Using Model Tracing and Evolutionary Algorithms to Determine Parameter Settings for Cognitive Models From Time Series Data such as Visual Scanpaths

Yunfeng Zhang (zywind@cs.uoregon.edu)
Anthony J. Hornof (hornof@cs.uoregon.edu)
Computer and Information Science, University of Oregon
Eugene, OR 97403 USA

Abstract

Time-series data such as eye movements or mouse movements contain rich information about the dependencies between successive human actions. This information can be potentially very useful for examining model assumptions and constraining parameter search. This paper explores the use of model tracing, which simulates a task by tracking observable human behaviors, to time-series data. We explore two aspects of tracing that are different from conventional cognitive modeling: (a) tracking the observed behaviors and (b) estimating the likelihood of the observed events. We demonstrate how these two features of tracing, along with the use of an evolutionary optimization algorithm, led to accurate and robust estimates for parameters of visual acuity functions needed by visual search models.

Keywords: cognitive modeling; model tracing; visual scanpath; visual search; genetic algorithms.

Introduction

Free parameters exist in almost all quantitative psychological theories. The parameters provide flexibility for models to capture the variability of the environment and the individuals. However, as many researchers pointed out (e.g., Roberts & Pashler, 2000; Howes, Lewis, & Vera, 2009), excessive free parameters also increase a model's degrees of freedom and reduce its predictive power.

One way to combat the loss of predictive power, particularly in the realm of cognitive modeling, is to fit the model to data at small timescales such as eye movement data or mouse movement data. Traditional psychometric data such as reaction time often do not place enough constraints on a model because cognitive models are constructed with mechanisms that operate on a much smaller timescale. For example, Hornof and Zhang (2010) demonstrated that a model with a plausible but unlikely multitasking strategy could accurately fit reaction time data, but not the eye movement data. Because data at small timescales are more directly related to some cognitive processes such as visual attention, there may be many fewer model parameter settings or strategies that can fit the data, and so the model that does fit is more likely to be correct.

One particular piece of information, the order of events in small timescales, is often overlooked, yet it may be extremely valuable for further constraining the free parameters. Modeling the order of events is also called tracing. Though this approach was initially introduced to the field of cognitive modeling by Anderson, Kushmerick and Lebiere (1993) in the early '90s, it has rarely been used by cognitive modeling researchers. This paper revives the method and applies it to fit data at small timescales.

Tracing involves predicting the next event given the task context that the participant experienced at a given point in time. A cognitive architecture has many context-, time-, or location-dependent mechanisms that will produce different predictions given different contexts. For example, in visual search, the location where the model looks next depends on the current gaze location (context-dependent), the contents of the visual working memory (time-dependent), and what objects and features can be perceived by peripheral vision (location-dependent). Correctly predicting a fixation location requires a model to adequately account for all three dependency factors. Predicting thousands of fixations, as a typical experiment would generate, will then place tight constraints on the plausible model parameter ranges.

Though tracing is potentially useful, it requires a different approach than conventional cognitive modeling. This paper introduces these approaches and demonstrates how tracing may be used to accurately estimate model parameters. The "case study" presented here is a visual search model, for which a set of parameters needs to be estimated to correctly represent the availability of object features in the peripheral vision. We use tracing and an evolutionary algorithm to search for the parameters that best fit the fixation scanpath data from a visual search task. The resulting parameters are validated by using them in a conventional cognitive model to predict summary eye movement statistics.

Model Tracing

To make the methods concrete and accessible, the following discussion about tracing will be grounded in the subjects of fixation scanpath tracing and visual search. However, the principle of these methods should easily apply to other applications of event sequence tracing.

When a conventional cognitive model is applied to predict a scanpath, two problems immediately emerge. First, the error accumulates very fast and can quickly lead to completely different scanpaths. This is because once the model's fixation location diverges from the participant's, the model will then have a different context in which to make decisions about where to look next, and hence is even less likely to correctly predict the participant's next fixation than if the model had remained aligned with the participant. Second, the scanpath data take on discrete, categorical values (e.g., the fixated object sequence may be A-B-C), but the conventional goodness-of-fit measures such as R^2 only apply to continuous values. This is one problem that Anderson et al.'s original tracing method did not address, and it will be addressed in this study by introducing a suitable goodness-of-fit measure to evaluate how well a model fits a scanpath.

The first problem—that of context—can be addressed by continually realigning the model to fixate the same object as the participant. With this technique, although the model still makes predictions about where the next fixation might be, it may or may not move its gaze to that predicted location. Rather, the model is set to match the behavior of the participants to make sure that when predicting the next fixation, the model always has the same context that the participant had. For example, if the model predicts that the next fixation is likely to be on Object B, but the participant actually fixated A, then the model's gaze is moved to A rather than to B.

The second problem—finding a suitable goodness-of-fit measure for the categorical scanpath data—can be addressed by having the tracing model generate likelihood predictions about the observed scanpath rather than generating an actual fixation sequence. Visual search, as well as many other tasks, is probabilistic in that at any point in time, there may be several equally good candidate next steps to take, in which case the model should be given the same credit for making any of the equally good choices. For instance, in the previous example, if the model predicts that both A and B are equally good choices (such as equally similar to the target), and the participant happened to choose A, then the model should score 50%, because in 50% of the time, the model will also choose A.

Visual Acuity Functions

Model tracing is applied in this study to estimate the parameters of visual acuity functions. These acuity functions describe how the visibility of object features gradually diminishes as objects move further from the point of gaze. Correctly characterizing this fundamental visual phenomenon is vital for comprehensive unified model of visual search, such as Halverson and Hornof (2011). Several researchers proposed different forms of visual acuity functions (e.g., Findlay & Gilchrist, 2003; Pomplun, Reingold, & Shen, 2003; Kieras, 2010; Nyamsuren & Taatgen, 2012). All proposed functions have free parameters to account for different feature characteristics, such as how color diminishes more gradually than shape. And yet it is very difficult to determine these parameters from real-world task data.

The particular visual acuity function examined here is proposed by Kieras (2010). This function assumes that for an object feature to be perceivable, the object size has to exceed a threshold that increases quadratically with increased object eccentricity (the angular distance between the object and the center of the gaze). The function is described as follows:

$$\begin{aligned} &threshold = ae^2 + be + c \\ &P(available) = P(s + X > threshold) \qquad (1) \\ &X \sim N(0, vs) \end{aligned}$$

where e represents the eccentricity, s represents object size, and X represents a noise that is sampled from a Gaussian distribution with a standard deviation of v times s. The parameters a, b, and c vary for different object features such as color, shape, and size to simulate different rates of visibility degradation.

Kieras showed that when incorporated into the EPIC cognitive architecture, such visual acuity functions can can contribute to accurate models of the data from the Williams (1966) visual search experiment; however, the good fit is somewhat questionable considering that the visual acuity functions alone had almost the same number of free parameters (10) as the number of data points (16) fitted. This problem is addressed here by model-tracing the thousands of fixations contained in the scanpath data for a re-running of the Williams experiment. Because, statistically speaking, estimating model parameters with more data points leads to smaller variance in the estimates (Cohen, 2003), fitting the model to a large set of scanpath data will result in more robust parameter settings than an estimate based on overall task time.

The Williams Visual Search Task (Replicated)

To obtain the complete scanpath data and to conduct a deeper analysis than was originally reported, we replicated the Williams (1966) experiment with contemporary eye tracking technology. Figure 1 shows a sample search display used in the experiment, which occupies a 39º by 30º rectangle area on the screen. The task was to search for a target in a grid of 75 objects that have different colors, shapes, and sizes. Each object has a unique two-digit number in the center. Search precues were shown before each trial and included the number of the target object and, depending on the precue condition, some combination of the target's color, size, and shape. The precue always included the target number and optionally included each of three features, resulting in eight possible precue conditions, such as "17 small blue cross" which was the "All" feature condition.

Each object in a search field had a unique combination of one of five colors, one of five shapes, and one of three sizes. Colors were blue, green, yellow, red, and purple. Shapes were circles, semi-circles, triangles, squares, and crosses. Sizes were small (0.8º), medium (1.6º), and large (2.8º), measured by the diameter of the circle for object size category; all shapes were normalized to occupy the same area as the circles in each size category.

Each participant was presented with ninety-six search fields that were grouped into four blocks. The trials of Blocks 1 and 2 and the trials of Blocks 3 and 4 were randomly selected from ninety-six preconfigured search fields, each with a fixed object arrangement. Each block had the same number of trials for every precue condition.

After finding the target, the participant clicked on it to proceed to the next trial. Monetary rewards were given for the correctly completed trials and were adjusted based on the search time and task difficulty to motivate good performance.

Eye movement data were collected for 22 participants using an L.C. Technologies binocular 120 Hz eye tracker. A chinrest reduced head movements and improved eye tracking accuracy. Fixations were identified using a dispersion-based algorithm with a maximum dispersion window size of 0.7 and a minimum fixation duration of 60 ms. A fixation was assigned to an object if it fell within a circular area of interest (AOI) around the object; otherwise,

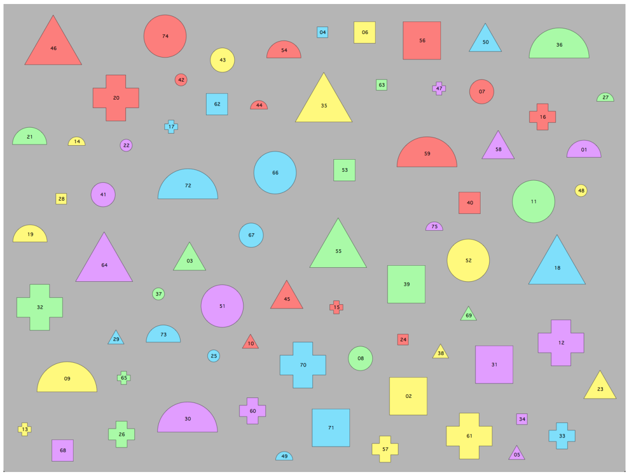

Figure 1. A sample search field used in the experiment.

it is classified as a between-object fixation. The size of the AOI was adjusted for objects of different sizes to properly distinguish fixations that were on versus between objects. More details about the experiment design and eye movement data processing can be found in Hornof and Zhang (2013).

The participants' performance is measure by three summary eye movement statistics, presented in Figures 3, 4, and 5 (black bars) alongside some model predictions (gray bars) that will be discussed later. Our experiment successfully replicated Williams' observation that color is more useful in guiding visual search than size and shape. This can be seen in Figure 4 in that the precue conditions that specified color had larger proportions of fixations landing on objects with the specified feature, which suggests that the participants may be able to see color in a wide area of their visual periphery and use that information to effectively plan their next saccade to objects that are likely to be the target.

Estimate Parameters Using Tracing

Our goal is to estimate visual acuity function parameters by tracing the fixation data collected in the replicated Williams visual search experiment. To implement tracing, we developed a standalone computational model that is dedicated to simulate this visual search task only. We call this new model the *scanpath tracing model*.

The scanpath tracing model adopts theoretical concepts the of visual acuity function and visual perceptual store (VPS) of the EPIC cognitive architecture, which have contributed to explanation of the original Williams results and other visual search tasks (Kieras, 2010; Nyamsuren & Taatgen, 2012). As discussed in the introduction, the visual acuity function determines whether an object feature is distinguishable based on the object size and its distance from the center of gaze. If the feature is determined to be available, it is then deposited in the VPS for a short time period (e.g, 300 ms). The model then uses the features in VPS to decide where to look next. The parameters for the two mechanisms include the coefficients of the visual acuity function and the feature decay time of VPS. They are the free parameters of the scanpath tracing model, and were estimated by fitting the model to the scanpath data.

With the implementation of the two visual mechanisms, the scanpath tracing model can simulate the task by cycling through three steps: (1) The model moves the gaze to the observed fixation location and sets the simulation time to the fixation time. (2) The model deletes from VPS the items that should have decayed based on the passing of time, and adds the objects and features that the visual acuity functions determine are available based on the current gaze position. (3) Based on the contents of VPS, the model calculates the likelihood that the next fixation would be at the observed location as opposed to at other locations. A higher likelihood indicates a better fit between the model and the human scanpath data.

In every cycle, the contents of VPS will contain some combination of the following:

- *Viable-candidates* – objects that have a feature in common with the target.
- *Non-targets* – objects that have a feature that is known and which makes it not possibly the target (such as a red object when looking for a blue target).
- *Unknown-objects* – objects that are visible but have no known color, size, or shape features.

For Step 3, calculating the likelihood of the next fixation location, the tracing model can encounter four possible VPS states. Table 1 shows these four states. These four states reflect the possibility that the VPS may or may not contain viable-candidates, and may or may not contain non-targets. The presence or absence of unknown-objects is not relevant for this decision because based on the screen size and acuity functions, they would always be present.

Table 1 shows, for each of the four VPS states, the likelihood that the visual search strategy will move the eyes to each of four possible destinations: viable-candidates, non-targets, unknown-objects, and the space between objects (outside of all of the AOIs). The table reflects the results of setting four numerical parameters: When neither viable-candidates nor non-targets are available (State 1), 66% of the fixations go to unknown objects, and 34% to the space between objects. If viable-candidates are not available but non-targets are available (State 2), 48% of the fixations go to non-targets, and the other half go to the remaining two categories based on the same proportions as in State 1. Any time that viable-candidates are available (States 3 and 4), 95% of the fixations go to the viable-candidates, and the rest are distributed based on the same proportions in the previous states.

These four parameters were set as follows: The 66% and 34% were assigned to unknown-objects and between-objects because one third of all long saccades (> 8º) that were made by participants fell between objects and these are the length of saccades the model will be making in State 1. The 48% and 95% were chosen by the authors based on their intuitions. The 95% is based on an assumption that people would almost always try to use feature information when it was available, but that the model needs to also account for a small amount of noise and error. The 48% is set to find a balance between (a) not fixating objects that are clearly not the target and (b) fixating nearby objects, to

Visual-Perceptual Store			The Search Strategy Prefers			
	Contains		Object Types as Follows			
State	Viable-candidates	Non-targets	Viable-candidates	Non-targets	Unknown-Objects	Between Objects
1	no	no	0.0%	0.0%	66.0%	34.0%
2	no	yes	0.0%	48.0%	34.3%	17.7%
3	yes	no	95.0%	0.0%	3.3%	1.7%
4	yes	yes	95.0%	2.4%	1.7%	0.9%

Table 1. The likelihood that the visual search strategy will move the gaze to the four possible destinations under each of the four visual-perceptual store states.

account for how visual search strategies tends to prefer nearby objects over far objects (Halverson & Hornof, 2011).

To calculate the likelihood of each observed fixation location, which is used to measure how well the model fits the scanpath, the model first determines which cell of Table 1 should be used based on the VPS state and the destination type of the observed location. Then, if the observed gaze location is between-objects, its likelihood is just the percentage designated in the table. If the object location is in one of the three other destination categories, its likelihood is *1/n* of the designated percentage, in which *n* is the number of objects in VPS that are also in that category.

The parameters of the visual acuity function directly affect the contents of VPS, and thus the likelihood generated for each observed fixation. Because a higher likelihood indicates a better fit to the data, our goal is to find the parameter settings that generate the highest likelihood for the observed scanpath.

The conventional method for finding good parameter settings is by trial-and-error (usually with a grid search), in which the analyst iterates through a set of different settings and finds one that fits the data well. However, because in this task, the visual acuity functions have 10 parameters (three coefficients for each acuity function, plus a noise parameter), it would be computationally impractical to iterate through the large parameter space with a grid search. Thus, we use an efficient evolutionary algorithm, specifically *differential evolution* (Vesterstrom & Thomsen, 2004), to find suitable parameter settings.

The differential evolution algorithm conducts a parameter search for the tracing model in the following four steps: (1) The algorithm instantiates a set of scanpath tracing models (100 models for our study) with random parameter settings (Generation 0). (2) It runs each instantiated tracing model, and each model calculates the likelihood of the scanpath data (the average log-likelihood of all fixations). For our study, the scanpath data include 24,821 fixations collected from the visual search trials that specified a single target feature. (3) The algorithm creates a new generation of parameter settings by moving the parameters that generated low likelihoods towards those that generated high likelihoods. The details of how the new parameters are created can be found in Vesterstrom and Thomsen (2004). (4) The algorithm repeats steps (2) and (3) for many generations until the termination condition has been reached. For this study, the search was set to terminate after 300 generations. Because in each generation the parameters are slightly improved, the parameters found after many generations should provide a sufficiently good fit to the scanpath data, though they are not guaranteed to be optimal.

To address the possibility that the differential evolution algorithm could become trapped in a local maxima, the parameter search procedure was repeated 12 times with different random number generator seeds. Each of the 12 runs produced a set of estimates for the 11 parameters of the tracing model (10 for visual acuity functions, and 1 for VPS). In all runs, the estimated parameters allowed the tracing model to fit the scanpath data better (average log-likelihood ranges from -3.61 to -3.614) than the original EPIC parameters (average log-likelihood is -3.74), suggesting that the parameter search with the differential evolution algorithm was successful. The parameters estimated from the 12 runs are very similar to each other (the *SD* ranged from 0.02 to 0.38 for the visual acuity function coefficients, and *SD* was 13 for the VPS feature decay time), indicating that the scanpath data provided sufficient constraints such that the good parameter settings were within a small region.

Results

The parameters of the best fitting model across all 12 runs are discussed in this section. The VPS feature decay time is estimated to be 73 ms, suggesting that the participants rarely held the available features in VPS for more than one fixation. The noise parameter v of equation (1) was estimated to be 0.05, indicating that the perceptual noise did not significantly affect an object's visibility.

Figure 2 shows the visual acuity functions estimated from tracing and from the original EPIC model (which can be found in Kieras, 2010). The curves determine the threshold object size for a feature to be available. That is, an object feature is available when it is above (or to the left of) that feature's curve. Both sets of functions show similar trends across the three features: Color is more visible than size, and size is generally more visible than shape. The main difference is that our parameters allow greater availability for all features than the original EPIC parameters. However, from this graph alone, it is difficult to tell which ones are better. Although our parameters fit the scanpath data better, we need to show that they are not overfitting and that they can be used to explain data at large-scales as well.

Validation of The Estimated Parameters

To validate the parameters estimated from tracing, we transferred them into Kieras' EPIC-based visual search

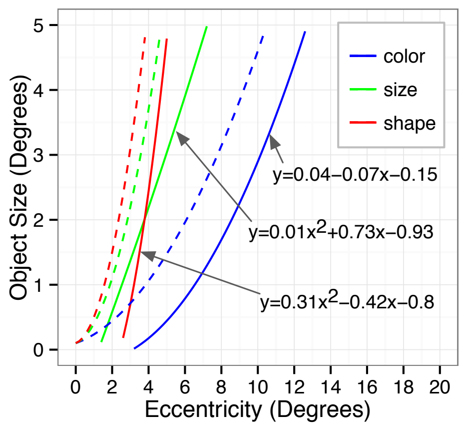

Figure 2. The visual acuity function estimated from tracing (solid) and from the original EPIC model (dashed). An object feature is available when it is above or to the left of that feature's curve.

model (with slight modifications discussed below) to see whether the EPIC model with the new parameter settings can fit the summary statistics of the eye movement data and better than the same model with the original parameter settings.

Three changes were made to Kieras' EPIC model: (a) The visual feature decay time was changed from 9 seconds to the estimated 73 ms. It is unlikely that the participant can actually remember the properties of many objects for as long as 9 seconds (Luck & Vogel, 1997). This was set in the original model to simulate how participants seemed able to remember objects that were examined recently and to avoid repeatedly examine the same objects. In this new EPIC model, repeat fixations are prevented by the next change. (b) The model inhibits fixations to the twenty most recently visited objects. Though this mechanism probably exceeds human capabilities as well, its effect might be achieved by some scanning strategies that remember the regions visited rather than individual objects. (c) When selecting one object from multiple objects, the model randomly selects one from the four nearest objects as opposed to from all objects. This change is made to account for the observation that people tend to prefer looking at nearby objects over far objects (Halverson & Hornof, 2011).

Figures 3, 4, and 5 compare the models' predictions with the observed data on three critical aspects of the visual search performance. The average absolute percentage error (AAPE) between the human data and the model predictions are listed in the figure captions.

Figure 3 shows the average number of fixations per trial across the eight precue conditions. The fewer the fixations, the better the search performance. Overall, the tracing-parameter model fits the data better than the original EPIC parameter model (as indicated by the AAPEs). Both models captured the main effect, that the conditions that specified color required many fewer fixations than other conditions. However, both models failed to predict how the number of fixations decreased from Shape to Size, and from Size to Size+Shape. It seems that the tracing parameters overestimated shapes' visibility, whereas the original parameters underestimated Size's visibility. Neither model

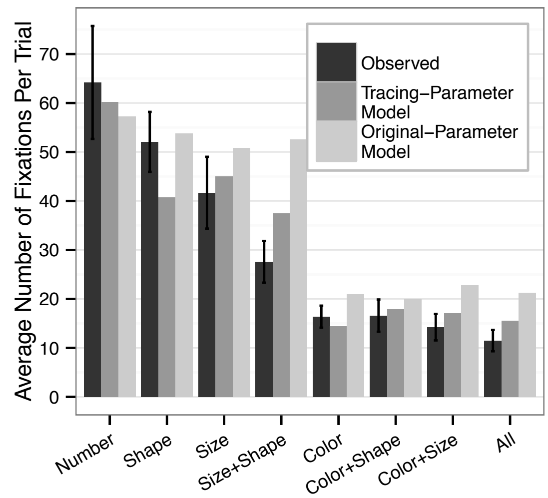

Figure 3. Average number of fixations per trial across eight precue conditions. The black bars represent human data, including the 95% confidence interval. The dark gray bars represent the model with parameters from tracing, and the light gray bars represent the the model with the original EPIC parameters. AAPE: Tracing, 18%; Original, 40%.

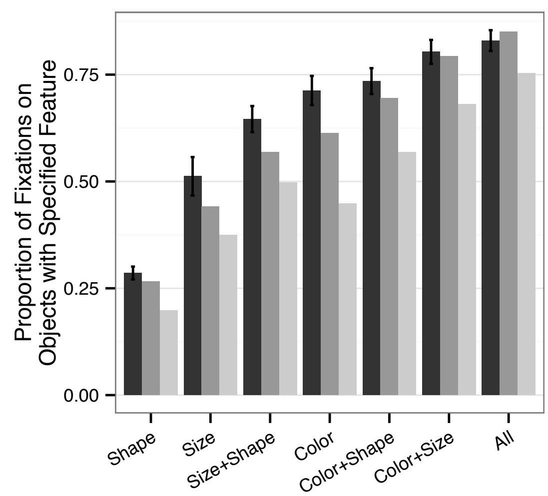

Figure 4. The proportion of fixations that landed on objects with at least one of the specified features in each of the seven precue conditions that provided object features. AAPE: Tracing, 8%; Original, 24%.

predicted the additive effect between Size and Shape, as seen in how Size+Shape required fewer fixations than the single feature conditions. Perhaps Size+Shape needs a separate visual acuity function to capture the effect.

Figure 4 shows the proportion of fixated objects that had at least one of the specified features across conditions. A higher proportion indicates that the participants could more easily find an object with the target feature in their visual periphery, leading to better search performance. Again, the tracing-parameter model fits the data better than the original-parameter model. Both models predicted smaller proportions than the observed in most conditions. This is perhaps because the between-object fixations were removed from the human data when calculating the proportions (to be

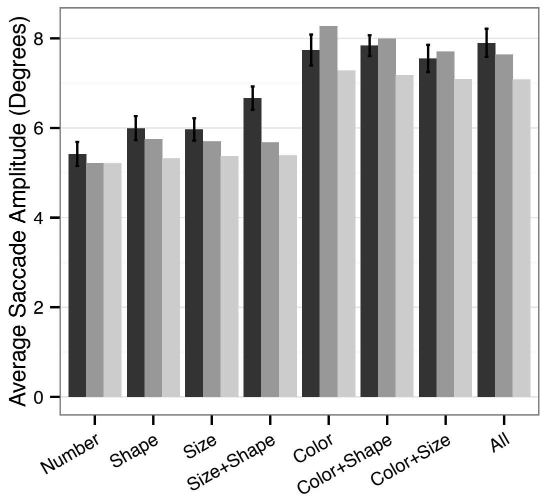

Figure 5. Average saccade amplitude across conditions. AAPE: Tracing, 5%; Original, 9%.

consistent with Williams' analysis), which inflated the proportion for the human data.

Figure 5 shows the average saccade amplitude across precue conditions, which also serves to illustrate the usefulness of the visual guidance provided by the different features because some features, such as color, may be more visible and useful than other features to guide vision at longer distances. Both models fit the data very well and predicted the average saccade amplitude for all conditions except the Size+Shape condition.

Overall, when used in the EPIC visual search model, the parameters estimated by the tracing model generated good fits, and in most cases, outperformed the original EPIC parameters that were specifically adjusted to fit the summary eye movement statistics.

Conclusion

This paper explores a novel approach to modeling. Rather than fitting summary statistics of empirical data, we develop tracing models that predict an event series, in this case fixation scanpaths, to robustly estimate model visual acuity function parameters, which are needed to accurately model visual search performance. The parameters were found using a genetic algorithm to maximize the likelihood of the observed scanpath data given the tracing model. The parameters are further validated in an EPIC visual search model. The results showed that the parameters estimated through tracing are accurate.

In developing the tracing model, we found a couple of elements that are needed and are not often seen in typical cognitive modeling. First, the tracing model needs to be able to track the observed state by making similar actions to what the participants did. This approach gives the model the chance to predict the next action with the correct context. Conventional approaches to cognitive modeling might benefit by making this event tracing approach readily available within cognitive architectures. Second, the model needs to predict the probability of the events to allow better assessment of the model's goodness of fit. This can be challenging because it may require the analyst to guess the probability of certain events, as we have done here in Table 1. Future research direction includes finding a more principled way to assign these probabilities.

In sum, event tracing is a novel and useful approach to explaining human data that may have great potential for developing and evaluating accurate computational cognitive models of human performance. This paper demonstrates event tracing for scanpath data, but it is possible to apply the same approach to other time series data as well.

Acknowledgements

This project was funded in part by NSF Grant IIS-1017593, though the opinions are strictly those of the authors. We thank David Kieras for providing a copy of his EPIC Williams task models.

References

Anderson, J. R., Kushmerick, N., & Lebiere, C. (1993). Navigation and conflict resolution. In: Anderson, J. R. L. (Ed.), *Rules of the mind*, Erlbaum, Hillsdale, NJ

Cohen, J. (2003). *Applied multiple regression/correlation analysis for the behavioral sciences (Vol. 1)*. Lawrence Erlbaum.

Findlay, J. M., & Gilchrist, I. D. (2003). *Active vision: The psychology of looking and seeing*. Oxford University Press.

Halverson, T., & Hornof, A. J. (2011). A computational model of "active vision" for visual search in human-computer interaction. *Human-Computer Interaction, 26*, 285–314.

Hornof, A. J., & Zhang, Y. (2010). Task-constrained interleaving of perceptual and motor processes in a time-critical dual task as revealed through eye tracking. In *ICCM 2010*, 97–102.

Hornof, A. J., & Zhang, Y. (2013). *The effect of target specification and visual acuity on objects fixated during visual search* (Tech. Rep. No. CIS-TR-2013-03). University of Oregon: Department of CIS Technical Report.

Howes, A., Lewis, R. L., & Vera, A. (2009). Rational adaptation under task and processing constraints: Implications for testing theories of cognition and action. *Psychological Review, 116*(4), 717–751.

Kieras, D. E. (2010). Modeling visual search of displays of many objects: The role of differential acuity and fixation memory. In *ICCM 2010*, 127–132.

Luck, S. J., & Vogel, E. K. (1997). The capacity of visual working memory for features and conjunctions. *Nature, 390*(6657), 279-281.

Nyamsuren, E., & Taatgen, N. A. (2012). Pre-attentive and attentive vision module. In *ICCM 1012*, 211-216.

Pomplun, M., Reingold, E. M., & Shen, J. (2003). Area activation: a computational model of saccadic selectivity in visual search. *Cognitive Science, 27*(2), 299-312.

Roberts, S., & Pashler, H. (2000). How persuasive is a good fit? a comment on theory testing. *Psychological Review, 107*(2), 358–367.

Vesterstrom, J., & Thomsen, R. (2004). A comparative study of differential evolution, particle swarm optimization, and evolutionary algorithms on numerical benchmark problems. In *Evolutionary computation, 2004. cec2004. congress on* (Vol. 2, pp. 1980–1987).

Williams, L. G. (1966). The effect of target specification on objects fixated during visual search. *Attention, Perception, & Psychophysics, 1*(5), 315–318.