Designing of Multimodal Feedback for Enhanced Multitasking Performance

Hyeongcheol Kim
College of Info. and Comm., Korea University
Anam-dong Seongbuk-gu, Seoul, Korea
blueskywithyou@gmail.com
+82 2 3290 3579

Gerard J. Kim
College of Info. and Comm., Korea University
Anam-dong Seongbuk-gu, Seoul, Korea
gjkim@korea.ac.kr
+82 2 3290 3196

ABSTRACT
In this paper, we explore the possibility of applying multimodal feedback to improve multitasking performance. For this purpose, we have devised a general multitasking test application, called the MSP-Blocks, which includes many basic elements of multitasking and can be used to carry out a variety of multimodal multitasking experiments. An experiment was run to study the effects of two factors (the number of jobs and types of multimodal feedback) to user task performance, specifically, interaction effort, concurrency, fairness and output quality. The results indicated that multimodal feedback did influence multitasking performance, and moreover, non-redundant multimodal feedback was more effective than no multimodality or redundant multimodality for tasks with reasonable difficulty, e.g. when the number of jobs was more than four.

Author Keywords
Multitasking, Multimodal interface, Experiment, Task performance, Concurrency.

ACM Classification Keywords
H5.m. Information interfaces and presentation (e.g., HCI): Miscellaneous.

General Terms
Experimentation, Human Factors, Performance.

INTRODUCTION
Multi-tasking generally refers to performing of more than one task at the same time [24]. Multitasking has become a way of life, especially fostered through the use of computers and mobile media devices which by design supports multi-tasking [10][11][15][24]. Concern has risen with regards to the potential harmful effects of multi-tasking such as low overall productivity, abated job qualities [11][15][24][32], and divided attention (and the ensuing danger e.g. in the classic example of "taking calls during driving") [7][29].

According to Salvucci et al., human's multitasking ability can be characterized by a continuum with completely parallel multitasking at one end and sequential (or interleaved/switched) at the other (see Figure 1) [26]. Several models have been proposed in attempts to explain the mechanism of multitasking; just to name a few, the ACT-R architecture [4], threaded cognition [27], memory for goals [3]. In all of these models, information and tasks can be processed in different modalities in parallel (in varying degrees). Tasks can also be interrupted, switched and resumed, e.g. by an executive control mechanism [18][25] and degraded performance in multitasking is often explained in terms of the retention overhead occurring during this task switching, the pause required to rehearse the problem representation and recall the previous task context [1][2][6].

Figure 1. The spectrum in the types of multitasking.

In this paper, we are interested in improving the multitask performance through proper multimodal interaction design. Modality refers to a way of representing information in some medium (e.g. visually, aurally, tactically, etc.) and multimodal interaction, to an interaction that has input and/or output and uses at least two different modalities [5]. Multimodal interaction is hypothesized to offer three avenues for our objective, i.e. improved overall task performance. First, multimodal encoding of the problem and context information can facilitate its rehearsal and recall process during the task switching [34]. Secondly, multimodal feedback can be used as a "reminder" to prevent the cognitive tunneling, the user tendency to get caught up in a single task and inadvertently fail to attend to other important tasks or events [15][37]. Lastly, multimodal interaction directly leverages on the human's innate capability of parallel (although limited to some degree) sensing/motor and (raw) information processing.

We posit that such a study can lead to "designing" of interactive tasks in the multimodal fashion to help user multitask more efficiently. In order to investigate such a design space, we have devised a test environment for general multitasking and conducted an experiment to observe human behavior and measure various task performances under different multitasking "and" multimodal feedback conditions.

This is particularly relevant nowadays as multimodal interfaces are also becoming readily available on the latest desktop and hand-held computing devices with enhanced computing power (e.g. for running real time vision based tracking, voice/gesture recognition), and various sensors/displays (e.g. camera, GPS, acceleration sensor, full color high resolution display, 5.1 surround sound effects, vibration and tactile feedback).

This paper is organized as follows. In the next section, we first review other research related to ours. Then, we describe the multimodal multitasking test platform we designed for the experiment. The next section describes the details of the actual experiment and the results. Finally, we summarize our findings and conclude with directions for future work.

RELATED WORK

Human Multitasking Models

Many of the aforementioned forms and perils of multi-tasking can be explained by the prominent human information processing models. In particular, multitasking has been studied much using the psychological refractory period (PRP) paradigm, a simple "dual" stimuli-response task [8][16]. Cognitive models such as EPIC [18] and ACT-R [4] have been applied to the PRP paradigm and have computationally emulated human multitasking. The task switch cost and interrupt overheads have been closely studied as well [26] in terms of memory and attention models. The switch cost is directly related the way the multiple tasks are represented and encoded and the amount of information for the tasks. For example, Borst et al. have shown that only one problem representation can be maintained concurrently [6] and Rubinstein et al. have shown that task switching overhead was reduced when task cueing was used and when switching to a familiar task [25]. Heterogeneous task representation and memory content interfere with one another during the task switch and cause not only delay but also confusion and error [6]. Another research by Altman and Gray has shown task switching performance depending on the access mechanism in the episodic memory representing the most recently cued task [1].

While the degree is debatable, the existence of parallelism in task processing generally accepted [8], especially for handling multiple simple independent tasks (also see Figure 1). The ability, effect, and degree of the parallel execution drops as the tasks become more difficult (or equally when the number of tasks increases) [6], and involve more cognitive activities [29]. Simply, humans are limited in their degree of parallel information processing, attentional capacity and sequential multitasking (i.e. in the number of tasks or workload they can handle) [9]. On the other hand, through practice and repetitive learning, the multitasking concurrency can also increase through integration [15][30] and humans develop various strategies in the process as well [7].

Wickens, on the other hand, in his multiple resource theory (MRT) proposes that the human operator does not have a single but several information processing resources (e.g. divided into different modalities) that can be tapped simultaneously [34][35][36]. Depending on the nature of the task, these resources may have to process information sequentially if the different tasks require the same pool (or modality) of resources, or can be processed in parallel if the task requires different resources. Wickens' theory views performance decrement in multitasking as a shortage and mismatch of these resources. Thus it seems plausible that structuring the tasks according to modalities (both within a task and among its subtasks, or across different tasks), can help user increase concurrency or task performance efficiency and reduce the degradation in job quality due to e.g. the divided attention.

It is also noteworthy that while divided attention and ensuing lack of overall concentration were cited as few of the fallouts of multitasking [33], there also exists a phenomenon, known as the cognitive tunneling, in which the user is too focused on one task (or interface) and not on the whole environment. This occurs when one task (or its interface) stands out to be so compelling that it consumes the majority of the attentional resource, so that there is not sufficient attentional capacity for others [37]. This phenomenon is similar to what is called, "starvation" in computer operating systems, where low priority tasks continuously get ignored.

Multimodal Interaction for Multitasking

Multimodal interaction has been known to improve task performance and usability in general [12][22]. Few researchers have also applied multimodal interaction for improving *multitasking* with the similar objective as ours. In particular, multitasking has been studied in the domain of multi-robot control. One of the prevalent approaches is to reduce the workload on the limited cognitive resources by streamlining and automating some of the subtasks [14]. Trouvain and Schlick [31] also have compared uni-modal and multimodal interfaces for a dual robot and payload tasks. In their case, the multimodal feedback (visual, aural and tactile) was given in a redundant fashion, and resulted in a better performance than when the uni-modal interface was used. The higher performance was observed for the robot control task, and the positive effect of multimodality toward multitasking was judged to be the reduced workload by the easier recognition of the environment situation (as also confirmed by the accompanied gaze study).

Multitasking performance has been assessed with working memory performance [30], dual task timing [17], and even attentional blinks [13]. Olsen and Wood has devised a timing based methodology for measuring interaction efficiency (of a given interface) in the context of multi-robot control, known as the fan-out equation [19][20]. "Fan-out" (which represents the degree of concurrency) is modeled by the time devoted to interaction with or control the target (IT: interaction time) vs. the time the target object carrying out meaningful action in response to the interaction (AT: activation time). The interaction time includes components like, (1) monitoring and task selection, (2) context switching, (3) problem solving and (4) command expression. IT is thus difficult to measure directly, and often, IE (interaction effort), a value proportional to the true IT, is derived instead by dividing AT by FO, as way for comparing different interfaces. IE represents how much effort is required to interact to achieve the given FO. They have empirically shown that the fan-out equation does model many of the effects of multitasking, and furthermore used to compare qualities of different human robot interaction (or extendedly to HCI) design. In our experiment, we use Olsen's framework to compare different multimodal interfaces under various multitasking conditions.

MULTITASKING TEST APPLICATION: MSP-BLOCKS

To experiment with effects of multimodal interaction and explore the design space of multimodal interfaces to and for multitasking, we have devised a general multitasking test environment, called the MSP-Blocks (Monitor, Schedule, and Process falling Blocks)." We first describe MSP-Blocks and explain how it possesses (or can be configured to possess) many of the important characteristics of multi-tasking, especially for the purpose of this study.

The interface for a single "job" of MSP-Blocks is composed of two windows (see Figure 2). The left window shows "numbered" blocks dropping from the top. The numbers represent the order of the blocks' appearances and also their ids. These "block appearance" events show up in the Event queue (in the right window) in the order of their appearances. When the blocks reach the bottom, they stay there until they are "processed" and taken off the window. This is accomplished by moving the corresponding events from the Event queue, placing them in the Command queue (via drag and drop operation), and making a final and timely click on it (see Figure 3). Note that while this "scheduling" action (placement of events in the Command queue) can be done ahead of time (by quick planning) before the respective block reaches the bottom, the "final processing" (clicking off the event from the Command queue) can only be done after the respective block has reached the bottom.

The blocks drop with varying speed and may reach the bottom in an order different from their appearances. Thus, the user must schedule the processing carefully. For instance, a processing command for a block that has not reached the bottom, but scheduled ahead of other blocks which has already reached the bottom and waiting to be processed, will stall the command execution flow. In such a case when the ordering in the Command queue is not right, they can be reshuffled by the same drag-and-drop interface.

Also, not all boxes make it to the bottom. Some may simply disappear in the middle, and thus the user may schedule to process such events, but later would have to remove them from the Command queue. The task of the user is e.g. to process and prevent these blocks from accumulating for a fixed number of blocks as soon as possible.

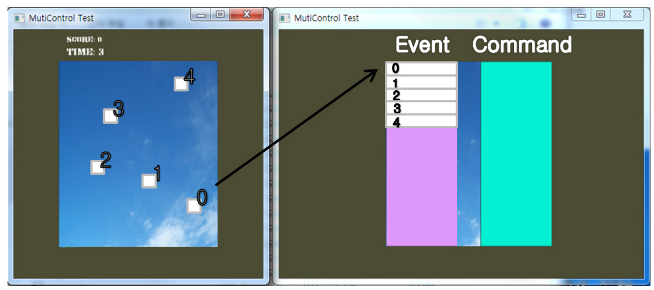

Figure 2. The basic look of the multitasking test application, MSP-Blocks. The users are to monitor the falling blocks in the left window and schedule and process them in the right window. Blocks drop in the left window and accumulate in the Event queue (right window) in their appearance order. Blocks become ready for processing (taken off the window) when they reach the bottom.

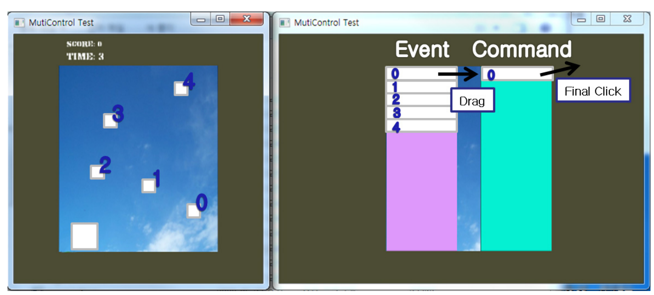

Figure 3. Events are clicked and dragged into the Command queue for the timely scheduling of processing the blocks. The block is processed by a final click off the Command queue when they actually reach the bottom.

The application is thus designed to contain many simultaneous "inter-dependent" subtasks at different temporal granularities as summarized in Table 1. A single instantiation of the application (henceforth referred to as a "job") has the two windows and is contained within one attentional or visual span (e.g. sufficiently within the field of view of the user).

Subtasks	Temporal granularity
1. Monitoring falling boxes	Continuous
2. Scheduling events	Frequent/Occasional
3. Reactive scheduling	Frequent/Occasional
4. Revising overall schedule	Occasional
5. Checking the schedule	Occasional
6. Executing commands	Frequent/Occasional

Table 1. Various subtasks and their temporal granularities in MSP-Blocks.

The Subtask 1 (monitoring the whereabouts of the blocks), which needs continuous sensory attention, is a candidate that might benefit from multimodal feedback and induce "parallel" multitasking at the same time. Other subtasks (2 – 6) that occur with less frequency may be candidates for sequential multitasking (interleaving/switching). Thus, in summary, a single MSP-Blocks job consists of a combination of task interleaving and parallel sensory processing.

The degree of multitasking can be varied by adjusting several parameters such as the average number of blocks appearing in unit time, the descending speed, and instantiating more than one job. A scoring system can be employed to motivate the user or influence one's behavior, e.g. assigning a score for each processed block, designating "gift" blocks that have higher scores, and giving penalty for blocks processed with delay (time since their arrival to the bottom).

MULTIMODAL EVENTS IN MSP-BLOCKS

The basic interface for MSP-Blocks is WIMP based (Windows, Icon, Mouse, Pointer), more specifically, visually based (single modality) for output and operated with the mouse for input. To experimentally test the effects of multimodal feedback, we defined several "special" events or situations that are to be notified to the user through multimodal output devices (multimodal input is not tested in this paper). The multimodal "notification" feedback is expected to be parallel processed using different modality resources and also as a distinct reminder, and prevent the user from entering the "cognitive tunneling."

Six (considering the capacity of human's short term memory) special events/situations are as follows,

1. Wrongly scheduled event/situation: This occurs when an event at the top of the Command queue is not correctly placed. To dequeue an event from the Command queue by the "final" click, the corresponding block must already have arrived at the bottom in the left window. Otherwise, clicking on it has no effect and it blocks the processing of all other awaiting events in the remainder of the queue. See Figure 4 (a).

2. Block waiting at the bottom: Often when the task gets difficult (e.g. by increasing the descending speed of the blocks, or increasing the number of jobs), the user may not be able to process all the events in a timely fashion. Some blocks may also be left unattended or unnoticed down at the bottom. Users are expected to show reactive behavior to handle newly noticed blocks that have accumulated and are waiting to be processed. See Figure 4(b).

3. Appearance of "gift" blocks: Occasionally, a gift box with a significantly higher score than other boxes may appear. Notifying this multimodally could help user attain higher score. See Figure 4(c). Note that gift blocks can only stay at the bottom (unprocessed) for a fixed amount of time and disappears afterwards (no points awarded). Users are expected to show reactive behavior and use short term strategies to handle "gift" blocks.

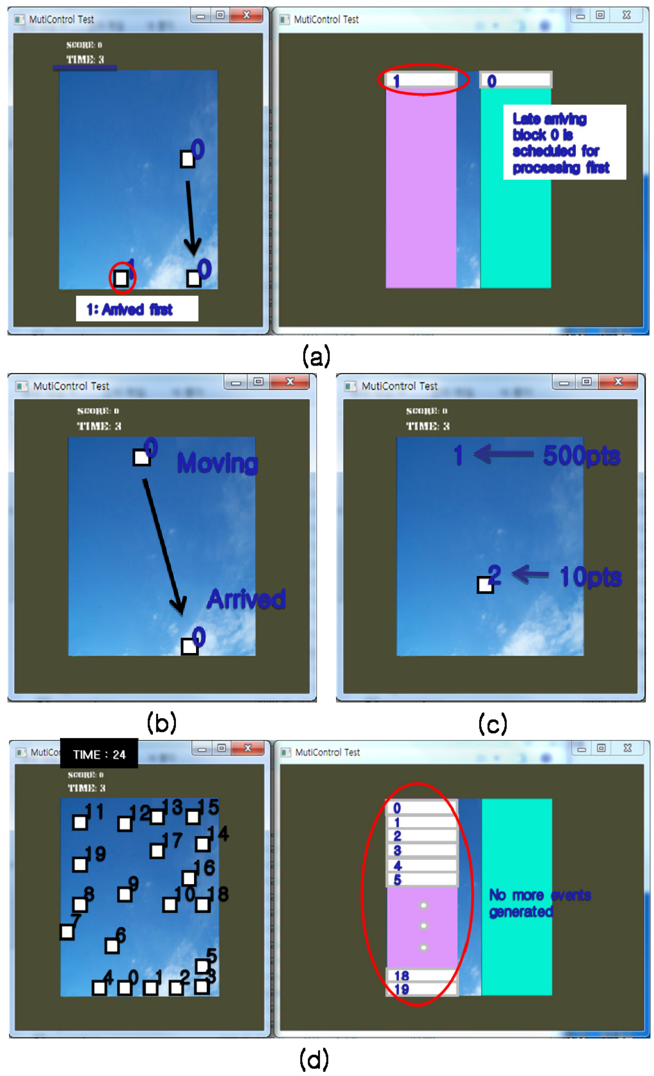

Figure 4. Depiction of four special events with multimodal feedback: (a) Special Event 1- Event 0 is wrongly scheduled ahead of Event 1. (b) Special Event 2 - Block 0 has reached to bottom and waits. (c) Special Event 3 - A gift block with 500 points has appeared. (d) Special Event 4 - Maximum number of events queued.

4. Maximum number of awaiting events reached: The maximum capacity of the Event queue is currently set to 20. When more than 20 events are accumulated, not scheduled nor processed by the user, no more events are generated. This is a situation that can happen when the overall task becomes too difficult (e.g. with more than one job) and the user gets to neglect a job. See Figure 4(d).

5. Disappearing events/blocks: Regular events (or blocks) may randomly disappear and not make it to the bottom. In this case, the user must remove this event from the command queue (if already entered).

6. New incoming events: When a new event/block is generated and starts dropping from the top, the user might want to be notified, to keep an eye on it, and decide the proper time to schedule it into the Command queue.

MULTIMODAL FEEDBACK FOR SPECIAL EVENTS

For this study, we tested feedback with combinations of three modalities: visual, aural and tactile. Note that this is in addition to the basic visual/graphics interface introduced so far (e.g. the two windows, graphic illustration of the blocks, graphic events and Event/Command queues). Here is the summary of how additional visual, aural and tactile feedback is presented to the user for the special event/situations explained above.

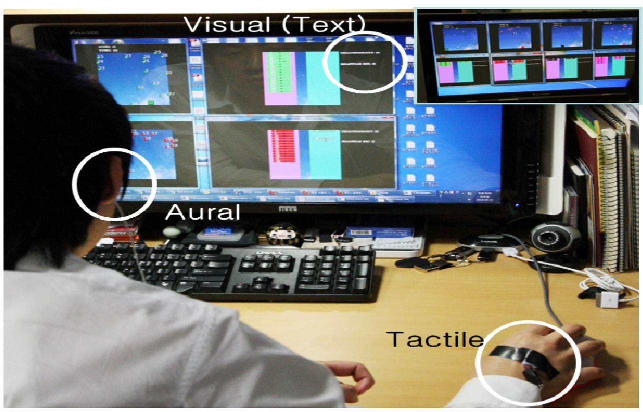

Figure 5. A snapshot of a subject carrying out the task (2 jobs) with multimodal feedback. The inset in the top right shows the screen when the number of jobs is four. The three multimodal feedbacks are also depicted.

1. Visual: when one of six events occurs, a pertaining message in text is shown in the right part of the command window. Also see Figure 5.

2. Aural: The same text message displayed visually is spoken aurally using a computer generated voice. However, since the event generation and event disappearance (special events 5 and 6) occur very frequently, familiar iconic sounds were used instead.

3. Tactile: Two vibratory motors were used only to signal the occurrences of the six events. Thus, it would not be possible to identify which event has occurred however, only with the existence of the vibratory tactile feedback itself. Instead, the identification would be accomplished with other accompanying modal output (visual or aural). Two vibration motors were used to discern between (almost) simultaneous events. Employing six different vibration patterns (corresponding to all the six special events individually) was not considered due to the low discerning power and unfamiliarity.

EXPERIMENT

To investigate the relationship between multitasking and multimodality, our experiment was run with two representative factors in each dimension. The first was the number of jobs. Note while a single job may contain inter-dependent subtasks, the number of job was varied to see the effect of multiple independent jobs as well. The second factor was the form of multimodal feedback. Note that we were not interested in identifying the specify type of the modality used (e.g. visual-tactile is better than visual-aural combination and etc.) but in the way they were matched to the subtasks (e.g. redundancy). Modality choice study is more proper with a particular domain application, rather than an abstract test application such as ours. Finally, we were interested in the possible interaction among these two factors, e.g. how to design the form of multimodal feedback according to varying multitask difficulty (i.e. no. of jobs).

Experimental Design

Given the experimental test bed as described above, the actual experiment was designed with two factors: (1) number of jobs (J) and (2) multimodal feedback configuration (M). For the first factor, we had set 3 levels, i.e. 1 (single job, but still multitasking at the micro-level, see further), 2, and 4 "jobs," and for the second factor, 4 levels in the degree and ways the multimodal feedback were given (Level 1 being uni-modal or basic visual feedback only condition). Each level for the second factor (M) is further illustrated in Table 2. Table 2 shows how the multimodal feedbacks were combined in different manner (for the six special events) to constitute each level for the factor of M. The Level 2 multimodality condition is designed such that each modality was more or less assigned to the special event in the non-redundant manner (i.e. unique feedback using only one modality for an event), while for the Level 3 and 4 multimodality conditions, at least two modalities were used for one special event in an increasingly redundant fashion

Note there are two levels of multitasking occurring in MSP-Blocks, one across the number of jobs and the other across the subtasks within a single job. The "number of jobs" factor should greatly influence the macro-level multitasking through job interleaving, and the "multimodal feedback" factor, the micro-level multitasking through sensory parallelism for handling the special events.

To summarize, the experiment was designed as a 2 factor (M=4 x J=3) within subject repeated measure (Table 3), totaling in 12 treatment groups.

Experimental Task

Using MSP-Blocks, the user was to carry out the following specific task. For each job, the user is asked to process 20 boxes (fixed). For each job, 20 boxes fell from the top and the user was to schedule the corresponding events by moving their icons from the Event queue into the Command queue in a proper order (and finally dequeue them when each event reaches their designated time). When the number of job was only one, the user would process 20 boxes for that job. When the number of job was more than one, it meant two, or four jobs would be instantiated simultaneously, and multiples of the 20 boxes had be processed (e.g. 40 or 80 boxes in total). As seen in Table 3, for each multitasking situation (according to the number of jobs), four multimodal feedback configurations were tested.

Level		Visual (Default)	Visual (Additional)	Aural	Tactile
1	6 events	O			
2	Event 1	O		O	
	Event 2	O			O
	Event 3	O		O	
	Event 4	O			O
	Event 5	O	O		
	Event 6	O	O		
3	Event 1	O	O	O	
	Event 2	O	O	O	
	Event 3	O		O	O
	Event 4	O		O	O
	Event 5	O	O	O	
	Event 6	O	O	O	
4	Event 1	O	O	O	O
	Event 2	O	O	O	O
	Event 3	O	O	O	O
	Event 4	O	O	O	O
	Event 5	O	O	O	O
	Event 6	O	O	O	O

Table 2. The 4 level feedback configurations for the "multimodality" factor (M) in the experiment. Level 1: ground condition, Level 2: non-redundant, Level 3: moderately redundant, Level 4: highly redundant.

Dependent Variables

Dependent variables were measured that reflected the level of user multitasking performance in terms of speed, concurrency, fairness, and quality.

1. Speed: As a way to measure multitasking performance, we apply the fan out equation developed by Olsen et al. [20]. The parameter of "fan out (FO)" roughly maps to the extent to which a user can multitask (e.g. how many robots can a human control simultaneously) and it is defined as the ratio of Activation time (AT) to Interaction time (IT). AT time refers to the time during which a robot (or equivalently a task) is activated, paid attention to, processing commands. In our case, it is the summation of the time durations for user initiated subtasks themselves (e.g. to schedule and process the block events), measured from the initiating to their finish. This is not to be confused with the time "taken" to interact for this to happen in the first place.

No. of Jobs / Multimodality	Single Job / Multitasking	Multiple Jobs / Multitasking	
	No. of Job = 1	No. of Job = 2	No. of Job = 4
Level 1	G1	G5	G9
Level 2	G2	G6	G10
Level 3	G3	G7	G11
Level 4	G4	G8	G12

Table 3. The 2 factor within subject repeated measure design.

To reiterate, the AT represents the time an object (being controlled, or in this case, the block event) being activated or treated by a single interactive command. The IT would correspond to the time needed to for the physical clicks plus the time for the mental effort. Therefore, as Olsen points out, it is difficult to directly measure this quantity [20]. We fix the value of the FO to 20 (which is the number of blocks for each job) instead, measure AT, and estimate the IE (a value proportional to IT). When the number of jobs increases to two or four (J = 2 or 4), then the FO is set to 40 or 80 accordingly. Since the FO is fixed, IE is proportional to the AT. We interpret that the less the AT (per unit FO) is, the more efficient the given interface is (i.e. less effort).

2. Concurrency: Another measure of multitasking performance is the degree of concurrency achieved in completing the jobs. The concurrency was measured by counting the number of times the visual attention has shifted between the jobs (in a multiple jobs situation only). Higher number of switches in a unit time means relatively higher level of concurrency induced through the given task and multimodal condition. It is the concurrency among the "jobs" that is measured. The concurrency at the subtask level is only indirectly deduced from the "speed" (i.e. the higher the IE, the more concurrency is attained).

3. Fairness: Similar to concurrency, another desirable feature in multitasking is the fairness (or avoidance of starvation), that is, not leaving certain blocks or jobs unnecessarily long unattended (note that in this experiment there was no priority among the different jobs). The wait time for the blocks (or jobs) to be processed was measured to assess this aspect.

4. Quality: Even though the FO or the number of total blocks to be processed was fixed, a differentiated score was possible because there existed the occasional (and additional) "gift" blocks. Three specially marked and high score block events were given equally to all subjects but in

random order and this produced a variation in the total score (i.e. scores for 20 boxes (fixed) + scores for three gift blocks). Differently from the regular blocks, the gift boxes would be lost if not treated in some fixed amount of time. The assessment of scores is only meaningful at each job level among different multimodality conditions.

Hypotheses

The first hypothesis on the projected outcome of the experiment is that multimodal feedback can help a user multi-task more efficiently. There were three bases for our hypothesis: (1) the multimodal feedback leverages on the humans' ability to parallel process sensory information to some degree within a single job boundary (among the subtasks), and secondly (2) multimodal feedback can act as a "reminder" to shift attention to unattended subtasks or jobs, and (3) multimodal feedback can help structure the encoding of the job or subtask context information and help reduce the task/job switch overhead.

However, as a second hypothesis, it is also expected not all forms of multimodality would be effective. Or put it differently, different forms of multimodality might be proper for different difficulty or multitasking situations. For one, users tend to have preferences toward certain modalities [28]. Therefore, depending on how the information is presented, there is a possibility, for example, modality masking and modality interference to occur at the same time [28]. That is, information structural differences across the modalities can interfere with one another and degrade the user performance.

Procedure

Sixteen paid subjects between of age 21 and 29 (14 men, 2 women, average age 24.5, no particular physical impairment) participated in the experiment. All subjects were recruited from the university campus and had used desktop computing and console games extensively before being familiar to basic graphical interfaces and multimodal feedback. This age group and background was chosen considering the need for minimum prior experience in multitasking.

There were a total of 12 treatments (4 x 3) and the subject carried out each task treatment two times (total of 24). The order of the treatment trials were balanced using the Latin Square and the whole session took approximately three hours per subject (including the resting time between the treatments to minimize the fatigue factor as much as possible). Prior to the actual task, the subject was given at least 30 minutes of training, familiarizing oneself with the task and trying it to a degree where they felt sufficiently competent and comfortable. For motivational and experimental purpose, the compensation was given according to the scores and the subjects were told about it beforehand. The dependent variables were captured through the test application program and later analyzed using one way ANOVA for the overall effects and Tukey's test for individual differences.

As for the experimental setup, MSP-Blocks was implemented (using OpenSceneGraph [38] for graphics, DirectShow [39] for audio) and executed on a PC (Intel Core i7-950 with DDR3 8G RAM, running Microsoft Windows 7) and presented with a 50 inch monitor (LG Xcanvas PDP, resolution: 1280x800) placed on a table. The user sat on a chair at the table (the viewing distance was approximately 60-80cm) and carried out the given jobs using a mouse (see Figure 5). For the multimodal feedback, the user also wore a headphone and a vibration device on one's dominant mouse controlling hand. The vibration device was implemented with flat coin-type vibrators controlled by a custom Arduino based board [40].

RESULTS

Interaction Effort per Subtasks (Blocks)

For each treatment, the AT was captured and averaged over the total number of block events (the target interaction object in this experiment). Note that IT equals AT over FO which was fixed at 20. Thus, lower activation time means lower IT (and IE), and indirectly higher interaction efficiency.

The Figure 6 shows that when only one or two jobs were given to the subject, no statistically significant differences in IE was observed across different modality configurations nor between the number of jobs. That is, multimodality did not improve nor degrade the performance regardless of its configuration when only one or two jobs were given.

The user performance did not really change either when the job was increased from one to two. However, when the number of job was increased to four ($J = 4$), there was statistically very significant difference (p-value < 0.000) in task performance compared to when only one or two jobs were given. In addition, multimodality had different effects, namely, the task performance was improved in the second multimodality condition (see circled portion of Figure 6, p-value $= 0.022$, t-value $= 2.36$) but degraded for the third and fourth multimodality condition, all compared to the ground condition of when no multimodality was used (specific test statistical values (other treatment pair-wise t-values and p-values are omitted for limited space). .

The second multimodality condition was designed such that each modality was more or less assigned to the special event in the non-redundant manner, while for the third and fourth multimodality conditions represented more redundant multimodal feedback, possibly incurring some kind of modality confusion and interference.

Oddly, the AT times seem to decrease with the increasing number of jobs. This is a mislead observation because in our experiment, when only one job was given, the job was sufficiently easy (plus no physical attentional shift was necessary) and the subjects did not have to shoot to be efficient (all they had to do was to process 20 blocks and score as much as possible) nor did they become confused with overly redundant multimodal feedback.

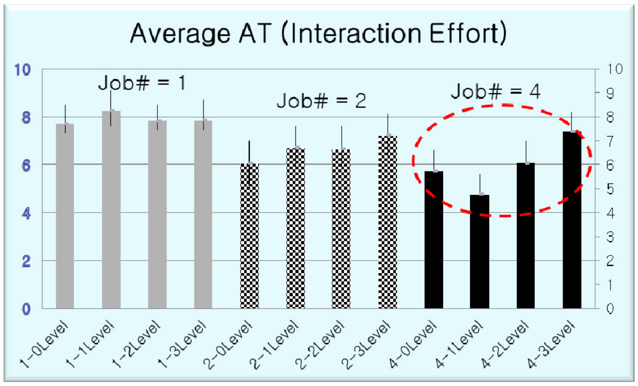

Figure 6. Average AT or interaction effort per target control object (block) over different number of jobs and multimodality conditions.

Concurrency among Jobs (for J=2 and 4 only)

The average degree of concurrency (number of job switch times, collected by recording the mouse positions) was measured and averaged over the number of total of jobs given. Figure 7 shows the results and when only one or two jobs were given, no statistically significant differences were shown in concurrency either between the number of jobs, nor across different multimodality conditions.

Note that when only one or two jobs were given, the target jobs and subtasks were within the user's field of view, thus not requiring any physical shifts in attention. It was easier for the user to parallel process the recognition of special events that drew attention to prevent tunneling on one job or subtask only. When four jobs were given to the user, the concurrency increased in a statistically significant manner (p-value=0.038). The concurrency was further different between when no multimodality (M=1) was used and when multimodalities were used (M=2-4, p-values<0.000). Despite the possibility of interference due to too many or too high degree of multimodality, it still had the effect of drawing attention to remind the subject to switch one's attention to other jobs and tasks.

Fairness (Event waiting time) among Subtasks (Blocks)

Depending on the difficulty of the job(s), the unprocessed and unattended block events may start accumulating and left unattended or unprocessed. We measured, if any, the time taken since block arrived at the bottom until it got processed. It reflects how timely the block events get processed without any delay and further measures the time of overall starvation. Figure 8 shows that the average waiting time per single event for different treatments. The event waiting time generally increased according to the number of jobs due to the increasing difficulty of the multi-job handling. Not much difference was observed for when the job number was only one or two. It seems again that the subject found these conditions to be sufficiently easy to handle and this is consistently reflected in other measures as well. For the condition of when the number of jobs were four, one can easily see the steep increase in the waiting time, and in addition, how the multimodality conditions helped reduce the average waiting time (within this J=4 condition), particularly so with the second multimodality configuration (p-value = 0.058). No statistical difference was found between 4-3 and 4-4.

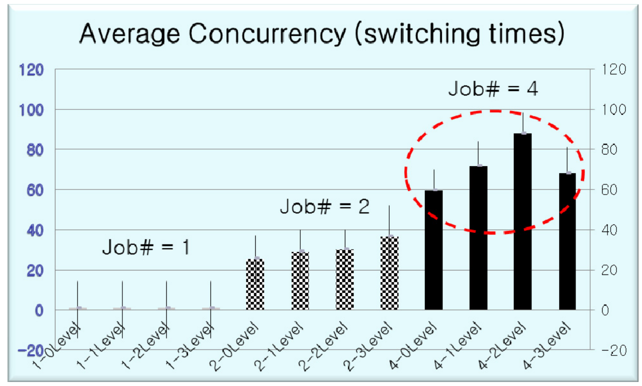

Figure 7. Concurrency among jobs for different multimodal feedback conditions.

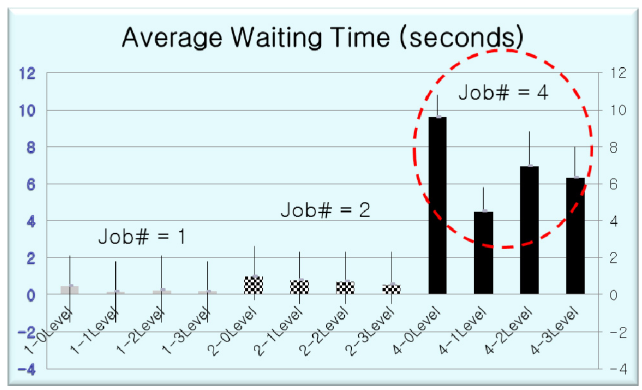

Figure 8. Average waiting time among subtasks or blocks for different number of jobs and multimodality conditions

Score for a Given Number of Jobs

The number of blocks to be processed per job was fixed at 20 (FO value being equal to 20 times the number of jobs). The score for each regular block is 10 and thus the basic score that the user can obtain is 10 times the number blocks (1000). However, we also inserted time to time gift blocks that had scores up to 500. These blocks were given to the user at random times but in equal number for all the subjects. Note that these gift blocks are notified to the user using multimodal feedback (for M=2-4). These blocks may be missed too if left unprocessed for a fixed amount of time. Thus, processing of these gift blocks and the resulting differentiated score can be used for another indication of the relative user performance or quality. Figure 9 shows the average scores. When only one or two jobs were given, the users seem to have found it sufficiently easy, and took care of all the gift blocks, resulting in no statistical differences in the score. When the number of jobs were four and when multimodality is not used (J = 4, M = 1, 4-1), we see that the user starts to miss few gift blocks and the average score drops a little bit, but increases again when multimodality is used (p-value = 0.052). Again, this is a result consistent with the other measures.

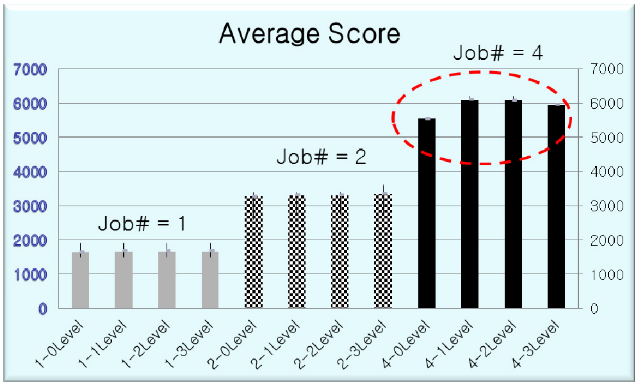

Figure 9. Average score for different number of jobs and multimodality conditions

Analysis across Number of Jobs

Graphs in Figure 6 through 9 can be rearranged for analysis of the effect of the number of jobs (J) for each multimodal condition. The analysis results showed that multimodality condition 2 and 3 were generally effective in lowering the AT, waiting time and increasing the concurrency and score values (we omit the details for limited space).

DISCUSSION

The first hypothesis that multimodality was expected to be of help to improve task performance was confirmed, but only for a task with reasonable difficulty. While in our experiment, for the conditions of J = 1 or 2 (sufficiently easy task), multimodal feedback did not negatively affect performance, it still could be a source of confusion. In our study, the individual jobs were independent but of same type. The similarity in the problem state should be of great help in the context switch, and thus multimodality would not be of much help when the task was relatively easy. There are also literatures pointing to a dual brain mechanism which allows the dual task fairly manageable [23].

The second hypothesis that a particular form of multimodality was expected of help was also confirmed. That is, providing multimodal feedback in a non-redundant manner was shown to be effective. While this phenomenon is explainable, e.g. with the Wickens' multiple resource theory, it is also true that there exists a general belief in the utility of redundant multimodal interfaces as well (that it helps recognition of an event [12][21][22]). Redundant multimodal interfaces also run the risk of mutual interference as well [28]. The least we can say is that redundancy in multimodality requires a careful design such that each modality complements one another very well.

Our study still has a long way to go. There are many avenues for future work. For instance, MSP-Blocks can be reconfigured easily to test multimodality effect with different task priorities, task performance and user behavior with time limit, multimodal input, etc. While there were different types of subtasks (e.g. reactive, planning, motor, mental) in our experiment, a detailed analysis in this respect has not been carried out yet.

CONCLUSION

In this paper, we have explored the possibility of applying multimodal feedback to improve multitask performance. MSP-blocks, the test multitasking application was carefully designed to include many basic elements of multitasking and can be used to carry out a variety of multimodal multitasking experiments. The first experiment described in this paper varied the number of jobs and types of multimodal feedback and studied their effect to user task performance, namely, interaction effort, concurrency, fairness and qualitative output. Our results indicated that non-redundant multimodal feedback was more effective than no multimodality or redundant multimodality for tasks with reasonable difficulty, e.g. when the number of jobs was more than four. We plan to continue to study other factors such as task priority, task type, level of automation, etc. We believe that the results of the study can contribute to establishing a guideline for multimodal interface design for multitasking applications.

ACKNOLWEDGEMENT

This research was supported in part by the Strategic Technology Lab. Program (Multimodal Entertainment Platform area) and the Core Industrial Tech. Development Program (Digital Textile based Around Body Computing area) of Korea Ministry of Knowledge Economy (MKE).

REFERENCES

1. Altman, E. & Gray, W. (2008). An Integrated Model of Cognitive Control in Task Switching. Psychological Review, 115 (3), 602-639.
2. Altman, E. & Trafton J. (2007). Time Course of Recovery from Task Interruption: Data and a Model. Psychonomic Bulletin & Review, 14 (6), 1079-1084.
3. Altmann, E. & Trafton, J. (2002). Memory for Goals: An Activation-based Model. Cognitive Science, 26, 39-83.
4. Anderson, J., Bothell, D., Byrne, M. D., Douglass, S., Lebiere, C., & Qin, Y. (2004). An Integrated Theory of the Mind. Psychological Review, 111, 1036-1060.
5. Bernsen, N. & Ole, D. (2010). Multimodal Usability (Human-Computer Interaction Series), Springer.
6. Borst, J., Taatgen, N. & Rijn, H. (2010). The Problem State: A Cognitive Bottleneck in Multitasking. Journal of Experimental Psychology: Learning, Memory, and Cognition, 36(2), 363-382.
7. Brumby, D.P., Salvucci, D.D., & Howes, A. (2007). Dialing While Driving? A Bounded Rational Analysis of Concurrent Multi-task Behavior. Proc. of the International Conf. on Cognitive Modeling, 121-126.

8. Bryne, M. & Anderson, J. (2001). Serial Modules in Parallel: The Psychological Refractory Period and Perfect Time Sharing. Psychological Review, 108, 847-869.
9. Damos, D. (1991). Dual-task Methodology: Some Common Problems. In Multiple-task Performance, D. Damos (Ed.), 101–119.
10. Foehr, U. (2004). Media Multitasking among American Youth: Prevalence, Predictors and Parings. Kaiser Family Foundation.
11. Healy. M. (2004). “We're all multi-tasking, but what's the cost?” Los Angeles Times (Home Ed.), Jul 19, F.1.
12. Jaimes, A. & Sebe, N. (2005). Multimodal Human Computer Interaction: A Survey. IEEE International Workshop on Human Computer Interaction.
13. Junina, I. & Taatgen, N. (2006). How Attentional Blink Facilitates Multitasking. Proc. of Annual Conf. of the Cognitive Science Society.
14. Kaber, D., Wright, M. & Sheik-Nainar, M. (2006). Investigation of Multimodal Interface Features for Adaptive Automation of a Human-Robot System. Intl Journal of Human Computer Studies, 64, 527-540.
15. Loukopoulos, L., Dismukes, K. & Barshi I. (2000). The Perils of Multitasking, Aerosafty World, 19-23.
16. McCann, R., Remington, R. & Van Selst, M. (2000). Lexical Decision and Naming: Evidence from the Psychological Refractory Period Paradigm. Journal of Experimental Psychology: Human Perception and Performance, 26, 1352-1370.
17. McElree, B. (2001). Working Memory and Focal Attention. Journal of Experimental Psychology: Learning, Memory and Cognition, 27(3), 817-835.
18. Meyer, D. & Kieras, D. (1997). A Computational Theory of Executive Control Processes and Human Multiple-task Performance: Part 1. Basic Mechanisms. Psychological Review, 104, 3-65.
19. Olsen, D. & Goodrich, M. (2003). Metrics for Evaluating Human-Robot Interactions, Proc. of the Performance Metrics for Intelligent Systems Workshop.
20. Olsen, D. & Wood, S. (2004). Fan-out: Measuring Human Control of Multiple Robots, Proc. of ACM CHI.
21. Oviatt, S. (2003). Advances in Robust Multimodal Interface Design. IEEE Computer Graphics and Applications, 23(5), 62-68.
22. Oviatt, S. (2002). Multimodal Interfaces. Handbook of Human-Computer Interaction, Ed. Jacko, J. & Sears, A., Lawrence Erlbaum.
23. Rettner, R. (2010). Why We Can't Do 3 Things at Once, LiveScience, http://www.livescience.com/health/brain-multitasking-limit-100415.html
24. Rosen, C. (2008). The Myth of Multitasking. The New Atlantis, Spring, 105-110.
25. Rubinstein, J., Meyer, D. & Evans, J. (2001). Executive Control of Cognitive Processes in Task Switching. Journal of Experimental Psychology - Human Perception and Performance, 27(4), 763-797.
26. Salvucci, D., Tattgen, N. & Borst. J. (2009). Toward a Unified Theory of the Multitasking Continuum: From Concurrent Performance to Task Switching, Interruption, and Resumption. Proc of ACM CHI.
27. Salvucci, D. & Taatgen, N. (2008). Threaded Cognition: An Integrated Theory of Concurrent Multitasking. Psychological Review, 115, 101-130.
28. Shimojo, S. & Shams, L. (2001). Sensory Modalities are not Separate Modalities: Plasticity and Interactions. Current Opinion in Neurobiology, 11, 505-509.
29. Strayer, D. (2007). Multi-tasking in the Automobile. In Applied Attention, Kramer, A., Wiegmann, D. & Kirlik, A. (Eds.), Oxford Univ. Press, 121-133.
30. Taatgen, N. (2005). Modeling Parallelization and Flexibility Improvements in Skill Acquisition: From Dual Tasks to Complex Dynamic Skills. Cognitive Science, 29, 421-455.
31. Trouvain, B. & Schlick, C. (2007). A Comparative Study of Multimodal Displays for Multirobot Supervisory Control, In Engin. Psychol. and Cog. Ergonomics (HCII 2007), D. Harris (Ed.), 184-193.
32. Vega, V., McCracken, K., Nass, C. & Labs, L. (2008). Multitasking Effects on Visual Working Memory, Working Memory and Executive Control. Annual Meeting of the Intl. Communication Association.
33. Ververs, P. & Wickens, C. (1998). Head-up Displays: Effects of Clutter, Display Intensity, and Display Location on Pilot Performance. Intl. Journal of Aviation Psychology, 8(4), 377-403.
34. Wickens, C., Lee, J., Liu, Y. & Gordon-Becker S. (2003). Introduction to Human Factors Engineering Second Edition: Cognition VI.
35. Wickens, C. (1991). Processing resources and attention. In Multiple-task Performance, D. Damos (Ed.), 3-34.
36. Wickens, C. (1980). The Structure of Attentional Resources. In: Attention and performance VIII, Nickerson, R.S. (ed.), 239-257.
37. Wickelgren, W. (1979) Cognitive Psychology, Prentice Hall.
38. http://www.openscenegraph.org
39. http://msdn.microsoft.com/en-us/directx
40. http://www.arduino.cc