An Initial Cognitive Model of a Radar Detection Task

Alexander R. Hough (alexander.hough.1@us.af.mil)
Christopher Stevens (christopher.stevens.28@us.af.mil)
Elizabeth Fox (elizabeth.fox.9@us.af.mil)
Christopher Myers (christopher.myers.29@us.af.mil)

Air Force Research Laboratory
Wright-Patterson AFB

Abstract

In adversarial operational environments like radar monitoring, humans have to monitor large amounts of information, multitask, and manage threats. They may also face electronic disruption or attacks aimed at degrading radar monitor effectiveness (a.k.a electronic warfare or EW). In these settings, it is unclear how frequent changes in personnel, training, and updates to visual displays affect an operator's readiness. A recent experiment used an analogous radar monitoring task to investigate effects of display density and electronic warfare on an operator's threat detection performance. Here, we present a cognitive model capable of completing a scaled-down version of that task to better understand the experimental results and underlying cognitive processes. Similar to the human experiment, our cognitive model completed conditions comprised of changes to the nature of the task(s), the number of targets to track, and the presence or absence of distractors, deemed 'friendlies'. Although this initial cognitive model uses primarily default ACT-R parameters, it was able to capture patterns in human performance across conditions. We present the results and discuss limitations to address in future work.

Keywords: Cognitive model; ACT-R; Multiple object tracking; Multitasking

Introduction

Many tasks in modern environments require individuals to maintain awareness of complex, evolving situations. For example, air-traffic-control, lifeguarding, and childcare are all situations in which an overlooked detail or event can lead to serious consequences. It is important to understand the way in which people maintain awareness in these complex situations, and the circumstances under which that awareness will be impaired. Here we consider a particular variety of radar monitoring that occurs in naval operational environments. In these settings, operators must monitor visual displays with many entities (tracks) and identify certain tracks for follow on action. These situations often involve large amounts information, multitasking, and continuous decision making. Operators rely on systems like the Aegis Combat System (i.e., ACS) that continuously update visual displays with information from multiple sources (Bath, 2020). Appropriate representation of entity types (e.g., hostiles and friendlies) in such systems is important for correctly identifying threats to avoid accidents (Pogue, 2016) and it is not clear how frequent changes to the system and training affects monitor's readiness (Fisher & Kingma, 2001).

Visual search literature provides constraints (Treisman & Gelade, 1980; Glavan, Haggit, & Houpt, 2020) and models (Wolfe, 2021; Nyamsuren & Taatgen, 2013; Fleetwood & Byrne, 2006) for representing human-like visual search. However, dynamic stimuli are out of scope for most models. Here, we present a cognitive model capable of completing a laboratory radar monitoring task with dynamic stimuli. We show how well it captures human performance and discuss its limitations to address in future work.

MOT-EW Task

The MOT-EW task (Fox et al., 2023) served as an AEGIS analog to investigate human performance in a laboratory setting. The multiple object tracking (MOT) task (Figure 1) involved four quadrants with moving hostiles and friendlies. Hostiles (i.e., targets) were presented as red circles and friendlies (i.e., distractors) were comprised of octagons and diamonds that were pink or magenta. Objects had protruding black track lines indicating the direction they were moving. Hostiles were slightly larger and had longer track lines.

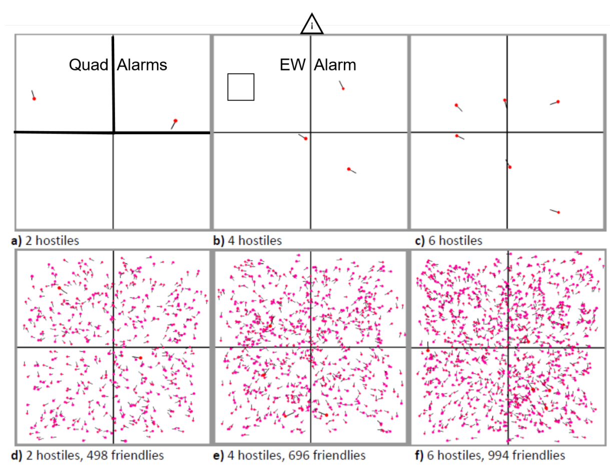

Figure 1: Depiction of the MOT-EW task, alarm states for quadrants, and EW attacks from Fox et al. (2023).

The task involved turning on quadrant alarms when at least one hostile was present and turning off alarms when hostiles were no longer present in that quadrant (See Figure 1a). The electronic warfare (EW) task used the same stimuli, but involved a hostile disappearing for 1-4 seconds once during 30 second windows. An EW alarm was turned on after a hostile had disappeared and off when it returned (see Figure 1b). In Fox et al. (2023), participants completed 18 conditions with three independent variables: number of hostiles (i.e., 2, 4, and 6), presence of friendlies (i.e., yes or no), and task (MOT, EW, or MOT-EW). Each condition was completed in a separate session that lasted approximately 12 minutes. Our goal was to develop a cognitive model capable of capturing human behavior in all of these 18 conditions.

MOT-EW Model

We implemented a cognitive model, we simply call the MOT-EW model, in the ACT-R cognitive architecture (Anderson, 2007). ACT-R includes both symbolic and sub-symbolic structures, and modules that represent systems of the mind. The MOT-EW model uses the goal, vision, motor, and procedural modules. The goal module serves as the models focus and stores goal relevant information. The vision module allows the model to perceive visual stimuli and direct attention. The motor module allows the model to turn on/off alarms using a keyboard. The procedural module uses condition-action rules (i.e., productions) to represent knowledge about how to do things and to drive the behavior of the model.

Our approach was to construct the simplest model without modifying parameters to test the "out-of-the-box" capability of ACT-R to complete the MOT-EW task. However, we had to make three modifications so the model could reasonably perform the task: 1) We modified the experimental task, 2) changed one parameter, and 3) deviated from typical visual perception methods allowing the model to reach and maintain human-like performance. We start by explaining the MOT-EW model task and then describe the model.

Model Task

Our overall goal for designing the model was to remain as faithful as possible to the experimental task, and to default ACT-R assumptions and design patterns. However, due to several constraints we had to make changes to both the virtual version of the task and the model. The first change we made to the task was reducing the number of friendlies on the screen. We reduced the number of friendlies for two reasons (Figure 2). First, if we included several hundred objects, the ACT-R visicon (i.e., collection of information available to the visual module "what" system) would get bogged down updating positions of moving objects. Interestingly, we did learn that the model is capable of handling at least 100 objects, but time and computation costs become unacceptable. Second, ACT-R has limited visual search capabilities. By default, it is possible to build a model that searches for more than one feature simultaneously (e.g. find a red circle). However, this will result in flat response time curves with respect to set size, contrary to what has been well established in the visual search literature for decades (Treisman & Gelade, 1980).

It would also result in the inability to account for effects of the presence of friendlies, which were observed in the original experiment (Fox et al., 2023). Alternatively, one could search for one feature at a time (e.g. find a red object) and search through the available objects linearly until an object is found that matches both desired criteria (Fleetwood & Byrne, 2006). This produces set size effects, but still becomes prohibitively slow for a display with hundreds of objects. These limitations have motivated previous extensions to the vision module, such as PAAV (Nyamsuren & Taatgen, 2013) and JSegMan (Tehranchi & Ritter, 2018). However, these capabilities do not currently exist in a form that is compatible with the version of ACT-R we used for the model (7.14). We also made two minor changes to the task representation. The model task uses colored letters instead of colored shapes and the location of alarms were modified. Letter locations can be modified, which ensured moving letters were still considered the same object. The model task is therefore, a scaled-down version of the experimental task.

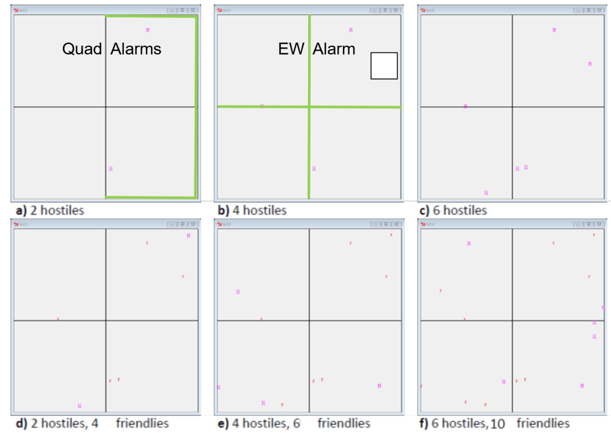

Figure 2: Depiction of model MOT-EW task and alarm states for quadrants and EW attacks.

Model Description

The model uses only four modules (i.e., goal, vision, motor, and procedural memory) and there is no semantic or procedural learning. There are several important parameters related to the task: visual attention latency, the speed of productions (i.e., processes), and visual finsts (i.e., number and span). All but one parameter, number of visual finsts, are left at default values. The number of visual finsts controls how many visual objects can be marked as attended and the span controls how long they remain marked. We have changed the number of visual finsts to 16 (default is 4). In addition, we deviated from the standard visual find-attend-encode loop. Typically, an object is found, the model shifts its attention to that object, and the object is then encoded (i.e., identified). In two instances, we allow the model to skip the attend step to simulate human's ability to extract information peripherally without directly fixating on it. We further explain why we adjusted the number of visual finsts and deviated from the standard visual processing loop in the following sections to provide context.

The MOT and EW tasks have their own set of processes, but share several general productions. They are completed separately in single task conditions or serially interleaved during dual task conditions. The model was provided information about the task (i.e., MOT, EW, or DUAL) at the start of each session (i.e., condition), but was not provided with information about the amount of hostiles or friendlies. We start by describing the MOT processes, followed by EW, and then the full model that interleaves both.

MOT Processes. For the MOT task, the model has to differentiate between quadrants, determine their unique alarm state, and identify which do and do not have a hostile. To accomplish this, the model searches through quadrants one at a time in a clockwise direction and makes alarm decisions (Figure 3). Once a quadrant is selected (e.g., search NW), the model orients to that quadrant's coordinates and checks the alarm state (i.e., check-alarm). Rather than attending and encoding the alarm state, the model finds the quadrant and encodes alarm state information or color peripherally without shifting attention directly. Alarm state information for the current quadrant is held in the goal buffer. After checking the alarm state, the model finds objects in the quadrant (i.e., find-object). If a friendly is found, it is marked as attended without shifting attention (i.e., friendly-no-attend), like the quadrant alarm state. The model continues to find objects until there are no new objects to find (i.e., all objects marked) or a hostile is found. In rare cases, the model shifts attention to a hostile, but there is no object at that location. This occurs because the object can potentially move beyond the focal area of the model. A reorient production (i.e., reorient) handles these rare cases and reorients the model to find the nearest object to the attended location, which should be the intended target. This represents a fixation that was slightly off and corrected.

Once all objects are searched or a hostile is found, the model makes an alarm decision. If the quadrant was completely searched and no hostile was found, the model turns off the alarm if it is currently on (i.e., turn-off-alarm), or moves to the next quadrant if it is already off (i.e., alarm-off-ok). If a hostile is found the model shifts attention to the location of the hostile (i.e., hostile-attend). The subsequent production encodes the hostile and either turns on the alarm if currently off (i.e., turn-on-alarm), or moves to the next quadrant if already on (i.e., alarm-on-ok). The model turns on and off alarms by pressing a keyboard key corresponding to the quadrant using the punch command that assumes fingers are resting on the home keys. After making a decision, the quadrant is marked as searched in a goal buffer slot and visual finsts are cleared. The model leverages visual finsts to ensure that quadrants are searched and setting the number of finsts to 16 ensures that the model does not get stuck in an endless loop within a quadrant (e.g., number of objects exceeds the parameter). Furthermore, clearing visual finsts ensures that the model does not skip over a marked object that moves into another quadrant within the default finst span time of 3 seconds. This could result in missing a hostile and either not turning on a quadrant alarm or incorrectly turning one off. The model completes the same process for each quadrant until all quadrants are searched and then model starts another quadrant search cycle (i.e., all-quads-searched).

Now that the MOT processes have been described, we provide an explanation for why we chose to skip the attend step for quadrant alarms and friendlies. The standard time for each production is 50ms, shifting attention takes 85ms, and punching (i.e. pressing a key) takes 210ms. The standard find-attend-encode loop would take 235ms to encode a single object or alarm state and changing an alarm state takes 260ms. Therefore, it would take 730ms just to encode a quadrant alarm state, encode a single object, and change an alarm state. Additionally, it would take 965ms to encode two objects, 1200ms for three, and 1435 for four. For comparison, the average human response time for changing quadrant alarm states is 900ms for the single MOT task and that includes conditions ranging from 2-6 hostiles and 498-994 friendlies. Reasonable response times are important, because they are related to alarm state change accuracy. By eliminating the need for the model to attend to alarm states and friendlies, we were able to achieve an average 1020ms response time across varied hostiles and friendlies in the scaled-down MOT model task. This decision also aligns with the human ability to extract information without direct fixation (Wolfe, 2021) and change alarm states quickly. These decisions are more important for EW processes, where the standard find-attend-encode loop would result in unreasonable response times, even in this scaled-down version of the experiment.

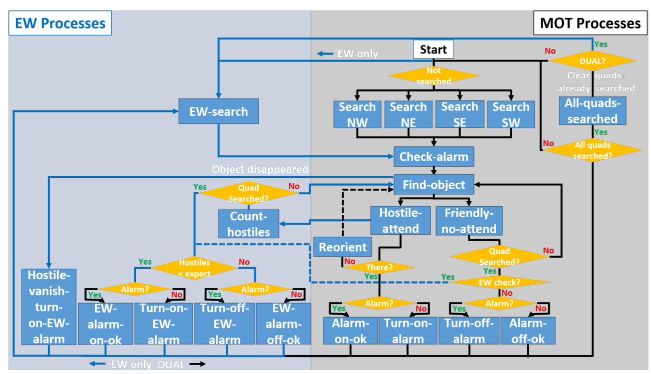

Figure 3: Diagram of processes to complete both MOT and EW tasks.

EW Processes. To complete the EW task, the model leverages MOT productions, which are considered general visual search processes (i.e., check-alarm and find-object). The model continuously checks whether an EW attack is occurring (i.e., a hostile has disappeared), and the EW alarm is turned on when an EW attack is ongoing and off when it ends (i.e., hostile reappears). The model starts the EW task by orienting to the entire screen (i.e., ew-search) and treating all four quadrants as the search area. Next, it checks the alarm state the same way as it does for MOT (i.e., check-alarm) and stores the alarm state in a slot in the goal buffer. The model then searches for objects (i.e. find-object), but contrary to the MOT, it is an exhaustive search. There are a maximum of 16 objects in one of the conditions (i.e., 6 hostiles and 10 friendlies) and this is why we set the number of visual finsts to 16. Although exhaustive, the model completes EW search similar to MOT. It perceives friendlies without attending and marks them as attended (i.e., friendly-no-attend). Using a find-attend-encode loop for all objects would be more of an issue for EW as there are more objects to search. It would take between 1200ms (i.e., 2 hostiles without friendlies) and 4255ms (i.e., 6 hostiles and 10 friendlies) to do a complete search and change the alarm state. For reference, the average human reaction time across number of objects was 734ms (1006ms for our model). Just like during MOT, if the model finds a hostile it shifts attention to its location (i.e., hostile-attend). However, rather than moving right to an alarm decision as in the MOT, the model encodes the hostile and keeps a count of how many hostiles have been attended (i.e., count-hostiles). Similar to participants, the model does not know how many hostiles to expect at the start of the session. For generality across conditions, the model updates a slot in the goal buffer that stores the amount of hostiles to expect. This slot is set to the highest number of hostiles counted during a session. For instance, if there are four hostiles and no EW attacks occur during the first five seconds, the model will count and set expected hostiles to four. If there was an EW attack at the start of the session, the model would count three and miss the EW attack. Once the model has performed its exhaustive search, it compares how many hostiles were counted with the amount expected (i.e,. conditions for alarm decisions). If it found the amount expected it turns off the EW alarm (i.e., turn-off-EW-alarm) if already on or it does nothing if the alarm is already off (i.e., EW-alarm-off-ok). If it found less hostiles than expected, it turns on the EW alarm if currently off (i.e., turn-on-EW-alarm) or does nothing if the alarm is already on (i.e., EW-alarm-on-ok). There is one additional alarm decision production (i.e., hostile-vanish-turn-on-EW-alarm) that is analogous to a person seeing a hostile disappear. A hostile could disappear after the model finds a hostile and starts shifting attention to that hostile. This hostile-vanish production handles this by turning on the EW alarm. Once a decision is made, the finsts are cleared and the model starts another EW check.

MOT-EW Processes. The complete model (Figure 3) is used for single and dual task conditions. During the dual task conditions, the model has to change quadrant alarm states based on hostile presence and change the EW alarm depending on whether an EW attack is occurring. The current model treats these as separate tasks and interleaves them. MOT processes are given priority and EW checks are initiated after all four quadrants have been searched. Therefore, a full quadrant search and EW check can be considered a complete cycle in the dual task conditions. After all quadrants are searched, the all-quads-search production fires, followed by the EW-search production that begins the EW check. One exception to this cycle is the production that turns on the EW alarm if a hostile disappears while shifting attention to a hostile (i.e., hostile-vanish-turn-on-alarm), which can supersede the cycle.

Results

We assess model performance and show how well it fits collected human data from the MOT-EW experiment (Fox et al., 2023). The experiment included 28 participants and a within-subjects 3 (amount of hostiles: 2, 4, and 6) x 3 (task: MOT, EW, and MOT-EW) x 2 (friendlies: present and not present) design. We simulated 25 participants for all 18 conditions. As the model does not possess individual differences, the model essentially simulates one participant completing the experiment 25 times. To assess how well the model captured the human data, we compared single and dual task performance for MOT and EW separately. We included both accuracy and response time for correct responses (i.e., time for correct alarm changes) as the dependent measures. For each comparison, we assess behavior patterns in dependent measures across variations in the number of objects (i.e., amount of hostiles and presence of friendlies). We used correlations to assess the ability for the model to capture patterns across conditions and use root mean squared error (RMSE) to assess the average difference between the model and human data.

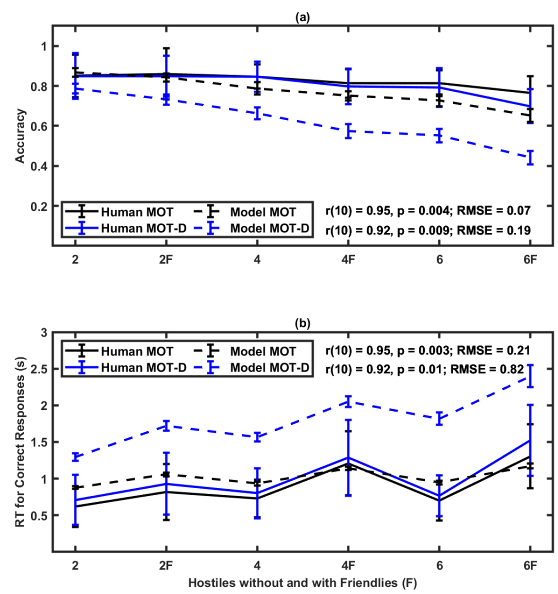

Figure 4: Model fit to human accuracy (a) and RT for correct responses (b) across all MOT conditions.

MOT Task

For the MOT (Figure 4), the model captured human behavior patterns for single and dual task conditions for both accuracy and response time ($p < .05$). However, the average difference for dual task accuracy (*RMSE* = .19) and response time (*RMSE* = 820ms) was higher than single task accuracy (*RMSE* = .07) and response time (*RMSE* = 210ms). The model had a larger difference between single and dual task accuracy (.15) and response time (787ms) than the human data (.02 and 106ms, respectively). Therefore, the model was better able to capture human performance in the single task MOT, then dual task. We also assessed the overall relationship between accuracy and response time in single and dual conditions for the human data and model. For the single task MOT, there were non significant negative relationships between accuracy and reaction time for both human ($r(10)$ = -.78, p = .07) and model data ($r(10)$ = -.63, p = .177). However, there were significant negative relationships in the dual task MOT for both human ($r(10)$ = -.82, p = .044) and model ($r(10)$ = -.91, p = .010).

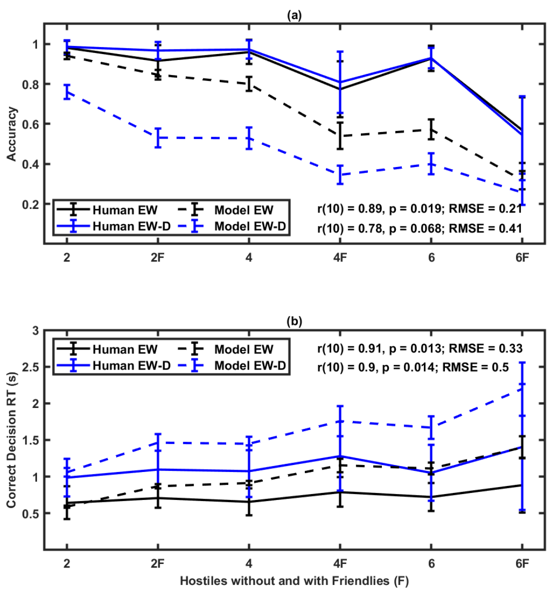

Figure 5: Model fit to human accuracy (a) and RT for correct responses (b) across all EW conditions.

EW Task

For the EW task, there were clearer average differences between human and model performance. The model captured the pattern for the EW single task accuracy, but not dual task accuracy ($p > .05$). The model had a higher average difference for EW accuracy in single (*RMSE* = .21) and dual task (*RMSE* = .41) compared to the MOT (*RMSE* = .07 and *RMSE* = .19, respectively). Similar to the MOT, the model had a greater difference in EW accuracy between single and dual task (.2) compared to humans (.01). The model was able to capture EW response time patterns for both single and dual task ($p < .05$) and similar to the MOT, the average difference was higher for the dual task (*RMSE* = .5) compared to the single (*RMSE* = .33). However, the average difference for EW dual task was lower than the MOT dual task (.5 compared to .82), suggesting EW processes had a stronger negative effect on MOT in the dual task conditions than vice versa. Again, the human data demonstrated a lesser difference between single (.01) and dual task (415ms) compared to the model (.2 and 594ms, respectively). We also assessed the relationship between accuracy and response time for the human data and model. In the single task EW, there were significant negative relationships for both human ($r(10)$ = -.98, p = .001) and model ($r(10)$ = -.98, p = .001). There were also significant negative relationships in the dual task EW for human ($r(10)$ = -.94, p = .004) and model ($r(10)$ = -.97, p = .001).

Table 1: Relationships between accuracy and RT.

	Condition	*df*	*r*	*p*
Human	Single MOT	10	-.78	.066
	Dual MOT	10	-.82	.044*
	Single EW	10	-.98	.001*
	Dual EW	10	-.94	.005*
Model	Single MOT	10	-.63	.177
	Dual MOT	10	-.91	.010*
	Single EW	10	-.98	.001*
	Dual EW	10	-.97	.001*

Discussion

The model was able to capture behavior patterns for both accuracy and response time, via significant correlations, for all but one task and condition (i.e., EW dual task accuracy). Although, the average difference in performance as measured by RMSE varied and in some cases, was rather high. MOT performance was captured better than EW, particularly with the single task. There was more deviation in MOT accuracy and response time for the dual task compared to the single task, which was not found in the human data. Interestingly, the model dual task average difference for MOT response time was greater than EW. This suggested that completing EW had a stronger effect on MOT performance in the dual task conditions than number of hostiles and presence of friendlies.

The model was not able to capture EW behavior as well. In contrast to MOT performance, model performance for EW is consistently lower for both single and dual. This suggests the EW model processes are not as well aligned with humans and is likely contributing to the performance decrement seen in the MOT during dual task. The model takes longer than humans to change EW alarm states, suggesting humans are doing EW checks differently or potentially adopting different EW strategies across conditions. For example, the model EW checks involve exhaustive search regardless of condition, and tasks are treated as separate and are interleaved serially instead of processed in parallel (e.g., multitasking). Given the nature of the model EW checks, there should be a rather linear decrease in EW performance as the number objects to search increases and this is evident in the EW figures (Figure 5). We see a similar pattern with the stronger negative trend in the MOT dual task accuracy. The trend for the MOT dual task reaction time is consistent with that of single task, but responses took an average of 787ms longer across conditions.

As stated, this model was intended to test out of the box capabilities of ACT-R and serve as a baseline. We believe we have achieved close to the best performance possible using canonical ACT-R without adjusting more than one parameter. There are of course, limitations with the existing model. Next we discuss these limitations and ideas to improve the model.

Limitations and Future Work

The model has several limitations: 1) There is no variation in task execution for single and dual task conditions (i.e., no strategies or individual differences), 2) the speed of processing and visual attention is notably slower than humans, and 3) EW performance is notably worse than MOT and is the likely cause for the decrease in dual task MOT performance.

The model has a rigid approach to completing the MOT, EW, and interleaving them in the dual task condition. There are no strategies and the model does not learn. In the human data, there is more variation within and between participants compared to the model. However, this is not surprising given the model could be considered one individual completing the experiment repeatedly. Using current and future human data could identify strategies, condition specific strategies, and perhaps clusters of individuals or types that have similar patterns of behavior. This would inform the ability of the model to capture individual differences and adding learning mechanisms enables strategy shifts across conditions. In addition, we could consider threaded cognition (Salvucci & Taatgen, 2008) as a method to enable multitasking rather than treating tasks as separately and interleaving them.

As mentioned, ACT-R does not currently have visual search capabilities beyond deterministic or featureless strategy-based search. To facilitate visual search, we deviated from the typical find-attend-encode loop and allowed the model to attend some stimuli peripherally. This decision was guided by the visual search literature (Wolfe, 2021) and was a plausible way to speed up visual search processes. Despite our efforts, it was clear that the model took longer to change alarm states than humans, which also relates to accuracy. Furthermore, the model interacted with a scaled-down version of the task presented to human participants. To address visual search capabilities and improve model performance, we plan to revive or implement features from the PAAV module (Nyamsuren & Taatgen, 2013). PAAV has both bottom up (e.g., color and shape salience) and top-down (e.g., strategy based) features that enable more directed visual search observed in humans. Furthermore, this should eliminate the need for exhaustive serial search. After making progress with visual search capabilities, we plan to scale up the model task to better align with the experimental task.

The model performed worse for the EW task, which also appeared to reduce MOT performance in the dual task conditions. We believe this resulted from: 1) flawed EW processes that are likely not as well aligned with human behavior and 2) limited visual search capabilities that encouraged exhaustive serial search. We plan to address both of these points with future work outlined in the above sections: 1) better understand and implement strategies and 2) make some improvements to visual search capabilities.

Conclusions

We successfully developed and implemented a simplified model in ACT-R capable of performing a complex radar detection task and testing hypotheses about the underlying cognitive processes. The model provided a reasonable fit to human data across 18 conditions and serves as a solid baseline by demonstrating the out of the box capabilities of ACT-R. Future work will extend this model to address identified limitations of the architecture, model performance, and the scaled-down model task.

Acknowledgments

This research was supported by the U. S. Air Force Research Laboratory's 711th Human Performance Wing, Cognitive Models and Sensory Systems Branches. The contents have been reviewed and deemed Distribution A. Approved for public release. Case number: AFRL-2023-1409. The views expressed are those of the authors and do not reflect the official guidance or position of the United States Government, the Department of Defense, or the United States Air Force.

References

Anderson, J. R. (2007). *How can the human mind occur in the physical universe?* Oxford University Press.

Bath, W. G. (2020). Overview of platforms and combat systems. *Johns Hopkins APL Technical Digest*, *35(2)*, 90–98.

Fisher, C. W., & Kingma, B. R. (2001). Criticality of data quality as exemplified in two disasters. *Information & Management*, *39*(2), 109–116.

Fleetwood, M. D., & Byrne, M. D. (2006). Modeling the visual search of displays: a revised act-r model of icon search based on eye-tracking data. *Human-Computer Interaction*, *21*(2), 153–197.

Fox, E. L., Stephenson, A., Stevens, C. A., & Bowers, G. (2023). Predictors of human efficiency in radar detection tasks. In *Proceedings of the 18th international conference on cyber warfare and security (iccws).* Towson, Maryland, USA.

Glavan, J. J., Haggit, J. M., & Houpt, J. W. (2020). Temporal organization of color and shape processing during visual search. *Attention, Perception, & Psychophysics*, *82*, 426–456.

Nyamsuren, E., & Taatgen, N. A. (2013). Pre-attentive and attentive vision module. *Cognitive systems research*, *24*, 62–71.

Pogue, D. (2016). 5 of the worst user-interface disasters. *Scientific American*.

Salvucci, D. D., & Taatgen, N. A. (2008). Threaded cognition: an integrated theory of concurrent multitasking. *Psychological review*, *115*(1), 101.

Tehranchi, F., & Ritter, F. E. (2018). Modeling visual search in interactive graphic interfaces: Adding visual pattern matching algorithms to act-r. In *Proceedings of iccm-2018-16th international conference on cognitive modeling* (pp. 162–167).

Treisman, A. M., & Gelade, G. (1980). A feature-integration theory of attention. *Cognitive Psychology*, *12*(1), 97–136.

Wolfe, J. M. (2021). Guided search 6.0: An updated model of visual search. *Psychonomic Bulletin & Review*, *28*(4), 1060–1092.