Available online at www.sciencedirect.com

SciVerse ScienceDirect

Cognitive Systems Research 24 (2013) 62–71

Cognitive Systems RESEARCH

www.elsevier.com/locate/cogsys

Pre-attentive and attentive vision module

Action editor: Nele Rußwinkel

Enkhbold Nyamsuren *, Niels A. Taatgen *

Department of Artificial Intelligence, University of Groningen, Nijenborgh 9, 9747 AG Groningen, Netherlands

Available online 16 January 2013

Abstract

This paper introduces a new vision module, called PAAV, developed for the cognitive architecture ACT-R. Unlike ACT-R's default vision module that was originally developed for top-down perception only, PAAV was designed to model a wide range of tasks, such as visual search and scene viewing, where pre-attentive bottom-up processes are essential for the validity of a model. PAAV builds on attentive components of the default vision module and incorporates greater support for modeling pre-attentive components of human vision. The module design incorporates the best practices from existing models of vision. The validity of the module was tested on four different tasks.

Keywords: Vision; Iconic memory; Cognitive architecture; ACT-R

1. Introduction

This paper introduces a general purpose vision module called PAAV, which stands for Pre-attentive And Attentive Vision. As the name suggests, the new module incorporates a greater support for bottom-up visual components that are considered pre-attentive in nature, such as multiple feature dimensions to describe visual objects, peripheral vision with differential acuity, iconic visual memory and a decision threshold. The module was developed as an integral part of ACT-R[1] cognitive architecture (Anderson, 2007) that provides a necessary top-down, attentive layer. By being part of ACT-R, PAAV should be able to model wide range of tasks where both top-down and bottom-up visual guidances are important. ACT-R already has a default vision module and a few extensions for it. However they have drawbacks that PAAV is aimed to solve.

ACT-R's default vision module can be described in terms of a *visicon* and two buffers: *visual-location* and *visual*. *Visual-location* and *visual* buffers essentially represent WHERE and WHAT components of a visual system. The *visicon* represents the visual scene containing visual objects with which an ACT-R model can interact. The *visicon* is considered to be a part of the environment (a monitor screen) rather than part of the model. A model can send a WHERE request to the *visual-location* buffer to find the location in the *visicon* of a potential visual object to encode. Within this request, the model can specify criteria for visual object such as its kind, color, coordinates or size. Given this request vision module randomly chooses one of the visual objects from the *visicon* that exactly matches the given criteria and puts its location information in the *visual-location* buffer. This entire process is instantaneous with no time cost. Next, model can send a WHAT request to the *visual* buffer to encode the object at the chosen location of *visicon*. A WHAT request assumes fixed execution times for both saccade and encoding that in total require 85 ms. This value, although it can be changed by the modeler through a dedicated parameter, is considered as a de facto standard in ACT-R.

* Corresponding authors. Tel.: +31 50 363 7450/6435; fax: +31 50 363 6687.

E-mail addresses: e.nyamsuren@rug.nl (E. Nyamsuren), n.a.taatgen@rug.nl (N.A. Taatgen).

[1] ACT-R stands for *Adaptive Control of Thought-Rational*.

1389-0417/$ - see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.cogsys.2012.12.010

EMMA (Salvucci, 2001) is arguably the most used extension to ACT-R's it default vision module. EMMA explicitly models saccades including preparation and execution times, path generation and variable landing points. However, EMMA's major contribution is in its ability to model covert attention shifts through variable encoding time dependent on visual object's frequency and eccentricity.

The disadvantage of the default vision module and EMMA is their optimization toward reading tasks or tasks with a relatively simple visual environment where bottom-up perceptual processes can be ignored without sacrificing the model's plausibility and performance. However, ACT-R's vision module is not suitable for tasks where visual stimuli are described with multiple feature dimensions. Such tasks often require theories of scene perception and visual search that are not part of current vision module. The issue is more pressing if one considers the importance of embodied cognition (e.g., Clark, 1997) in problem-solving tasks (Nyamsuren & Taatgen, 2013) and in everyday human activities in general (Land, Mennie, & Rusted, 1999). Embodied cognition assumes that cognitive control is not purely goal based, but it is also driven perceptually. The simplest example of it is an interference of the salient feature during the task (Theeuwes, 1992). When subjects are asked to look at the scene they tend to look at the most salient parts first. Those salient parts of the scene can interfere with task even if subjects are explicitly asked to not to look at them.

2. Architecture of PAAV module

2.1. Feature dimensions

In PAAV every visual object can be characterized by five basic features: color, shape, shading, orientation and size. The features are chosen because of their pop-out nature and importance in guiding visual attention (Wolfe & Horowitz, 2004). Each of those features can have a wide range of values, such as, red and green for color; and oval and rectangle for shape. Currently, PAAV does not support modeler specified custom features. However, it is included as a future implementation milestone.

2.2. Peripheral vision

The current implementation of ACT-R's vision assumes that everything in a visicon is visible to the vision module and consecutively available for information processing. However, human vision is limited in what it can see, especially in the extra-foveal region (Rayner, 1998). PAAV introduces limitations on visibility by assuming that a visual object is only visible if at least one of five features of that object is visible. Visibility of a feature is calculated with an acuity function. We have adopted a modified version of the psychophysical acuity function proposed by Kieras (2010). Kieras' original acuity function states that for an object's feature to be visible the object's angular size s, with some Gaussian noise added to it, must exceed a threshold calculated as a function of eccentricity e:

$$threshold = ae^2 + be + c$$

$$P(available) = P(s + X > threshold)$$

$$X \sim N(0, vs)$$

The free parameters a, b, c and v are to be adjusted for each particular feature. The function works quite well for modeling differential acuity of features. However, the quadratic form in the function makes it less suitable when the object size is particularly small. For example, in their feature search experiment for color, Treisman and Gelade (1980) used visual stimuli of $0.8° \times 0.6°$ in size scattered over area of $14° \times 8°$. This feature search experiment cannot be replicated with the above acuity function for color unless parameter a is assigned an extremely low value that is well below the 0.035 used by Kieras (2010).

PAAV uses a modified version of the acuity function to mitigate issue above:

$$threshold = ae^2 - be$$

$$P(available) = P(s > threshold)$$

The constant c has been removed since it has no significant influence when object size is reasonably large and too much influence when object size is quite small. Similarly, the Gaussian noise has been removed because of its tendency to introduce too much or too little acuity variation depending on the object size. Next, the coefficient b has an opposite sign. It results in less steeper increase in threshold when an eccentricity increases. It also removes the necessity of giving unreasonably small value to coefficient a when object size is small. The free parameter a has been refitted again to 0.104, 0.147, 0.14 and 0.142 for color, shading, size and shape respectively. The parameter b has been fitted to 0.85 for color and 0.96 for all other features. We are still in process of fitting parameters for the orientation feature.

2.3. Iconic visual memory

Everything PAAV perceives from the *visicon* is stored in iconic memory. Visual features of every object visible via peripheral vision are stored in this memory. As such, the content of iconic memory is not necessarily a complete or even a consistent representation of the objects in the *visicon*.

Information in iconic memory is not treated as consciously perceived visual properties. It is rather perceived as bottom-up visual stimuli on which bottom-up processes can operate. Iconic memory is trans-saccade persistent. Items in iconic memory are persistent for a short duration of time if they are not visible through peripheral vision anymore. The parameter for persistence time is currently set to 4 s, as determined by Kieras (2009), to be a lower bound for a visual memory.

Iconic memory is a model's internal representation of a *visicon*, otherwise visual scene. As such, all WHERE

requests are handled with respect to the content of iconic memory via a newly defined *abstract-location* buffer, a replacement to now depreciated *visual-location* buffer. A request may include desired criteria including any of the five feature dimensions or location.

2.4. Visual activation

Each visual object in iconic memory is assigned an activation value. The location of the visual object with the highest activation value is returned upon a WHERE request. The activation value is calculated as a sum of bottom-up and top-down activation values. It is adapted from the concept of an activation map used by Wolfe (2007) in his model of a visual search.

2.4.1. Bottom-up activation

The bottom-up activation for a visual object i is calculated based on its contrast to all other objects in iconic memory with respect to each feature dimension k:

$$BA_i = \sum_{j}^{visual\ \ objects} \sum_{k}^{features} \frac{dissim(v_{ik}, v_{jk})}{\sqrt{d_{ij}}}$$

The $dissim(v_{ik}, v_{jk})$ is the dissimilarity score of two feature values of the same dimension. It is a simplification of a bottom-up activation based on the difference in channel responses used in Guided Search 4.0 (Wolfe, 2007). If two values are the same then $dissim(v_{ik}, v_{jk}) = 0$, otherwise $dissim(v_{ik}, v_{jk}) = 1$. The dissimilarity is weighted by a square root of a linear distance d_{ij} between two objects. Thus the objects farther away contribute less to a contrast-based saliency of the visual object i than the objects closest to it.

2.4.2. Top-down activation

In a WHERE request a model can provide feature values as desired criteria for the next visual object to be located. Those feature criteria are used to calculate the top-down activation value for each visual object in iconic memory. Given k feature criteria the top-down activation for visual object i is calculated as:

$$TA_i = \sum_{k}^{feature\ \ criteria} sim(f_{ik}, f_k)$$

$sim(f_{ik}, f_k)$ is a similarity score of the feature value f_k in WHERE request to a value f_{ik} with the same feature dimension in visual object i. This similarity score is 1 for an exact match and 0 for a mismatch. If the value f_{ik} is not accessible from iconic memory then the similarity score is 0.5. Thus uncertainty is preferred to certain dissimilarity.

2.4.3. Total visual activation

The total activation for visual object i is the sum of bottom-up and top-down activations:

$$VA_i = W_{BA} * BA_i + W_{TA} * TA_i + N$$

W_{BA} and W_{TA} are the weight parameters for the bottom-up and top-down activations respectively. They are set to 1.1 and 0.45. In correspondence with Wolfe (2007), those weights control the unintentional and intentional attentional captures. The bottom-up activation is given a higher weight to compensate for the distance d_{ij} adjustment, which results in the lower bottom-up activation value in comparison to the top-down activation value. N is noise from a logistic distribution with variance σ^2 calculated as a function of a parameter s: $\sigma^2 = s^2\pi^2/3$. s is set to 0.2 by default.

2.5. Saccade and encoding

After a visual object has been located with a WHERE request, a model can send a WHAT request. This is essentially the same encoding processes of a visual object from the *visicon* as in ACT-R's default vision module. However, PAAV assumes that the saccade that precedes the encoding has a variable execution time dependent on the saccade's amplitude. Prior to a saccade execution, PAAV calculates its duration and landing point. Salvucci (2001) described a set of formulas to calculate those variables. For calculating the execution duration, we used EMMA's default parameters: 20 ms as a base execution time plus additional 2 ms for an every degree of angular distance covered by a saccade. Differently from Salvucci (2001), we have used two Gaussian distributions around the center of the object to calculate saccade's landing position. The standard deviation for distribution along X axis is calculated as s_g times of the object's linear width, where s_g is a gaze noise parameter set to 0.5. In a similar manner, the standard deviation for Y axis is calculated using object's linear height. Such implementation is in accordance with theory that the saccade's landing position depends on the size of a visual stimulus (Rayner, 1998).

Upon completion of a saccade, PAAV starts encoding. The parameter for encoding time is 50 ms. It is in line with findings that the sufficient information is encoded in the first 45–75 ms of a fixation for an object identification to occur (van Diepen, De Graef, & d'Ydewalle, 1995). Except eccentricity, Salvucci (2001) used word frequency to calculate variable encoding time. However, we believe this approach is not applicable to PAAV where visual object is defined along multiple dimensions. Hence, further study is needed to investigate the object's encoding process in more details sufficient for proper computational modeling.

2.6. Visual decision threshold

One of the challenging problems in a visual perception is how does the visual system recognize the absence of a desired visual object. For example, humans can spot the absence of a salient object as fast as its presence in a visual field (Fig. 1). Similarly, given a WHERE request with specific criteria, how does PAAV know that the desired object is not in iconic memory. One obvious solution is to attend every object in *visicon* and stop when there are no more

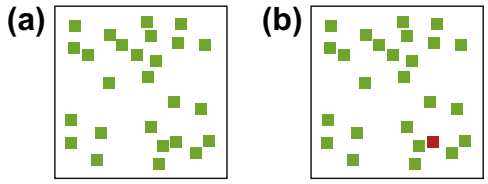

Fig. 1. Humans can spot an absence (a) of a red object in field of green objects as fast as its presence (b). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

objects to attend. However, visual search paradigms, such as feature search, show that it is not the case. The visual system is much more efficient and does not require fixation on every item to detect an absence of a target (Treisman & Gelade, 1980; Wolfe, 2007).

PAAV incorporates the concept of a visual decision threshold to decide whether any of the objects in iconic memory will match a given WHERE request. A partial solution is to ignore every object that has zero top-down activation due to complete mismatch. However, results from tasks, such as conjunction search, show that a visual search can be efficient even when distracters partially match the target. PAAV should also be able to filter out objects that match only partially. This is done via simulation of visual grouping based on top-down activation. Given a WHERE request, PAAV returns some object i. Let's assume that, at the time of WHERE request, the distance between object i and the gaze position was d_{Th}, and object i's top-down activation was TA_{Th}. When object i is encoded these two values are stored and used as a threshold for the consecutive WHERE requests. In the following WHERE requests PAAV completely ignores every object j in iconic memory that has $TA_j \leqslant TA_{Th}$ and $d_j \leqslant d_{Th}$ where d_j is a distance between object j and gaze position. Top-down activation serves as a natural threshold for object selection. Every time a model encodes an incorrect object, the acceptance threshold for the next WHERE request increases up to the activation value of that object. The distance d_{Th} provides a measure that PAAV uses to judge whether it can reliably compare two top-down activation values. It is a simulation of a visual grouping where a cluster of similar objects is grouped together. The d_{Th} can be viewed as an approximate radius of the cluster.

2.6.1. Step by step example

Let us consider an example in which the model is looking for a red square, but there are only three green squares in the iconic memory. The example is depicted in Fig. 2. In this example the model is able to notice the absence of a red square after only one fixation.

When the model sends the first WHERE request, the module calculates distance d_j between each object j in iconic memory and model's current gaze position (depicted as a black cross). It also calculates the top-down activation TA_j for every object (for the sake of simplicity the bottom-up activation is ignored). All objects receive a top-down activation of one for matching the requested shape feature. Since all objects have the same activation values, let us assume that the module randomly returns the location of the second object as the next object to be fixated. The state of iconic memory after the first WHERE request is depicted as state A in Fig. 2.

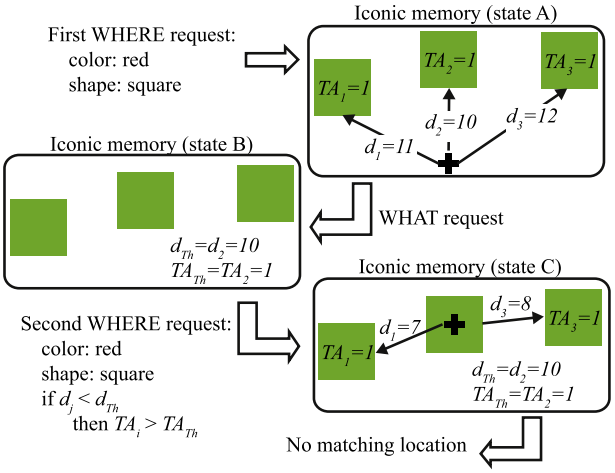

Fig. 2. Example usage of a visual decision threshold.

The first WHERE request is followed by WHAT request. Given this request, the module stores the value of d_2, the distance between the current gaze position and the second object, as the distance threshold d_{Th}. The module also stores the second object's top-down activation value TA_2 as an activation threshold TA_{Th}. After those steps, the module triggers a saccade execution, changes the gaze position to the location of the second object and encodes the object. At this point iconic memory transitions into state B.

Since the encoded object is not a red square, the model sends a second WHERE request. However, this time the model includes the distance and top-down activation thresholds as request parameters along with the color and shape values. The threshold parameters state that if the object's distance from current gaze location is less than distance threshold d_{Th} then the object's top-down activation should be higher than the activation threshold TA_{Th} for the object to be considered a next valid destination to be attended. So, in the current example, there are two unattended objects in iconic memory (state C in Fig. 2). The distances to both objects from the current gaze location, seven and eight respectively, are less than threshold distance of 10. Therefore, both objects should have a top-down activation that is higher than activation threshold of one. This is not the case since both objects again have a top-down activation of only one because of the color mismatch. Hence, the PAAV module lets the model know that there are no more locations to attend. In turn, the model knows that there is no red squared object in iconic memory.

In the example, three green objects are treated as a cluster of similar objects rather than three individual objects

each needing separate attention. The distance threshold d_{Th} can be viewed as a maximum radius of the cluster, while activation threshold TA_{Th} is a maximum dissimilarity threshold within which objects can be considered members of the cluster.

2.7. Spreading activation from iconic memory

Lastly, PAAV module introduces spreading activation from visual iconic memory to declarative memory. It has long been observed that visual stimuli can influence the result of a memory retrieval (Wais, Rubens, Boccanfuso, & Gazzaley, 2010). PAAV's spreading activation mechanism was developed to replicate this cognitive process.

ACT-R's declarative memory is a long term memory where knowledge is stored in the form of chunks with slots. One chunk usually represents one concept, while concept properties can be described through values assigned to chunk slots. The model can retrieve only one chunk at the time, and, when there are several chunks that match the retrieval request, the one with the highest activation value has the highest probability of retrieval. There are can be different sources of activation for a chunk, and chunk's total activation is a sum of activations from all available sources.

In the PAAV module, visual objects in iconic memory also serve as sources of activation. Visual feature values from all visual objects spread activation to all chunks in declarative memory that have the same visual feature values as slot values. For example, each green object in iconic memory spreads activation to all green objects in declarative memory. Let us assume there is a chunk k in declarative memory, and it receives a total spreading activation of S_k from iconic memory. Then S_k is calculated as:z

$$S_k = W * \sum_{i \in V} (S + ln(1 + fan_{ik}))$$

V is a set of all slots from chunk k that have any visual feature value as a slot value. fan_{ik} is a normalized value indicating a number of visual objects in iconic memory that have the same feature value as the chunk k in its slot i. In ACT-R fan_{ik} has to be normalized because a chunk, technically, can have infinite number of slots and the same value in two or more slots. We will not go into the details of normalization since it is ACT-R specific. S, a parameter for the minimum associative strength, indicates the minimum amount of activation that should be spread. W, a parameter for association weight, is a weight of total spreading activation from iconic memory. By default, S and W are set to 0 and 0.7 respectively. With the addition of S_k the default activation equation for declarative memory changes to:

$$A_k = B_k + S_k + P_k + \varepsilon_k + S_k$$

3. Validation models

This section describes two models that do visual search tasks and a more complex model of a player for a game of SET that requires both top-down and bottom-up cognition. All models are based on ACT-R with the default vision module replaced by the PAAV module. The tasks are simple, yet demand complex cognitive and perceptual processes, and require most of the components of the PAAV module described in this paper. Hence, those tasks serve as a good way to validate the PAAV module. All models use the same default values for PAAV parameters described in this paper with the only exception that top-down activation weight W_{TA} is increased to 3.0 in the model of SET to account for a higher top-down cognitive load.

3.1. A model of feature and conjunction searches

The first model was created to do feature and conjunction searches. Both of these visual search tasks involve finding a target among a set of distracters. In a feature search task the target differs from distracters by a single feature such as color (Fig. 3a). In a conjunction search the target can differ from distracters by either of two features (Fig. 3b). A feature search is usually an efficient search with reaction time being independent of a number of distracters. On the other hand, reaction time in a conjunction search increases with a number of distracters. Those results are consistent among different studies (e.g., Treisman & Gelade, 1980; Wolfe, 2007; Wolfe, Cave, & Franzel, 1989).

The goal in feature search was to find a red rectangle among green rectangles. In a conjunction search, the model had to find a red rectangle among green rectangles and red ovals. In each trial values for both shape and color were present in near equal amount.

The following experimental conditions were set for the model. In both types of visual search tasks, the set size ranged from 1 to 30. For each set size, there were 500 trials where a target was present and another 500 trials where a target was replaced with a distracter. In total, there were 6000 trials in each of feature and conjunction search tasks. The screen size was 11.3° × 11.3°, and the size of each object was 0.85° both in width and height. Within the screen, objects were positioned in a random pattern with the constraint that they should not overlap. The model had to press either "P" or "A" for target being either present or absent. The time of key press was considered as trial end time. The model was reset after each trial.

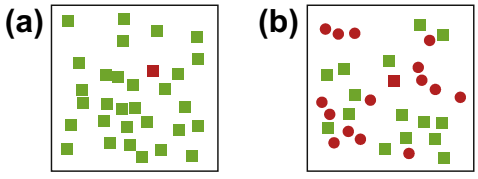

Fig. 3. Examples of feature search (a) and conjunction search tasks (b). In both tasks the red rectangle is a target. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

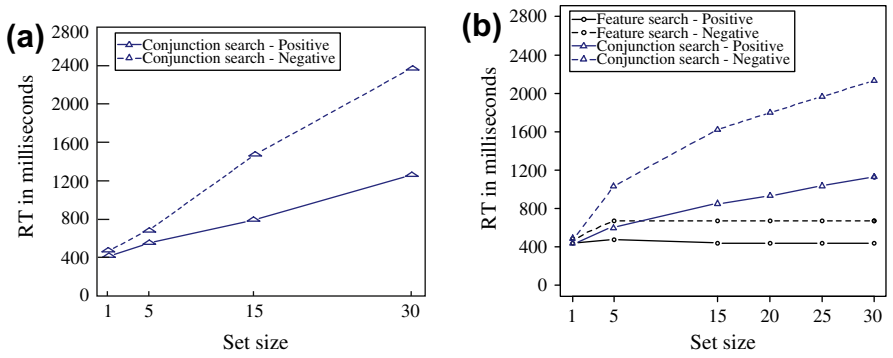

Fig. 4. (a) Mean reaction times of human subjects in conjunction search as reported by Treisman and Gelade (1980); (b) mean reaction times in feature and conjunction search tasks produced by our model.

Fig. 4b shows the model's mean reaction times in both feature and conjunction search tasks each averaged over trials of the same set size. The black solid line is for feature search task where target was present, and black dashed line is for feature search task where target was absent.

In feature search task the model was asked to find any red object. The resulting RT is mostly independent of set size and averages to 446 ms when a target is present and 641 ms when a target is absent. It is consistent with experimental findings where RT for positive trials is also around 430 ms and for negative trials is 550 ms (Treisman & Gelade, 1980; Wolfe, 2007). The model RT remains the same in positive trials due to very high bottom-up activation the target receives due to its color contrast to homogeneous surrounding objects. Top-down activation from the matching color also contributes to the overall saliency of the target. However, bottom-up activation alone is enough to make the target salient enough to attract almost immediate attention. In negative feature search trials all objects in iconic memory have zero top-down activation. It takes the model few fixations to realize absence of a top-down activation after which the model stops searching. As a result, model also produces flat RT line independent of a set size, although slightly higher than in positive trials.

In a conjunction search task the model was asked to find any red rectangle. Fig. 4 compares the RT produced by the model to the RT[2] obtained by Treisman and Gelade (1980) from their experiment with human subjects. The standard errors for the model RT are too small, and thus are not shown in Fig. 4b. As the blue[3] lines in Fig. 4 indicate the RT in both positive and negative trials rise as the set size increases. The slopes, however, are different with negative trials having a significantly higher slope. Linear regression of model's RT on set size gives intercept of 459 ms and 646 ms for positive and negative trials respectively. The slopes are around 23.2 ms/item and 53.8 ms/item. The model results can be compared to those obtained in previous studies (Table 1).

Table 1
Comparison of the results of the model's linear regressions of RT on set size to results of linear regression from similar experiments with human subjects.

	Trial type	Slope (ms/item)	Intercept (ms)
Model data	Positive	23.2	459
	Negative	53.8	646
Treisman and Gelade (1980)	Positive	28.7	398
	Negative	67.1	397
Wolfe et al. (1989)	Positive	7.5	451
	Negative	12.6	531

In this task the distracters are not homogenous. They vary by both color and shape. As a result, there is no guarantee in positive trials that a target will have a higher bottom-up activation than distracters. However, the target always receives higher top-down activation than any other object in iconic memory since it has both matching color and shape. When a set size is small the target's top-down activation is enough to compensate for smaller bottom-up activation, and the target almost immediately attracts attention as the most salient object. When the set size is big, there is a higher chance that the target will get significantly lower bottom-up activation than a distracter, which then cannot be compensated by higher top-down activation. Consecutively, those distracters with a higher overall activation are attended first which results in RT increasing with set size.

In negative conjunction trials the model should know when to stop the search and report the absence of the target. Since most of the distracters either match color or shape with a target, there are few objects that have zero top-down activation. Hence, the model had to rely on visual decision threshold to filter out partially matching distracters. The model requires on average 53.8 ms/item in negative trials indicating that the model does not need to fixate on every object to realize the absence of a target. Hence, top-down activation serves quite well as a visual decision threshold.

[2] Confidence intervals or standard errors are not available for human data in feature, conjunction and comparative visual search tasks due to lack of the data in original papers.

[3] For interpretation of color in Figs. 4 and 9, the reader is referred to the web version of this article.

Considering the variations between different studies, the model gives a good fit to experimental findings from previous studies with a slightly higher intercept for negative trials than that found in experiments with human subjects. This is probably due to the fact that the corresponding RT line (Fig. 4b) is not strictly linear, and as a result has an elevated intercept for an entire linear function. We are still in process of investigating what causes the slightly increased RT for those trials.

3.2. A model of comparative visual search

The second model does a comparative visual search, a paradigm proposed by Pomplun et al. (2001). The task involves detecting a mismatch between two, otherwise equal, halves of a display referred to as hemifields (Fig. 5). The task is a simplified version of the traditional picture matching task (Humphrey & Lupker, 1993) with a major difference that it does not require image processing.

For the model of comparative visual search, we set the screen size to 24° × 16°, and the size of each object was 0.6° both in width and height. Those are the same conditions used in the original experiment (Pomplun et al., 2001). The screen was divided vertically in two halves, hemifields. Each hemifield contained 30 objects varying in shape (rectangle, oval and triangle) and color (red, green and blue). Each color and shape value was represented in a trial in an equal quantity. Positions of the objects were generated randomly with minimum margin of 10 pixels from the boundaries of the screen. Two hemifields were identical except one object, the target, which mismatched in either color or shape. The target was chosen at random among 30 objects as well as the type of mismatch.

In total, the model had to do 10,000 trials where half of the trials had targets that mismatched color and the other half that had targets with mismatched shape. The model was not aware of the type of mismatch it had to find in a trial. The model was reset after each trial.

The model used a very simple algorithm to do visual search. The model starts from a top-left corner of a screen and does following steps:

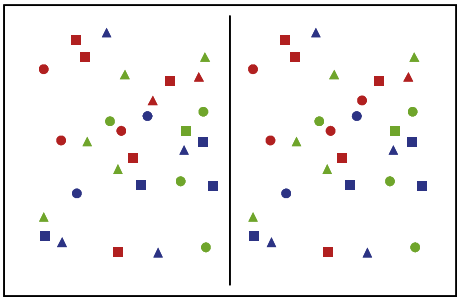

Fig. 5. An example comparative visual search task where targets are red triangle and red oval in left and right hemifields respectively. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Table 2
Comparison of model's mean RTs to those reported by Pomplun et al. (2001). All RTs are in ms.

	Color	Shape	Total
Model	9051	9197	9124
Pomplun et al. (2001)	9903	11,997	10,950

1. Fixate on any unattended object (further referred to as O1) in the current hemifield.
2. Fixate on any object (referred as O2) in the opposite hemifield that has the same y coordinate as the O1.
3. If O1 and O2 are the same then go to step 1.
4. If O1 and O2 are different then:
   a. Fixate on an object NO2 nearest to O2.
   b. Fixate on O1.
   c. Fixate on an object NO1 nearest to O1.
   d. If NO1 and NO2 are the same then end the trial.
   e. If NO1 and NO2 are not the same then go to step1.

The steps 4a to 4e are necessary to ensure that the module is comparing a correct pair of objects. This uncertainty comes from the fact that when locating a target's twin in the opposite hemifield the model knows only its y coordinate and not the x coordinate. Therefore, it is possible for the model to fixate on a wrong object that by chance had the same y coordinate. To detect such mistakes model also compares two objects from two hemifields that are closest to respective target objects.

The model's mean RT over all trials was 9124 ms (Table 2). On average, the model needed 9051 ms (SE = 79) and 9197 ms (SE = 80) to finish trials where the difference was either in color or in shape respectively. This is a reasonable fit to reaction times reported by Pomplun et al. (2001). The current model was unable to show difference between trials where the mismatch was either in color or in shape.

Fig. 6a shows a histogram of reaction times from original experiment done by Pomplun et al. (2001). This histogram can be compared to a histogram of reaction times produced by our model depicted in Fig. 6b. Both graphs show a plateau of short RT between 3 and 10 s, indicating that the distribution of RT produced by the model closely fits the distribution from the original experiment. On average, the model made 37.4 (SE = 0.23) fixations during a trial. This is a close match to 39.6 fixations reported by Pomplun et al. (2001). The model produces nicely structured scanpath (Fig. 7) even though there is no explicit control of which object should be chosen as O1.

3.3. A model of a SET player

In our previous study (Nyamsuren & Taatgen, 2013) we have described how human players play the card game of

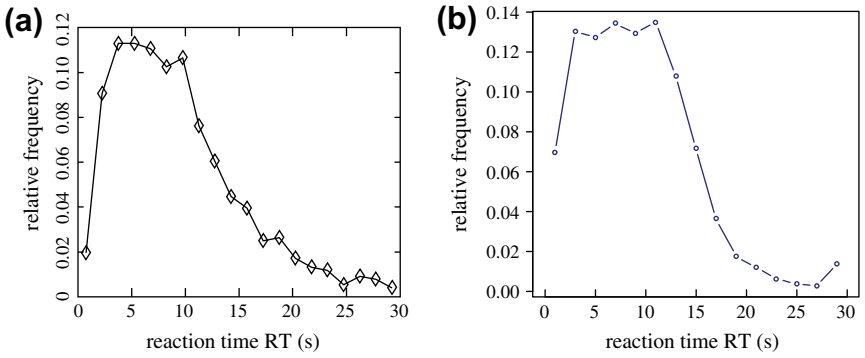

Fig. 6. (a) Histogram of reaction times in original comparative visual search experiment (Pomplun et al., 2001); (b) histogram of reactions times from 10,000 model trials in comparative visual search.

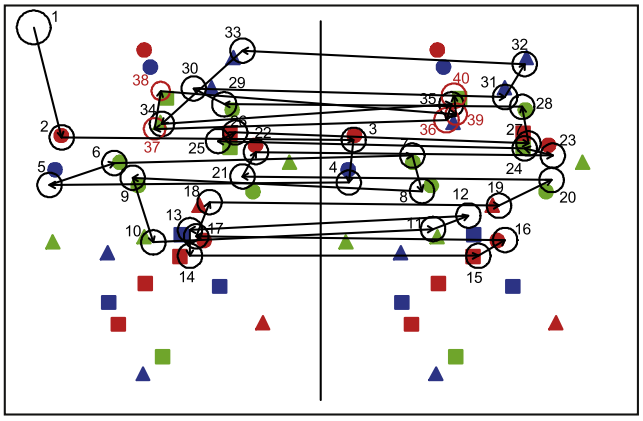

Fig. 7. Example scanpath produced by the model. Open circles indicate fixations while arrows indicate saccade directions. Numbers are positions of fixations in the fixation sequence. Targets are blue and green triangles at 36th and 37th fixations. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

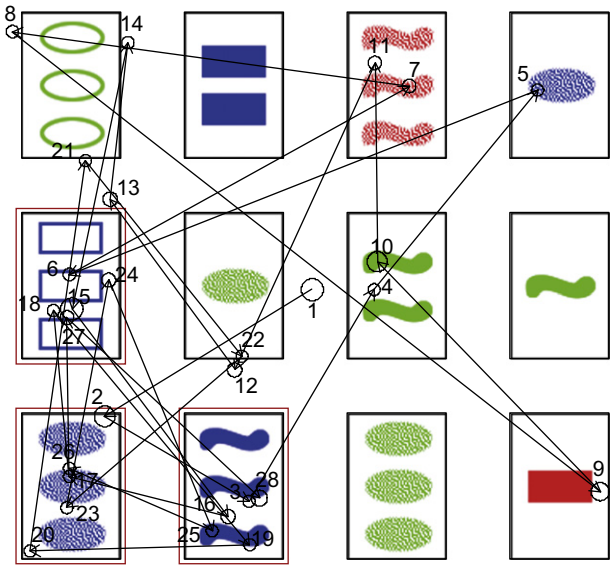

Fig. 8. An example array of 12 cards where cards with red borders make up a set. Also shown is an enumerated fixation sequence produced by the model. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

SET[4] and how human behavior in that game can be replicated and further explained by an ACT-R model. In this study we have used ACT-R's default vision module and compensated for lacking functionalities with custom code specifically written for that model. In this study we have changed the original model to work with PAAV module. We show how PAAV module helps to describe and explain one of the interesting effects found in original study. See the original article for a more detailed description of the study.

In each SET trial, 12 cards are dealt face up, as shown in Fig. 8. Each card differs from other cards by a unique combination of four features: color, shape, shading and the number of shapes. Each feature can have one of three distinct values. From those 12 cards, the subject should find a unique combination of three cards, further referred to as a *set*, satisfying a rule stating that in the three cards the values for each particular feature should be all the same or all different. We refer to the number of different features in a set as the *set level*. The set level has a significant effect on the human player's reaction times with higher level sets requiring more time to find (Fig. 9b).

SET players have a tendency to use a dimension reduction strategy while playing a game (Jacob & Hochstein, 2008). That is, they prefer to look for a set among cards that share a common feature value thus effectively reducing the search space by one feature dimension. For example, subjects might look for a set among the cards that have the color green. The choice of a common value heavily depends on an attribute type. For example, an analysis of fixations (Nyamsuren & Taatgen, 2013) indicates that color, as shown in Fig. 9a, is used for dimension reduction twice as much as any other feature. The new model easily explains this effect using PAAV's spreading activation from iconic memory and differential acuity. The model also serves well in validating these two functionalities of PAAV module.

In the new model we used three different values for size feature to mimic number of shapes on a card. The actual

[4] SET is a game by Set Enterprises (www.setgame.com).

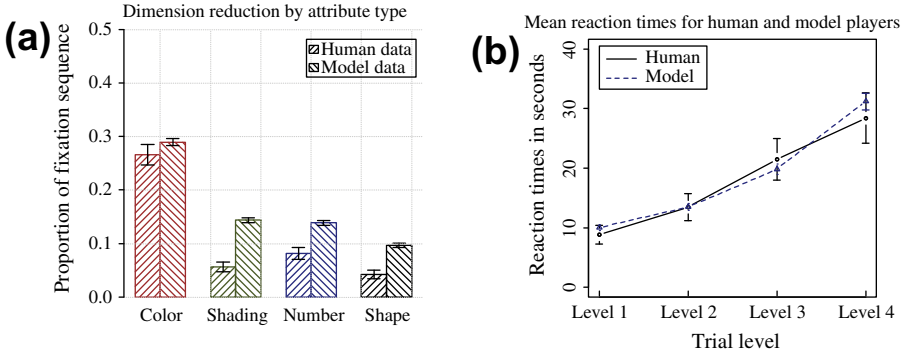

Fig. 9. (a) Dimension reduction usage by feature type shown both for human players and model; (b) mean reaction times for human players and the model.

size of a stimulus representing a card in the visicon also varied based on the number of shapes on the card. Sizes were 9.67°, 23.43° and 37.04° for one, two and three shapes respectively. The model chooses a feature value for dimension reduction by retrieving any of 12 possible values from declarative memory. This retrieval is heavily influenced by a spreading activation from iconic memory. For example, if oval shape is a dominant feature value in iconic memory then the model is more likely to retrieve oval. However, availability of feature values in iconic memory is limited by feature's differential acuity. Therefore, even if the shape value is the dominant value in the visicon, the color value can become the dominant value in iconic memory because it has lower visibility threshold. Therefore, overall color is used more often by model for dimension reduction than other features (Fig. 9a). The model is not only able to replicate the effect of dimension reduction, but also provides a nice overall fit to human players' mean reaction times (Fig. 9b).

As our model shows, the tendency of human players to prefer color can be explained with embodied cognition, influence of an external world on our decision making, and the limitations of human peripheral vision.

4. Conclusion

There are many existing models of the human visual system. We have greatly leveraged from those models by adopting different concepts and integrating them into one module that became PAAV. Our main goal is not to reinvent the wheel, but to create a tool that allows modelers to create cognitively plausible models of tasks that require comprehensive visual system. This is the major difference between PAAV and existing models of a visual system. Models, such as a three-level model of comparative visual search (Pomplun & Ritter, 1999) or Guided Search 4.0 (Wolfe, 2007), were created to perform very specific set of tasks. On the other hand, PAAV was developed to be general enough to model a wide range of tasks. For example, PAAV is highly customizable due to the possibility to adjust any parameter mentioned in this paper. This is why we prefer to call PAAV a module rather than a model. Furthermore, PAAV is not a stand-alone tool, but rather a part of a cognitive architecture. For example, Guided Search 4.0 excels at modeling feature and conjunction search tasks. However, an absence of a general cognitive theory makes it hard to investigate top-down influence in these tasks. On the other hand, ACT-R imposes limitations on what PAAV is allowed to do, but it also gives additional layer of plausibility. The source code for the PAAV module and the models of the visual search tasks described in this paper can be downloaded via http://www.ai.rug.nl/~n_egii/models/.

References

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

Clark, A. (1997). *Being there: Putting brain, body and world together again*. Cambridge, MA: MIT Press.

Humphrey, G. K., & Lupker, S. J. (1993). Codes and operations in picture matching. *Psychological Research, 55*, 237–247.

Jacob, M., & Hochstein, S. (2008). Set recognition as a window to perceptual and cognitive processes. *Perception & Psychophysics, 70*(7), 1165–1184.

Kieras, D. (2009). The persistent visual store as the locus of fixation memory in visual search tasks. In A. Howes, D. Peebles, & R. Cooper (Eds.), *9th International conference on cognitive modeling – ICCM2009*. Manchester, UK.

Kieras, D. (2010). Modeling visual search of displays of many objects: The role of differential acuity and fixation memory. In *Proceedings of the 10th international conference on cognitive modeling* (pp. 127–132).

Land, M., Mennie, N., & Rusted, J. (1999). The roles of vision and movements in the control of activities of daily living. *Perception, 28*, 1311–1328.

Nyamsuren, E., & Taatgen, N. A. (2013). Set as instance of a real-world visual-cognitive task. *Cognitive Science, 37*(1), 146–175.

Pomplun, M., & Ritter, H. (1999). A three-level model of comparative visual search. In *Proceedings of the twenty first annual conference of the cognitive science society* (pp. 543–548).

Pomplun, M., Sichelschmidt, L., Wagner, K., Clermont, T., Rickheit, G., & Ritter, H. (2001). Comparative visual search: A difference that makes a difference. *Cognitive Science, 25*(1), 3–36.

Rayner, K. (1998). Eye movements in reading and information processing: 20 years of research. *Psychological Bulletin, 124*, 372–422.

Salvucci, D. D. (2001). An integrated model of eye movements and visual encoding. *Cognitive Systems Research, 1*(4), 201–220.

Theeuwes, J. (1992). Perceptual selectivity for color and form. *Perception & Psychophysics, 51*, 599–606.

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

van Diepen, P. M., De Graef, P., & d'Ydewalle, G. (1995). Chronometry of foveal and peripheral information encoding during scene perception. In J. M. Findlay, R. Walker, & R. W. Kentridge (Eds.), *Eye movement research: Mechanisms, processes and applications* (pp. 349–362). Amsterdam: North Holland.

Wais, P. E., Rubens, M. T., Boccanfuso, J., & Gazzaley, A. (2010). Neural mechanisms underlying the impact of visual distraction on long-term memory retrieval. *Journal of Neuroscience, 30*(25), 8541–8550.

Wolfe, J. M., Cave, K. R., & Franzel, S. L. (1989). Guided search: An alternative to the feature integration model for visual search. *Journal of Experimental Psychology: Human Perception and Performance, 15*, 419–433.

Wolfe, J. M. (2007). Guided search 4.0: Current progress with a model of visual search. In W. Gray (Ed.), *Integrated models of cognitive systems* (pp. 99–119). New York: Oxford.

Wolfe, J., & Horowitz, T. (2004). What attributes guide the deployment of visual attention and how do they do it? *Nature Reviews Neuroscience, 5*(6), 495–501.