Advances in Cognitive Systems 6 (2018) 1-18 Submitted 6/2018; published 8/2018

SpatioTemporal Template-based Search: An Architecture to Model Human Search for Spatiotemporal Targets

Ellis Brown, II ELLIS.L.BROWN@VANDERBILT.EDU
Soobeen Park SOOBEEN.PARK@VANDERBILT.EDU
Noel Warford NWARFORD@OBERLIN.EDU
Electrical Engineering and Computer Science, Vanderbilt University, Nashville, TN 37235 USA

Adriane Seiffert A.SEIFFERT@VANDERBILT.EDU
Psychology, Vanderbilt University, Nashville, TN 37235 USA

Kazuhiko Kawamura KAZ.KAWAMURA@VANDERBILT.EDU
Electrical Engineering and Computer Science, Vanderbilt University, Nashville, TN 37235 USA

Joseph Lappin JOE.LAPPIN@VANDERBILT.EDU
Discerning Technologies, LLC, and Psychology, Vanderbilt University, Nashville, TN 37235 USA

Maithilee Kunda MKUNDA@VANDERBILT.EDU
Electrical Engineering and Computer Science, Vanderbilt University, Nashville, TN 37235 USA

Abstract

Visual search for a spatiotemporal target occurs frequently in human experience—from military or aviation staff monitoring complex displays of multiple moving objects to daycare teachers monitoring a group of children on a playground for risky behaviors. In spatiotemporal search, unlike more traditional visual search tasks, the target cannot be identified from a single frame of visual experience; as the target is a spatiotemporal pattern that unfolds over time, detection of the target must also integrate information over time. We propose a new computational cognitive architecture used to model and understand human visual attention in the specific context of visual search for a spatiotemporal target. Results from a previous human participant study found that humans show interesting attentional capacity limitations in this type of search task. Our architecture, called the SpatioTemporal Template-based Search (STTS) architecture, solves the same search task from the study using a wide variety of parameterized models that each represent a different cognitive theory of visual attention from the psychological literature. We present results from initial computational experiments using STTS as a first step towards understanding the computational nature of attentional bottlenecks in this type of search task, and we discuss how continued STTS experiments will help determine which theoretical models best explain the capacity limitations shown by humans. We expect that results from this research will help refine the design of visual information displays to help human operators perform difficult, real-world monitoring tasks.

1. Introduction

Visual search is something that humans do all the time; we look for our keys in the morning, we look for a particular file among the clutter of icons on our computer desktop, we look for the reddest basket of strawberries at the grocery store. These examples involve visual search for a static visuospatial target, whose existence can be defined by features that are visual and are fixed in time. The process of searching for these targets is often described as a series of spatial shifts in attention, with some visual target-detection process taking place at each location (Wolfe, 1994).

Humans also perform visual searches for dynamic targets, whose existence can be defined by features with a temporal dimension. For example, a lifeguard monitoring a swimming pool is watching not just for a particular visual feature (e.g., a swimmer being fully submerged underwater), but for the combination of this feature with temporal information (e.g., a swimmer being submerged for some length of time). This type of search—*visual search for a spatiotemporal target*—has different information processing requirements than traditional visual search. In particular, while search is still likely to proceed using a series of spatial shifts in attention, what differs is that the target-detection process itself has a temporal dimension. Targets cannot be detected from a single visual "frame" of perceptual experience. Instead, the information coming in from a particular location must be monitored continuously for some duration in order to decide whether the target is detected.

Despite its importance in everyday human activities, visual search for spatiotemporal targets has been less widely studied than traditional visual search—in psychology and artificial intelligence (AI). In this paper, we present a computational cognitive architecture called the *SpatioTemporal Template-based Search (STTS) architecture*, designed to support computational research into this type of visual search.

As a first step, we designed the STTS to solve a basic spatiotemporal search task, which we call the "chase task." Members of our research team recently studied the chase task with human partic-

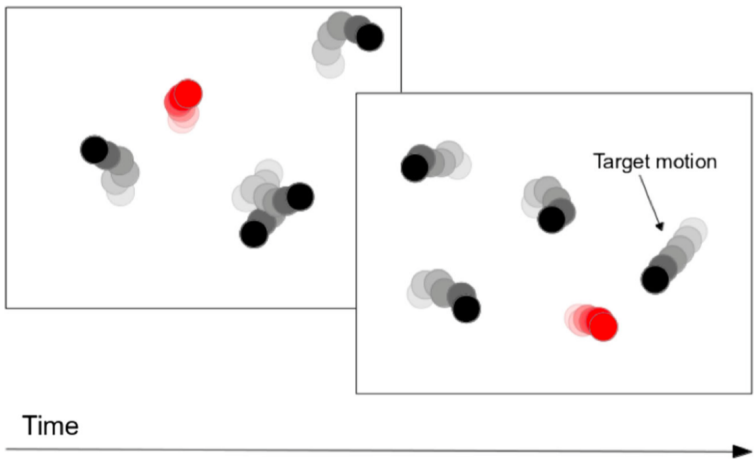

Figure 1. Schematic illustration of the chase task. At first, all of the black chaser dots move randomly and independently. At an unpredictable time during each trial, the motion of one chaser changes from random to pursuit of the red prey dot. This "chase" event is the spatiotemporal target that must be detected.

ipants, finding interesting evidence of attentional capacity limitations related to the spatiotemporal nature of the task (Lappin et al., 2016). Figure 1 shows a schematic illustration of this task.

In the chase task, a human participant watches a computer screen where some number of dots are moving around. One dot—the "prey"—is red, while all of the other dots—the "chasers"—are black. At first, all of the dots move randomly (Brownian motion with added momentum). Then, at some point during the trial, one of the black chaser dots will stop moving randomly and instead begin moving directly towards the red prey dot. This "chase" event is the spatiotemporal target that must be detected by the participant. Note that the participant cannot respond based on an instantaneous glance at the display; the target is defined as a spatiotemporal chase event that requires the participant to monitor the display for some duration in order to make a successful detection.

The participant responds with a key press as soon as they detect that the chase is taking place. If they do not detect the chase by the time the chaser has caught the prey, then the trial is scored as a false negative or missed detection. If they respond but no chase has yet occurred, then the trial is scored as a false positive. (Every trial has a chase event, and so there are no true negative trials.)

Here, we describe the design of the STTS architecture and how it can be used as a computational platform to explore different hypotheses about processes of attention and search on the chase task. This paper focuses primarily on the design of the architecture itself; future work will include more realistic agents and detailed comparisons with the human performance data. While the chase task is relatively simple, especially compared to spatiotemporal search in complex, naturalistic environments, we expect that studying this task will lay the foundations for continued research into more difficult spatiotemporal search tasks in the future.

2. The STTS Architecture

There are many ways that a human or an AI system could solve the chase task shown in Figure 1. The SpatioTemporal Template-based Search (STTS) architecture adopts one particular strategy out of many, but we acknowledge and emphasize that there are many other variations that would be possible and indeed plausible as accounts of human cognition. We expect that future work both by our group and by other research groups building computational models of human attention will help to uncover and characterize the diversity of attentional processes that are likely in use by humans across different individuals, tasks, and settings.

The approach currently employed by STTS, and which is the focus of this paper, is rooted in informal, introspective observations made by members of the research team about how they seemed to solve the chase task, and also builds on previous work on the role of visual imagery in reasoning and problem solving (Kunda et al., 2013; Kunda & Ting, 2016; Kunda, 2018).

In particular, the chase event often "looks" like straight line motion of a chaser towards the target, especially when the chase begins and the chaser is still far from the prey. (As the chaser nears the prey, the motion of the chaser becomes more curved.) STTS attempts to detect this straight line motion by storing information about the target as a multidimensional, spatiotemporal *template*—essentially as a "mental movie" that the architecture generates and compares to visually perceived information. This template-based formulation, along with the fact that the template is represented as a distributed, spatiotemporal entity (i.e., as an iconic, image-based representation instead of a

more abstract propositional representation) form the core parts of the cognitive theory that we are investigating. For the chase task explored in this paper, the template is defined as straight-line motion of a chaser towards the prey. For other tasks that we will explore in future work, the template could contain arbitrary motion patterns.

2.1 STTS Theory

There are three key components of the STTS theory proposed here: representing the search target as a spatiotemporal template in working memory, systematically deploying the template onto a search environment using selective attention, and making decisions about whether the search target has indeed been detected. We describe each of these components briefly in turn.

Spatiotemporal template. The internal representation of the search target, stored in working memory in the STTS architecture, is a spatiotemporal template that is iconic, i.e., the representation bears a structural resemblance to what it represents (Nersessian, 2008; Shimojima, 1999). In other words, the search template serves as both a spatial and temporal analogue to the actual search target in the outside world (i.e., in the external search environment within which STTS operates). This approach contrasts with other methods that parameterize aspects of the search target into a condensed set of features. While the distributed nature of a template-based representation makes it more memory intensive than a compressed, feature-based representation, a template-based encoding can represent arbitrary spatiotemporal targets with no change in memory usage. Parameterized representations generally require additional parameters to represent spatiotemporal patterns of increasing complexity.

Selective attention. In order to effect a visual search process, STTS has a control process that deploys the spatiotemporal template onto specific locations in the search environment at specific times during the search task—a form of selective attention. As in most computational theories of visual search, overall search performance (accuracy and efficiency) is heavily influenced by the effectiveness of this control process, which in turn is largely determined by the system's resource limitations, such as how much information working memory can store, the number of locations to which it can attend simultaneously, and the heuristics used to inform selection of specific targets for attention and switching of attention from one target to the next.

Decision making. Once a spatiotemporal template has been deployed onto the search environment, STTS decides whether it has detected the search target at that time and location. This step involves a comparison between the internal, template-based representation of the search target and the external visuospatial information perceived from the environment. Unlike static visual search, the dynamic, spatiotemporal nature of the search target means that this decision-making process must also unfold over time. STTS adopts an information-integration approach: as the internal spatiotemporal template is "run" in working memory, a comparison process evaluates a measure of similarity/difference between the template and the outside world. STTS also includes a decision threshold that determines whether a given output of this comparison corresponds to "detection" of the search target.

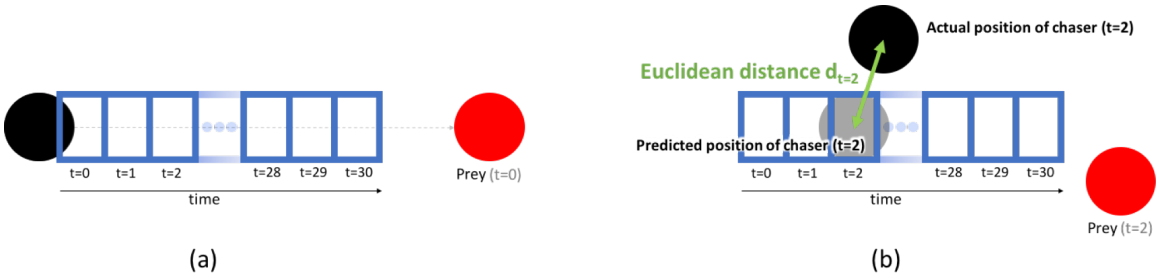

Figure 2. Illustration of steps in STTS algorithm. (a) A spatiotemporal template is generated to represent the predicted movements of a single chaser if it were, in fact, engaging in a chase. (b) At each timestep, the Euclidean distance between the chaser's predicted position and its actual position is calculated. These values are summed over all timesteps within the template to generate a cumulative difference measure.

2.2 Details of STTS Algorithm for the Chase Task

The general STTS approach given in the above section could be applied to any task that involves visual search for a spatiotemporally-defined target. In this section, we specify the details of the STTS algorithm as implemented to solve the chase task described in the introduction.

Let $t = 0$ be the timestep at which STTS begins to "attend" to a particular chaser C to detect whether a chase event is in progress. Let C_t represent the position of the chaser at timestep t after $t = 0$, and likewise let P_t represent the position of the prey.

First, a new template T is initialized to represent the predicted "mental movie" of what chaser C *would* be doing if it were chasing the prey. The template T consists of a time-indexed series of k predicted spatial positions, labeled here as $T = [T_0, T_1, T_2, ..., T_k]$. The initial position of the template T_0 is set to be equal to C_0, and the remaining positions $T_1, ..., T_k$ represent a motion path of constant velocity along a straight line from the current position of the chaser C_0 towards the current position of the prey P_0. Figure 2a shows an illustration of the initialized template.

Then, as STTS continues to observe the trial, it computes the cumulative difference $\Delta(C, T)$ between the predicted positions of chaser C, represented by positions in the template T, and the actual positions of chaser C. This cumulative difference is calculated as:

$$\Delta(C, T) = \sum_{t=1}^{k} distance(T_t, C_t) \tag{1}$$

where $distance$ is calculated as Euclidean distance between the two points. Figure 2b illustrates the comparison between C_t and T_t at one time step.

Finally, the calculated cumulative difference $\Delta(C, T)$ is compared to a pre-defined threshold value d. If $\Delta(C, T)$ is greater than or equal to d, then no chase has been detected. If, on the other hand, $\Delta(C, T)$ is less than d, then a chase has been detected, and STTS can generate a response accordingly. Essentially, a low $\Delta(C, T)$ value means that the actual movement of the chaser C was close enough to the predicted movement represented by T to suggest that a chase did, in fact, occur.

This approach is not perfect. For example, as shown in Figure 2, representing a chase event as a straight line between the chaser and the prey at a single timestep $t = 0$ does not capture more subtle

aspects of the chasing motion; as the prey continues to move after time $t = 0$, the straight-line-estimate becomes less and less accurate. However, as described more fully in the results section, this straight-line procedure is sufficiently accurate to generate successful performance on many of the chase task trials. Some failures can occur if the chase begins when the chaser is already quite close to the prey, which results in the chasing motion being much more strongly curved.

In addition, the performance of this method depends heavily on the specific assignments of many parameters within the architecture, including the template length k, the decision threshold d, and others. The following section describes these parameters in more detail.

2.3 STTS Parameters

The STTS architecture explores how the spatiotemporal, template-based target detection mechanism described in the previous section can be deployed in various ways to solve the chase task. Different strategies using this template approach are parameterized within the STTS architecture to represent different theoretical models of human attention and search. Parameters that can be used to customize the basic behavior of the architecture include:

- **Number of templates** n — Maximum number of templates that can be simultaneously deployed at any given timestep.
- **Template length** k — Number of timesteps for which the template predicts a chaser's path.
- **Decision threshold** d — At timestep t, the distance from a chaser's actual position C_t to its projected template position T_t is added to a cumulative distance measure $\Delta(C, T)$, as described in Equation 1. After k timesteps, the agent decides that a chase has begun if this cumulative difference $\Delta(C, T)$ is strictly less than a predefined threshold d.
- **Selection method** — How the next chaser is selected after an existing template expires:
  - *Serial*: Selects chasers in a fixed, arbitrary order.
  - *Uniform*: Selects chasers randomly with uniform probability.
  - *Linear*: Selects chasers randomly with probabilities linearly proportional to distance between prey and each chaser.
  - *Quadratic*: Selects chasers randomly with probabilities quadratically proportional to distance between prey and chaser.
- **Inhibition of return** (on/off with parameter c) — For the random selection methods (Uniform, Linear, Quadratic), this parameter can optionally be used to enforce a minimum duration c between any two selections of the same chaser.
- **Early kill** (on/off) — Determines whether templates are deployed for a fixed length of time k, or if they can be terminated early if the decision threshold has already been exceeded before the cumulative difference measure has been computed over the full template length.

In the rest of the paper, we use the term *agent* to refer to any one specific assignment of these parameters within the architecture, which represents a single, concrete, runnable search strategy. The initial set of agents presented in this paper can all loosely be described as models of conscious processing—simulated by deploying one or more templates to various chasers, with additional parameters for template length, decision thresholds, etc., as listed above.

3. Experimental Methods

Each trial of the chase task begins with the chasers and prey moving randomly. At some randomly-selected time, one of the chasers begins chasing the prey; we call this the *chase time*. The trial ends when the chaser "catches" the prey—i.e., runs straight into it—at the *catch time*. An agent being tested on a trial can generate a response at any time during the trial. Importantly, the response is just a simple yes/no; the agent does not need to specify which chaser was doing the chasing.

We create each trial by generating the positional coordinate data for the chasers and the prey for every timestep, using identical code to what was used to generate the visual displays for human participant studies. STTS currently takes these coordinate lists as input; future work will expand STTS to include a perceptual system, so that STTS can accept pixel-based visual displays as input.

For all experiments in this paper, we collected data from the same 1,000 trials in each experimental condition, as defined by the independent variables listed below. Across each set of 1,000 trials, we analyzed agent performance in terms of the dependent variables listed below.

Independent variables. The STTS architecture enables defining agents with different parameter settings, as described above. The selection of parameters for each particular agent is one independent variable in our experiments.

The chase task itself can be run with different numbers of chasers. For people, the task becomes more difficult—accuracy decreases, and/or reaction time increases—as the number of chasers goes up. The number of chasers thus represents the second independent variable in our experiments. Here, we evaluate each agent against trials containing 1 to 12 chasers, for a total of 12,000 trials.

Other task parameters that can be varied include the speed of each dot, its pattern of motion, whether the prey is red or black, etc., many of which have been explored in human studies (Lappin et al., 2016), but not yet using our model.

Dependent variables. During each trial, the agent has the option to generate a response at any time step. Scoring for each trial is defined as a combination of accuracy—did the agent correctly detect the chase?—and reaction time—how long after the start of the chase did the agent make its response (if at all)? Response accuracy is classified as:

- *True Positive* (TP) — a response was given between the chase and catch times (a.k.a. hit).
- *False Positive* (FP) — a response was given before the chase time (a.k.a. false alarm).
- *False Negative* (FN) — no response was given before the catch time (a.k.a. miss).

True negative outcomes are not possible in the chase task, because every trial has a chase event. The main accuracy measure we analyzed is the proportion of true positive (TP) responses given across each set of 1,000 trials.

As part of our analyses, we also tracked "lucky guess" TP scenarios, where the correct response was given but the agent made its decision off of a chaser that was not actually performing a chase. In observations of human behavior, it would be impossible to distinguish a true TP from a "lucky guess" TP, without asking the person to provide additional information about which chaser they were responding to. In experiments with the STTS architecture, we can differentiate the two types of TP responses, since the full decision-making trace is available for inspection. We found that the overall proportion of such lucky guesses was low across all of our experiments, though future

work will investigate how the likelihood of lucky guesses might change across variations in STTS parameter settings or experimental conditions of the task.

Reaction times are calculated as the time elapsed from the *chase time* until the agent's response. The main response time measure we analyzed is the median response time for true positive (TP) responses only, across each set of 1,000 trials.

4. Results and Discussion

We conducted two different types of computational experiments using STTS. In the first set of experiments, we show how specific parameters (in this case, template length k and decision threshold d) affect agent performance by creating a simple agent configuration and then examining various settings of these parameter values. In the second set of experiments, we show how theories of attention can be modeled in STTS by creating four "sample agents" that each represents a single (albeit simplified) theory of human attention, whose performance can then be compared to study performance trade-offs among these various theories.

4.1 Single-Parameter Experiments

In order to show how certain parameters affect performance of agents, we created a basic agent configuration and then varied specific parameters over a range of reasonable values. The basic parameter configuration we used is as follows:

- *Number of templates:* 1
- *Template length:* 20
- *Decision threshold:* 50
- *Early kill:* On
- *Selection method:* Uniform
- *Inhibition of return:* Off

This agent configuration attends to a single chaser at a time, for a maximum of $k = 20$ timesteps. At each timestep, it calculates the cumulative difference between its prediction of chaser position and the actual position (as given in Equation 1). If this cumulative difference value exceeds the decision threshold $d = 50$ at any time within the $k = 20$ timesteps, it will abandon the template and move on to the next chaser—selected at random from a uniform distribution. This basic configuration was identified using initial experiments with the STTS architecture as one that could generate reasonably successful performance on the chase task.

We conducted two experiments based on this basic agent configuration. Holding all other parameters constant, the first experiment studied the effects of varying the template length k, and the second experiment studied the effects of varying the decision threshold d.

4.1.1 Parameter Experiment 1: Template Length (k)

We first examine the effects of varying the length k of the template, which we evaluated in increments of ten, ranging from ten to fifty timesteps. The length of the template is analogous to the amount of time the agent attends to each chaser.

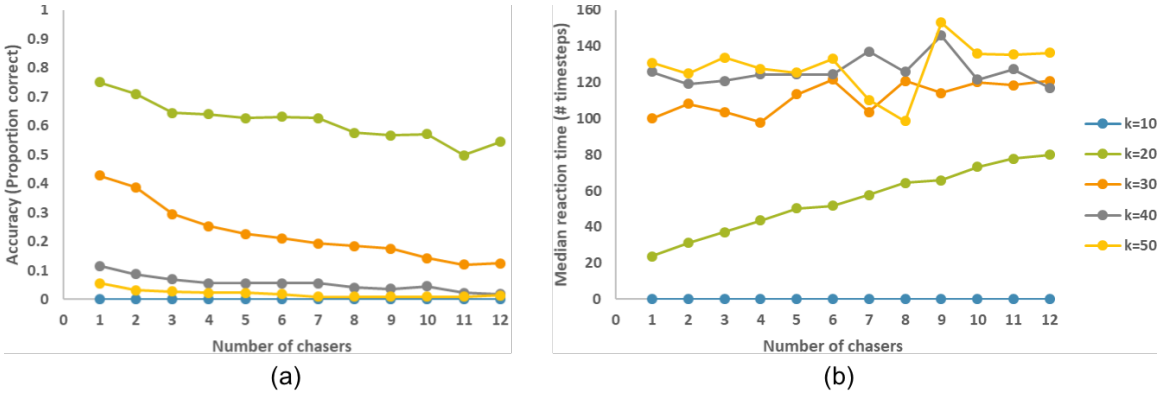

Figure 3. (a) Average accuracy, measured as proportion of true positive responses, and (b) median reaction time for correct responses, as a function of number of chasers for five different template lengths (k) ranging from ten to fifty. Each data point represents data from 1,000 trials.

We expect that on the low end of the template length spectrum, the agent will not pay attention long enough to any one chaser to notice the chase event. On the high end of the spectrum, we expect to see the agent focusing for too long on specific chasers that may be following a promising movement pattern (close to the straight line template projection), and therefore potentially missing a chase event occurring on one of the unattended chasers. Towards the middle of the spectrum, we expect to find a "sweet spot" for k such that the agent attends promising chasers for long enough to notice a chase event but not too long to miss chase events occurring with other chasers.

Results from this experiment are shown in Figure 3, showing average accuracy (left) and median reaction times (right). For all five settings of k, we notice that the accuracy tends to decrease as the number of chasers increases. This is an intuitively expected result. If there are more chasers to pay attention to, then it is more difficult to accurately respond to the chase when it occurs. For $k = 10$, the agent's accuracy is zero for all experiments, because the agent is responding to all trials with false alarms. For $k = 20$, the agent shows much better accuracy. However, for $k = 30$, the agent's accuracy decreases relative to $k = 20$. This is most likely caused by two factors. First, agents with longer template lengths spend too much time (relatively) on each chaser, and so are more likely to miss the occurrence of the chase event, especially with larger numbers of chasers. Second, when varying template lengths using the same absolute distance threshold, agents with shorter templates have a higher "relative threshold" than agents with longer templates. The further decreases in accuracy for $k = 40$ and $k = 50$ support this interpretation.

In the graph of median reaction times, the $k = 10$ agent has no correct responses and thus no measurable reaction time data. The median reaction time of the $k = 20$ agent shows a pattern of increasing median reaction time with increasing number of chasers, as expected.From $k = 20$ to $k = 30$, there is an increase in median reaction time. This is also an expected result, because paying attention to each chaser for a longer time means that the agent will take longer overall to cycle through all of the chasers. For agents with template lengths 30, 40, and 50, there is not much difference in median reaction times. This is likely due to the early kill parameter, which means that

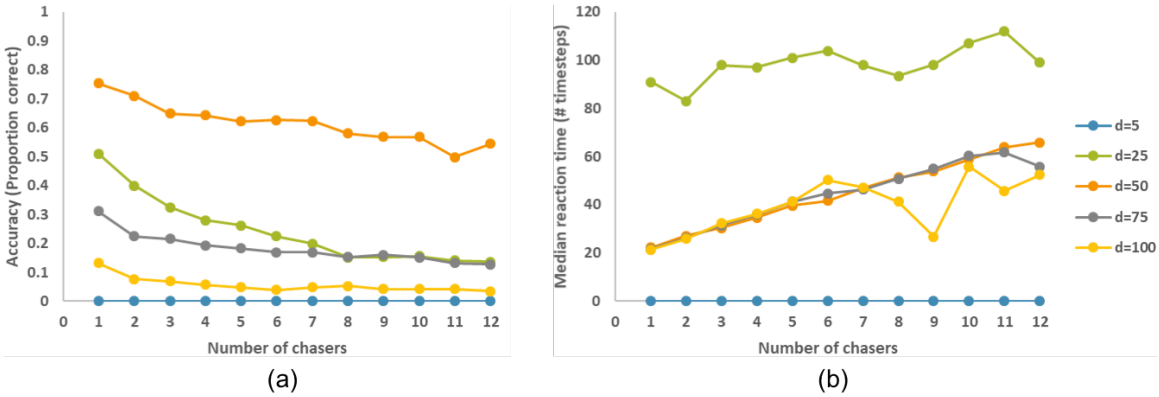

Figure 4. (a) Average accuracy, measured as proportion of true positive responses, and (b) median reaction time for correct responses, as a function of number of chasers for five different decision thresholds (d) ranging from 5 to 100. Each data point represents data from 1,000 trials.

agents do not have to wait for each template to be completed in its entirely for chasers that have already exceeded the decision threshold.

It is worth noting that the distance threshold is defined in these agents as an absolute distance, and not as a variable proportional to template length. Because a longer template allows a greater distance to accumulate, future work will investigate defining thresholds that can vary proportionally as a function of template length.

4.1.2 Parameter Experiment 2: Decision Threshold (d)

Next, we studied the effect of varying the decision threshold d across values of 5, 25, 50, 75, and 100. The threshold parameter is analogous to how closely the agent expects the chaser to follow its predicted path during a chase event.

We expect that on the low end of the decision threshold spectrum, the cumulative difference measure $\Delta(C, T)$ (see Equation 1) will quickly surpass the threshold—as the agent is expecting the chaser to follow a nearly perfectly straight path—thus causing the agent to shift attention to other chasers rapidly and frequently. Thus, low, overly strict thresholds should result in the chaser missing most of the true chase events. On the high end of the decision threshold spectrum, the agent will allow the chaser to follow the estimated straight-line path much less closely, and thus the agent will attend to each chaser for longer amounts of time. We expect this increased leniency to cause the agent to trigger false positive responses more often, as more chasers pass the decision threshold. Towards the middle of the spectrum, we expect to find a "sweet spot" where the agent is lenient enough to attend to each chaser for a reasonable length of time but is also strict enough to avoid false positive responses.

Results from this experiment are shown in Figure 4, showing average accuracy (left) and median reaction times (right). In this experiment, we again notice that the accuracy tends to decrease as the number of chasers increase, which, as noted before, is an expected result. The agent with threshold 5 produces no correct responses at all; all of its responses are false negatives. A threshold of 25

produces some correct responses, and increasing the threshold from 25 to 50 results in a further increase in accuracy. However, from threshold settings of 50 to 75 to 100, the accuracy decreases. These results are in line with our expectations about how accuracy would vary with threshold, as described above.

In the graph of median reaction times, the agent with a threshold of 5 has no correct responses and thus no reaction times to report. For every other agent, there is a general trend of increase in median reaction time with increase in number of chasers. A threshold of 25 results in higher median reaction times than a threshold of 50, while agents with thresholds 50, 75, and 100 report similar median reaction times. The decrease in reaction time from threshold of 25 to higher thresholds is somewhat surprising. This result could be due to the fact that with a threshold of 25, the agent will be switching attention from one chaser to the next more frequently, and thus will be slower in detecting a chase event once it has begun. Even with a single chaser, the likelihood that a chase begins within the time frame of a single template-based attentional period will be lower with a shorter threshold, though further study of this finding is warranted. Since reaction times are only collected for successful true positive response trials, total number of true positive results would also be helpful to provide context for this result.

4.2 Sample Agent Experiment

The parameters built into the STTS architecture are expressive enough to capture a very wide variety of agents. We selected four specific agent definitions to sample a diversity of attention/search strategies, with parameters listed below and visualizations shown in Figure 5.

- *Agent 1: Simple Serial Exhaustive* – one template at a time, no early killing, serial selection, and no inhibition of return.
- *Agent 2: Simple Serial Sophisticated* – one template at a time, early killing, quadratic selection, and inhibition of return.
- *Agent 3: Limited Multiple* – maximum of four templates at a time, early killing, quadratic selection, and inhibition of return.
- *Agent 4: Omniscient* – as many templates as chasers and early killing. The selection method is irrelevant, since all chasers have a template at all times.

These four definitions of agents are intended to demonstrate interesting tradeoffs between accuracy, reaction time, and computational resources for performing the chase task.

Agent 1 is a proof of concept agent that has the most simple setup possible. We expect it to perform poorly on the task (low accuracy, high reaction time), but be very lightweight to run—as the agent only needs to attend to a single template at a time. Agent 2 is an improved version of Agent 1. It is intended to show the improvements that can be made by more sophisticated parameter settings (e.g., early kill and inhibition of return) while still having a light computational workload. On the opposite end of the spectrum, Agent 4 is intended to demonstrate optimal performance at the expense of computational resources—the agent attends to all chasers at all times. Agent 3 is a happy medium between Agents 3 and 4, and represents a more realistic parameter setting with solid performance and computational workload. The idea of having 3–4 attentional slots in visual working memory is also roughly in line with studies of human cognitive limitations; future work

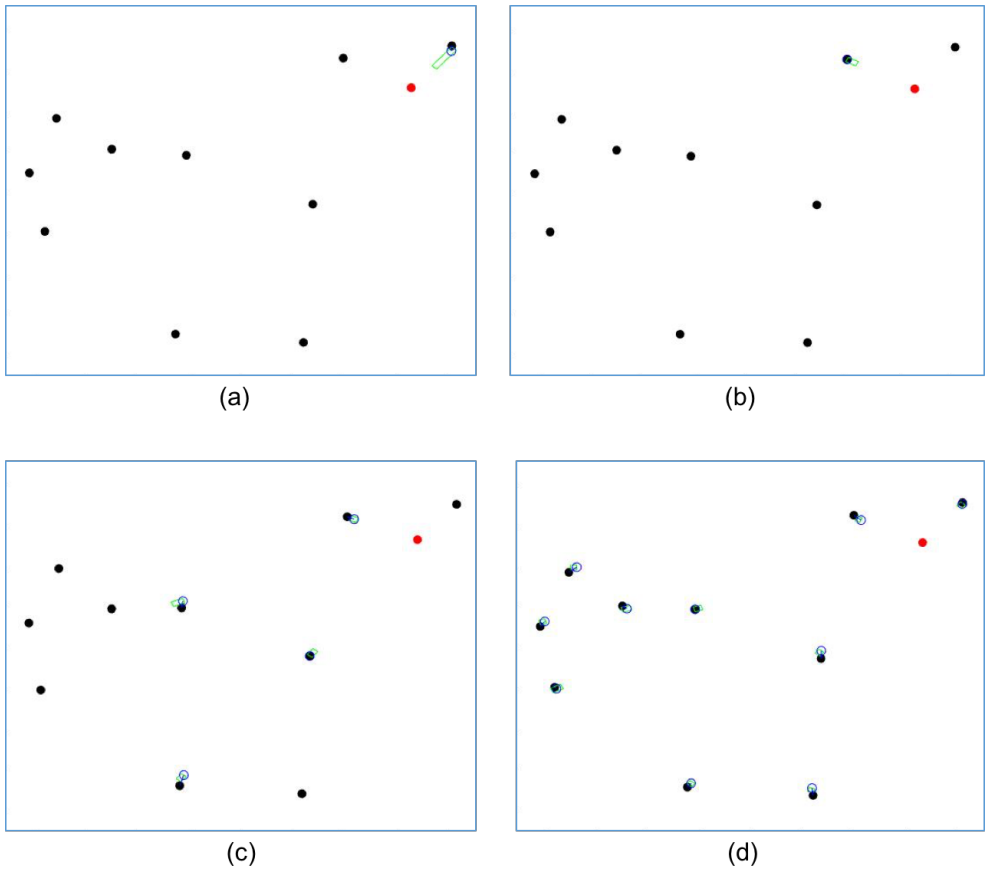

Figure 5. Visualizations of templates deployed by four sample agents: (a) Agent 1: Simple serial exhaustive; (b) Agent 2: Simple serial sophisticated; (c) Agent 3: Limited multiple; and (d) Agent 4: Omniscient. All panels show the same moment in time from a single trial of the chase task.

with STTS will also incorporate the notion of continuous (versus discrete), flexible resource pools as a model of memory capacity (Luck & Vogel, 2013).

We ran each of these four agents against the same 12,000 chase task trials that we used for our earlier parameter experiments. Results are summarized in Figure 6.

From the accuracy results shown in Figure 6a, we can see that Agent 2 (Simple Serial Sophisticated) performs better than Agent 1 (Simple Serial Exhaustive), which is to be expected given its improvements outlined above. Agents 3 (Limited Multiple) and 4 (Omniscient) have the same behavior when we vary the number of chasers between 1 and 4, because they are in fact the same agent for these particular task conditions. (An agent cannot have more than one template per chaser, and so for 1 to 4 chasers, Agents 3 and 4 each also deploy 1 to 4 templates, respectively.) However, for numbers of chasers higher than 4, we see that Agent 4 indeed performs better than Agent 3, though it expends significantly more computational resources to do so, computing templates for every single chaser all of the time for what turns out to be a fairly modest improvement in accuracy.

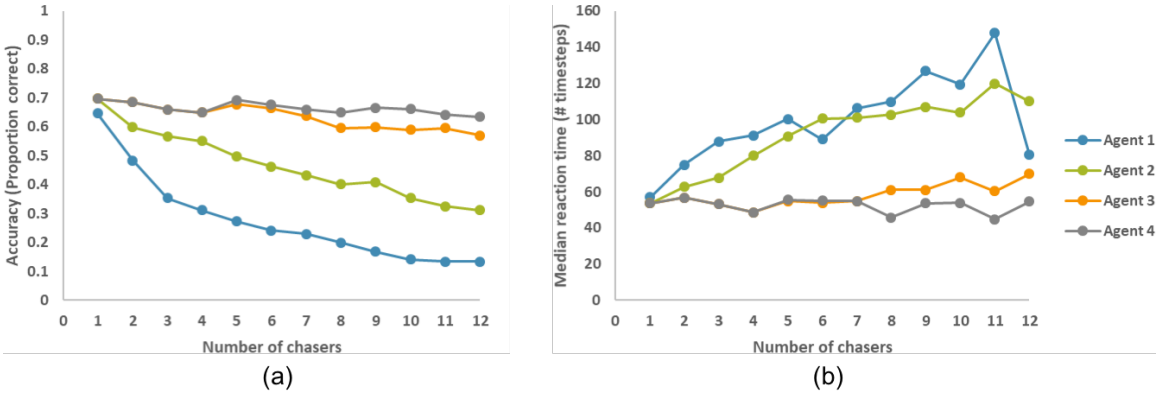

Figure 6. (a) Average accuracy, measured as proportion of true positive responses, and (b) median reaction time for correct responses, as a function of number of chasers for four different agents: Agent 1 (Simple serial exhaustive), Agent 2 (Simple serial sophisticated), Agent 3 (Limited multiple), and Agent 4 (Omniscient). Each data point represents data from 1,000 trials.

In the graph of median reaction times shown in Figure 6b, we see that our two most primitive agents—Agents 1 and 2—had the longest reaction times, as expected. Agent 1 generally had longer median reaction times than Agent 2.The more sophisticated agents—Agents 3 and 4—had faster reaction times than Agents 1 and 2, which is also an intuitive result. We see that the most sophisticated agent—Agent 4—outperformed all other agents in terms of getting the lowest median reaction time, which is expected, though (as with accuracy) a fairly modest improvement over the much more computationally efficient Agent 3.

5. Related Work

The STTS architecture differs from previous computational models of attention and search because it focuses on tasks in which target detection must must integrate information over time. This work relates to many research areas across psychology, neuroscience, and AI, which we briefly summarize in this section. Ongoing work with STTS includes more detailed comparisons to previous models of attention from each of these fields.

Research on attention and visual search includes behavioral studies (Treisman & Gelade, 1980), neuroimaging (Hopf et al., 2000), and computational modeling (Wolfe, 1994). The type of spatiotemporal search that is addressed in this paper relates to work on motion detection (Borst & Egelhaaf, 1989) and motion discrimination (Hanks et al., 2006), especially in the context of "chasing" behaviors in visually perceived inputs (Gao et al., 2009; van Buren et al., 2017) and multiple object tracking (Fencsik et al., 2007; Howe & Holcombe, 2012; Luu & Howe, 2015). This work is also relevant to findings from neuroscience that neurons in the visual cortex integrate information over time to capture spatiotemporal perceptual information (Buracas et al., 1998).

Researchers have also developed computational models of human visual search, including the role of visual salience (Bruce & Tsotsos, 2009; Itti & Koch, 2000), planning of gaze shifts (Rao

et al., 2002), and how target information is stored and accessed (Kunda & Ting, 2016). Strategies for visual search have also been studied within the context of general-purpose cognitive architectures such as EPIC (Hornof, 2004) and ACT-R (Fleetwood & Byrne, 2006), as well as specialized, attention-specific frameworks like Arcadia (Bridewell & Bello, 2016). However, the majority of these models focus on static visual search tasks, in which the targets can be defined using features at a single instant. As a result, they often focus on the spatial aspects of the task.

Regarding the temporal nature of search tasks, research in video processing has investigated techniques for recognizing actions (Le et al., 2011; Ciptadi et al., 2014), events (Johnson & Hogg, 1996), and behaviors (Dollár et al., 2005). Most of these efforts use feature-based representations that encode only some aspects of the target, usually ones that discriminate it from other elements in the environment. In other words, the selected features may *differentiate* the target from other entities, but they do not represent the target in its entirety. In contrast, template-based search is commonly used for static search tasks (Jain et al., 1998; Brunelli, 2009), including recognition of faces (Brunelli & Poggio, 1993), traffic signs (Gavrila, 1998), and medical images (Hill et al., 1994). There has been extensive research on flexible ways to represent templates and compute similarity, including the use of deformable templates (Yuille et al., 1992).

The STTS architecture provides an important bridge between these various areas of research in AI, i.e., computational cognitive models of human visual search, methods for spatiotemporal pattern recognition in videos, and techniques for template-based search in static visual search.

6. Contributions and Next Steps

In this paper, we have presented a computational cognitive architecture called the SpatioTemporal Template-based Search (STTS) architecture, for modeling visual search processes. Using this architecture, we designed and implemented a variety of parameterized "agents" that each can solve a spatiotemporal search task called the chase task (Lappin et al., 2016). Each agent essentially represents a concrete, runnable instantiation of a particular theory of attention/search. We present results from three different computational experiments using STTS as a first step towards understanding the computational nature of attentional bottlenecks in this type of search task.

One important finding that is implicit across all of our experiments is that the proposed STTS system is *capable* of solving the chase task (at least reasonably well) using the representations and processes described in this paper. Even though the system only represents an approximation of "chasing" behavior—i.e., straight line motion from a chaser's current position towards the prey, whereas the true motion of the chaser is actually a curve—this approximate representation is sufficient for achieving 70 to 80 percent accuracy in many cases.

In the previous human study that motivated this research, people can (with extensive practice) achieve accuracy rates in the upper 90 percents (Lappin et al., 2016). It remains to be seen whether further accuracy gains can be achieved using the STTS architecture as-is, perhaps by continuing to find more effective combinations of system parameters, or whether the ceiling of achievable accuracy is lower due to the nature of the STTS representations and processes themselves.

For instance, we have observed that the straight-line approximation does not work well when a chase event happens to start when the chaser is already quite close to the prey; in these cases, the true

chasing motion is more strongly curved. It may be that a more sophisticated type of template, one that updates the anticipated shape of the chase as the prey continues to move, may be instrumental in boosting STTS performance to higher levels.

Another limitation of the current STTS architecture is that all of the system parameters are fixed, and must be assigned *a priori* by the system designers. There is no learning that takes place from trial to trial. However, as mentioned above, people who undergo repeated practice on the chase task can eventually reach very high levels of accuracy, and are almost certainly "tuning" their own search strategies to achieve such high performance. The evolution of human performance from novice to expert on the chase task has not been explicitly studied but is quite interesting from a learning standpoint. How do people choose attentional strategies on the chase task? When, and how often, do they switch between strategies? How are their strategies affected by the feedback they receive over the course of completing many trials?

In order to model learning effects in the STTS architecture, we plan to incorporate reinforcement learning in order to enable agents to learn to optimize their own search strategies (Sprague & Ballard, 2004) in response to task feedback, which has potential as a model of practice/expertise effects in people. If the STTS architecture can vary its own parameters from trial to trial, instead of relying on a fixed set of parameters, then we should be able to study the effects of different task formulations on strategy selection and optimization. For example, if false positive responses receive strongly negative feedback, then the system ought to "select" more conservative parameter settings that prioritize accuracy over reaction time. If, on the other hand, false positive responses are not strongly penalized, but long reaction times are, then the system ought to learn to respond more aggressively and quickly, at the risk of generating more false positives.

Another important extension of STTS will address how attentional resource limitations are modeled in the architecture. Currently, resource limitations are modeled in STTS as a capacity limit on the number of templates that can be simultaneously deployed, and the length/size of each individual template in working memory. These two parameters are fixed, and so, for example, STTS cannot contain templates of varying lengths, or vary the number of templates deployed in response to changing task complexity. However, some research in psychology and neuroscience has proposed conceptualizing attentional and memory resources as continuous pools instead of discrete slots, so that, for example, a person can pay attention to fewer things with higher precision or to more things with lower precision (Van den Berg et al., 2012).

The STTS architecture offers a straightforward way to study these tradeoffs in memory models by designing agents that can flexibly allocate attentional resources according to current task demands. For example, instead of limiting the total number of templates all having some fixed length, we could instead place a limitation only on the total number of template timesteps that can be stored in working memory across any number of templates, and let the number and size of each template serve as additional parameters that could be optimized by a learning process within the system.

Continued research with the STTS architecture on the chase task will additionally include more detailed and sophisticated definitions of agents, based on prevailing psychological theories, as well as comparisons with human performance data. We have also begun experiments with the STTS architecture to explore models that incorporate subconscious (i.e., pre-attentive) processing—in which a layer of subconscious perceptual processes run continuously, and conscious attentional

processes then sample from the outputs of these pre-attentive processing stages. Finally, while the STTS architecture currently only addresses the visually simplistic chase task, incorporating a more sophisticated visual perception module that can detect rich visual features in naturalistic videos will enable extending the range of spatiotemporal search tasks that can be performed.

We expect that in the long run, results from this research will not only advance the basic cognitive science of human attention but will also inform the design of visual information displays to help people perform more efficiently and effectively on difficult, real-world monitoring tasks.

Acknowledgements

This work was supported in part by Contract #N00014-15-0024 to Discerning Technologies LLC and Vanderbilt University from the Office of Naval Research, which is not responsible for the content and conclusions contained herein.

References

Van den Berg, R., Shin, H., Chou, W.-C., George, R., & Ma, W. J. (2012). Variability in encoding precision accounts for visual short-term memory limitations. *Proceedings of the National Academy of Sciences*, *109*, 8780–8785.

Borst, A., & Egelhaaf, M. (1989). Principles of visual motion detection. *Trends in Neurosciences*, *12*, 297–306.

Bridewell, W., & Bello, P. (2016). A theory of attention for cognitive systems. *Advances in Cognitive Systems*, *4*, 1–16.

Bruce, N. D., & Tsotsos, J. K. (2009). Saliency, attention, and visual search: An information theoretic approach. *Journal of Vision*, *9*, 1–24.

Brunelli, R. (2009). *Template matching techniques in computer vision: Theory and practice*. Chichester, United Kingdom: John Wiley & Sons.

Brunelli, R., & Poggio, T. (1993). Face recognition: Features versus templates. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *15*, 1042–1052.

Buracas, G. T., Zador, A. M., DeWeese, M. R., & Albright, T. D. (1998). Efficient discrimination of temporal patterns by motion-sensitive neurons in primate visual cortex. *Neuron*, *20*, 959–969.

van Buren, B., Gao, T., & Scholl, B. J. (2017). What are the underlying units of perceived animacy? Chasing detection is intrinsically object-based. *Psychonomic Bulletin & Review*, *24*, 1604–1610.

Ciptadi, A., Goodwin, M. S., & Rehg, J. M. (2014). Movement pattern histogram for action recognition and retrieval. *Proceedings of the European Conference on Computer Vision* (pp. 695–710). Zürich, Switzerland: Springer.

Dollár, P., Rabaud, V., Cottrell, G., & Belongie, S. (2005). Behavior recognition via sparse spatio-temporal features. *Proceedings of the Second Joint IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance* (pp. 65–72). Beijing, China: IEEE.

Fencsik, D. E., Klieger, S. B., & Horowitz, T. S. (2007). The role of location and motion information in the tracking and recovery of moving objects. *Perception & Psychophysics*, *69*, 567–577.

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*, 153–197.

Gao, T., Newman, G. E., & Scholl, B. J. (2009). The psychophysics of chasing: A case study in the perception of animacy. *Cognitive Psychology*, *59*, 154–179.

Gavrila, D. M. (1998). Multi-feature hierarchical template matching using distance transforms. *Proceedings of the Fourteenth International Conference on Pattern Recognition* (pp. 439–444). Brisbane, Queensland, Australia: IEEE.

Hanks, T., Ditterich, J., & Shadlen, M. (2006). Microstimulation of macaque area LIP affects decision-making in a motion discrimination task. *Nature Neuroscience*, *9*, 682.

Hill, A., Cootes, T., Taylor, C., & Lindley, K. (1994). Medical image interpretation: A generic approach using deformable templates. *Medical Informatics*, *19*, 47–59.

Hopf, J.-M., Luck, S. J., Girelli, M., Hagner, T., Mangun, G. R., Scheich, H., & Heinze, H.-J. (2000). Neural sources of focused attention in visual search. *Cerebral Cortex*, *10*, 1233–1241.

Hornof, A. J. (2004). Cognitive strategies for the visual search of hierarchical computer displays. *Human-Computer Interaction*, *19*, 183–223.

Howe, P. D., & Holcombe, A. O. (2012). Motion information is sometimes used as an aid to the visual tracking of objects. *Journal of Vision*, *12*, 10–10.

Itti, L., & Koch, C. (2000). A saliency-based search mechanism for overt and covert shifts of visual attention. *Vision Research*, *40*, 1489–1506.

Jain, A. K., Zhong, Y., & Dubuisson-Jolly, M.-P. (1998). Deformable template models: A review. *Signal Processing*, *71*, 109–129.

Johnson, N., & Hogg, D. (1996). Learning the distribution of object trajectories for event recognition. *Image and Vision Computing*, *14*, 609–615.

Kunda, M. (2018). Visual mental imagery: A view from artificial intelligence. *Cortex*, *105*, 155–172.

Kunda, M., McGreggor, K., & Goel, A. K. (2013). A computational model for solving problems from the raven's progressive matrices intelligence test using iconic visual representations. *Cognitive Systems Research*, *22*, 47–66.

Kunda, M., & Ting, J. (2016). Looking around the mind's eye: Attention-based access to visual search templates in working memory. *Advances in Cognitive Systems*, *4*, 113–129.

Lappin, J. S., Morse, D. L., & Seiffert, A. E. (2016). The channel capacity of visual awareness divided among multiple moving objects. *Attention, Perception, & Psychophysics*, *78*, 2469–2493.

Le, Q. V., Zou, W. Y., Yeung, S. Y., & Ng, A. Y. (2011). Learning hierarchical invariant spatio-temporal features for action recognition with independent subspace analysis. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 3361–3368). Colorado Springs, CO: IEEE.

Luck, S. J., & Vogel, E. K. (2013). Visual working memory capacity: From psychophysics and neurobiology to individual differences. *Trends in Cognitive Sciences*, *17*, 391–400.

Luu, T., & Howe, P. D. (2015). Extrapolation occurs in multiple object tracking when eye movements are controlled. *Attention, Perception, & Psychophysics*, *77*, 1919–1929.

Nersessian, N. J. (2008). *Creating scientific concepts*. Cambridge, MA: MIT Press.

Rao, R. P., Zelinsky, G. J., Hayhoe, M. M., & Ballard, D. H. (2002). Eye movements in iconic visual search. *Vision Research*, *42*, 1447–1463.

Shimojima, A. (1999). The graphic-linguistic distinction: Exploring alternatives. *Artificial Intelligence Review*, *13*, 313–335.

Sprague, N., & Ballard, D. (2004). Eye movements for reward maximization. *Proceedings of the 2004 Conference on Advances in Neural Information Processing Systems* (pp. 1467–1474). Vancouver, British Columbia, Canada: MIT Press.

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

Wolfe, J. M. (1994). Guided search 2.0: A revised model of visual search. *Psychonomic Bulletin & Review*, *1*, 202–238.

Wolfe, J. M. (1998). What can 1 million trials tell us about visual search? *Psychological Science*, *9*, 33–39.

Yuille, A. L., Hallinan, P. W., & Cohen, D. S. (1992). Feature extraction from faces using deformable templates. *International Journal of Computer Vision*, *8*, 99–111.