# Pre-attentive and Attentive Vision Module

**Enkhbold Nyamsuren (e.nyamsuren@rug.nl)**
**Niels A. Taatgen (n.a.taatgen@rug.nl)**
Department of Artificial Intelligence, University of Groningen,
Nijenborgh 9, 9747 AG Groningen, Netherlands

**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 three different tasks.

**Keywords:** vision; iconic memory; cognitive architecture; ACT-R.

## Introduction

This paper introduces a general purpose vision module called PAAV, which stands for **P**re-attentive **A**nd **A**ttentive **V**ision. 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 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.

EMMA (Salvucci, 2001) is arguably the most used extension to ACT-R's 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 in their optimization toward tasks that involve reading or working with items of a user interface. Those are the tasks with relatively a simple visual environment where bottom-up perceptual processes can be ignored without sacrificing 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, 2011) 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.

## Architecture of PAAV Module

### 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; oval and rectangle for shape and etc. Currently, PAAV does not support modeler specified custom features. However, it is included as a future implementation milestone.

### 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°x0.6° in size scattered over area of 14°x8°. 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 0.035 and 0.1 for color and shape respectively. The parameter $b$ has been fitted to 0.601 for both color and shape. We are still in process of refitting parameters for the rest of the features.

## 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. This persistence time is currently set to 4 s 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 request may include desired criteria including any of the five feature dimensions or location.

## 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.

**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.

**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 considered to be 0.5. Thus uncertainty is preferred to certain dissimilarity.

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

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

$W_{BA}$ and $W_{TA}$ are the weights for the bottom-up and top-down activations respectively. In correspondence with

Wolfe (2007), those weights control the intentional and unintentional attentional captures. Those weights are set to 1.1 and 0.45. 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.

### 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 20 ms as a base execution time and additional 2 ms for an every degree of angular distance between gaze position and the center of the object to be fixated. This is exactly the same method used by Salvucci (2001). 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 0.5 times of the object's linear width. In a similar manner, the standard deviation for Y axis is calculated using object's 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 encoding time takes a fixed 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, DeGraef, & 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.

### 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 (Figure 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 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 $TA_j \leq TA_{Th}$ and $d_{jg} \leq d_{Th}$ where $d_{jg}$ 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.

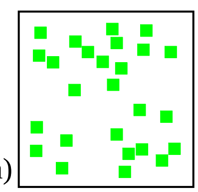
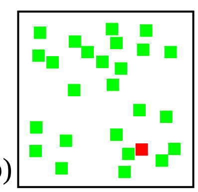
(a)  (b) 

Figure 1: Humans can spot an absence (a) of a red object in field of green objects as fast as its presence (b).

## Validation Models

This section describes two models that do common visual tasks. The models are based on ACT-R where the default vision module was replaced with the PAAV module. The tasks are simple, yet demand complex cognitive and perceptual processes, and require most of the components of PAAV module described in this paper. Hence, those tasks serve as a good way to validate the PAAV module.

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 (Figure 2a). In a conjunction search the target can differ from distracters by either of two features (Figure 2b). 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, Cave & Franzel, 1989; Wolfe, 2007).

The second model does a comparative visual search, a paradigm proposed by Pomplun, Sichelschmidt, Wagner, Clermont, Rickheit and Ritter (2001). The task involves detecting a mismatch between two, otherwise equal, halves of a display referred to as hemifields (Figure 3). 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.

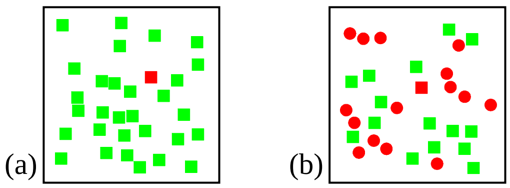



Figure 2: Examples of feature search (a) and conjunction search tasks (b). In both tasks the red rectangle is a target.

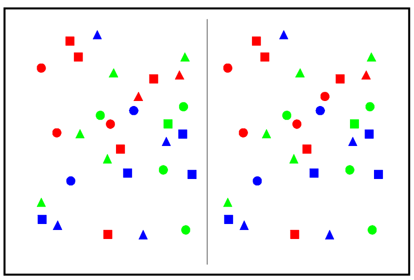


Figure 3: An example comparative visual search task where targets are red triangle and red oval in left and right hemifields respectively.

## A Model of Feature and Conjunction Searches

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°x11.3°, 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.

Figure 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 439 ms when a target is present and 640 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. Figure 4 compares the RT produced by the model to the RT obtained by Treisman and Gelade (1980) from their experiment with human subjects. As the blue lines in Figure 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 440 ms and 689 ms for positive and negative trials respectively. The slopes are around 19.6 ms/item and 72.8 ms/item. The model results can be compared to those obtained in previous studies (Table 1).

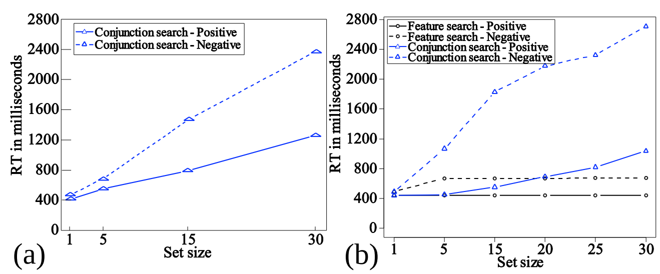



Figure 4: (a) Mean reaction times of human subjects in conjunction search as reported by Treisman and Gelade (1980); (b) Mean reaction times in feature and conjunction search tasks produced by our model.

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.

The main challenge for the model in negative conjunction trials is to 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 72.8 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.

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 (Figure 4b) is not completely linear, and the elevation for trials with set size of 15 and 20 results in an elevated intercept for an entire linear function. We are still in process of investigating what causes the slightly increased RT for those trials.

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 by Treisman and Gelade (1980) and Wolfe, Cave and Franzel (1989).

| | Trial type | Slope (ms/item) | Intercept (ms) |
|---|---|---|---|
| Model data | Positive | 19.6 | 440 |
| | Negative | 72.8 | 689 |
| Treisman and Gelade, 1980 | Positive | 28.7 | 398 |
| | Negative | 67.1 | 397 |
| Wolfe, Cave and Franzel, 1989 | Positive | 7.5 | 451 |
| | Negative | 12.6 | 531 |

## A Model of Comparative Visual Search

For the model of comparative visual search, we set the screen size to 24°x16°, 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 10000 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:

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 4.a to 4.e 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 9089 ms (Table 2). On average, the model needed 9007 ms and 9170 ms 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). However, the current model was unable to show difference between trials where the mismatch was either in color or in shape.

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 | 9007 | 9170 | 9089 |
| Pomplun et al. (2001) | 9903 | 11997 | 10950 |

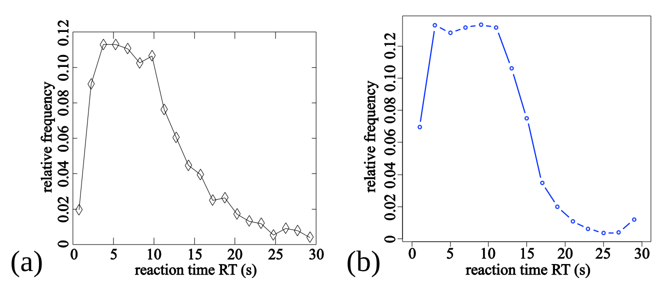



Figure 5: (a) Histogram of reaction times in original comparative visual search experiment (Pomplun et al., 2001); (b) Histogram of reactions times from 10000 model trials in comparative visual search.

Figure 5a 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 Figure 5b. Both graphs show a plateau of short reaction times between three and ten seconds, indicating that the distribution of RT produced by the model closely fits the distribution from the original experiment. On average, the model made 37.3 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 (Figure 6) even though there is no explicit control of which object should be chosen as O1.

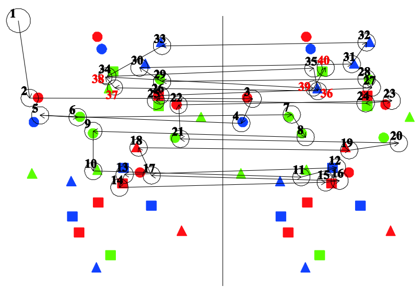


Figure 6: Example scanpath produced by the model. Open circles indicate fixations while arrows indicate saccade directions. Numbers are positions of fixations in the fixation sequence. Targets are blue and green triangles at 36th and 37th fixations.

## 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. 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://ai178174.ai.rug.nl/iccm2012/.

## 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.

Kieras, D. (2010). Modeling Visual Search of Displays of Many Objects: The Role of Differential Acuity and Fixation Memory. *Proceedings of the 10th International Conference on Cognitive Modeling*, (pp. 127-132).

Kieras, D. (2009). The Persistent Visual Store as the Locus of Fixation Memory in Visual Search Tasks. In A. Howes, D. Peebles, & R. Cooper (Ed.), *9th International Conference on Cognitive Modeling.* Manchester, UK.

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. (2011). Top-down Planning and Bottom-up Perception in a Problem-solving Task. *Proceedings of the 33rd Annual Conference of the Cognitive Science Society* (pp. 2685-2690). Austin, TX: Cognitive Science Society.

Pomplun, M., & Ritter, H. (1999). A Three-level Model of Comparative Visual Search. *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.

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. 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., & Horowitz, T. (2004). What Attributes Guide the Deployment of Visual Attention and How Do They Do It? *Nature Reviews Neuroscience , 5* (6), 495-501.