated by (1), the visual-service effectiveness metric requires three boundary values: the visual acuity (a_u), maximum visual field (θ_u), and maximum viewing angle (ψ_d) to check the visibility conditions. We obtained the visual acuity of each participant either from the survey or approximated it from experimental results. However, the maximum visual field and maximum viewing angle were largely unavailable. Therefore, we attempted to determine the maximum visual field and maximum viewing angle that maximized the accuracy of the metric for each participant.

Fig. 8 shows the accuracy using a range of values for the maximum visual field and maximum viewing angle for each participant. The x- and y-axes in the figure represent the maximum visual field and maximum viewing angle, respectively, whereas the accuracy is shown on the z-axis, where dark blue indicates the highest accuracy. The results show that the values associated with the highest accuracy varied for each participant. Moreover, the maximum visual field differed among the participants, ranging from 67° to 114° (82° on average). This occurred because the visual field is a user-related factor affected by the visual capability of each user. Note that the visual field of each participant was affected not only by visual capability but also by how closely the participant followed the instructions of the experiment because we asked participants to restrict the neck rotation in the horizontal direction. In contrast, the maximum viewing angle was 67° for most of the participants, except for one, because the viewing angle is a service-related factor that depends highly on the device's characteristics.

Fig. 9 shows the estimation performance (accuracy, precision, and recall) of the visual-service effectiveness metric for each participant. The results show that the accuracy was generally high for all participants and that precision and recall were generally high for most participants. However, the recall of two participants was lower than 0.8, whereas their precision was higher than 0.9. Such an imbalance may have occurred because we used the maximum visual field values found from the previous result, which maximized accuracy without considering precision and recall.

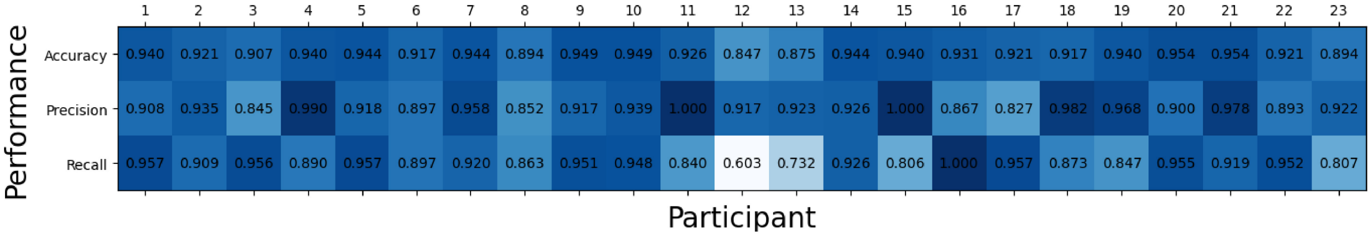


Fig. 9. Estimation performance for each participant.

TABLE II
CONTINGENCY TABLE OF THE EFFECTIVENESS ESTIMATION

		Perceived		Sum
		Effective	Ineffective	
Estimation	Effective	1849	154	2003
	Ineffective	220	2745	2965
Sum		2069	2899	4968

Table II compares the effectiveness perceived by the participants with that estimated by the metric. A total of 4968 cases were collected from the participants, of which 1849 were true positives, 154 were false positives, 220 were false negatives, and 2745 were true negatives. The results show an accuracy of 0.925, precision of 0.923, and recall of 0.894, indicating that the metric successfully estimated the perceived effectiveness of the participants.

Commonly, users may not know their maximum visual field, and the visual field may change because of various contexts. Therefore, we also attempted to identify an average boundary value that could be applied to users without significant performance degradation. As shown in Fig. 10, the accuracy is maximized when the maximum visual field and maximum viewing angle are set to 80° and 70°, respectively. This means that the range of the participant's visual field was approximately 160° horizontally, and the maximum viewing angle of the display devices was approximately 70°, which is slightly less than the device specification (89°). The accuracy was 0.897, which would not be considered low, but was less than the highest accuracy of 0.925, which was obtained using the personalized maximum visual field, as shown in Table II. In other words, the metric must be personalized by considering different maximum visual field values of the participants for best performance. However, when the maximum visual field of the user is unknown, visual-service effectiveness can be measured using the average value of the maximum visual field without severe degradation of the performance.

Fig. 11 shows the receiver operating characteristic (ROC) of the visual-service effectiveness metric. Because the visual-service effectiveness metric has multiple boundary values, namely, visual acuity (a_u), maximum visual field (θ_u), and maximum viewing angle (ψ_d), we plotted the ROC by using dots rather than curves for improved visualization; multiple curves may overlap with each other when multiple boundary values are used. The results show that the area under the curve (AUC) value exceeded 0.9, which implies that the metric can separate effective and ineffective cases stably.

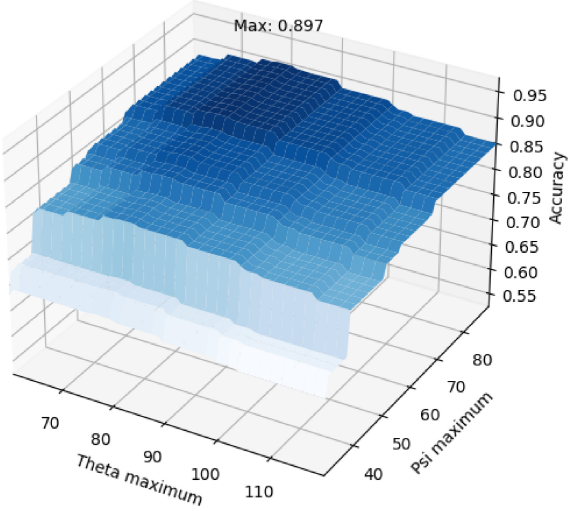


Fig. 10. Accuracy by general boundary values.

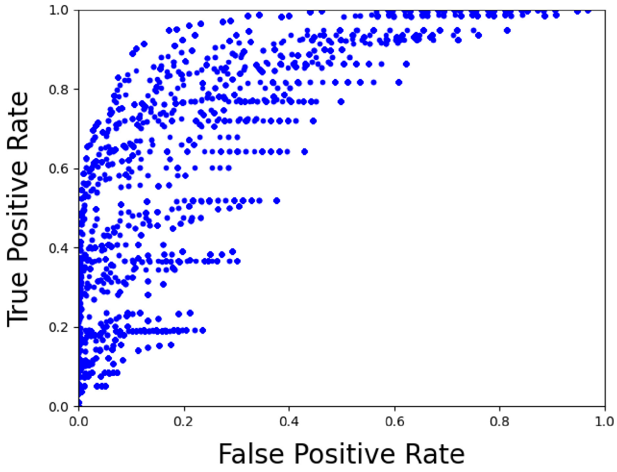


Fig. 11. ROC with different boundary values.

3) Threats to Validity: The implemented laboratory environment was sufficient to evaluate the metric as it included every component required to test all three visibility conditions considered primarily in the metric. The first visibility condition, the visual field, was tested by rotating the participants' orientations. The second visibility condition, the viewing angle, was tested by locating participants such that the viewing angles ranged from 0° to 90°, especially toward display 2, which was installed obliquely toward the participants, as shown in Fig. 7. The third visibility condition, the visual angle, was tested by locating the participants at short, medium, and long distances from the displays.

Because the participants were asked to answer a simple and short question about whether the text shown on the display was readable or not, we thought that the participants could keep consistent over different conditions of the experiment. We also collected enough data from many participants to minimize the effects of potential inconsistency that might occur. Furthermore, we gave enough time for the participants to stabilize their emotional states before starting the experiment. The experiment was conducted similarly to the common visual acuity test. Therefore, we thought that the effects of very brief eye contact and conversations between the inspector and the participants could be ignorable.

Note that we only consider the horizontal movement of a user because the range of the vertical movement is relatively narrow and has less impact than that of the horizontal movement in typical IoT environments. Moreover, to reduce the effects of font legibility, we fixed the font type used for the experiment to Arial, which is one of the most common and readable font types.

During the experiments, we restricted the horizontal rotation of the participants' necks by 30° to the left and right to approximate mobile users' behavior. However, the constraint was not applied strictly, and eight participants claimed that the 30° requirement was too ambiguous to follow. We plan to extend the experimental implementation to real mobile users, in which case these constraints can be safely discarded.

We used only four simple phrases to check readability by reducing the bias caused by the different degrees of legibility of the phrases. However, six participants asserted that the text pool was too small and that they could read each text by their silhouette despite their vision being imperfect. We may enlarge the text pool to reduce these learning effects, but it would have to be designed carefully to avoid placing excessive emphasis on the legibility of each text.

D. Discussions on the Metric

Our visual-service effectiveness metric requires the user's location and orientation as inputs, which may not be available using the current infrastructure. The locations of users can be measured by using the global positioning system (GPS) outdoors and an indoor positioning system (IPS) indoors, such as Sewio's indoor tracking real-time location system (RTLS)[1] based on ultrawideband (UWB). Furthermore, it is expected to become possible to measure users' orientation by using wearable devices, such as embedded and sufficiently small inertial measurement units (IMUs), such as the Sewio IMU tag and eSense.[2]

In the metric, visual acuity and visual field may be affected by situational contexts of users, and the viewing angle may be affected by the spatial structure of the environment. For instance, the visual field will be narrow when a user is nervous because of safety-critical situations. Our effectiveness metric can deal with such cases by modifying the personalization variables; however, determining the exact values automatically in real time is a challenging problem owing to the excessively large number of factors. Therefore, it may be infeasible to customize the effectiveness metric manually for each service type [4]. We are currently investigating the use of recent machine-learning techniques to customize the effectiveness metric in an automated manner by utilizing the data collected from an IoT environment. As two of the participants suggested, implicit feedback, such as the time spent for reading the text, can be utilized to infer service effectiveness without interrupting users.

The visual-service effectiveness metric does not consider content-related factors, such as the color, font types, and distortion of visual content that may affect the readability of textual contents. Furthermore, user-related factors, such as emotional state and cognition may affect the visual-service effectiveness. We plan to extend the metric in our future work to consider these by conducting an in-depth investigation of the effect of each of the factors. In particular, we only considered textual content in this work, although many other types of visual content could be used, such as illustrative images, informational signs, or even videos. For textual content, visual-service effectiveness can be defined as a readability measurement of texts, but this is not appropriate for other types of content.

The current version of the metric ignores physical obstacles and application-specific factors to focus solely on the physical characteristics of visual output services and the human visual system. We will further extend the metric to consider more complex factors based on the findings of this study in future work.

Furthermore, the metric is defined to evaluate the effectiveness of delivering the physical effects of an IoT service to a single user. However, it is also an important issue to deliver service effects to multiple users who share the same service [37]. To support group users, it is necessary to mediate the personal factors of the users so that the physical effects of the service can be effectively consumed by all the users at the same time. We plan to extend the visual service effectiveness metric to support multiple users in our future work.

A participant suggested that other senses, such as the auditory senses, should be considered. We plan to study how these various types of physical effects may constructively or destructively interfere with each other, thus extending our visual-service effectiveness concept to general service effectiveness. Because visual and acoustic effects are delivered in different ways from output services to users, a metric of acoustic service effectiveness would need to be designed based on different domain knowledge, but follow a structure similar to that of visual-service effectiveness. These other types of physical effects should not be considered in isolation, because they can also cause interference. This interference may be constructive and synergistically improve the overall effectiveness of the service, or it may be destructive and degrade the overall effectiveness. The incorporation of an interference model into our service effectiveness concept will allow the selection of composite output services to be achieved more effectively from

[1] www.sewio.net/product/indoor-tracking-rtls-uwb-wi-fi-kit

[2] esense.io

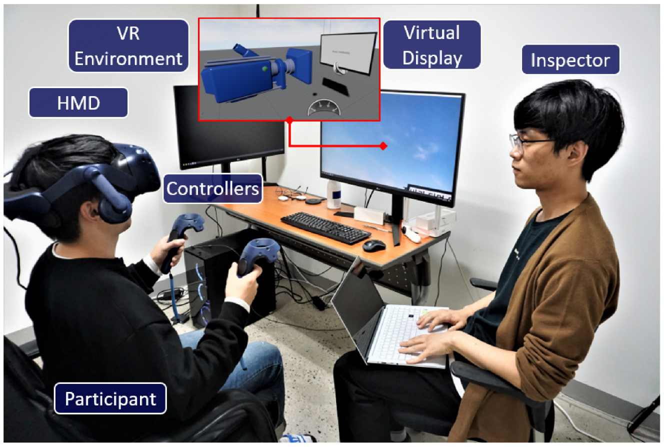


Fig. 12. Experiment in the VR environment.

the user's perspective. This approach would hopefully lead to synergistic effects among the services and avoid destructive interference.

We defined the visual-service effectiveness metric at a specific snapshot without considering the time axis and simply added the measured values to represent the overall effectiveness of service provisioning. However, a simple sum may not sufficiently reflect real users' perceived effectiveness. To improve the metric, we plan to extend the definition of service effectiveness to the time axis to obtain a more practical and improved representation of service effectiveness.

E. VR Experiment for Reproducing the User Study Results

To reduce the high cost of evaluating the metric practice, we are investigating the use of VR technologies known to be effective in obtaining realistic and interactive replications of physical experiments [8], [38], [39]. Recent experiments on visual acuity testing [38], simulation of escape signs for evacuation [8], and visual field examination [39] were successfully reproduced in VR with better efficiency than experiments in physical environments. Therefore, we conducted a similar user study in a VR environment to determine whether we could obtain results consistent with those observed in the laboratory environment.

We conducted the VR-based study with the same participants as the user study in the previous section. Each participant was asked to wear a Vive[3] Pro head-mounted display (HMD), which is considered high-quality VR equipment. We utilized the Unreal Engine[4] to implement 3-D VR environments, which is one of the most popular game engines for developing VR games. As shown in Fig. 12, we placed a virtual display device and spawned an avatar in the same locations and orientations as the laboratory studies. For each location and orientation, we presented a text on the virtual display device, and the participants were asked to report the perceived effectiveness in terms of readability by using controllers. The trigger of the right controller was for reporting the text was readable, and the trigger of the left controller was for reporting the text was not readable.

TABLE III
Contingency Table on the Reported Effectiveness

		Laboratory		Sum
		Effective	Ineffective	
VR	Effective	1779	339	2118
	Ineffective	290	2560	2850
Sum		2069	2899	4968

Additionally, the participants completed a questionnaire on the immersive tendency and the sense of the presence of the VR environment. We adopted the questionnaire of Witmer and Singer [40] to measure these. We expected that the immersive tendency and the sense of the presence of a participant may affect the consistency of the results between laboratory and VR experiments. For instance, a person with a low immersive tendency may show low consistency and should be avoided as a participant in VR experiments.

Table III shows the contingency table on the reported effectiveness in laboratory and VR experiments. Among the 4968 collected cases, 1779 were true positive, 339 were false positive, 290 were false negative, and 2560 were true negative. The measured accuracy was 0.873, precision was 0.840, and recall was 0.860, which means that the results of experiments in the laboratory and VR were highly consistent. As a result, we concluded that VR could accurately imitate the physical environment and reproduce the results of our experiments on visual service effectiveness. We may utilize the VR environment in our subsequent studies to perform experiments efficiently under more challenging and practical environmental conditions, such as emergency evacuation scenarios, which are difficult to develop.

Fig. 13 shows the consistency between laboratory and VR experiments for each participant. The consistency was high in terms of accuracy, precision, and recall. However, some of the recall and precision values of participants were lower than 0.7. Such low consistencies were due to the problems of the HMD used in the experiments. A participant claimed that the Field of View (FoV) was restricted by the HMD narrower than the FoV of real human vision. Another participant claimed that the HMD was too heavy to rotate the head freely, and left-right weights were not balanced. Such characteristics of HMD result in the reduced visual field of participants, which may negatively affect consistency. Because improving VR equipment, including HMD, is beyond the scope of this work, we expect that this hardware limitation may be resolved in the future.

Fig. 14 shows the correlation between consistency and immersive tendency/sense of presence measured by using the questionnaire [40]. Note that we calculated the total immersive tendency and presence score by summing scores from each question on a 7-point Likert scale. The results show that consistency measured for each participant was not strongly correlated with immersive tendency or presence and exhibit a low r-square score. This means that the VR environment can be applied to various participants regardless of personal

[3] www.vive.com
[4] www.unrealengine.com

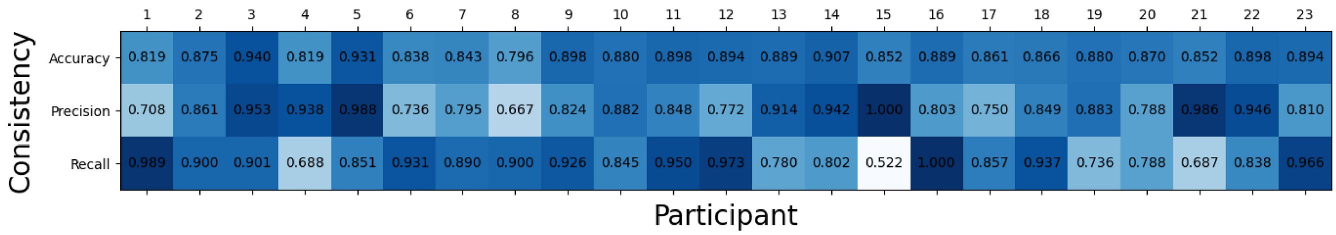


Fig. 13. Consistency between laboratory and VR experiments for each participant.

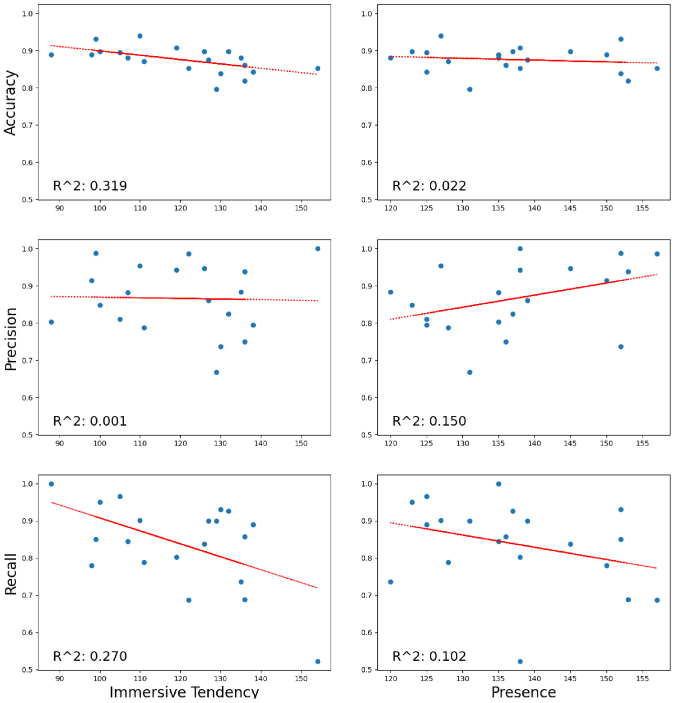


Fig. 14. Correlation of consistency and immersive-tendency/presence.

immersive tendency and presence because it shows high consistency with previous results even with participants who have low immersive tendency and presence.

IV. Dynamic and Effect-Driven Selection of Output-Services Using Deep Reinforcement Learning

In this section, we define the dynamic and effect-driven output-service selection problem based on the visual-service effectiveness metric defined in the previous section. Subsequently, we propose DEOSA, which makes decisions regarding the selection of output services. We used a reinforcement-learning algorithm to train DEOSA to learn a service selection policy from simulated experiences.

A. Dynamic Effect-Driven Output-Service Selection Problem

We formulate the dynamic and effect-driven output-service selection problem as a partially observable Markov decision process (POMDP) $\{\mathbb{S}, O, \mathbb{A}, Tr, R, \gamma\}$ consisting of the state $\mathbb{S}$, observation function O, action $\mathbb{A}$, transition probability function Tr, reward function R, and discount factor γ, which is the standard form of the reinforcement learning problem. An agent may traverse POMDP by performing actions and transitioning from one state to another to collect rewards.

The state of the problem is defined as the state of the environment at a specific time t, $\text{env}(t)$, consisting of the user $u(t)$, visual IoT service $i(t)$, and output services $\mathbb{D}$. The observation of the state may be limited to specific candidates and factors because of environmental constraints. For example, factors such as user orientation may not be measured practically using available infrastructures. In our problem definition, we assume that the user's orientation is hidden. Additionally, the information based on which the user currently selects the output service is provided to satisfy the Markov property. Otherwise, the action of the agent to solve the problem may be affected by past states visited by the agent, for which the reinforcement learning technique requires additional variations.

The action of the problem is the selection of output services among the discovered candidates based on the observed information. After a selection is made, the environment transitions from the current state $\text{env}(t)$ to the next state $\text{env}(t+1)$ by replacing the current output service $d_s(t)$ with the newly selected output service, $d_s(t+1)$, and moving the user's location according to the mobility $m_u(t)$. Note that the transition on the environment functions discretely and periodically; thus, the selection is performed in increments of 1 s. If the selected output service at $d_s(t)$ is different from the previous one $d_s(t-1)$, it indicates that a replacement occurred at time t.

The problem involves two optimization goals: 1) maximizing the visual-service effectiveness metric $e(u, i, t)$ and 2) minimizing the number of replacing output services over the runtime. First, the effectiveness of the visual service should be high during service provision because the physical effects should be delivered to the user with high effectiveness. Second, the number of replacements should be minimized because the replacement process produces additional overhead and inconvenience. Based on the optimization goals, we designed a reward function by combining the visual-service effectiveness metric and the overhead of replacements, as follows:

$$R(t) = \begin{cases} e(u, i, t) & d_s(t-1) = d_s(t) \\ e(u, i, t) - 1/2 & d_s(t-1) \neq d_s(t) \end{cases} \tag{8}$$

which means subtracting an overhead value of 1/2 from the visual-service effectiveness metric if a replacement occurs. We can collect and consider users' preferences by calculating the weighted sum of optimization goals, but we set the weight of replacement overhead manually as half of the maximum effectiveness value to avoid an excessive penalty for

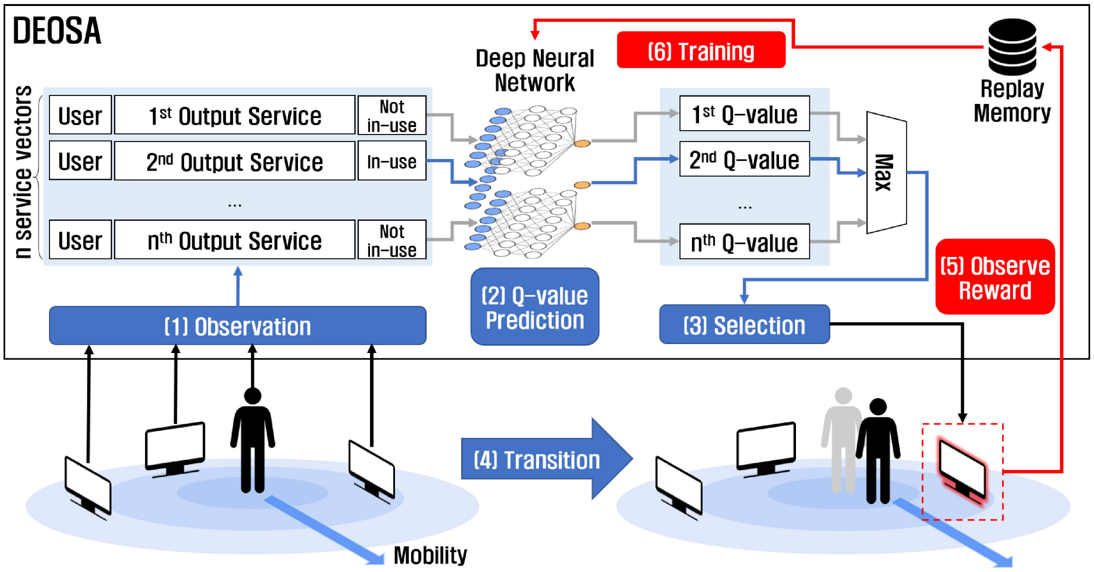


Fig. 15. Internal structure of DEOSA.

the replacement. When the agent performs a service selection, it receives a reward $R(t)$ according to the effectiveness of the selected service and a replacement penalty. The goal of the agent is to learn a policy that maximizes the cumulative rewards gained over service provision by selecting the most promising output service among the candidates in the long term. The discount factor (γ) is a value in the range [0, 1) that represents how the POMDP considers the future state for the calculation of the cumulative reward.

B. Output-Service Selection Using Reinforcement Learning

1) Output-Service Selection Agent: DEOSA observes the state of the POMDP and performs selections on output services for each step of the time series, as shown in Fig. 15. DEOSA can be associated with any type of service consumers, such as user agents and software agents, to delegate the service selection process. While designing DEOSA, we referred to DQN [23], which is one of the most popular value-based deep reinforcement learning algorithms, with a recent extension called Munchausen reinforcement learning [24] that improves the stability of the training process. The DQN-based deep neural network is a core part of DEOSA for predicting the Q-value of each service selection according to the observed factors of the user and environment, where a Q-value is the mathematical expectation of the cumulative reward when selecting a service. After selecting an output service based on the prediction, DEOSA receives a reward according to the POMDP and stores the result of the selection in the experience memory. The collected experiences are used to update the parameters of the neural network in the direction of decreasing the prediction error. While updating the parameters of the neural network, DEOSA minimizes the loss between the predicted Q-value and observed Q-value based on bootstrapped future values [23]. Furthermore, DEOSA considers the maximum entropy [41] and scaled log-policy [24] to improve the stability of the learning process. After training the neural network with sufficient experience to predict the Q-value accurately, selection can be performed simply by selecting the most promising candidate that has the maximum Q-value. The number of available actions varies according to the number of available candidates, in contrast to other common reinforcement learning problems. To deal with the variable number of available actions, we design the neural network to receive information about each candidate and predict Q-value individually [42].

2) Training Algorithm of DEOSA: As shown in Algorithm 1, we utilized the DQN-based algorithm improved with the Munchausen reinforcement learning technique [23], [24] to train DEOSA. DEOSA includes the probability of randomly selecting an output service that decreases during the training process, which is called the ε-greedy policy. Such an ε-greedy policy is for exploration to obtain various experiences, which reduces the risk of falling into the local-optimal policy. The training algorithm begins by initializing the memory to store experiences (line 1), deep neural networks for Q-value prediction (lines 2 and 3), and ε as the probability of random selection (line 4). The algorithm maintains two neural networks for the high stability of the learning [23]: the main neural network is for the prediction of Q-value during simulation, and the updated values of the main neural network are reflected in the target neural network gradually. The entire training process consists of many iterative simulations (lines 5–26), and a simulation consists of several selections (lines 8–24). Each simulation was conducted by resetting the environment to a new configuration, which enabled the agent to be generalized to various environmental conditions and avoid overfitting to a specific condition (line 6).

For each simulation, DEOSA performs selections iteratively and receives rewards as the results (line 8–24). At the beginning of a simulation, the values of the target neural network are copied to the main neural network (line 7). A selection is performed based on the observed information of the user

Algorithm 1 Training Algorithm of DEOSA

```
1:  memory.initialize()
2:  main_network.initialize()
3:  target_network.initialize()
4:  $\varepsilon = 1.0$
5:  for simulation = 1, $N_{simulation}$ do
6:      env.reset()
7:      main_network.copy_from(target_network, 1)
8:      for selection = 1, $N_{selection}$ do
9:          $o$ = env.observe()
10:         if get_random_value() $< \varepsilon$ then
11:             $d$ = random_selection($o.\mathbb{D}$)
12:         else
13:             $Q$ = main_network.predict_Q($o.u$, $o.\mathbb{D}$)
14:             $d$ = argmax(Q)
15:         end if
16:         $o'$, reward = env.selection($d$)
17:         memory.add($o$, $d$, reward, $o'$)
18:
19:         if memory.is_full() then
20:             batch = memory.sample(batch_size)
21:             main_network.update(batch)
22:             target_network.copy_from(main_network, $\tau$)
23:         end if
24:     end for
25:     $\varepsilon = \varepsilon \times \varepsilon_{decay}$
26: end for
```

and candidate output services (line 9). Following the ε-greedy policy, DEOSA randomly selects an output service (line 11) if the sampled random value within the interval [0, 1] is less than ε (line 10). Otherwise, DEOSA selects the output service with the highest Q-value (line 14) based on the prediction results of the Q-values (line 13). Each selection triggers the transition of the environment to the next state with a reward value. Each selection occurs during the transition of the environment to the next state, and a reward value is provided to the DEOSA (line 16).

The results of selections are stored in the memory of DEOSA as experiences for the further learning process (line 17). DEOSA collects experiences until the experience memory becomes full (line 19), and it updates the parameters of the deep neural network at the end of each selection process (line 21) by sampling a set of experiences from the experience memory (line 20). Updated values of the main neural network are partially copied to the target neural network by τ (line 22). The mathematical details of updating the parameters of the neural networks were adopted from Munchausen-DQN [24]. At the end of each simulation, ε is slightly decreased (line 25) to reduce the probability of random selection as the prediction of the Q-value becomes more accurate and, thus, ε becomes small in the final training phase.

C. Evaluation of DEOSA

1) Configuration of Experiment: We evaluated the performance of DEOSA in simulated IoT environments on a machine running the Windows 10 Enterprise operating system with an Intel i7-9700k processor, two nVidia GeForce RTX 2070 graphics processing units (GPUs), and 32 GB of random access memory (RAM). The implementation of the simulation is publicly accessible,[5] written in Python 3.6.5, along with GPU-accelerated TensorFlow 2.1.0,[6] which is the most popular machine learning framework. The details of the experimental settings are listed in Table IV, where some settings were set based on the results of the user study in Section III, and others were set empirically.

TABLE IV
DETAIL EXPERIMENT SETTINGS OF DEOSA EVALUATION

Type	Name	Value
Environment	Number of simulations	1000
	Length of each simulation	50 s
	Space	100 × 10 × 5
	Number of output services	100
	Maximum viewing angle (ψ_d)	70°
	Scaling factor (f_d)	1.0 - 5.0
User	User's height	1.7 m
	User's speed	2 m/s
	Maximum visual field (θ_u)	80°
	Visual acuity of user (a_u)	0.0
Training	Size of memory	1000
	Size of batch	10
	Discount factor (γ)	0.99
	Optimizer	Adam [43]
	Learning rate	1e-3
	Epsilon decay rate (ϵ_{decay})	0.99
	Minimum epsilon (ϵ_{min})	0.1
	Target network update rate (τ)	0.001
	Entropy temperature [24]	0.1
	Alpha [24]	0.9
Neural Network	Number of hidden layers	5
	Size of each hidden layer	128
	Activation function	ReLU

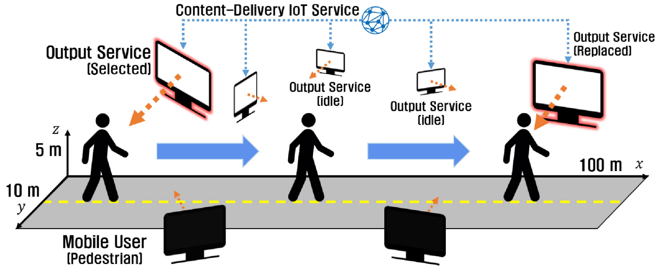


Fig. 16. Target scenario of simulation.

For simulations, we followed the motivation scenario discussed in Section I by further simplifying it for DEOSA, that is, a pedestrian user walks along a street and receives content from visual output services, as depicted in Fig. 16. The width,

[5] https://github.com/Blest215/DEOSA

[6] https://www.tensorflow.org/

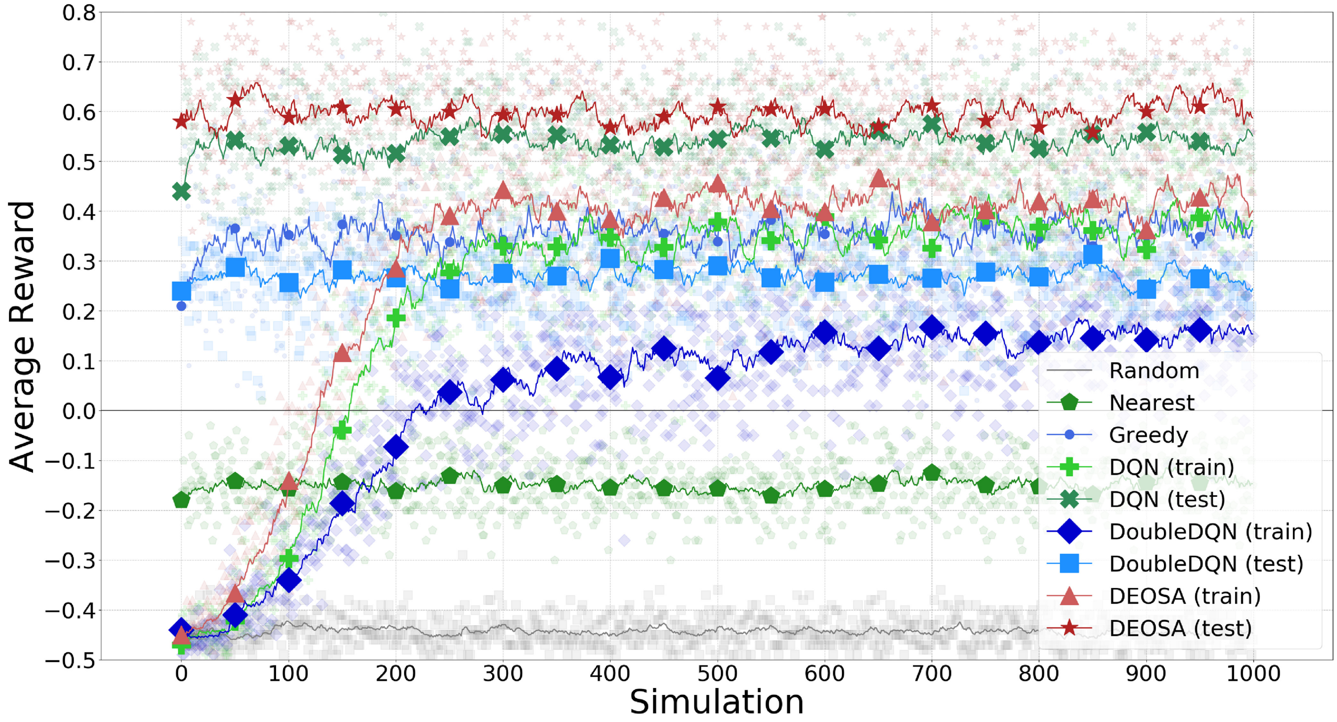


Fig. 17. Comparison of average reward gained during each simulation.

length, and depth of simulated IoT environments were 100, 10, and 5 m, respectively (100 × 10 × 5), to mimic a straight street. A total of 100 output services are distributed over the environment with its associated display device, where the location, orientation, and scaling factor are randomly assigned and do not change throughout the simulation. The maximum viewing angle ψ_d of the output services is set as 70°, as found in the user study in Section III-C. The unit of time is second, so the state of an environment, such as the location of the user, is updated every second. We utilized a neural network consisting of two hidden layers for the prediction of the Q-value, and each hidden layer has 128 units with the rectified linear unit (ReLU) function. The training process of DEOSA used the Adam optimizer [43], which is one of the most popular optimizers for neural networks. We set the learning rate of the neural network 1e-3, the discount factor 0.99, the size of the experience memory 1000, and the size of batch 10. Munchausen reinforcement learning uses two more parameters to stabilize the training process: entropy temperature and alpha [24], which we set as 0.1 and 0.9, respectively.

When a simulation begins, the user starts walking from the left-most side of the environment and moves toward the right direction along the x-axis. The speed of the user was 2 m/s, which is slightly faster than the typical walking speed of a pedestrian in an urban environment. The user's maximum viewing angle θ_u is set as 80°, as found by the user study in Section III-C, and visual acuity is set as a standard value of 0.0. In the simulations, DEOSA can observe the user's parameters, except the orientation. Each simulation was performed for 50 s with 50 selections on the output service. For each second, the user's orientation is randomly changed by rotating the user's mobility direction horizontally, where the rotation angle was sampled from a normal distribution by a set mean of 0 and a standard deviation of 0.75, to restrict the user's orientation to within 80° from the user's mobility direction.

We compared the performance of DEOSA to that of the random selection, nearest selection, greedy selection, DQN-based, and double-DQN-based state-of-the-art algorithms, as baselines. The first baseline, the random selection algorithm, randomly selects an output service as the simplest solution. The nearest selection algorithm selects the output service that is located nearest to the user. The greedy-selection algorithm partially calculates the visual-service effectiveness metric of each output service except the visual field condition that requires the user's orientation and selects from among the output services that are expected to be highly effective based on the partial calculation of the metric. Note that the user's orientation is hidden from DEOSA to evaluate DEOSA under a more practical configuration; thus, it is not possible to calculate the metric completely. The DQN-based algorithm is the previous version of DEOSA [3] that calculates the Q-value of the candidate services based on vanilla DQN [23]. We consider the double-DQN-based algorithm as the state of the art, which is known to be the best solution to the problem of selecting the most appropriate service provider among functionally equivalent services [32]. The double-DQN-based algorithm calculates the Q-value of the candidate services based on double-DQN [33].

2) Performance Results of DEOSA: Fig. 17 shows the average reward gained by each agent for each of the simulations. DEOSA was tested after the training, so, the results from the testing phase show the actual performance of DEOSA. For better visualization, we apply an exponential moving average filter to reduce the fluctuation resulting from the high

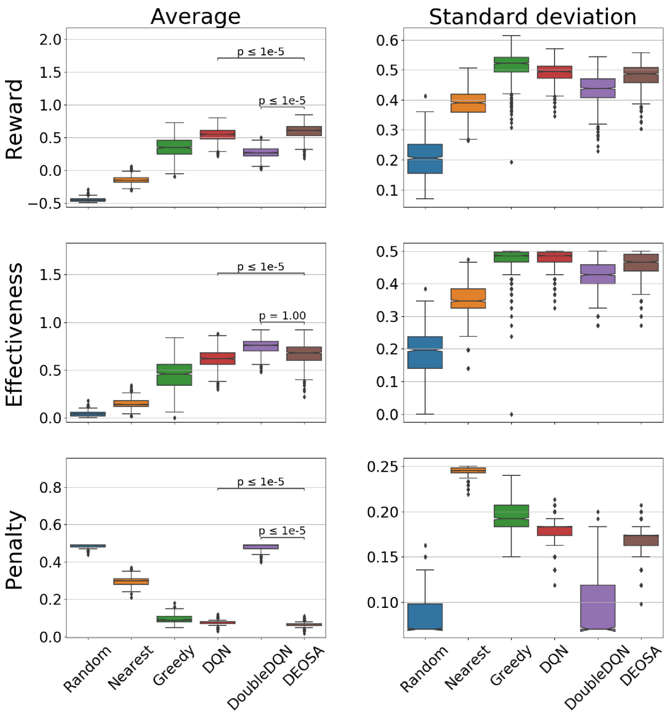


Fig. 18. Statistics of service effectiveness and replacement overhead.

randomness of the simulations. Specifically, the curve of the average reward for each simulation fluctuates because the output services are relocated to a random location at the beginning of a new simulation.

The results show that the random-selection algorithm suffered from the high replacement overhead, resulting in negative average rewards. The nearest selection algorithm shows better average rewards than the random selection but is still lower than zero because of the high replacement overhead. The greedy-selection algorithm shows higher rewards than other baselines by estimating the visual-service effectiveness metric, but it is imperfect because it lacks the user's orientation.

In early simulations of its training, DEOSA performed more poorly than the baselines in terms of average rewards collected from each simulation. However, the performance of DEOSA increased by training its service selection policy over simulations and achieving high rewards, even though some factors were hidden from DEOSA. During the testing phase, DEOSA achieved higher performance than the baselines. Note that DEOSA was tested in randomly generated new environments different from the environments of the training phase, which means that the service selection policy DEOSA learned is generally applicable to new environments without an overfitting problem.

The results also show that DEOSA learns faster than both the DQN-based algorithm and the double-DQN-based algorithm, which is the state of the art and achieves the best performance earlier in the training phase. Finally, DEOSA achieves a higher performance than the double-DQN-based state-of-the-art in the testing phase. Therefore, we can conclude that the adoption of Munchausen reinforcement learning [24] effectively improves DEOSA in terms of the stability of the learning process and the performance in terms of collected reward.

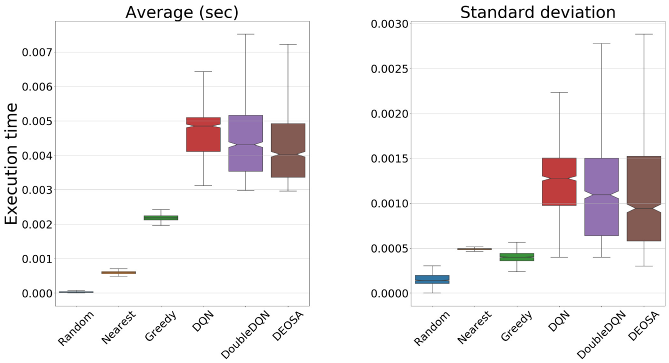


Fig. 19. Comparison of execution time.

Fig. 18 shows the statistical results in the form of the average and standard deviation values of the: 1) reward; 2) visual-service effectiveness; and 3) replacement overhead of DEOSA and the baselines. The average reward of DEOSA is higher than that of the baselines with statistical significance. We tested the statistical significance by conducting one-sided Mann–Whitney U tests because we could not assume a normal distribution. The low effectiveness of the nearest selection algorithm implies that a simple distance-based selection strategy is insufficient for evaluating the quality of the actual delivery of physical effects. Visual-service effectiveness of the state-of-the-art algorithm is higher than that of DEOSA but suffered from high replacement penalties, which implies that the state-of-the-art algorithm failed to learn the influence of the replacement penalty.

Fig. 19 shows the statistics of the execution time taken to conduct a selection for DEOSA and the baselines. The results show that DEOSA requires a longer execution time than the baselines because the calculation of the deep neural network is computationally more intensive than the baselines. However, the average execution time of DEOSA is slightly shorter than that of the state-of-the-art algorithm, near 0.005 s. This result shows that even though the training of the agent may be time-consuming because of the learning process, service selection using neural networks is sufficiently practical with a short execution time. We expect that the execution time of DEOSA can be effectively reduced by pruning the redundant parameters of the trained neural network.

3) Threats to Validity: We evaluated DEOSA using a pedestrian scenario in randomly simulated environments because reinforcement learning in a real-world setting is a critical challenge [44], especially when the action of the agent affects the environment dynamically. In addition, we considered that users have limited mobility, with a fixed movement direction and speed. This is because the current design of DEOSA focuses on reflecting the physical settings in IoT environments rather than considering the physical movements of users. The service-related factors, such as the location, orientation, and scaling constant of output services, were randomized to simulate various service conditions. Therefore, simulations under thousands of random environmental conditions were sufficient to evaluate DEOSA in terms of optimization and generalization.

We implemented the state-of-the-art algorithm [32] that we explained in Section II. We used Tensorflow, one of the most reliable machine learning libraries, to implement the core double-DQN part of the state-of-the-art algorithm. We compared the state-of-the-art and our algorithms under the same configurations of simulations. We ran enough simulations to compare the performance of the algorithms under various environmental conditions and to show the difference between DEOSA and the state-of-the-algorithm is statistically significant.

D. Discussions on DEOSA

The current version of DEOSA was designed for a single user and ignores other users who may compete to obtain visual output services that can serve only one user at a time. However, complex interactions of users may affect the decisions, such as negotiation that decides who would utilize the output service or cooperation that shares output services. Therefore, we intend to extend the design of DEOSA by adopting multiagent reinforcement learning techniques to deal with nonstationarity caused by other agents and uncertainties of the environment.

Moreover, while extending the design of DEOSA to a multiagent setting, resolving conflicts among agents is an important challenge. Distributed conflicts between the user–user, user–provider, or provider–provider should be resolved locally by performing negotiations [45], [46]. However, existing negotiation mechanisms are mostly about adjusting prices; for instance, a user may ask for a lower price, or service providers may compete to get a user. As we discussed for the service effectiveness metric, detecting and resolving conflicts on the level of destructive interference among physical effects from users, such as noise or glare is one of our future research directions.

The design of DEOSA is model-independent and metric-agnostic; therefore, DEOSA can be applied to any other effectiveness metric or environment via retraining. To make DEOSA adaptable for various types of services and environmental conditions, we plan to extend it by adopting meta-learning techniques [47] that reuse previous knowledge to accelerate the retraining process. Furthermore, in our formulation of the dynamic and effect-driven output-service selection problem, a selection is performed for each time step in a discrete manner. Such a formulation may lead to a redundant selection process; therefore, we may formulate the problem in a continuous manner such that it also decides the timing at which the selection is to be performed.

V. Conclusion

The dynamic selection and replacement of services in an effect-driven manner is essential for providing the content of IoT services that deliver physical effects to users as their outputs. However, existing studies on dynamic service selection only consider network QoS and do not evaluate the delivery of physical effects generated by output services. Furthermore, most service selection algorithms do not support the predictive replacement of services, which is essential for physical output services in IoT environments.

In this study, we have improved the visual-service effectiveness metric presented in our previous work [3] to measure how effectively a visual output service delivers visual effects to the user. We referred to the human visual system to design the metric. Based on the metric, we have defined a dynamic and effect-driven output-service selection problem, which requires the predictive selection of output services considering the future states of the environment. As a solution, we have developed a reinforcement learning agent named DEOSA that selects and replaces output services dynamically to maximize the visual-service effectiveness metric and minimize the overhead of replacements.

We evaluated the practicality of the visual-service effectiveness metric by performing a user study in a laboratory environment. The results showed that the metric successfully estimates users' perceived effectiveness of visual output services with high accuracy, precision, and recall in terms of delivering physical effects. We investigated the use of VR techniques to imitate the IoT environments to improve the efficiency of experiments, and the results showed that the experiments in VR environments could reproduce the results of the experiments in laboratory environments consistently. We also evaluated DEOSA by simulating service selection processes in randomly generated simulations of IoT environments. The simulation results showed that DEOSA achieved higher visual-service effectiveness and lower overhead of service replacements than other baseline agents.

In the future, we aim to extend the service effectiveness metric to consider other types of physical effects, such as acoustic effects and physical interference. Second, we intend to extend the visual-service effectiveness metric to consider the temporal aspect of IoT service provision. In other words, we plan to design a metric that measures the overall effectiveness of delivering service effects for a specific period. Third, we aim to enhance the service selection algorithm to support multiple users using multiagent reinforcement learning techniques.

References

[1] I. Yen, F. Bastani, S.-Y. Hwang, W. Zhu, and G. Zhou, "From software services to IoT services: The modeling perspective," in *Proc. Int. Conf. Serviceol.*, 2017, pp. 215–223.

[2] I.-Y. Ko, H.-G. Ko, A. J. Molina, and J.-H. Kwon, "SoIoT: Toward a user-centric IoT-based service framework," *ACM Trans. Internet Technol.*, vol. 16, no. 2, pp. 1–21, 2016.

[3] K. Baek and I.-Y. Ko, "Effect-driven dynamic selection of physical media for visual IoT services using reinforcement learning," in *Proc. IEEE Int. Conf. Web Services (ICWS)*, 2019, pp. 41–49.

[4] W. Kim, S. Lee, Y. Chang, T. Lee, I. Hwang, and J. Song, "Hivemind: Social control-and-use of IoT towards democratization of public spaces," in *Proc. 19th Annu. Int. Conf. Mobile Syst., Appl., Services*, 2021, pp. 467–482.

[5] J. Coenen, N. Wouters, and A. V. Moere, "Synchronized wayfinding on multiple consecutively situated public displays," in *Proc. 5th ACM Int. Symp. Pervasive Displays*, 2016, pp. 182–196.

[6] P. Knierim, S. Maurer, K. Wolf, and M. Funk, "Quadcopter-projected in-situ navigation cues for improved location awareness," in *Proc. CHI Conf. Human Factors Comput. Syst.*, 2018, pp. 1–6.

[7] M. Endrei *et al.*, *Patterns: Service-Oriented Architecture and Web Services* (International Technical Support Organization). New York, NY, USA: IBM Corp., 2004.

[8] K. Krösl, D. Bauer, M. Schwärzler, H. Fuchs, G. Suter, and M. Wimmer, "A VR-based user study on the effects of vision impairments on recognition distances of escape-route signs in buildings," *Vis. Comput.*, vol. 34, no. 6, pp. 911–923, 2018.

[9] H. Xie, L. Filippidis, S. Gwynne, E. R. Galea, D. Blackshields, and P. J. Lawrence, "Signage legibility distances as a function of observation angle," *J. Fire Protect. Eng.*, vol. 17, no. 1, pp. 41–64, 2007.

[10] M. Alrifai, D. Skoutas, and T. Risse, "Selecting skyline services for QoS-based Web service composition," in *Proc. 19th Int. Conf. World Wide Web*, 2010, pp. 11–20.

[11] M. Moghaddam and J. G. Davis, "Service selection in Web service composition: A comparative review of existing approaches," in *Web Services Foundations*. New York, NY, USA: Springer, pp. 321–346, 2014.

[12] T. H. Tan et al., "Optimizing selection of competing services with probabilistic hierarchical refinement," in *Proc. IEEE/ACM 38th Int. Conf. Softw. Eng. (ICSE)*, 2016, pp. 85–95.

[13] S. Deng, H. Wu, D. Hu, and J. L. Zhao, "Service selection for composition with QoS correlations," *IEEE Trans. Services Comput.*, vol. 9, no. 2, pp. 291–303, Mar./Apr. 2016.

[14] P. Brooks and B. Hestnes, "User measures of quality of experience: Why being objective and quantitative is important," *IEEE Netw.*, vol. 24, no. 2, pp. 8–13, Mar./Apr. 2010.

[15] L. Purohit and S. Kumar, "A study on evolutionary computing based Web service selection techniques," *Artif. Intell. Rev.*, vol. 54, no. 2, pp. 1117–1170, 2021.

[16] S.-Y. Hwang, E.-P. Lim, C.-H. Lee, and C.-H. Chen, "Dynamic Web service selection for reliable Web service composition," *IEEE Trans. Services Comput.*, vol. 1, no. 2, pp. 104–116, Apr.–Jun. 2008.

[17] Y. Liu, A. H. Ngu, and L. Z. Zeng, "QoS computation and policing in dynamic Web service selection," in *Proc. 13th Int. World Wide Web Conf. Alternate Track Papers Posters*, 2004, pp. 66–73.

[18] D. Lin, C. Shi, and T. Ishida, "Dynamic service selection based on context-aware QoS," in *Proc. IEEE Ninth Int. Conf. Services Comput.*, 2012, pp. 641–648.

[19] A. Altaf, H. Abbas, F. Iqbal, M. M. Z. M. Khan, and M. Daneshmand, "Robust, secure, and adaptive trust-oriented service selection in IoT-based smart buildings," *IEEE Internet Things J.*, vol. 8, no. 9, pp. 7497–7509, May 2021.

[20] H. Alayed, F. Dahan, T. Alfakih, H. Mathkour, and M. Arafah, "Enhancement of ant colony optimization for QoS-aware Web service selection," *IEEE Access*, vol. 7, pp. 97041–97051, 2019.

[21] M. Chandra and R. Niyogi, "Web service selection using modified artificial bee colony algorithm," *IEEE Access*, vol. 7, pp. 88673–88684, 2019.

[22] F. Dahan, H. Mathkour, and M. Arafah, "Two-step artificial bee colony algorithm enhancement for QoS-aware Web service selection problem," *IEEE Access*, vol. 7, pp. 21787–21794, 2019.

[23] V. Mnih et al., "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, 2015.

[24] N. Vieillard, O. Pietquin, and M. Geist, "Munchausen reinforcement learning," in *Advances in Neural Information Processing Systems*, vol. 33. Red Hook, NY, USA: Curran Assoc., 2020, pp. 4235–4246.

[25] H. Wang, Q. Wu, X. Chen, Q. Yu, Z. Zheng, and A. Bouguettaya, "Adaptive and dynamic service composition via multi-agent reinforcement learning," in *Proc. IEEE Int. Conf. Web Services*, 2014, pp. 447–454.

[26] H. Wang et al., "Integrating reinforcement learning and skyline computing for adaptive service composition," *Inf. Sci.*, vol. 519, pp. 141–160, May 2020.

[27] L. Ren, W. Wang, and H. Xu, "A reinforcement learning method for constraint-satisfied services composition," *IEEE Trans. Services Comput.*, vol. 13, no. 5, pp. 786–800, Sep./Oct. 2020.

[28] H. Liang, X. Wen, Y. Liu, H. Zhang, L. Zhang, and L. Wang, "Logistics-involved QoS-aware service composition in cloud manufacturing with deep reinforcement learning," *Robot. Comput.-Integr. Manuf.*, vol. 67, Feb. 2021, Art. no. 101991.

[29] N. Swetha and G. Karpagam, "Reinforcement learning infused intelligent framework for semantic Web service composition," *Appl. Intell.*, vol. 52, no. 2, pp. 1979–2000, 2022.

[30] K.-D. Baek and I.-Y. Ko, "Spatially cohesive service discovery and dynamic service handover for distributed IoT environments," in *Proc. Int. Conf. Web Eng.*, 2017, pp. 60–78.

[31] K. Baek and I.-Y. Ko, "Spatio-cohesive service selection using machine learning in dynamic IoT environments," in *Proc. Int. Conf. Web Eng.*, 2018, pp. 366–374.

[32] G. Sun, K. Liu, G. O. Boateng, G. Liu, and W. Jiang, "Intelligent cruise guidance and vehicle resource management with deep reinforcement learning," *IEEE Internet Things J.*, vol. 9, no. 5, pp. 3574–3585, Mar. 2022.

[33] H. Van Hasselt, A. Guez, and D. Silver, "Deep reinforcement learning with double Q-learning," in *Proc. AAAI Conf. Artif. Intell.*, vol. 30, 2016, pp. 2094–2100.

[34] H. Strasburger, I. Rentschler, and M. Jüttner, "Peripheral vision and pattern recognition: A review," *J. Vis.*, vol. 11, no. 5, p. 13, 2011.

[35] I. L. Bailey and J. E. Lovie, "New design principles for visual acuity letter charts," *Amer. J. Optometry Physiol. Opt.*, vol. 53, no. 11, pp. 740–745, 1976.

[36] H. Snellen, *Probebuchstaben zur Bestimmung der Sehschärfe*, vol. 1. Berlin, Germany: Hermann Peters, 1873.

[37] A. E. S. Tafreshi, K. Marbach, and M. C. Norrie, "Proximity-based adaptation of Web content on public displays," in *Proc. Int. Conf. Web Eng.*, 2017, pp. 282–301.

[38] S. Shekar, P. R. Pesaladinne, S. A. Karre, and Y. R. Reddy, "VREye: Exploring human visual acuity test using virtual reality," in *Proc. Int. Conf. Human-Comput. Interact.*, 2020, pp. 415–429.

[39] S. Tsapakis et al., "Visual field examination method using virtual reality glasses compared with the Humphrey perimeter," *Clin. Ophthalmol.*, vol. 11, p. 1431, Aug. 2017.

[40] B. G. Witmer and M. J. Singer, "Measuring presence in virtual environments: A presence questionnaire," *Presence*, vol. 7, no. 3, pp. 225–240, 1998.

[41] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, "Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor," in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 1861–1870.

[42] J. He et al., "Deep reinforcement learning with an action space defined by natural language," in *Proc. Workshop Tracks Int. Conf. Learn. Represent. (ICLR)*, 2016, pp. 1621–1630.

[43] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2015, *arXiv:1412.6980*.

[44] G. Dulac-Arnold et al., "Challenges of real-world reinforcement learning: Definitions, benchmarks and analysis," *Mach. Learn.*, vol. 110, no. 9, pp. 2419–2468, 2021.

[45] B. Shojaiemehr, A. M. Rahmani, and N. N. Qader, "Cloud computing service negotiation: A systematic review," *Comput. Stand. Interfaces*, vol. 55, pp. 196–206, Jan. 2018.

[46] S. Son and K. M. Sim, "A price-and-time-slot-negotiation mechanism for cloud service reservations," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 42, no. 3, pp. 713–728, Jun. 2012.

[47] T. M. Hospedales, A. Antoniou, P. Micaelli, and A. J. Storkey, "Meta-learning in neural networks: A survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 9, pp. 5149–5169, Sep. 2022.

KyeongDeok Baek received the B.E. degree in computer science from the School of Computing, Korea Advanced Institute of Science and Technology, Daejeon, South Korea, in 2016, where he is currently pursuing the integrated program for M.E. and Ph.D. degrees with the Web Engineering Lab, School of Computing.

His research interests include human-centric service-oriented computing, Internet of Things services, service selection, and reinforcement learning.

In-Young Ko (Member, IEEE) received the Ph.D. degree in computer science from the University of Southern California, Los Angeles, CA, USA, in 2003.

He is a Professor with the School of Computing, Korea Advanced Institute of Science and Technology, Daejeon, South Korea. His research interests include service computing, Web engineering, and software engineering. His recent research focuses on service-oriented software development in large-scale and distributed system environments, such as the Web, Internet of Things, and edge cloud environments.