Inferring Multitasking Breakpoints from Single-Task Data

Peter Bogunovich (pjb38@drexel.edu)
Drexel University, Department of Computer Science
Philadelphia, PA, USA

Dario D. Salvucci (salvucci@cs.drexel.edu)
Drexel University, Department of Computer Science
Philadelphia, PA, USA

Abstract

Recent research has shown that computer users placed in a deferrable multitasking situation generally postpone secondary-task interruptions until points of low mental workload in the primary task. Studies examining this phenomenon have relied on empirical data that explicitly show user switch points in the course of multitask performance. This paper addresses a related question: Can these same switch points, found empirically in a multitasking context, be inferred solely from single-task data? We investigate this question and propose an approach that analyzes a particular behavioral signature in single-task data—outliers in the distributions of time between task actions—to infer multitasking breakpoints. We evaluate this approach using behavioral data from a user-interface task, showing how the proposed method's inferences from single-task data match well to the real switch points observed during multitask performance.

Keywords: Multitasking; task analysis; data analysis.

Introduction

Multitasking is a concept that is familar to most computer users. It is not uncommon for a user to switch computing tasks every few minutes. In many cases switching is initiated by an interruption of the current task. For example, a notification of a newly received email may appear on the screen prompting a user to stop what he is doing and look at his email before continuing his previous task. Research has shown that interruptions can increase the overall time spent on a single task. One important source of this increase is the *resumption lag*, or time required to switch back to the task and resume after the interruption has been addressed (Trafton, Altmann, Brock, & Mintz, 2003; Monk, Boehm-Davis, Mason, & Trafton, 2004). Recently it has been shown that it is more beneficial to interrupt at certain points than at others (Adamczyk & Bailey, 2004; Bailey & Konstan, 2006; Cutrell, Czerwinski, & Horvitz, 2000). One particularly strong result states that the performance loss associated with interruption is reduced when interruptions occur at points of low mental workload (Iqbal & Bailey, 2005). This result has obvious importance when considering *forced interruptions* in which the user is required to address the interruption immediately before moving on with the primary task.

The relationship between mental workload and interruptibility has been strengthened in further studies of *deferrable interruptions* (Salvucci & Taatgen, 2010) in which a user is notified of a secondary task but the user can defer processing of this task until a later (presumably more comfortable) time. For example, it has been shown (Salvucci & Bogunovich, 2010) that in this situation users tend to defer switching tasks until a point where there is a drop in mental workload. As exemplified by these studies, a detailed analysis of when users switch tasks is critical to a deeper understanding of human multitasking behavior. A particular goal in this line of research involves the prediction of breakpoints, the points in a task sequence where the user can most conveniently switch tasks.

One approach to breakpoint prediction combines expert coding, feature detection and model prediction (Iqbal & Bailey, 2007). This approach begins by observing users in some natural multitasking environment. An expert manually examines user actions and identifies specific features which appear to signal breakpoints. A statistical model is then developed based on these features. Promising results have been obtained with his method, however it requires the human coders to identify the perceived breakpoints and features, and does not necessarily make use of the relationship between cognitive load and interruptibilty. A successful related approach that makes use of mental workload is to examine the typical execution structure of an action in advance and use this structure to estimate opportune breakpoints (Bailey, Adamczyk, Chang, & Chilson, 2006). This method still requires expert analysis and it may fail when variation in strategy is introduced.

There exists a well-known relationship between cognitive load and pupil dilation (Beatty, 1982). Researchers have made use of this link in another approach to breakpoint detection (Bailey & Iqbal, 2008). In this approach, pupil dilation data is recorded as users perform a task, and subtask boundaries, where there is an assumed drop in cognitive load, are estimated by changes in dilation. The result is a more general and more automatic estimation of good potential breakpoints that relies less on pre-computed models or experts. Despite these findings, it may not be possible to obtain pupil-dilation in practice for many tasks.

In this paper we attempt to infer multitasking breakpoints in a automatic, data-driven manner. In this respect our approach is most similar to (Bailey & Iqbal, 2008), but instead of relying on typically inaccessible equipment like eye-trackers, our goal is to come up with the good estimates using only data logs of system events generated by users performing a single primary task. Our analysis focuses on the distributions of elapsed time between recorded event pairs, using single-task data collected for a customer-support task

(Salvucci & Bogunovich, 2010). From our analysis of the recorded data, and particularly the estimation of observed outliers in distribution tails, we were able to infer breakpoints that closely mirror actual deferred user breakpoints as they arose in a multitasking context.

Task and Data

The task that we analyzed is taken from a recent experiment in which users performed a mail-based customer-support primary task while occasionally being interrupted by chat (instant-message) questions. The primary task simulated a typical customer-service scenario where a user receives email inquiries for the prices of a variety of products. The simulation was comprised of a simulated email program and a browser window used for looking up product prices, shown in Figure 1. Each email in the inbox contained a request for the price of a single product. Once the user read the email and became aware of the request, he or she had to look up the product in the browser to obtain the correct price. Each product consisted of a real manufacturer name and a fictitious model identifier (for example, "Canon H-44", or "Sony M-76"). To find the price of a product, the user had to first click on the proper manufacturer name from the top-level of the browser, and then click on the proper model identifier from a secondary browser level. The user could have at any time returned to the top level of the browser by clicking "home" button. Once the price of the product in question had been located, the user sent a reply email containing the requested information. The users were also asked to manually move the replied to emails to a "replied" bin by clicking and dragging.

In the multitasking setting a secondary chat task was introduced which simulated a typical instant messenger conversation. A chat window was included in which the users were occassionally asked questions about recent films by a simulated interlocutor. The users were notified of a new question by having the chat window flash, but it was up to the users to decide when to break from the primary mail task to address the questions once the notification was received.

It is important to note that in both the single mail task and dual mail and chat task situations, the simulation windows were arranged so that only the window that was currently being focused on could be seen. For example, while looking up a product price in the browser window, the name of the product given in the email window was obscured. This required the users to commit sub-task relevant information to memory.

For our analysis, we look specifically at single mail task data collected from six participants in this experiment. This data was collected in a sesssion where the chat simulation was not present. In particular, our goal is to analyze the single-task data, infer and estimate breakpoints from these data, and then evaluate our estimates by comparing the results to the also collected multitask data. The data recorded for the mail task (both single- and dual-task contexts) comprises a sequence of time-stamped events occurring in the task. Table 1 lists and describes these events. The full data recorded for a single event includes the event type, as given in Table 1, the time of the event, and any auxilary information about the event (for example, which character was typed, or which product link was clicked); we use only the event type and time information here.

mail-select:	Select (click on) an email from a list.
mail-move:	Move (drag) an email to the "Replied" bin.
browser-focus:	Change focus to browser window.
browser-home:	Press "browser home" button.
mfr-link:	Click on "product manufacturer" link.
model-link:	Click on "product model" link.
reply-button:	Press "Reply" button to open a new window to compose response.
reply-type:	Type characters in a response email.
reply-send:	Press the "Send" button to send response email.
reply-focus:	Change focus to an opened response window.

Table 1: User events in the mail customer-support task.

Analysis of Recorded Event Data

Starting with the recorded single-task data, we tried several theoretically-motivated approaches for analyzing the data and inferring multitasking breakpoints. In the following sections we discuss several of the approaches that we took. Motivations and limitations associated with each approach are given.

Frequency of Sequences

When relying solely on the frequencies of occurrence of given event sequences, perhaps the most naive hypothesis is that good locations for breakpoints are found between pairs of consecutive events that were observed infrequently. The motivation is that sequences which appear frequently consist of events that are strongly linked together, and thus switching tasks between the events is less desirable or at least less likely.

Problems with this hypothesis arise immediately, however, in noting that it is extremely unlikely or impossible for many pairs of events to occur consecutively. For instance, in the mail task, it is not possible for to observe the event "*model-link*" followed immediately by the event "*mfr-link*" due to the design of the task interface. Other pairs of consecutive events are unlikely due not to the design of the simulation, but simply because they make little sense for any user attempting to complete the mail goal. For example, the sequence "*mfr-link* → *browser-home*" is not useful in looking up a product price, since the price is not obtained until the "*model-link*" event. Any occurrences of "*mfr-link* → *browser-home*" are likely due to an error by the user and there is little reason to believe that this is a good place to switch tasks.

While it is clear that pairs of events with no or few occurrences do not necessarily represent good breakpoints, it still

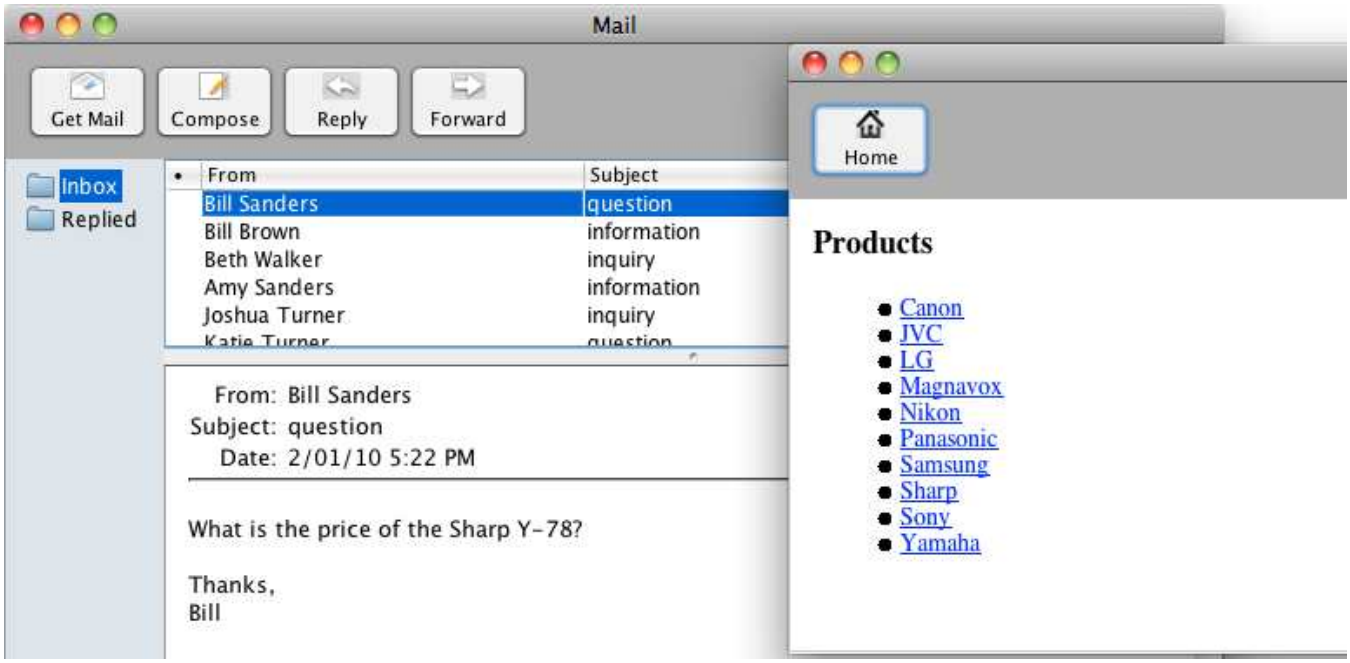


Figure 1: The customer service mail simulation.

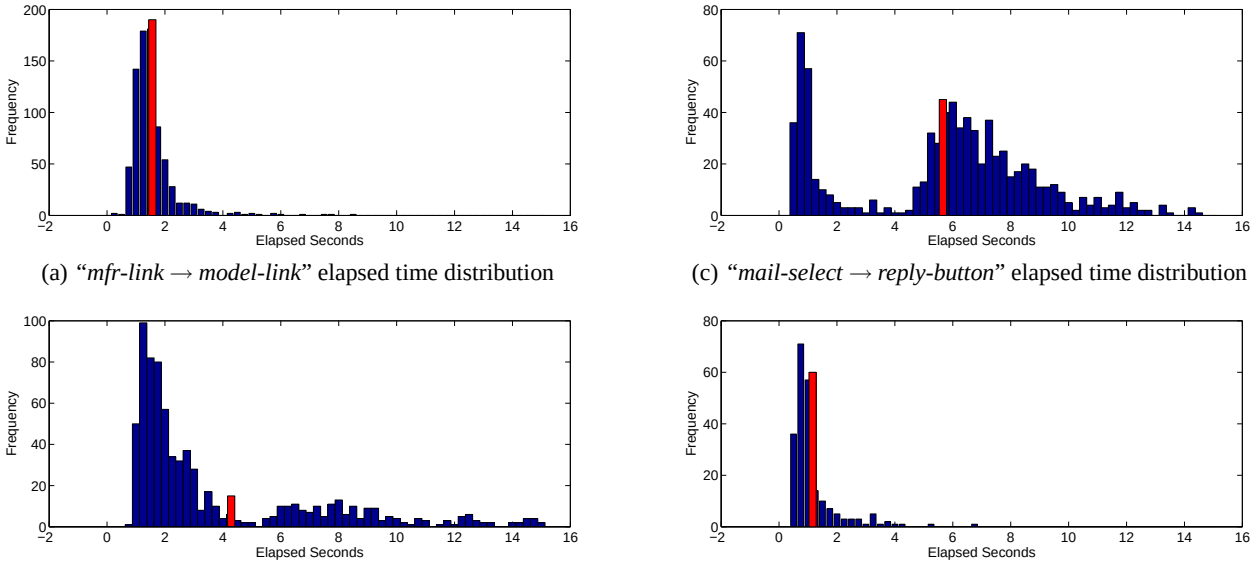
seems possible that pairs of consecutive events with high frequency represent strongly linked events and that task switching should not occur between them. This argument is supported when we compile a list of the most frequent consecutive event pairs and observe that one of the highest frequency consecutive event pairs is "*mfr-link* → *model-link*". It makes sense that we should link together these actions as they are the ordered steps required to look up a product's price. There should not be a drop in cognitive load after the "*mfr-link*" event since the model number is still required for the following "*model-link*" event and we should not expect task switching here. On the other hand, another high frequency consecutive event pair is "*reply-send* → *mail-select*". While these events appear to be strongly linked together, this pair actually does present a reasonable breakpoint. The "*reply-send*" event signals that a response email has been sent and a customer inquiry is completed. Handling a new customer inquiry is always marked by selecting a new mail from the list, or a "*mail-select*" event. It follows that the pair "*reply-send* → *mail-select*" is a task boundary and a drop in cognitive load should accompany it, making this a good breakpoint.

Mean Elapsed Time

A second attempt at identifying breakpoints involves considering the mean elapsed time between events. The hypothesis is similar to the frequency hypothesis: A low mean elapsed time between two events signals a strong link between them that should not be broken, while a large mean elapsed time between events indicates a weak link that may be broken when an interruption occurs.

For a given pair of events such as "*A*" and "*B*", it is not immediately clear how to construct the frequency distribution. We could look at all occurrences of "*A*" followed by a "*B*" any time thereafter, with the possibility of some events in between. This approach is appealing since it introduces some robustness to "noisy" user errors in the recorded events. We see some positive evidence supporting this choice in the distributions shown in Figure 2(a) and Figure 2(b). In both of these distributions, the mean of the histogram is indicated by a red (lighter) bar. The distribution shown in Figure 2(a) corresponds to the event pair "*mfr-link* → *model-link*", which as a sequence makes sense in a goal strategy and does not represent an expected boundary of cognitive subtasks. The mean of this distribution is about 1.56 seconds elapsed between the occurrence of the two events. The distribution shown in Figure 2(b) corresponds to the event pair "*model-link* → *reply-button*", which occurs when the user has completed the task of looking up the price of a product and is about to begin the process of responding to the inquiry. The mean of this distribution is 4.29 seconds of elapsed time between events. The larger mean found here supports the hypothesis, since this pair of events should straddle a subtask boundary and a drop in cognitive load should accompany it.

The idea of considering all occurrences of "*A*" followed some time later by "*B*" begins to break down, however, when we consider the distribution shown in Figure 2(c). This distribution corresponds to the elapsed time between the events "*mail-select*" and "*reply-button*". The mean elapsed time is 5.66 seconds, which seems to indicate that the events are not strongly linked. The problem with this assessment becomes clear when we take into consideration the variations in task strategies taken by different users. The consequence



(a) "*mfr-link* → *model-link*" elapsed time distribution

(c) "*mail-select* → *reply-button*" elapsed time distribution

(b) "*model-link* → *reply-button*" elapsed time distribution

(d) "*mail-select* → *reply-button*" distribution for one strategy

Figure 2: Distributions of elapsed time between pairs of events. The locations of the means are indicated by a red (lighter) bar.

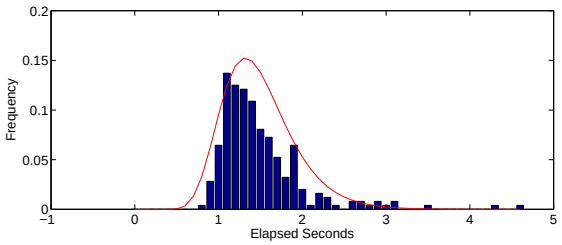
of this is that the sequence "*mail-select* → *reply-button*" is strongly linked together in several task strategies, but it is not found in all of them. This explains the two peaks seen in the histogram. The first peak (and the surrounding bins) correspond to the instances of the strategies which make use of the "*mail-select* → *reply-button*" sequence, while the second peak corresponds to the remaining strategies. Analyzing this sequence simply based on the mean of all of the possible occurrences does not provide a clear understanding of the data.

Addressing Multiple Strategies

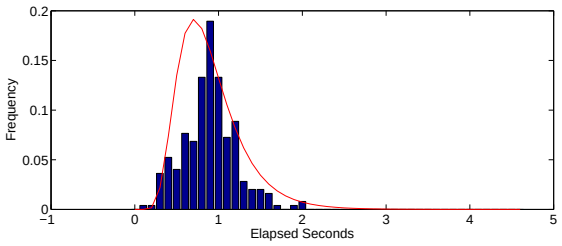
Regardless of the usefulness of the mean elapsed time in indicating the breakpoints, the observation concerning the multiple strategies needs to be addressed in any analysis of distributions. It seems that our distributions represent a classic example of a mixture distribution, which should lead us to consider a method such as expectation maximization (EM) (Moon, 1996) to fit a mixture model to the histogram. Once we've found a mixture model, we could then perform clustering to obtain only the instances of event sequences which should correspond to a single strategy. Another approach would be to use the T-Patterns method for identifying the critical interval (Magnusson, 2000) of elapsed time that we should consider acceptable for a given event pair. Both of these approaches present advantages and disadvantages for our data, and are likely to prove both useful and necessary in analyzing tasks containing variation in general.

We decided to use a much simpler approach to identifying the valid instances of a sequence. Based on the task that was assigned, we note that each task trial—the processing of a single email—must begin with a "*mail-select*" event to view the email. Furthermore, that once a new email has been selected, another "*mail-select*" event is very unlikely before this first email has been completely addressed. Following these assumptions, we can segment our raw event data stream into individual mail task instances by using each "*mail-select*" event as a boundary and consider unique sequences separately. This method is supported by Figure 2(d), where only the instances of the sequence "*mail-select* → *reply-button*" which are part of a strategy using those consecutive events are considered in the distribution. When compared to Figure 2(c) we now see a single a peak with a mean of 1.30 seconds versus two peaks and a mean of 5.66 seconds.

By considering instances of consecutive event pairs which are part of a particular observed task strategy, a lot of unexpected behavior in the elapsed time distributions is removed, but not enough to make the mean elapsed time a completely useful indicator of cognitive load or interruptibility. One reason for this lies in the simple nature of the data that was recorded. By comparing just the elapsed time between events "*A*" and "*B*", the analysis does not have at its disposal vital information about possible subtasks being performed. Consider once again the "*mfr-link* → *model-link*" sequence. Generally this sequence is observed when the user is looking up the price of a product for a customer inquiry. For one strategy which uses this sequence (actually all strategies must use this), we get a mean of 1.49 and a relatively large st. dev. of 0.59. Based on our hypothesis we should expect both a small mean and variance for such a strongly linked pair of events, but in fact we see a relatively large variance. This contradiction is explained when we consider that after a "*mfr-link*" event, a user completing this action is required to perform the relatively time-consuming task of reading through the list of model numbers to find the link for the model in question, before the "*model-link*" event can occur. A similar statement could be made about any event preceding the "*mfr-link*"



(a) "*model-link* → *reply-button*" distribution for one strategy



(b) "*reply-button* → *reply-type*" distribution for one strategy

Figure 3: Distributions for instances of pairs found in one strategy. The histograms are shown with a fitted log-normal distribution curve. Note that in 3(a) more of the mass falls in the far right end of its tail than in 3(b).

event. (More detailed data, such as eye-movement recordings, would further inform such an analysis—but again, such detailed data are not available in the general case.)

Tail Mass of Elapsed Time Distributions

Since basic statistics of our elapsed time distributions do provide an adequate signature with respect to multitasking breakpoints, we decided to take a closer look at the form of the distributions. When we compare the histogram distributions for different pairs of events, it becomes clear that certain histograms appear to have longer tails than others. To obtain a better picture of this, we could look at the amount of the histogram mass that falls several standard deviations to the right of the mean. We can also observe modeling the histogram with a normal distribution may not be the best choice, since there can be no negative elapsed times and typically the distributions exhibit an early peak followed by a right end tail. The log-normal distribution has these properties and we can easily find a maximum likelihood log-normal distribution to fit to our observations. Figure 3 shows two pair histograms that have been fitted with log-normal distributions. Figure 3(a) shows the distribution for the pair "*model-link* → *reply-button*", which corresponds to the boundary between the price lookup task and the email reply task and is a reasonable breakpoint. Figure 3(b) shows the distribution for the pair "*reply-button* → *reply-type*", which form consecutive events in the mail reply task and probably is not a good breakpoint. Notice that a significantly larger portion of the total observed mass in Figure 3(a) appears in the far right tail of the fitted distribution than does the mass in Figure 3(b). Another way to put it is that the "*model-link* → *reply-button*" distribution contains significantly more outliers than the "*reply-button* → *reply-type*" distribution.

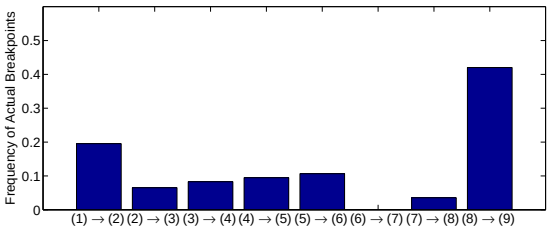
The hypothesis resulting from this analysis is that the amount of observed mass in the far end of the tails (outliers) of distributions of elapsed time between event pairs is a good indicator of the interruptibility between the events. We suspect that the underlying reason relates to people taking short mental breaks between these task steps: by resting for a short time (up to a few seconds) between actions, a person can mentally regroup for the next component of the task. It seems reasonable that such a mental regrouping would occur at higher-level task boundaries, or equivalently at places of low mental workload. Whatever the underlying reason, the tails of the distributions seem to serve as a good signature for multitasking breakpoints, as we detail in the next section.

To identify the outlier observations, we can simply fit the model to our observations and see how many observations fall n standard deviations to the right of the mean. Since we are specifically interested in outliers in the far right end of the tail, we should set n to be large, possibly $n = 3$ or 4. This simple method will certainly identify some outliers, but we can improve the method by performing it iteratively. In the iterative approach, we first fit the model, find the estimated std. dev., remove outliers n standard deviations from the mean from the distribution, and repeat. At each iteration the estimated mean will shift slightly to the left and we will consider more observations to be outliers. For large fixed n the estimates converge after a few iterations (i.e., no new outliers are found). At that point we have a good estimate of the percentage of the total observations which can be considered outliers.

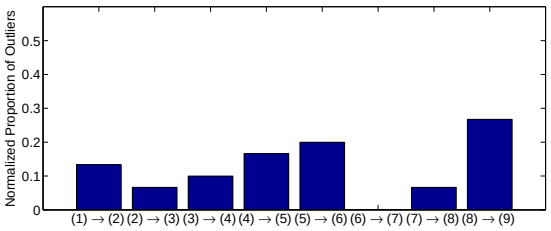
Results

To evaluate the outlier-based inference of multitasking breakpoints, we selected the events corresponding to the most frequently observed mail task strategy that we obtained from our data segmentation procedure. The complete sequence has the form: *mail-select*, *browser-focus*, *browser-home*, *mfr-link*, *model-link*, *reply-button*, *reply-type*, *reply-send*, *mail-select*. We calculated the outliers for each pair of consecutive events, and formed a normalized histogram of breakpoint likelihoods where the frequency of each bin is based on the number of outliers that were found. Our results were obtained using a log-normal distribution to fit our elapsed time distributions and a value of $n = 3.75$ standard deviations for the identifying outliers. Using the accompanying multitasking (mail and chat task) data, we also constructed a similar histogram of the actual deferred breakpoints that we taken by users while employing this strategy.

Both of the resulting histograms are shown in Figure 4. The inferred results match reasonably well to the observed breakpoints, $R = 0.83$. We obtained similar but not as good results using the normal distribution, and for several observed secondary strategy sequences.



(a) Observed Proportions of Breakpoints while Multitasking



(b) Proportions of Breakpoints Inferred from Single-Task Data

Figure 4: Comparison of actual breakpoints taken in (Salvucci & Bogunovich, 2010) with the outlier inferred breakpoints for the most frequent strategy: (1) *mail-select*, (2) *browser-focus*, (3) *browser-home*, (4) *mfr-link*, (5) *model-link*, (6) *reply-button*, (7) *reply-type*, (8) *reply-send*, (9) *mail-select*.

Discussion

To summarize, we found that the outliers (tails) of the distributions of time between task actions in a single task setting served as a good indicator of multitask breakpoints, were a secondary task to be introduced: The presence (or lack) of outliers in the tails of the distributions correlated well with people's tendency to switch away from a task between two given actions. These conclusions build on the results of (Bailey & Iqbal, 2008) which showed that users produce evidence of potential interruptibility in a single-task setting, but the proposed method was able to identify similar evidence using solely time and event data (rather than pupil-dilation or other data that may be more difficult to obtain). Our results suggest that when performing a task, users may occasionally take a short breaks (up to a few seconds) when a cognitive subtask is completed and before beginning a new subtask. Analysis based on this idea agrees well with multitask data from (Salvucci & Bogunovich, 2010) and hints at a strong relationship between distribution outliers and boundaries of cognitive subtasks.

Acknowledgments

This work was funded by ONR grant #N00014-09-1-0096.

References

Adamczyk, P. D., & Bailey, B. P. (2004). If not now, when?: the effects of interruption at different moments within task execution. In *Chi '04: Proceedings of the sigchi conference on human factors in computing systems* (pp. 271–278). New York, NY, USA: ACM.

Bailey, B. P., Adamczyk, P. D., Chang, T. Y., & Chilson, N. A. (2006). A framework for specifying and monitoring user tasks. *Computers in Human Behavior*, *22*(4), 709–732.

Bailey, B. P., & Iqbal, S. T. (2008). Understanding changes in mental workload during execution of goal-directed tasks and its application for interruption management. *ACM Trans. Comput.-Hum. Interact.*, *14*(4), 1–28.

Bailey, B. P., & Konstan, J. A. (2006). On the need for attention-aware systems: Measuring effects of interruption on task performance, error rate, and affective state. *Computers in Human Behavior*, *22*(4), 685–708.

Beatty, J. (1982). Task-evoked pupillary responses, processing load, and the sturcture of processing resources. *Psychological Bulletin*, *91*(2), 276–292.

Cutrell, E. B., Czerwinski, M., & Horvitz, E. (2000). Effects of instant messaging interruptions on computing tasks. In *Chi '00: Chi '00 extended abstracts on human factors in computing systems* (pp. 99–100). New York, NY, USA: ACM.

Iqbal, S. T., & Bailey, B. P. (2005). Investigating the effectiveness of mental workload as a predictor of opportune moments for interruption. In *Chi '05: Chi '05 extended abstracts on human factors in computing systems* (pp. 1489–1492). New York, NY, USA: ACM.

Iqbal, S. T., & Bailey, B. P. (2007). Understanding and developing models for detecting and differentiating breakpoints during interactive tasks. In *Chi '07: Proceedings of the sigchi conference on human factors in computing systems* (pp. 697–706). New York, NY, USA: ACM.

Magnusson, M. (2000). Discovering hidden time patterns in behavior: T-patterns and their detection. *Behavior Research Methods, Instruments and Computers*, *32*, 93–110.

Monk, C. A., Boehm-Davis, D. A., Mason, G., & Trafton, J. G. (2004). Recovering From Interruptions: Implications for Driver Distraction Research. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, *46*(4), 650–663.

Moon, T. (1996, Nov). The expectation-maximization algorithm. *Signal Processing Magazine, IEEE*, *13*(6), 47–60.

Salvucci, D. D., & Bogunovich, P. (2010). Monotasking and multitasking: the effects of mental workload on deferred task interruptions. In *Proc. CHI 2010*. ACM.

Salvucci, D. D., & Taatgen, N. A. (2010). *The multitasking mind*. New York: Oxford University Press.

Trafton, J. G., Altmann, E. M., Brock, D. P., & Mintz, F. E. (2003). Preparing to resume an interrupted task: effects of prospective goal encoding and retrospective rehearsal. *International Journal of Human-Computer Studies*, *58*(5), 583–603.