Performance of reactive interfaces in stimulus rich environments, applying formal methods and cognitive frameworks

Li Su and H. Bowman

Abstract

Previous research has developed a formal methods-based (cognitive-level) model of the ICS central engine, with which we have simulated attentional capture in the context of Barnard's key-distractor Attentional Blink task. The same core system would be at work when human operators interact with computer interfaces. Thus, we have used this model to evaluate the performance trade-offs that would arise from varying key parameters in Stimulus Rich Reactive Interfaces (SRRIs). The results of these evaluations are presented in this paper. A strength of formal methods is that they are abstract and thus, the resulting specifications of the operator are general purpose, ensuring that our findings are broadly applicable.

Introduction

In this paper we will be looking at a particular type of stimulus rich reactive interface (SRRIs), where the stimuli are presented rapidly (Wyble et al., 2006). In order to evaluate the performance of the interface, we have made the following assumptions. Firstly, we assume that all stimuli are presented for the same length; we call each stimulus an item. Secondly, we consider two presentation rates, 1) 120ms per item, which is similar to the 10 items per second rate of Barnard's key-distractor attentional blink task (Barnard et al., 2004), and 2) 240ms per item. Finally, items are either words or blanks. There are two types of words: target words and background words. The only difference between target and background words is that they belong to different categories. In our experiments, targets are jobs or occupations, e.g. waitress, and background words are things in the nature environment, e.g. river.

A fundamental problem in perceiving such stimuli in real-time is information overload. Humans have limited attentional resources, which ensure some stimuli are processed more extensively than others. In the case of SRRIs, human operators will

potentially miss some targets. This is supported by experimental evidences on spatial and temporal attention (Driver, 2001; Duncan, 2000; Barnard et al, 2004; Barnard et al, 2005, Wyble and Bowman, 2005). For example, Barnard et al. have discovered that human subjects may miss target words following key-distractors by a few hundred milliseconds (Barnard et al, 2004). This pheromon is known as the attentional blink (Raymond, Shapiro, & Arnell, 1992), which has been reported by many researchers in many different forms.

Many real world applications require both high reliability of detecting targets and also the ability to perceive information within a bounded time. These requirements lead to trade-offs when we vary some parameters in the interface. For example, one simple way to ensure that human operators have successfully perceived all targets is to ask them to fill in a "check list" at the end of the presentation. The system can then use this feedback to re-present the missed targets. This might even involve physiological feedback prediction of a users cognitive and perceptual state (Wyble, Craston & Bowman, 2006). However, there is a significant problem with this solution, i.e. the feedback step "pauses" the interface and delays the information to be processed. Alternatively, we can restrict the interface so that it avoids presenting targets, which can potentially interfere with other targets. We call the first solution a reactive approach, and the second a constructive approach. In this paper, we will concentrate on the second approach. A simple constructive approach is to separate targets by either blanks or background items, to ensure a target can be fully perceived before a further item arises. An obvious drawback is that such an approach also delays the information flow. Hence, it is necessary to explore these trade-offs during the design of efficient SRRIs.

Modelling SRRIs

In this section, we model the interface from the perspective of computer systems. We can use computer networks as a metaphor. Hence, a stimulus can be seen as a data packet and a stream of stimuli can be modelled as some type of traffic, e.g. network or disk I/O traffic. There are many models of traffic proposed by computer scientists, for example, the Poisson arrivals model or the self-similar model. We choose a particular traffic model called the *b*-model (Wang et al., 2002). It is a simple model with very few parameters, but it can generate self-similar and bursty traffic for a given time scale. The traditional Poisson arrival model is only bursty within a short time scale, but it will smooth out if it is applied to large time scales. However, the environment often has observable bursts on all time scales. Hence, self-similar models are desirable.

The b-model
There are a number of methods to generate self-similar traffic, such as the multiple ON/OFF source aggregation process (Wang et al., 2002). However, many of these are complex and hard to implement. In contrast, the *b*-model proposed by Wang et al. has relatively few parameters, e.g. the burstyness of the traffic can be characterised by a single parameter. Intuitively speaking, the *b*-model unevenly and randomly allocates events to two time periods with equal length. The model repeats this process until all events have been distributed. Wang et al. (2002) introduced the following algorithm to generate traffic using the *b*-model.

Inputs:

- Bias b, which determines the burstyness of the traffic. The traffic is least bursty when $b = 0.5$. It becomes more bursty when the bias moves towards 0 or 1.
- Aggregation level n, which determines the total number of time intervals or the length of the traffic, i.e. $l = 2^n$.
- Volume N, which denotes the total number of events.

Output: Number of events distributed in the 2^n time intervals following the b-model.

Algorithm: The generation of the traffic uses a binary tree and a stack is used to traverse the tree.

1) Push $(0, N)$ onto the stack.
2) If the stack is empty, stop.
3) Pop (k, v) from the stack. If $k = n$, output v and go to Step 2), else, move to the next step.
4) Flip a coin. If heads, push $(k+1, v \times b)$ and $(k+1, v \times (1-b))$, else, push $(k+1, v \times (1-b))$ and $(k+1, v \times b)$. Go to Step 2).

An example of its result is shown in Figure 1. A complete description of the model is presented in (Wang et al., 2002).
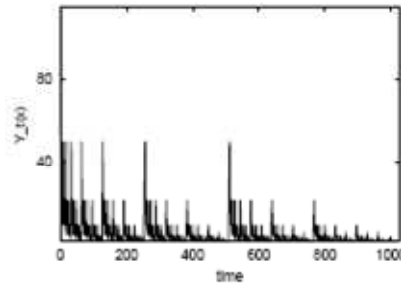
A sample b-model data

Figure 1 Self-similar traffic (Wang et al., 2002)

The System

We suggest that the traffic generated from the b-approach is a good model of how stimuli occur in the environment. One important feature of the traffic is that multiple events/stimuli can arrive simultaneously. A computer interface can display them in different spatial locations or display them by using different modalities. However, we assume that our computer system avoids this by using a buffer, which stores events and presents them to the user serially. This assumption restricts our research to the area of temporal attention, rather than to spatial or cross-modality attentional issues. This approach enables us to focus here on temporal issues, as a precursor to follow-up research addressing more complete simulations of human interaction.

Modelling Human Operators

Our previous research has developed a formal methods-based (cognitive-level) model of the ICS central engine, with which we have simulated attentional capture in the context of Barnard's key-distractor Attentional Blink task. The same core system would be at work when human operators interact with computer interfaces. The top level "box and arrow diagram" of our human model is shown in Figure 2, which is a subset of ICS (Barnard, 1985). The propositional subsystem (prop) and implicational subsystem (impic) are two of the central subsystems. The propositional subsystem considers one level of meaning, which encodes referentially specific relationships in semantic space. The implicational subsystem considers another level of meaning, which encodes an abstract schematic model. The source and the sink summarise the perceptual and response subsystems respectively. More details of this theory is presented in (Barnard, P.J. & Bowman, H., 2004) and (Bowman, H., Li Su & Barnard, P.J., 2006).
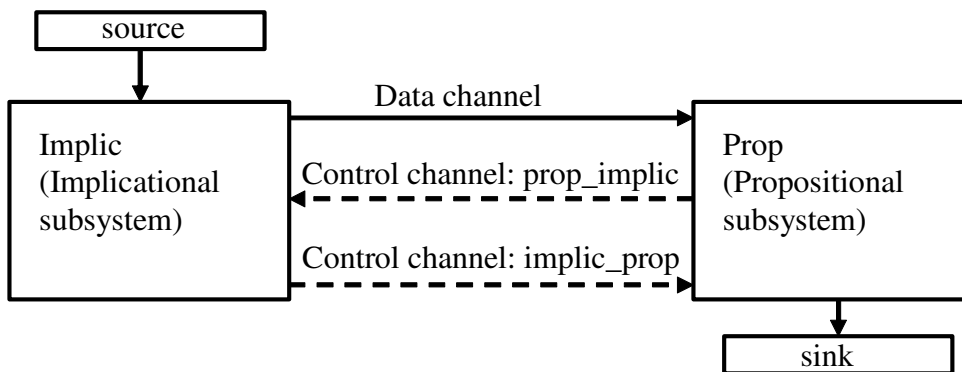

Figure 2 Top-level structure of the human model

The distinction between prop and implic is critical in the study of the attentional blink (Barnard, P.J. & Bowman, H., 2004; Bowman, H., Li Su & Barnard, P.J., 2006). Here though we evaluate the performance of the human computer interface. This performance is an emergent property of the entire system, which involves both the computer system and human operator. Hence, we will regard the human model as a whole and test its performance in conjunction with the computer model explained previously.

Accordingly, we will not discuss the interactions between implic and prop. Instead, we will view these as underlying mechanisms that are effectively hidden from the analysis we consider here. The only explicit knowledge available about the human operator will be the blink curve shown in Figure 3, which is an overall (global) property of the human operator. (In fact, we use this curve to derive a blink window size, which is used by the interface. We will explain this in more depth in the relevant sections.) The aim of this study is to investigate the change in performance after we have applied this explicit knowledge to the design of the interface. A strength of formal methods is that they are abstract and thus, the resulting specifications of the human operator are general purpose, ensuring that our findings are broadly applicable. For example, the models that were developed from the cognitive psychology perspective are used in this HCI context.
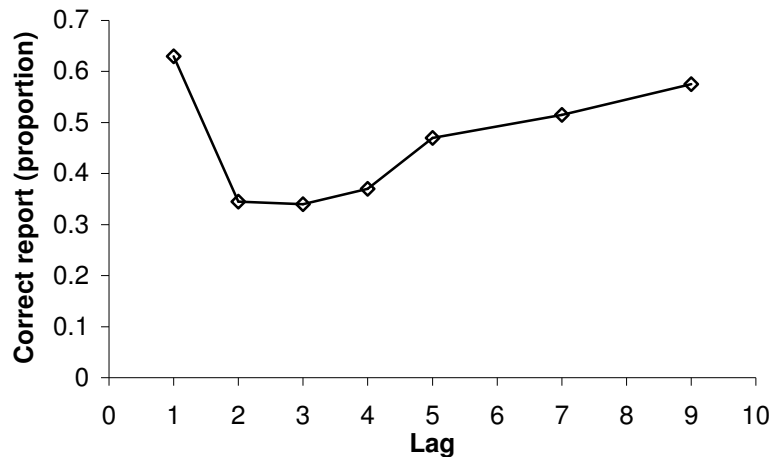
Figure 3 The basic blink curve (Barnard et al,. 2004). X-axis denotes the serial-position that the second target follows the first and Y-axis denotes accuracy as the probability of correct report of second target conditional on correct report of first target.

Performance Evaluation

In this section, we will vary the parameters in the *b*-model and measure the probability of detecting targets by the human model. In our experiments, the interface is only active for a bounded period, i.e. the length of 200 items. All items in the buffer will be discarded after this period. Studies of HCI have developed a large number of techniques to enhance the salience of the stimuli presented to human users. However, a fundamental problem is that the computer may not be confident of what information is actually important for the user. This is why the human operator's reaction/selection is essential. The human attentional mechanism results in an information bottleneck. So, it is not a good idea to overload the system with unnecessary warnings. In the next part of the paper, we will vary different parameters in our model and explore the consequences on performance.

Varying Number of Targets
In the first experiment, the presentation rate is 120ms/item, the aggregation level of the traffic is fixed at 6, i.e. the events are distributed to the first 64 (2 to the power of 6) time points. At each time point, the computer system inputs the events that occur. It stores those events in its buffer. If there is no event at a particular time point, the buffer will be filled with a blank item. At the same time, the computer outputs the items to the human user from its buffer. It is shown in Figure 4 that when there are a small number of targets and the traffic is not very bursty, the probability of detecting a target is 0.6, which is close to the baseline performance of the AB experiments. The performance then decreases as the number of targets increases. It eventually stabilises on a level between 0.2 and 0.3, which is close to the worst performance in AB experiments. Note that the performance does not decrease any more after 60 targets. The reason for this is that the buffer contains a maximum of 64 targets if the aggregation level is 6. Hence, increasing the number of targets will only force the targets beyond the aggregation level. As a result, the performance stabilises when the number of targets is more than 60. However, we can predicate that the performance will become worse when the number of targets is more than 200. If we double the

presentation length from 120ms to 240ms, we will observe very similar results (see Figure 5) except for a baseline shift. The reason for the shift is that increasing the presentation time will make the task easier.

When we decrease the *b* value, the traffic becomes burstier. Hence, the performance decreases too, since targets are closer to each other and thus compete more in time with each other for limited attentional resources. Most of the AB experiments have very small numbers of targets, i.e. normally less than three. However, in the case of SRRIs, the human user may have to cope with much larger numbers of targets. This may potentially exceed the working memory capacity, and impair performance of the system. Note, during our simulations, we have ignored working memory capacity. It indeed may have a dramatic influence on the probability of reporting targets.
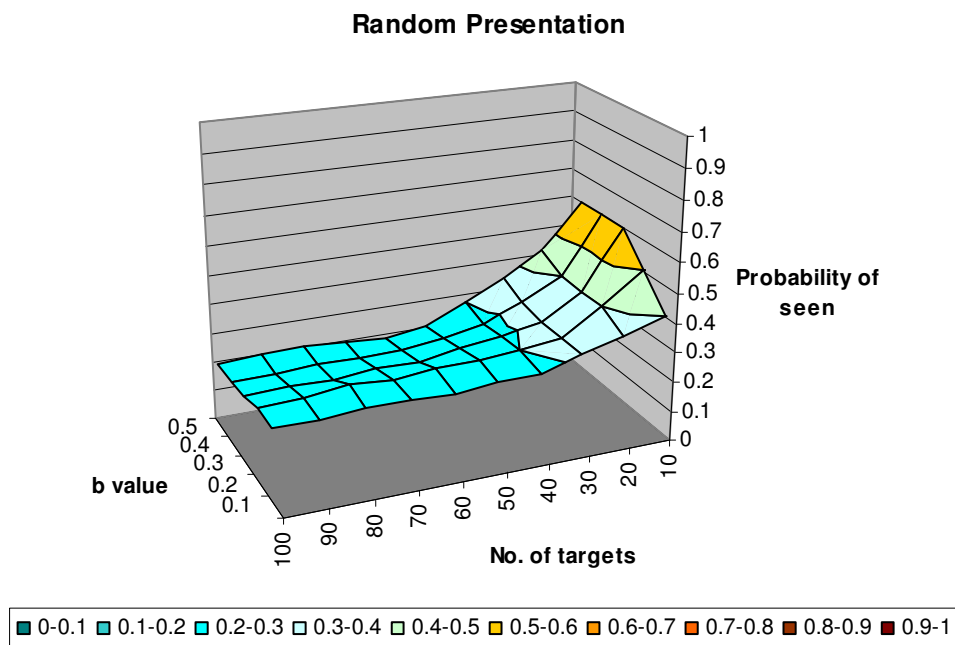

Figure 4 Probability of detecting a target with presentation rate of 120ms/item (Non-AB aware condition). A smaller b-value indicates increasing burstyness.
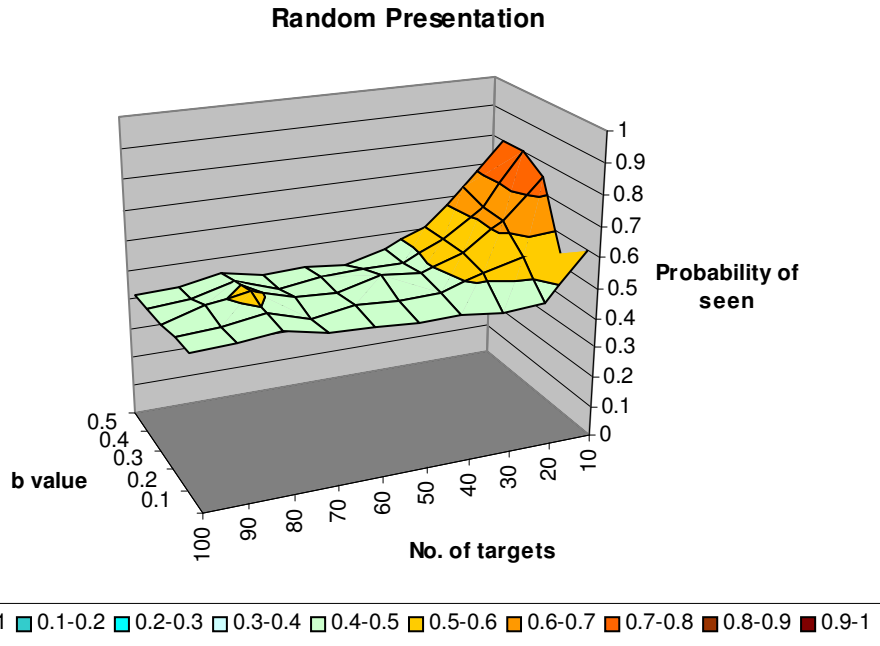
Random Presentation

Figure 5 Probability of detecting a target with presentation rate of 240ms/item (Non-AB aware condition). Reducing the b-value represents increased burstyness.

From the study of temporal attention and AB experiments (Barnard et al., 2004), we have learned that the processing of previous targets impairs detection of the current target. Hence, a straightforward approach to improve performance is to separate the targets, so that their interference is minimised. We have also learned that performance is maximally impaired during lags 2, 3, and 4 (i.e., the serial-position that the second target follows the first) when the presentation rate is around 100ms/item. So, we should design our interface to avoid presenting targets during this blink window. We also do not want to take advantage of lag-1 sparing, since this only occurs when the SOA is around 100 ms and such SOAs are not typical outside the laboratory setting. So, in the case of SRRIs, it may not be possible to extract meaning in this case.

We have designed an interface, which does not present any two targets within the blink window, as just defined. Figure 6 shows the probability of detecting targets using such an AB aware interface. The blink window size is set to the length of 5 items. Figure 7 shows the difference in performance between the two systems. We find that, in general, performance is indeed improved when we make the system blink aware. There is little difference between the performances of these two systems when the number of targets is small. This is because, due to random perturbations, the targets are (in general) already well separated in the Non-AB aware condition and thus, the AB aware system changes the presentation stream very little. The AB aware system becomes more beneficial when we increase the number of targets. However, the performance of the AB aware system starts to decrease after the number of the targets gets more than 40. This is because the AB aware system is not able to present 40 targets within the total time bound of 200 items.
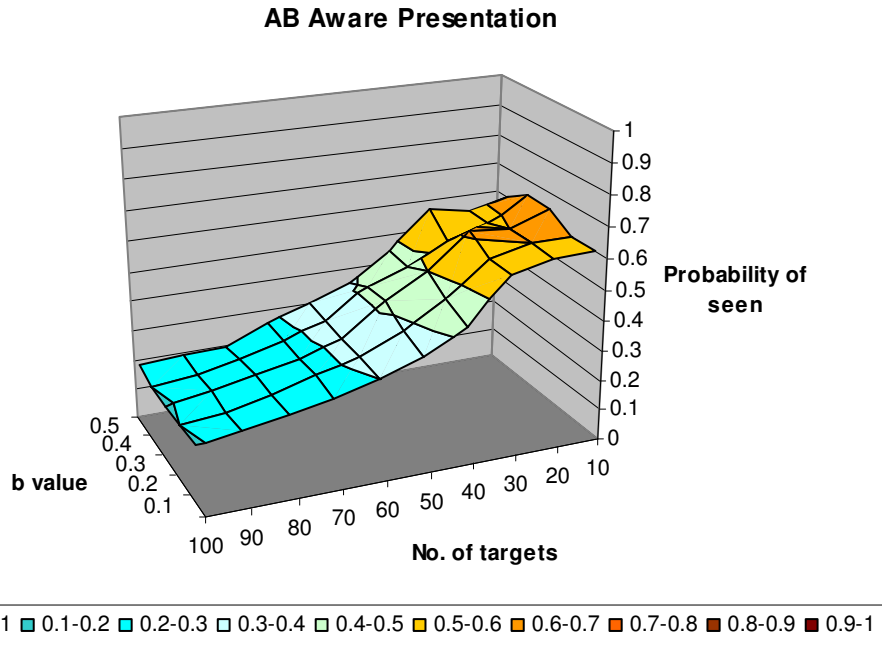
AB Aware Presentation

0-0.1 ■ 0.1-0.2 ■ 0.2-0.3 □ 0.3-0.4 □ 0.4-0.5 □ 0.5-0.6 ■ 0.6-0.7 ■ 0.7-0.8 ■ 0.8-0.9 ■ 0.9-1

Figure 6 Probability of detecting a target with presentation rate of 120ms/item (AB aware condition). Burstyness increases as b-value decreases.

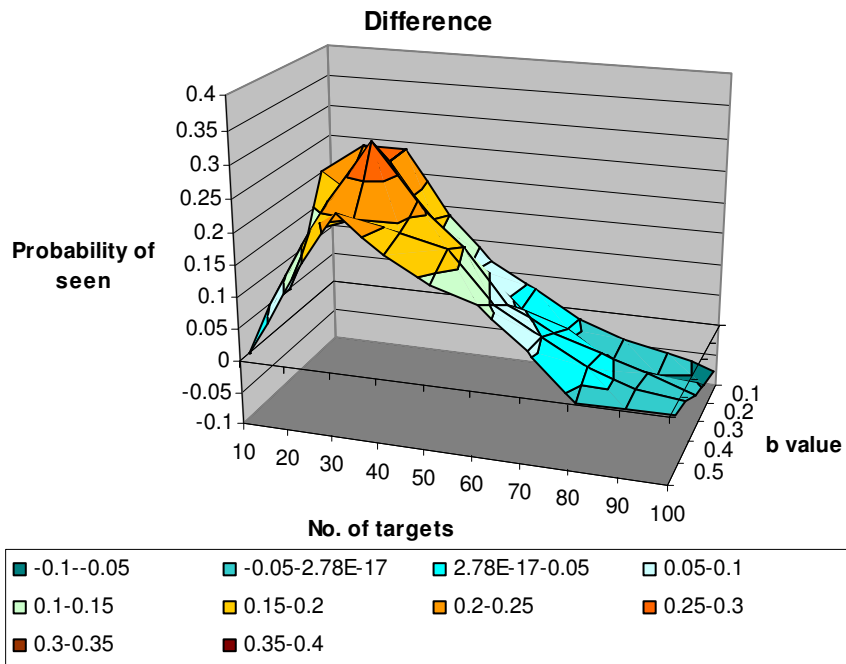**Difference**

■ -0.1--0.05   ■ -0.05-2.78E-17   ■ 2.78E-17-0.05   □ 0.05-0.1
□ 0.1-0.15   □ 0.15-0.2   ■ 0.2-0.25   ■ 0.25-0.3
■ 0.3-0.35   ■ 0.35-0.4

Figure 7 Performance difference between AB aware and Non-AB aware system with presentation rate of 120ms/item. Burstyness increases as b-value decreases.

Varying window size varies the value of the AB aware system Figure 6. We find that performance for the AB aware system is not dramatically influenced by changes in burstyness (expressed using the *b* value). Hence, we average results across different *b* values. Figure 8 shows the performance of the Non-AB aware system (which could be seen as a special case of the AB aware system with a blink window size 0) and AB aware systems using different blink window sizes. There is a trade-off in the AB aware system, i.e. if we want to ensure that the human user perceives as many targets

as possible, we should increase the blink window size. However, a large window size (for example, 10) will only improve the performance when the umber of targets is small. The performance will become slightly worse with a larger number of targets.
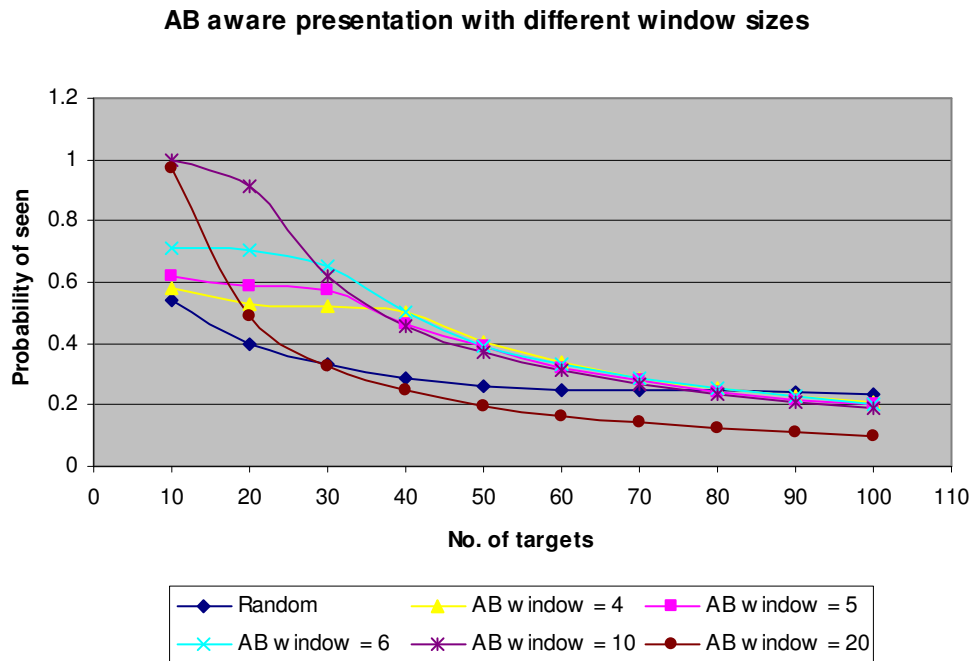
AB aware presentation with different window sizes

Figure 8 Probability of detecting a target using AB aware system with different window sizes. Random denotes the non-blink aware system.

We will observe very similar results in Figure 9 and Figure 10 if we double the presentation length from 120ms to 240ms and set the window size to 5[1]. We can also see a baseline shift, which partly arises for the same reason. However, there is an additional reason. This is because the length of 5 items is 1200ms when the presentation rate is 240ms/item, and it is 600ms when the presentation rate is 120ms/item. In other words, targets are further apart when the presentation rate is slow. Hence, the trade-off between perceiving targets accurately and also perceiving as many as possible is more pronounced now.

---

[1] In the rest of the paper, we use 5 as the default size for the blink window.

AB Aware Presentation

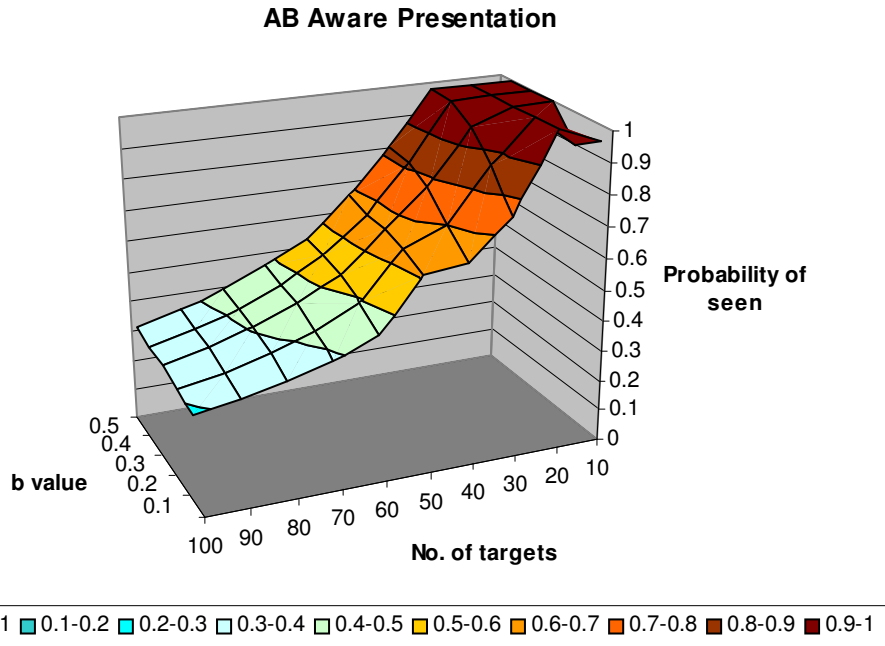Legend: 0-0.1 ◼ 0.1-0.2 ◻ 0.2-0.3 ◻ 0.3-0.4 ◻ 0.4-0.5 ◻ 0.5-0.6 ◻ 0.6-0.7 ◼ 0.7-0.8 ◼ 0.8-0.9 ◼ 0.9-1

Figure 9 Probability of detecting a target with presentation rate of 240ms/item (AB aware condition). Burstyness increases as the b-value decreases.
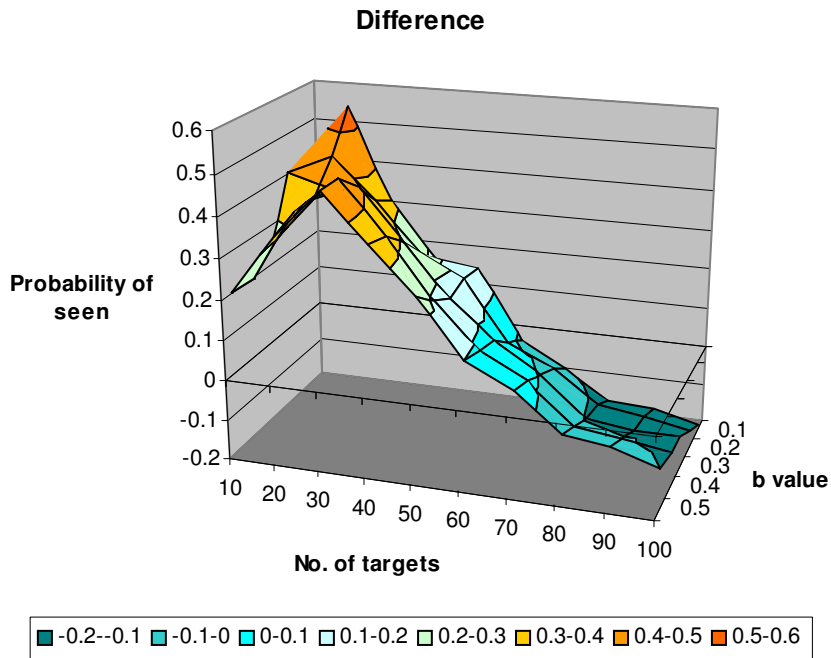
Difference

Legend: -0.2--0.1 ◼ -0.1-0 ◻ 0-0.1 ◻ 0.1-0.2 ◻ 0.2-0.3 ◻ 0.3-0.4 ◼ 0.4-0.5 ◼ 0.5-0.6

Figure 10 Performance difference between AB aware and Non-AB aware system with presentation rate of 240ms/item

Varying Aggregation Level

In this section, we fix the number of targets to 20 and vary the aggregation level. The presentation rate is set to 120ms/item in this experiment. As is shown in Figure 11 the performance improves as we increase the aggregation level until a level of 8 is reached. This is because targets are better separated when the aggregation level is high. We can also see that performance is impaired when the *b* value decreases. This is consistent with our previous experiment. Note that performance does not improve continuously as the aggregation level increases. In particular, it starts to drop after the

10

aggregation level 8, since our simulation only presents the first 200 items (2 to the power of 7.645 is nearly 200) to the user and any item that arrives later will be discarded. When we change the presentation rate to 240ms/item, we obtained similar results, see Figure 12, except for a baseline shift.

Random Presentation
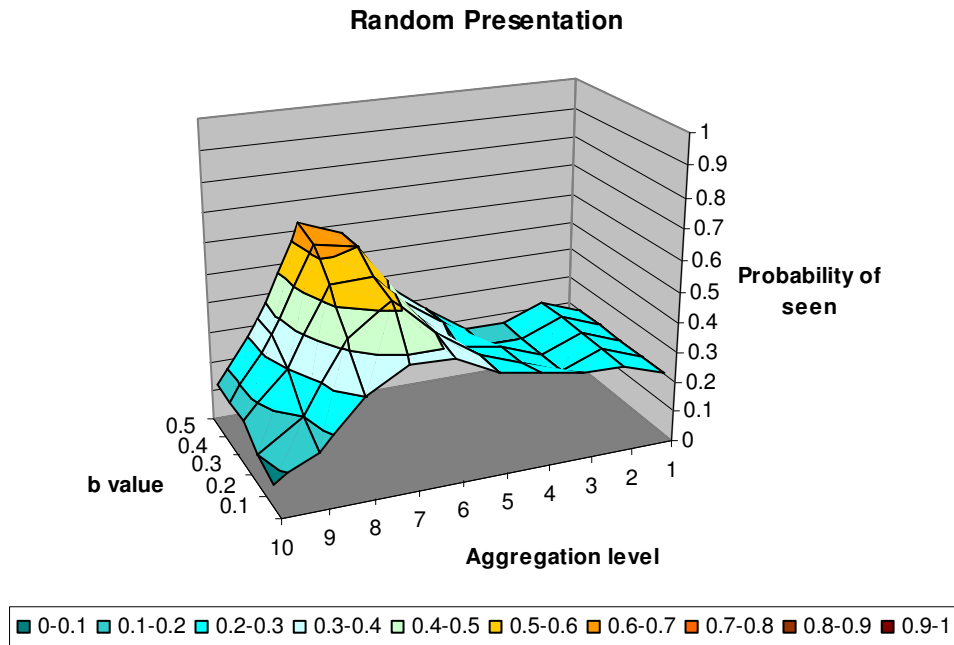

Figure 11 Probability of detecting a target with presentation rate of 120ms/item (Non-AB aware condition). Burstyness increases as the b-value decreases.
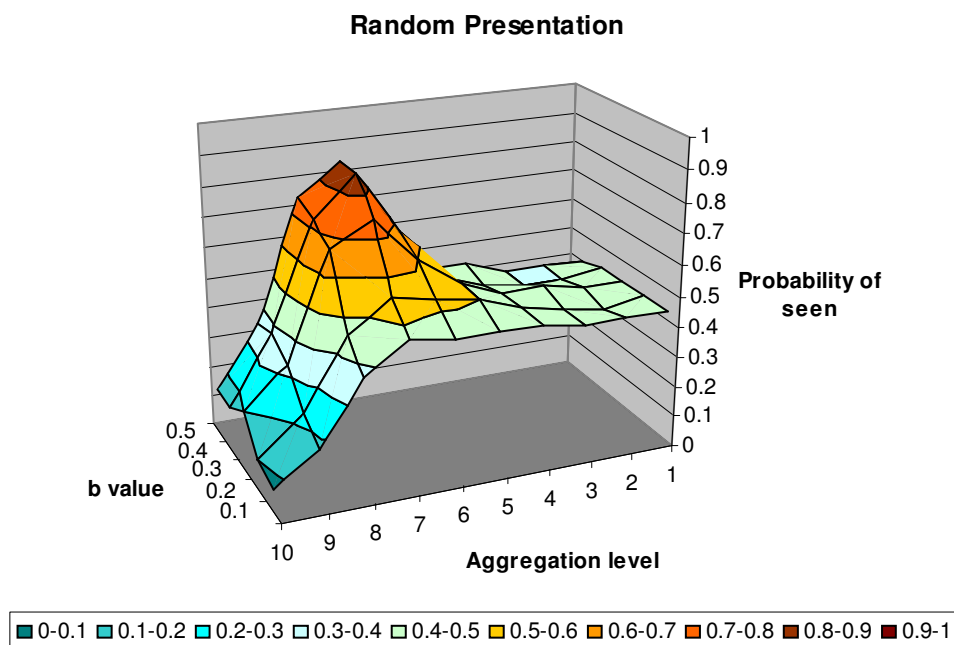
Random Presentation

Figure 12 Probability of detecting a target with presentation rate of 240ms/item (Non-AB aware condition. Burstyness increases as the b-value declines.

We use the same AB aware system (blink window size 5) introduced in the previous section and evaluate its performance again. We can observe in Figure 13 that

11

performance improves before an aggregation level of 8, but it gets worse after that. The reason for this is the same as with the Non-AB aware system, i.e. some items arrive too late to be presented. This trade-off between accuracy and urgency occurs in almost every one of our experiments. Figure 14 shows the difference between the Non-AB aware and AB aware systems. It can be seen that beyond aggregation level 8, the performance is worse in the AB aware system than in the Non-AB aware system, since the AB aware system attempts to separate the targets. Hence, targets are more likely to arrive too late to be presented.
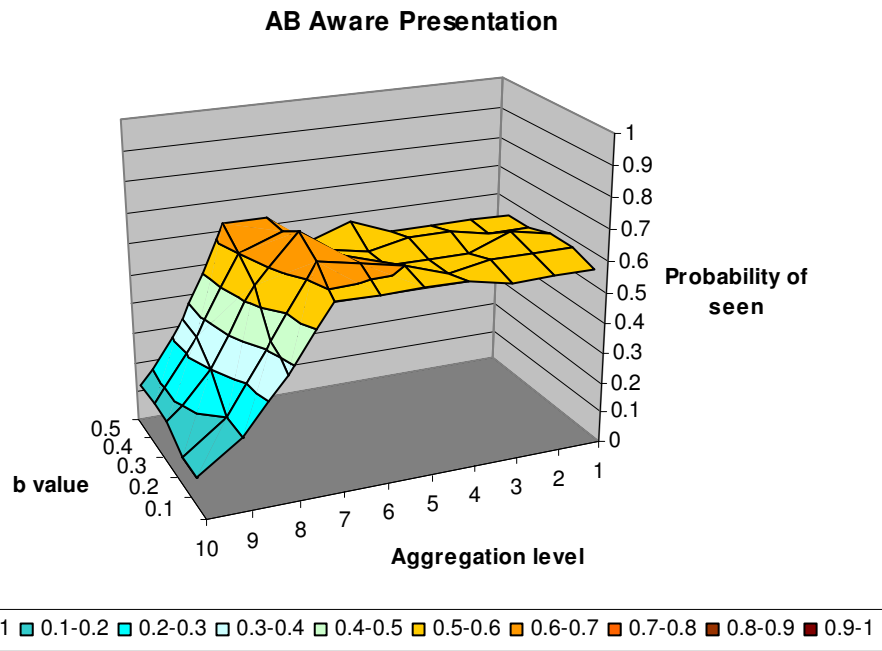
AB Aware Presentation

Figure 13 Probability of detecting a target with presentation rate of 120ms/item (AB aware condition. Burstyness increases as b-value declines.
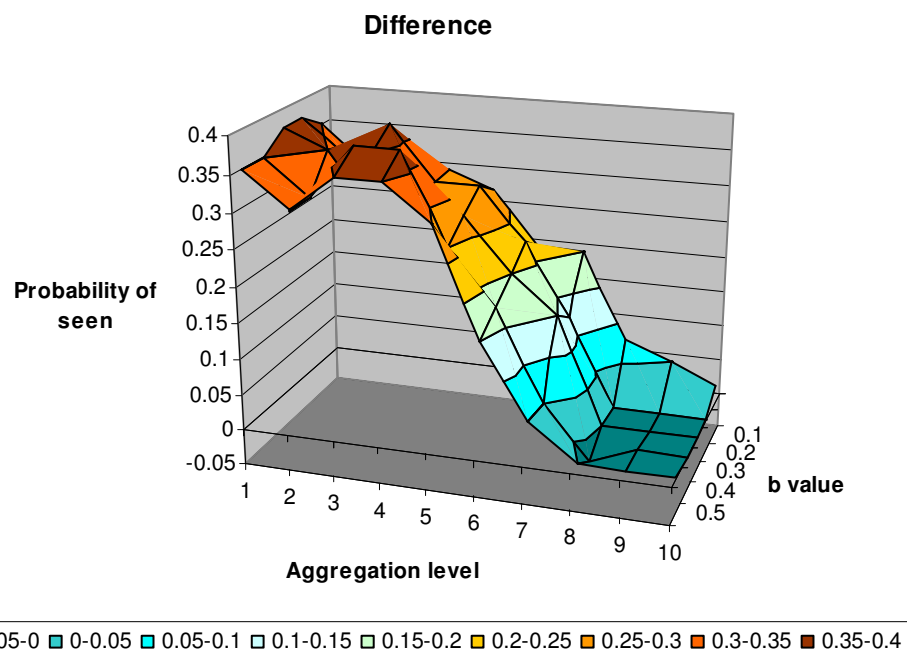
Difference

Figure 14 Performance difference between AB aware and Non-AB aware systems with presentation rate of 120ms/item. Burstyness increases as b-value declines.

As shown in Figure 15 and Figure 16, we have obtained similar results when the presentation rate is 240ms/item.
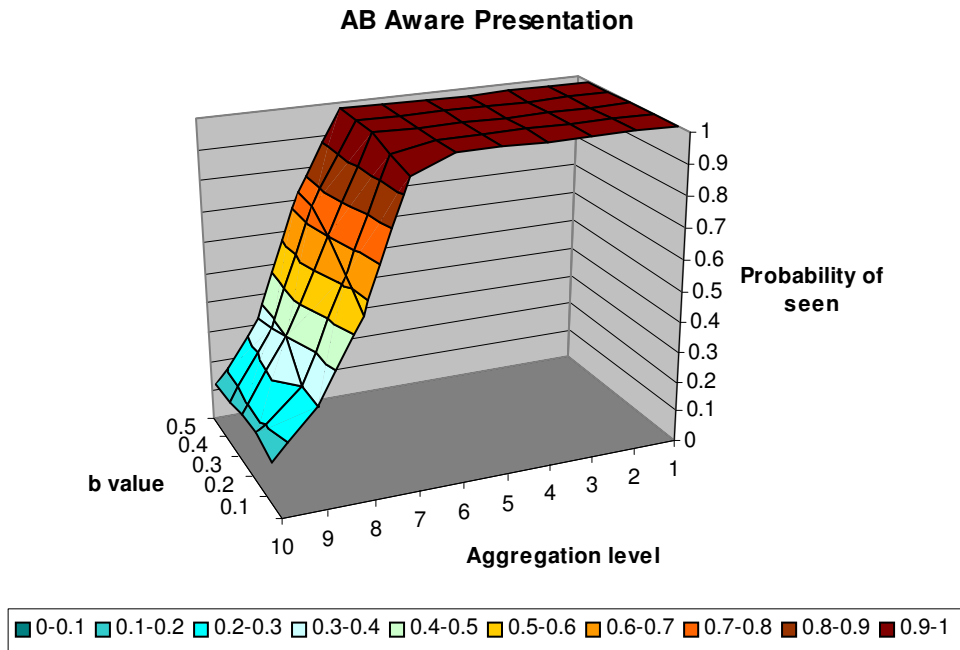
AB Aware Presentation

Figure 15 Probability of detecting a target with presentation rate of 240ms/item (AB aware condition). Burstyness increases as b-value declines. Note, performance is at ceiling at aggregation levels up to 7.
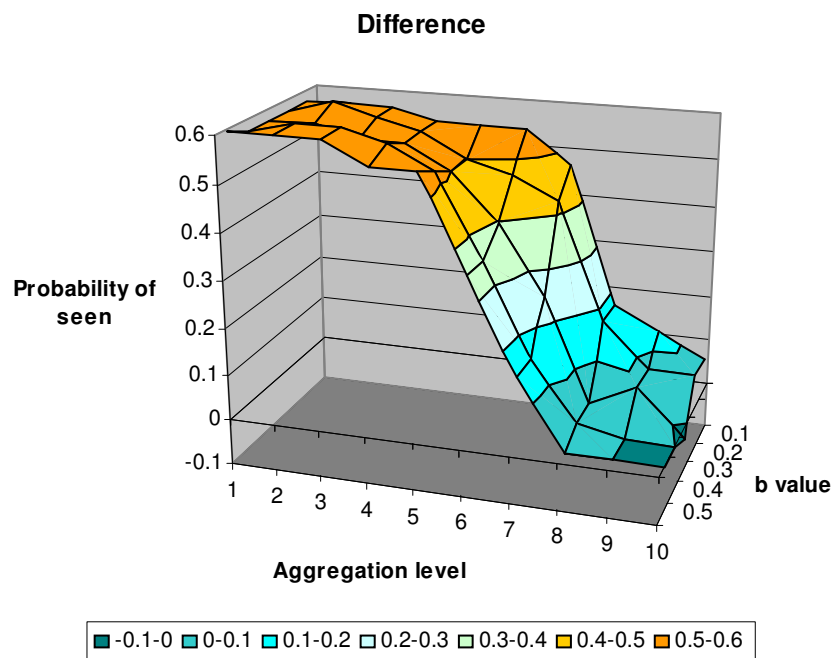
Difference

Figure 16 Performance difference between AB aware and Non-AB aware systems with presentation rate of 240ms/item. Burstyness increases as b-value decreases.

Varying Traffic Length

In this section, we increase, at the same speed, both the number of targets and the aggregation level. As a result, the only two variables in the traffic are the length and the burstyness. As is shown in Figure 17, this traffic gives a very unstable performance when the length of the traffic is very short, since missing one target can have dramatic effects on performance. However, intermediate lengths show stable performance. As discussed in previous sections, performance declines when the aggregation level is bigger than 8.

Random Presentation

0-0.1 ■ 0.1-0.2 ■ 0.2-0.3 □ 0.3-0.4 □ 0.4-0.5 □ 0.5-0.6 ■ 0.6-0.7 ■ 0.7-0.8 ■ 0.8-0.9 ■ 0.9-1
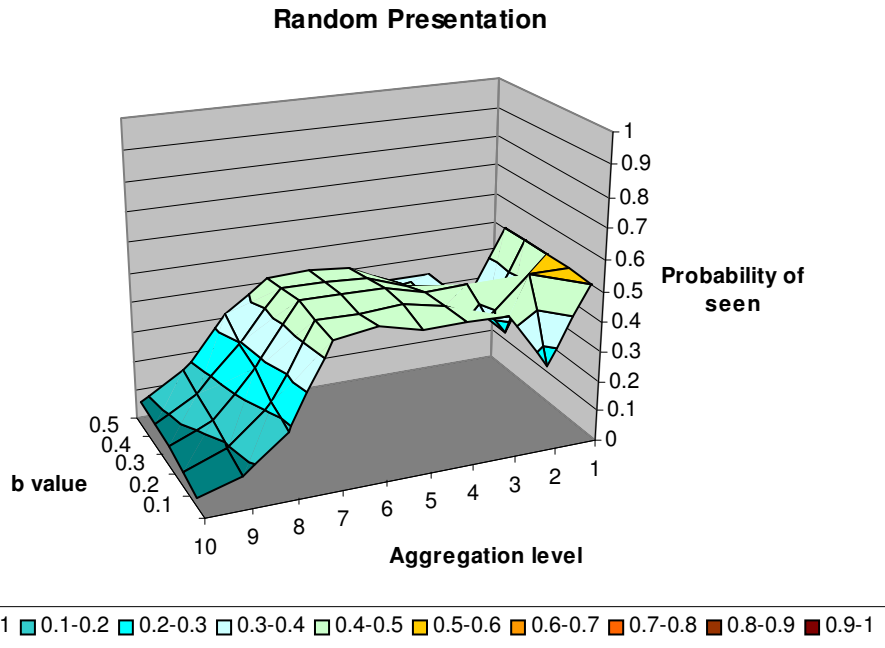
Figure 17 Probability of detecting a target with presentation rate of 240ms/item (Non-AB aware condition). The number of targets increases with the aggregation level. So, the overall effect is that the length of the traffic increases witht the affregation level and the rate of traffic is constant. Burstyness increases as the b-value declines.

Figure 18 and Figure 19 show that the AB aware system improves performance consistently for the short and intermediate traffic length. However, the performance of the AB aware system is worse than the Non-AB aware system for long traffic lengths. This demonstrates a clear trade-off between accuracy and urgency. Finally, there is no significant effect of burstyness.
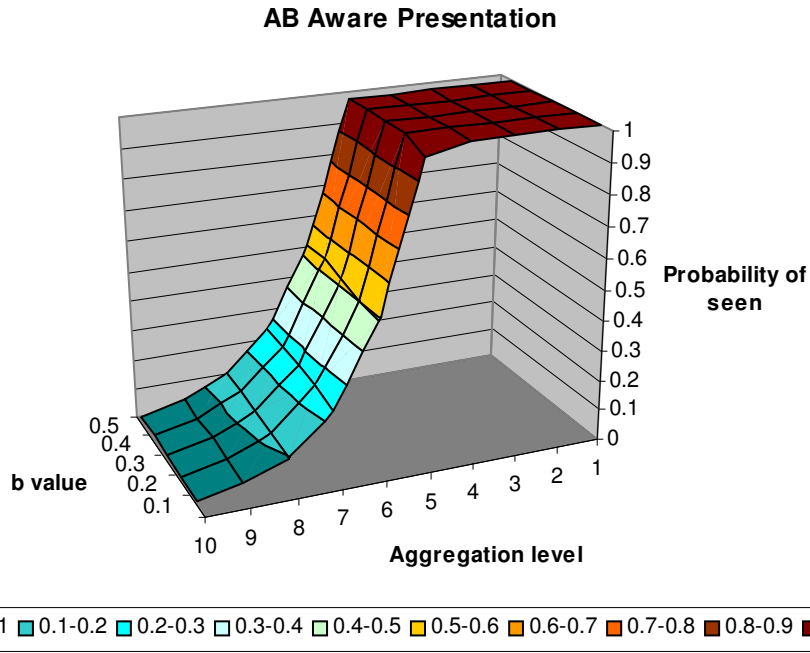
14

AB Aware Presentation

Figure 18 Probability of detecting a target with presentation rate of 240ms/item (AB aware condition). Burstyness increases as the b-value declines.
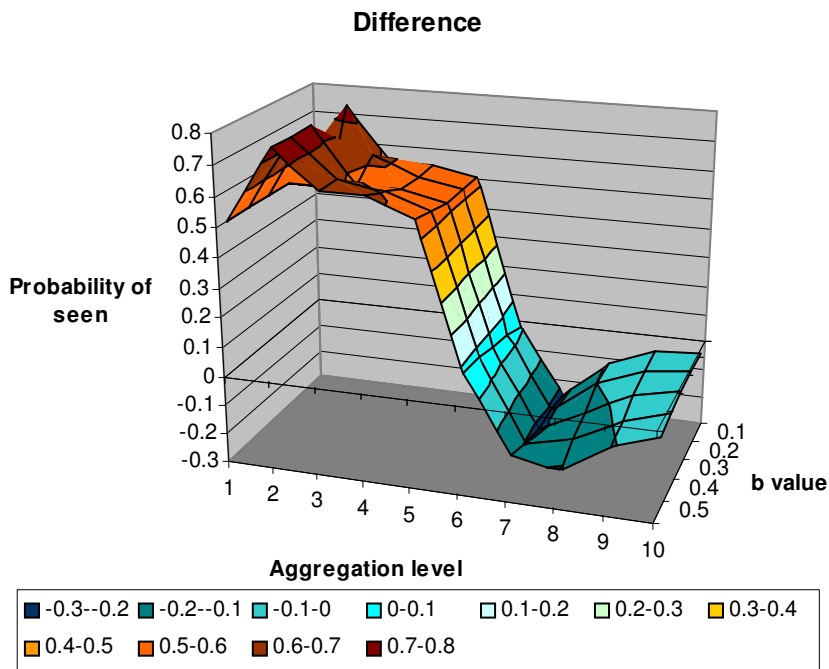
Difference

Figure 19 Performance difference between AB aware and Non-AB aware systems with presentation rate of 240ms/item. Burstyness increases as the b-value declines.

Conclusion

We have evaluated the performance of SRRIs in the context of temporal attention capture. Based on studies of the attentional blink phenomenon, we have designed an interface, which is called the AB-aware system. It attempts to avoid presenting target items when the human operator is not ready to perceive them. We have varied a

number of parameters in the stimulus stream, which is generated using the *b*-model. Then we evaluated the performance trade-offs arising in such an AB-aware system. As a result, we have discovered a number of properties of AB-aware interfaces.

- They can reduce the effect of burstyness by smoothing traffic. In a sense, they buffer stimuli until the user is ready to perceive them. Although this is done in a predictive (constructive) manner, i.e. the system predicts when the user will be ready to perceive an item.
- They can improve the probability of detecting targets when urgency is ignored.
- Their disadvantage is that they often delay the presentation of targets and breakdown absolute timing of the events.

Future research may consider different approaches to improve SRRIs, for example, using EEG feedback as an acknowledgement from the human operator. This approach is introduced in (Wyble et al, 2006). Such EEG feedback cannot always accurately predict that the user has perceived the targets. Hence, we should extend our existing models to produce artificial EEG signals and use these signals to guide the interface. We will also evaluate how accurately the acknowledgement should be in order to make such an approach worthwhile.

Reference

Barnard, P. (1985). Interacting Cognitive Subsystems: A Psycholinguistic Approach to Short-term Memory. In A. Ellis (Ed.), *Progress in the Psychology of Language, Vol. 2* (pp. 197-258). Hove, UK: Lawrence Erlbaum Associates Ltd.

Barnard, P.J. & Bowman, H. (2004). Rendering Information Processing Models of Cognition and Affect Computationally Explicit: Distributed Executive Control and the Deployment of Attention. *Cognitive Science Quarterly*, 3(3), 297-328.

Barnard, P. J., Scott, S., Taylor, J., May, J., & Knightley, W. (2004). Paying attention to meaning. *Psychol Sci, 15*(3), 179-186.

Barnard, P. J., Ramponi, C., Battye, G., & Mackintosh, B. (2005). Anxiety and the Deployment of Visual Attention over Time. *Visual Cognition, 12*(1), 181-211.

Bowman, H., Li Su & Barnard, P.J. (2006). Semantic Modulation of Temporal Attention: Distributed Control and Levels of Abstraction in Computational Modelling. Technical Report 9-06, Computing Lab, University of Kent at Canterbury.

Driver, J. (2001). "A selective review of selective attention research from the past century." British Journal of Psychology **92**(1): 53-78.

Duncan, J. (2000). Visual attention in mind and brain. Brain mechanisms of perception, learning and memory: Essays for Gabriel Horn. J. J. Bolhuis. Oxford, Oxford University Press**:** 49-68.

Raymond, J. E., Shapiro, K. L., & Arnell, K. M. (1992). Temporary Suppression of Visual Processing in an RSVP Task: An Attentional Blink. *Journal of Experimental Psychology: Human Perception and Performance, 18*(3), 849-860.

Wang, M., Madhyastha, T., Chan, N. H., Papadimitriou, S. & Faloutsos, C. (2002). Data Mining Meets Performance Evaluation: Fast Algorithms for Modeling Bursty Traffic. *18th International Conference on Data Engineering*.

Wyble, B., & Bowman, H. (2005). Computational and experimental evaluation of the attentional blink: Testing the simultaneous type serial token model. In Bruno G. Bara, Lawrence W. Barsalou, and Monica Bucciarelli, editors, *CogSci 2005, XXVII Annual Conference of the Cognitive Science Society*, pages 2371-2376. Cognitive Science Society, Cognitive Science Society through Lawrence Erlbaum.

Wyble, B., Craston, P., & Bowman, H. (2006). Electrophysiological feedback in adaptive human computer interfaces. Technical Report 8-06, Computing Lab, University of Kent at Canterbury.