Supporting Daily Living Activities using Behavior Logs and Augmented Reality

Yuki Nakamura[1], Shinya Yamamoto[2], Morihiko Tamai[1], Keiichi Yasumoto[1]
[1]Graduate School of Information Science, Nara Institute of Science and Technology, Ikoma, Nara 630-0192, Japan
[2]Tokyo University of Science Yamaguchi, Yamaguchi 756-0884, Japan
Email: [1](yuki-na, morihi-t, yasumoto)@is.naist.jp, [2]shiny-ya@ed.yama.tus.ac.jp

Abstract—**In this paper, we propose a support system for daily living activities based on a user's behavior log in an indoor environment. Generally speaking, daily living activities hardly remain in our memory, so that, determining the order and priority of daily tasks can be difficult. There are studies supporting daily chores based on a behavior log and aiming to improve the work efficiency by using a memory aid. However, most of the existing studies require pre-installation of costly devices such as cameras and sensors, and are limited to supporting a particular task or purpose. There have been proposed many methods based on Augmented Reality (AR) which annotate captured images of the real world with texts, but text-based information may frustrate users. More intuitive user interfaces are required. In this paper, we first propose a user's behavior log management technique which associates each activity log with an object/location in the target indoor space. Then, we propose a method for reproducing AR-based visual effects on the object/location determined based on the log; the method naturally navigates the user to carrying out the intended task. We have implemented a prototype of the proposed method and have conducted subjective surveys to evaluate the effectiveness of the proposed visual effects. We thereby, confirmed that the proposed visual effects can make users more intuitively aware of the intended task than the text-based annotation method.**

I. Introduction

Daily living activities, such as cleaning up a room, hardly remain in our memory, so that, determining the order and priority of daily chores can be difficult. It may be useful to show users ways of imitating professionals or point out tips found on web pages. However, the way of handling daily chores varies from person to person depending on the ways and customs of each household. So it is difficult to recommend a best way of carrying out a task. There are studies for supporting daily activities and improving work efficiency using a memory aid [1]–[3]. Those studies, however, adopt a text-based presentation of information. Text information may cause mental stress and frustration, since it takes time to read and understand text. Small texts may impede intuitive and immediate understanding. It is known that daily life behavior is deeply affected by stress [4]. We need intuitive system outputs that alleviate users frustration. Furthermore, most of the existing studies require pre-installation of costly devices such as cameras and RF-ID dedicated to supporting a particular task or purpose. Therefore, they are costly and not flexible.

In this paper, we propose a flexible daily activity support method. We target tasks such as cleaning, machine maintenance, pet care and replacement of consumables. Our proposed system provides intuitive visual effects using Augmented Reality (AR) on the display of a mobile device equipped with a camera. Our proposed system first calculates the priority of each daily task, which, using the behavior log, is associated with a location/object. Next, if the priority of the location/object is high, the system applies *insistent effect* such as spotlight which highlights the object/location on the display (Hereafter, we refer to such effects as *Spatial Decoration Effects* or *SDEffects*). If the priority is low, the system applies a *restrain effect*, such as fog, which makes the object/location inconspicuous on the display. Our proposed system aims to reduce stress and frustration by utilizing intuitive visual effects on the target object/location so as to guide the user's gaze naturally to the location/object of high priority. To this end, it is necessary to provide an adequate SDEffect for each pairing of location/object and priority. We propose a behavior log management method utilizing a *container* which is a virtual cube tagged to the location/object. The system calculates the priority for each container, and applies to the container an adequate SDEffect. Realizing the proposed system requires accurate identification of containers on the captured image. There are many existing methods to solve this problem, and among them, we adopt the AR-marker based method since it is cheap and easy to install.

We implemented a prototype of our proposed system and conducted experiments with the system in which users make decisions on priorities and the locations for cleaning our laboratory. As a result, we confirmed that our proposed method outperforms a conventional method using text-based AR annotation.

II. Related Work

In this section, we describe the related research on AR and support systems for daily living activities.

There have been many studies utilizing AR technology such as a navigation system [5], an interface [6], visualization of the radio field strength [7], and rehabilitation or medical treatment [8], [9]. Recently, some applications using AR technology operate on a mobile device platform [10]–[12], owing to miniaturization and advances in the performance of mobile devices such as smart phones. This fact accelerates the rapid expansion of applications of AR technology.

One of main challenges for realizing applications of AR is to estimate the position of user and objects, and there are many studies such as [13], [14]. The solution using AR-markers, especially, is cheap and installation is easy. To cope with the drawback that attaching AR-markers spoils the landscape, invisible AR-markers [15] and visual tags like CyberCode [16] are proposed. Another problem with AR applications is the accuracy and visibility of an AR tag's displayed position. When multiple AR tags overlap each other, it will be hard for users to read the tags. There is a study which explicitly considers the displayed position of AR tags in the depth direction [17]. Another study proposes a method to calculate the position to display AR tags so that they do not overlap on objects [18]. However, many existing studies adopt text-based annotation, while there are a few studies which discuss visual information presentation.

Many systems have been proposed to support daily life activities. For example, there is a support system for finding lost objects by spotlighting the location where the object is likely to be [1], and a system for finding items in storage boxes using captured images [2], [3]. However, these systems target a particular activity like finding lost objects. Also they require installation and operation of various devices.

To overcome drawbacks of existing studies, we propose an intuitive AR-based information presentation system which gives users intuitive hints with visual effects and requires only a mobile device such as a smart phone.

III. A Method for Supporting daily life activities

In this section, we describe the requirements and the basic idea for realizing a system for daily life behavior support, and then propose methods for realizing the support system.

A. Requirements and Basic Idea

The basic idea is to have each user (i) record their behavior in a container so that to-do tasks and their importance are identified from the log, and (ii) get an intuitive hint for the task.

For (i), we introduce a virtual cube, called a *container*, which is associated with a real object or a location such as a table or a window. A behavior log is recorded in the corresponding container so that the system calculates the urgency/priority of the tasks for the object/location. We describe details of the behavior log management method in section III-B.

For (ii), since the human's perception is the most affected by visual information [19], we provide users with adequate visual effects depending on the urgency/priority of the task calculated from the behavior log using AR technology. Details are described in section III-C.

B. Behavior log management method

We show symbols and terms used in this section in Table I. Every position in the target space is represented by a three dimensional orthogonal coordinate system. We assume that there are the set of *target objects* such as a window and a desk, denoted by O. We assume that each object o has its base position $o.pos$ and a name denoted by $o.name$. Our proposed log management method is based on the idea of dividing the target space by containers which are associated with existing objects. We denote the set of containers by C where each container is a virtual cube specified by the user. We denote by C_D the set of containers in the view of the mobile device's camera. We assume that each container c is associated with an object o and denote it by $c = \langle p, o, ttd, log, E\rangle$. Here, p is the priority for container. ttd is the time to deadline which is an attribute specified by the user depending on the task type such as a cleaning cycle or an expiration date. Log is the behavior log. log consists of multiple tuples where each tuple is denoted by $\langle date, what\rangle$. Here, $date$ is the time when the tuple is entered into the system and $what$ is the task type.

TABLE I
Definitions

	notation	definition
target space	S	
target object	O	$O = \{o_1, ..., o_n\}$, $o = \langle o.pos, o.name\rangle$
object name	$name$	
container	C	$C = \{c_1, ..., c_n\}$, $C_D = C \cap DisplayView$, $c = \langle p, o, ttd, log, E\rangle$
behavior log	Log	$Log = \{log_1, ..., log_t\}$, $log = \langle date, what\rangle$
SDEffects	E	$E = \{e_1, ..., e_n\}$, $e = \langle Effecttype, EV, CP\rangle$

For each container c, we denote by $c.E$ the set of *SDEffects* which can be applied to c. We suppose that $c.E$ is given for each container c. We also denote by $C.E$ the set of *SDEffects* which can be applied to the set of containers C. We denote each SDEffect by $\langle Effecttype, EV, CP\rangle$. Here, our proposed method utilizes *insistent effect* and *restrain effect* to give users intuitive hints for tasks to be carried out. Insistent effect decorates the space around the container to attract the user's interest, e.g., with a spotlight by AR technology. Conversely, restrain effect is the visual effect to avert the user's attention. Examples of these effects are shown in Fig. 3 (c) and (d). $Effecttype$ specifies the type of intended *SDEffects* and the spacial range of those effects represented by the maximum/minimum radii. When there are multiple *SDEffects* applicable to the container, the system needs to decide on the best one. EV (effectiveness value) represents the effectiveness degree of each SDEffect when applying it to the container. We derive EV of each SDEffect through experiments.

We also propose combinatorial use of two *SDEffects* to reproduce more intuitive visual effects. Here, we expect that there would be effective/ineffective pairs of *SDEffects*. Then, we denote by CP the compatible pairs of *SDEffects* that can be used together. In section V-C, we derive the effectiveness

Algorithm 1 SDEffect Decision Algorithm

```
1: while $H \neq \emptyset$ do
2:     Pick out $h$ with the highest container priority from $H$.
3:     if $I = \emptyset$ then
4:         add $h$ to $I$
5:     else if $I \neq \phi \wedge I.E \cap h.E \neq \emptyset$ then
6:         add $h$ to $I$
7:     end if
8: end while
9: decide $i.e$ by selecting the SDEffect with the highest EV
10: while $L \neq \emptyset$ do
11:     Pick out $l$ with the lowest container priority from $L$
12:     if $R = \emptyset$ then
13:         add $l$ to $R$
14:     else if $I.E.CP \cap l.E \neq \emptyset$ then
15:         add $l$ to $R$
16:     end if
17: end while
18: decide $r.e$ by selecting the SDEffect with the lowest EV
19: calculate relative distance between containers
20: if the distance between containers of $I$ and $R$ is close then
21:     remove elements with priority near 0.5 from $I$ and $R$
22: end if
```

value of each SDEffect (EV) and the effective SDEffects pairs (CP) through experiments.

C. Decision of SDEffects

We provide different *SDEffects* for different priority containers. We calculate the priority $c.p$ for container c by the following equation.

$$c.p = \begin{cases} 1 & (t_{now} - c.log.date_k > c.ttd) \\ \frac{t_{now} - c.log.date_k}{c.ttd} & (otherwise) \end{cases}$$

Here, t_{now} is the current time, $c.log.date_k$ is time of the latest record in container c's log. $c.ttd$ is the time to deadline for c. If there are a sufficient number of records in the log, the system automatically updates the value of $c.ttd$ as follows: $ttd = \mu + \sigma$ according to a normal distribution of time intervals between subsequent records in $c.log$ with average μ and standard deviation σ. Then, we classify the set of containers C_D into two sets H and L according to the priority of each container and the threshold TH by the following equation.

$$H = \{h | h \in C_D \wedge h.p > TH\}$$
$$L = \{l | l \in C_D \wedge l.p < 1 - TH\}$$

Here, TH is specified by the user between 0.5 and 1.0 in advance. H is the set of high priority containers with $p > TH$, to which insistent effects are applied, while L is the set of low priority containers with $p < 1 - TH$, to which restrain effects are applied.

If $H = \emptyset$, the system cannot provide visual effects which attract the user's attention. To avoid this, we select a container c_{max} with the highest priority among C_D and choose the SDEffect with the highest EV from $c_{max}.E$ and apply it to c_{max}.

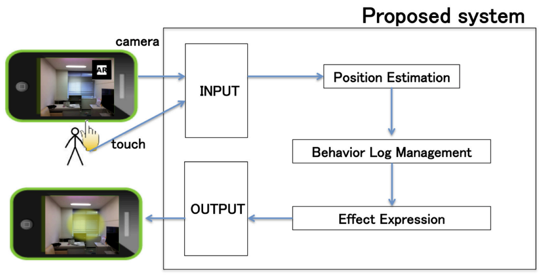


Fig. 1. System Overview

The system output creates visual effects by applying *SDEffects* to these sets of high priority containers. If the resulting visual effects on the display are too complex, the user may not understand the output intuitively. If $|H| > 1$, different SDEffects may be applied to containers of H. This impedes intuitive understanding of the priority of containers (tasks) because the visual effects are confusing. To cope with this problem, we propose the *SDEffect Decision Algorithm* shown in Algorithm 1 to select only one *SDEffect* for containers of H (also for L) according to the effectiveness value of each SDEffect (EV). As mentioned before, the effectiveness value of each SDEffect ($e.EV$) is known value through experiments in section V-C. In the proposed algorithm, we use variables I and R representing the sets of containers to which insistent effects and restrain effects are applied, respectively. In addition, we use $I.E = \bigcup_{i \in I} i.E$, and $R.E = \bigcup_{r \in R} r.E$ for representing the sets of insistent effects and restrain effects, respectively.

The system decides and simplifies the output of *SDEffects* by the process described above. We show an example of applying the algorithm in section V-C.

IV. Design and Implementation

In this section, we show the system architecture to realize our proposed method. We composed a system of three components: *Position Estimation*, *Behavior Log Management* and *Effect Expression*, shown in Fig.1.

Our proposed behavior log management requires identification of containers on the display of the user's mobile device. So we apply the position estimation technique by triangulation with AR-markers. We suppose that there are a sufficient number of AR-markers with different IDs in the target space and that each AR-marker's size and position are given in advance. The Position Estimation component calculates the distance and camera angle on each AR-marker shown on the captured image, estimates the camera position and direction, and identifies containers on the display. The behavior logs of these containers are retrieved, and appropriate *SDEffects* for the containers are decided. Finally, the Effect Expression component applies to the containers the SDEffects decided on, such as spotlight by AR technology as shown in Fig.1.

TABLE II
RESULTS OF THE EXPERIMENT TO EVALUATE THE DIFFERENCE IN EFFECTIVENESS AMONG FOUR ANNOTATION METHODS.

	Accuracy (%)	Correspondence Score	Inclusiveness Score	Time Spent (s)	Understandability	Comfort	Decidability	Usability
(a) List	86	4.7	3.0	36.4	3	3.5	4.3	3.6
(b) Label	53	2.8	2.0	32.5	4.1	3.6	4.4	3.5
(c) Light	79	4.4	2.7	26.3	3.4	3.5	3.2	3.4
(d) Fog	30	0.6	2.1	41.1	2.1	2.1	1.8	2.1

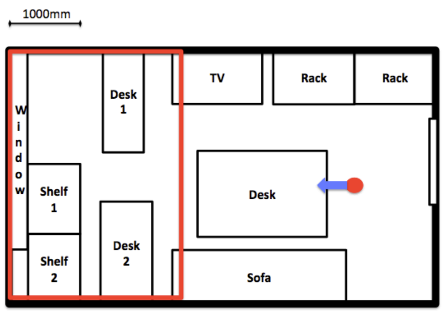


Fig. 2. Layout of the room used in the experiments.

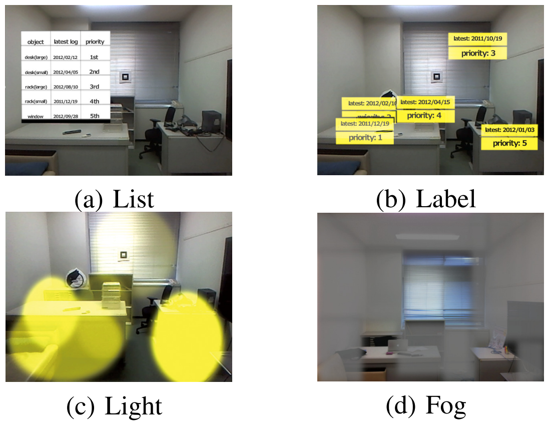


Fig. 3. Different annotation methods in experiment 1.

When the display of the mobile device is touched by the user for log input, the Position Estimation component identifies the containers from camera position and direction, and the position on the screen touched by the user. When there are two or more containers in the depth direction from the camera, the system selects the container which has line-of-sight (i.e., which is nearer to the camera). Then, the log input form is shown for the selected container so that the user can input the behavior log.

V. EVALUATION

For evaluation, we implemented a support system for cleaning a room, which shows SDEffects on the objects that have not been cleaned for a long time. The system is implemented using NyARToolkit [20], which is a Java wrapper of ARToolkit [21]. The experiments are conducted using a room in our laboratory, whose layout is shown in Fig. 2.

A. Experimental setup

Our experiments are conducted by using the following devices: a web camera (Logicool HD Pro Webcam C910), a laptop PC (CPU: Core2Duo 1.86GHz, Memory: 4GB, OS: Windows7), and a AR-marker ($10cm \times 10cm$). The participants in the experiments are 10 graduate students of our laboratory. They stand at the location that is marked with a red circle in Fig. 2, holding the laptop PC. The camera of the laptop PC is directed to the window of the room, and the region marked with a red rectangle is captured by the camera.

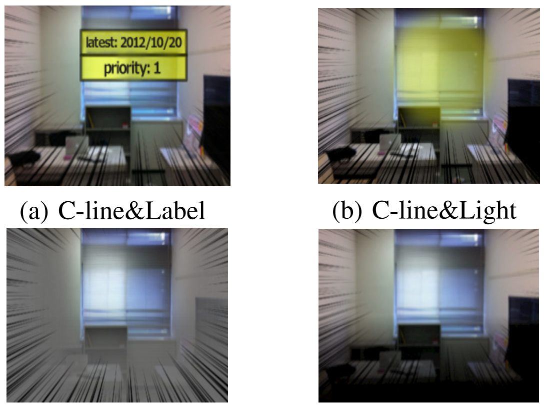
B. Experiment 1: Evaluation of the effectiveness of SDEffects

In this experiment, we evaluated the effectiveness of SDEffects through comparison with text-based annotation methods. Each method shows information for the following five objects in the room: $Window$, $Desk_1$, $Desk_2$, $Shelf_1$, and $Shelf_2$. The participants determined the appropriate order of cleaning these objects. The following four annotation methods are used and compared:

(a) List (Text information)
(b) Label (Text-based AR tag)
(c) Light (Proposed insistent effect)
(d) Fog (Proposed restrain effect)

Here, we used simple text-based annotation methods without smart placement of AR tags. "Light" applies the brighter effect as the priority increases, and "Fog" applies a stronger fog effect as the priority decreases. Examples of these methods are shown in Fig. 3. For each method, participants determined the top three locations that should be cleaned. We measured the speed with which, the participants determined the locations, and the accuracy of their choices. We also asked the participants to fill out a questionnaire about the usability of each annotation method.

The result is shown in Table II. In this table, a *correspondence score* is given for each participant's answer indicating its correspondence to the correct answer. Correspondence scores of 3, 2 and 1 are earned for each of the objects that are correctly ranked 1st, 2nd, and 3rd in the answer, respectively. Also, an *inclusiveness score* is the total numbers of correct answers selected by the participant. Note that the maximum values for the correspondence score and the inclusiveness score are 6 and 3, respectively. The *accuracy* in the table is calculated by: $\frac{Score}{Score_{max}} \times 100$, where $Score$ is the sum of the correspondence and the inclusiveness scores, and $Score_{max}$



(a) C-line&Label (b) C-line&Light

(c) C-line&Fog (d) C-line&Curtain

Fig. 4. Examples of combined SDEffects used in experiment 2 (The window is the target object that is intended to attract the user's attention).

TABLE III
COMPARISON AMONG DIFFERENT COMBINATIONS BETWEEN SDEFFECTS.

	Accuracy(%)	Decid-ability	Comfort	Favor-ability
(1) Label	70	0.35	0.44	0.28
(2) Light	90	0.28	0.59	0.49
(3) Fog	90	0.37	0.32	0.3
(4) Curtain	50	0.22	0.29	0.22
(5) C-line	100	0.23	0.46	0.34
(6) Label & Light	85	0.26	0.49	0.53
(7) Label & Fog	95	0.45	0.43	0.39
(8) Label & Curtain	60	0.47	0.52	0.49
(9) Label & C-line	95	0.42	0.53	0.68
(10) Light & Fog	90	0.82	0.58	0.52
(11) Light & Curtain	90	0.62	0.54	0.51
(12) Light & C-line	100	0.61	0.45	0.69
(13) Fog & Curtain	30	0.60	0.39	0.31
(14) Fog & C-line	100	0.28	0.29	0.36
(15) Curtain & C-line	100	0.39	0.37	0.47

is the sum of the maximum values of the two scores (i.e., $6 + 3 = 9$).

Table II shows the result of the questionnaire, which asks participants to evaluate the system from four perspectives: *Understandability*, *Comfort*, *Decidability* (how easy it is to decide the priority of cleaning), and *Usability*, each of which is represented and answered as an integer value in the range of 1 (lowest) to 5 (highest). The values for each row of the table are the average values among 10 participants.

From Table II, we can see that "List" got the highest accuracy (86%). However, for this annotation method, there was an opinion that the list covers a large space on the screen, and this is not a desirable property. On the other hand, the accuracy of "Label" was 53%. This is because it is difficult to read labels correctly in a situation where the labels overlap each other. The accuracy of "Light" was 79%, and from this we can confirm the effectiveness of the SDEffect. In contrast to this, the accuracy of "Fog" was 30%. This indicates that the shadiness of the fog effect does not differentiate priorities among objects appropriately.

The participants spent significantly less time to identify the tasks using the "Light" effects, which also achieved higher scores in the questionnaire. From this result, we confirmed that the SDEffects using insistent effects provide accurate locations and priorities of objects with intuitive interface for users.

In this experiment, however, we can not confirm the effectiveness of the restrain effects. In the next subsection, we evaluate the effectiveness of combinatorial use of insistent effects and restrain effects.

C. Experiment 2: Evaluation of combinatorial use of SDEffects

In this experiment, we show how effectiveness varies among combinations of SDEffects, aiming to obtain a concrete effectiveness value (EV) and compatibility pairs (CP) of SDEffects. We used the following five annotation methods:

(1) Label (Insistent effect using text labels)
(2) Light (Insistent effect using a spotlight)
(3) Fog (Restrain effect using fog)
(4) Curtain (Restrain effect using a dark tone curtain)
(5) C-line (Insistent effect using a concentration line)

By combining pairs of these methods, we can derive 20 different combinations of SDEffects. Four examples of the combined SDEffects are shown in Fig. 4, in which the window is intended to be the target object that should be cleaned immediately. In the experiment, for each of the combined SDEffects, participants choose one object that should be cleaned. Similar to the previous experiment, we calculated the accuracy of the participant's answers. We provided a questionnaire that evaluates from three perspectives: *Decidability* (how easy it is to decide the priority of cleaning), *Comfort*, and *Favorability*, each of which is represented and answered as a ranking value in the range of 1 (highest) to 5 (lowest).

The result is shown in Table III. For comparison, the results of stand-alone use of the SDEffect are also measured and shown in the table. Each value in Table III is averaged over 10 participants. *Accuracy* in the table is the ratio of the correct answers to all the answers.

From Table III, we can see that the overall results of combined SDEffects achieved relatively higher scores compared to the stand-alone SDEffects. Nevertheless, some of the combined SDEffects showed significantly lower scores than the stand-alone SDEffects. For example, the accuracy of the combined SDEffects of "Fog" and "Curtain" is 30%, implying that they are not compatible. These two effects are both restrain effects, which suggests that we need to use a combination of restrain effects with caution.

To make the system comprehensive, it is required that the system can automatically select the combinations of SDEffects that are mutually compatible. For this purpose, we obtain the effectiveness (EV) for each SDEffect based on the results from the above experiment, and confirm that the system can derive the combined SDEffects that are mutually compatible from EV by applying the algorithm described in Subsection III-C.

In this experiment, we calculated the EV of a SDEffect (expressed as e) as the average of the accuracies of all combined SDEffects that involve e. For example, EV of the effect "Label" is calculated by the sum of the accuracies from (6) to (9) in Table III. We can obtain the following values from Table III: $EV_{Label} = 84$, $EV_{Light} = 91$, $EV_{Fog} = -79$, $EV_{Curtain} = -70$, and $EV_{C-line} = 99$. Note that EV of a restrain effect is a negative value as described in section III-C.

We show an example of deriving the combined SDEffects. We assume the situation where five containers exist. We also assume the simple case that the container of the window is categorized to H, while the other objects are categorized to L, and all containers of L have the same priority. In these conditions, by applying Algorithm 1, the container of the window is selected as h at line 1. Then, the container of the window is added to I at line 4. Then, the "C-line" is selected as insistent effect at line 9. Similarly, other containers are selected as l at line 11, and they are added to R at lines 10-17. Then, the "Fog" is selected as restrain effect at line 18. From Table III, we can see that this combination of "C-line" and "Fog" achieves 100% accuracy, and thus we can confirm that the algorithm decides the effective pairs of SDEffects.

In the above example, although we assumed that all of the five SDEffects can be applied to any container, for the cases that each container has a different set of available SDEffects or that the multiple containers are categorized to H, other combination of SDEffects may be selected based on the algorithm. For example, if there are multiple containers categorized to H, the system selects "Light", which has the second highest EV, instead of "C-line", because the use of multiple "C-lines" confuses users in understanding which objects are important. From the above discussion, we can conclude that our proposed algorithm effectively derives the combined SDEffects based on an empirical measurement of EV.

VI. Conclusion

In this paper, we proposed a support system for daily life tasks by AR-based visual effects to attract user's attention to the high-priority tasks in a natural way. The main ideas of the proposed method consist of a behavior log management based on containers associated with locations/objects and a selection method of AR-based visual effects applied to the containers. We conducted experiments in a laboratory environment where various combinations of visual effects were evaluated and compared with text-based annotation methods.

As a result, we confirmed that (1) our method allows users to identify the locations (tasks) and the order of their urgency with similar accuracy but about 25% faster, compared to the text-based annotation methods, and (2) that using effective combinations of different SDEffects can help users identify tasks and their order more precisely than stand-alone use.

In our future work, we will conduct more experiments to evaluate the system further with users using the system continuously, and we will carry out a more thorough statistical evaluation (e.g., confidence levels) with a larger number of users. We also plan to extend our system to include a sharing of behavior logs among multiple users, and we plan to adapt the system to enable it to synchronize with a user's calender for realizing context-aware support.

VII. Acknowledgment

This work was partly supported by JSPS Grant-in-Aid for Young Scientists (B), No. 23700094.

References

[1] T. Nakada, H. Kanai and S. Kunifuji: "A support system for finding lost objects using spotlight," *Proc. of the 7th conference on Human-computer interaction with mobile devices and services(MobileHCI'05)*, (2005).

[2] M. Komatsuzaki, K.Tsukada and I. Siio: "BoxFinder: Finding items in boxes using images and visual markers," *Proc. of Pervasive 2010 (Demo)*, pp. 45–48, (2010).

[3] M. Komatsuzaki, K. Tsukada and I. Siio: "DrawerFinder: finding items in storage boxes using pictures and visual markers," *Proc. of the 16th Int'l conference on Intelligent user interfaces(IUI'11)*, pp. 363–366, (2011).

[4] E. Rothblum, L. Solomon and J. Murakami: "Affective, cognitive, and behavioral differences between high and low procrastinators," *Journal of Counseling Psychology*, pp. 387–394, (1984).

[5] R. Tenmoku, M. Kanbara and N. Yokoya: "A Wearable Augmented Reality System for Navigation Using Positioning Infrastructures and a Pedometer," *Proc. of the second IEEE and ACM Int'l Symposium on Mixed and Augmented Reality (ISMAR'03)* , (2003).

[6] A. Ajanki, M. Billinghurst and H. Gamper, et al.: "An augmented reality interface to contextual information," *Virtual Reality*, Vol. 15, No. 2-3, pp. 161–173, (2003).

[7] D. Claros, M. Haro and M. Dominguez, et al.: "Augmented Reality visualization interface for Biometric Wireless Sensor Networks," *Computational and Ambient Intelligence Lecture Notes in Computer Science*, Vol. 4507/2007, pp. 1074–1081, (2007).

[8] M. Juan, M. Alcaniz and C. Monserrat, et al.: "Using Augmented Reality to Treat Phobias," *Computer Graphics and Applications*, Vol. 25, Issue 6, pp. 31–37, (2005).

[9] A. Atif, C. Jongeun and E. Mohamad, et al.: "Evaluating the post-stroke patients progress using an Augmented Reality Rehabilitation system," *Proc. of the IEEE Int'l Workshop on Medical Measurements and Applications, 2009(MeMeA 2009)*, (2009).

[10] R Malinda, G Ann-Sofie and H Anders, et al.: "A Novel Interface to Sensor Networks using Handheld Augmented Reality," *Proc. of the 8th conference on Human-computer interaction with mobile devices and services(MobileHCI'06)*, pp. 145–148, (2006).

[11] S. Branislav and J. Radovan: "3D Interface based on Augmented Reality in Client-Server Environment," *Journal of Information, Control and Management Systems*, Vol. 8, No. 3, (2010).

[12] W. Daniel and S. Dieter: "ARToolKit on the PocketPC Platform," *Proc. of the Augmented Reality Toolkit Workshop, 2003. IEEE International*, pp. 14–15, (2003).

[13] H. Jefrey and B. Gaetano: "Location Systems for Ubiquitous Computing," *Technical Report UW-CSE 01-08-03*, Vol. 34, Issue. 8, pp. 57–66, (2001).

[14] M. Kourogi and T. Kurata: "Personal Positioning based on Walking Locomotion Analysis with Self-Contained Sensors and a Wearable Camera," *Proc. of the second IEEE and ACM Int'l Symposium on Mixed and Augmented Reality (ISMAR'03)*, Vol. 34, Issue. 8, pp. 57–66, (2001).

[15] Y. Nakazato and M. Kanbara: "Localization System for Large Indoor Environments Using Invisible Markers," *Proc. of Localization System for Large Indoor Environments Using Invisible Markers(VRST'08)*, (2003).

[16] J. Rekimoto and Y. Ayatsuka: "CyberCode: Designing Augmented Reality Environments with Visual Tags," *Proc. of Designing augmented reality environments(DARE2000)*, pp. 1–10, (2000).

[17] K. Uratani, T. Machida and K. Kiyokawa, et al.: "A Study of Depth Visualization Techniques for Virtual Annotations in Augmented Reality," *Virtual Reality, 2005*, (2005).

[18] K. Makita, M. Kanbara and N. Yokoya: "View management of annotations for wearable augmented reality," *Proc. IEEE Int. Conf. on Multimedia and Expo (ICME2009)*, pp. 982–985, (2009).

[19] M. Zimmermann and F. Robert: "Fundamentals of Sensory Physiology," *KINPODO*, Chapter 2, pp. 33–80, (1980).

[20] NyARToolkit, http://nyatla.jp/nyartoolkit/wp/.

[21] ARToolkit, http://www.hitl.washington.edu/artoolkit/.