Combining Low-Level Perception with Expectations in CHREST

Peter C. R. Lane (Peter.Lane@bcs.org.uk)
Department of Computer Science, University of Hertfordshire,
College Lane, HATFIELD AL10 9AB, UK

Anthony Sykes (axs@psychology.nottingham.ac.uk)
School of Psychology, University of Nottingham,
University Park, NOTTINGHAM NG7 2RD, UK

Fernand Gobet (frg@psychology.nottingham.ac.uk)
School of Psychology, University of Nottingham,
University Park, NOTTINGHAM NG7 2RD, UK

Abstract

The ability of humans to reliably perceive and recognise objects relies on an interaction between information seen in the visual image and prior expectations. We describe an extension to the CHREST computational model which enables it to learn and combine information from multiple input modalities. Simulations demonstrate the presence of quantitative effects on recognition ability due to cross-modal interactions. Our simulations with CHREST illustrate how expectations can improve classification accuracy, reduce classification time, and enable words to be reconstructed from noisy visual input.

Introduction

The direct association of visual perception with cognition has long been recognised by cognitive scientists, although understanding of its nature has changed. Marr (1982) proposed a model of visual perception which is essentially one-dimensional: visual stimuli are processed in a sequence of stages, until the final representation is passed to high-level cognition. This conception is undermined by experiments indicating the role that expectations play in altering bottom-up processes of classification. These expectations may be conceptual, based on familiar patterns in the input, or due to priming with, for instance, verbal cues. The challenge for computational modellers is to provide a framework in which information from multiple modalities may be combined and used.

A further motivation is found in the importance of application areas, such as Human-Computer Interaction and image analysis, which highlight the need for a greater understanding of how humans relate their high-level conceptual knowledge to what they perceive. These applications are reflected in recent extensions to the ACT-R architecture, supporting perceptual-motor actions (Byrne, 2001). However, although ACT-R/PM uses expectations in the form of high-level schemata, the model must still be 'programmed' with its initial information.

In this paper, we are interested in how information from multiple modalities may be *learnt* and *combined* in a manner supporting the interplay between perceived and expected information. We develop some extensions to the CHREST (Chunk Hierarchy and REtrieval STructures) computational model of perception and learning. CHREST implements a theory of how humans learn hierarchical categories from naturalistic input (see Gobet, Lane, Croker *et al.*, 2001), and models the process by which experts learn *perceptual templates* (Gobet & Simon, 2000). We aim to combine the perceptual-learning processes already existing in CHREST with mechanisms for handling multiple input modalities.

Expectations in Perception

We focus on three important phenomena demonstrating the role of expectations in perception. The first of these is that expected objects are recognised with greater accuracy than unexpected objects, particularly in domains where noise affects the quality of the input stimulus. For instance, characters may be badly formed, ambiguous, or simply 'damaged' or partially hidden. An expectation that characters are from a standard alphabet enables correct identification of characters which would otherwise be ambiguous (Neisser, 1966; Richman & Simon, 1989).

The second phenomenon shows that expectations may relate to complex collections of objects, or *schemata*. Perceptual classification of objects within a familiar schema can be quicker than when the objects are not in the schema. For instance, Biederman (1981) describes an experiment in which participants took longer to identify a fire-hydrant when positioned above street level than when at its expected position. A similar result can be found in reading: identifying the 'K' in a word such as 'ANKLE' is quicker than in a non-word such as 'XGKAL'.

A third phenomenon is that of *reconstructive memory*, whereby a set of partially obscured objects may be identified based on their being recognised as a composite. For instance, a collection of partially obscured characters may be identified as a word (Lindsay & Norman, 1972), even though each individual character may be ambiguous.

Although it must be admitted that some of these phenomena are difficult to cleanly replicate in experimental settings, it is clear that people do not simply scan an input stimulus in a serial fashion whilst looking for a given item. Instead, humans employ higher-order constraints, based on expected schemata, to constrain the range of potential matches. In other words, perception is not a one-dimensional, bottom-up process, as proposed by Marr (1982), but instead interprets what is being seen

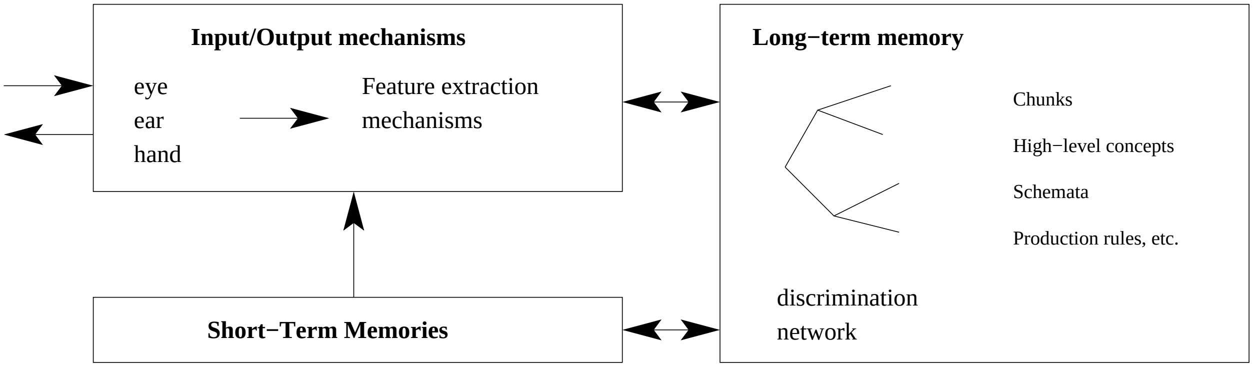

Figure 1: The CHREST Model

in the light of what is currently expected. We now describe some extensions to the CHREST computational model which support such interactions.

The CHREST Model of Learning

Figure 1 illustrates the three main components of CHREST, which are: mechanisms for interacting with the external world; multiple short-term memories (STMs), to hold information from different input modalities; and a long-term memory (LTM), where information is held within a discrimination structure known as a *chunking network*.

CHREST's interface with the external world can include a variety of input/output mechanisms. In this paper, we use two input modalities: visual, and verbal. For visual input, we use patterns consisting of stylised characters, which may be presented to the model in a sequence. For the verbal input, we use a simple character input, representing a single input entity. As with visual input, verbal input can occur in sequence. Thus, CHREST may be presented with words, in the form of a sequence of characters, either via a visual input, or via a verbal input, or through both simultaneously.

Each STM contains pointers to information held in LTM, and its contents vary as the model carries out learning and searching operations. In this paper, CHREST is given two STMs, one for information relating to the visual input, and one for the verbal input. The STMs are important in learning as they support the construction of links between information, as explained below.

CHREST uses a *chunking network* in its LTM to store learnt information. A chunking network consists of a collection of nodes, with each node holding a learnt pattern (or *chunk*) in its *image*. Nodes are primarily interconnected with *test links*, which impose a discrimination structure on LTM. Beginning from the root node, a given input pattern is sorted to a node with a matching image by following those test links whose tests match the input pattern. When the pattern cannot be sorted any further, then the node is pushed onto the appropriate STM.

Apart from test links, chunking networks also support associations between nodes from disparate parts of the network. These associations are made with lateral links, two of which are:

naming links A node representing visual information is *named* by linking it with a node representing verbal information.

sequence links A node is linked to a second *of the same modality*, to indicate that the first is followed by the second *in sequence*.

The CHREST model, like its predecessor, EPAM (Elementary Perceiver and Memorizer: Feigenbaum & Simon, 1984), has proven successful in modelling a wide range of cognitive phenomena, including: chess expertise (e.g. see de Groot & Gobet, 1996; Gobet & Simon, 2000), diagrammatic reasoning (Lane, Cheng & Gobet, 2001), and language learning (e.g. see Freudenthal, Pine & Gobet, 2002). More details and references for CHREST can be found at: `http://homepages.feis.herts.ac.uk/~comqpcl/chrest/`

Learning Multiple Input Modalities

As mentioned above, we provide CHREST with two input modalities, visual and verbal. In this section, we define the input patterns and describe how CHREST learns from two modalities, separately and in combination.

Example Data

We use one form of visual input, a stylised character, and one form of verbal input, consisting of character names. The verbal input can be used as a *name* for the visual input. Thus, the string "A" presented on the verbal input would be used to name the visual pattern for a character 'A'.

Our design for the visual input is motivated by three factors. First, a symbolic form of input is preferred for ease of implementation. Second, the individual characters should be arrangeable in a sequential array, so the model can learn that certain characters usually appear in a given sequence. Third, the representation should support noisy or occluded (hidden) inputs in a controlled

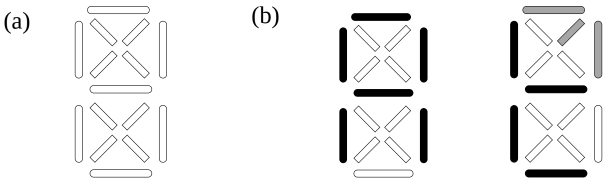

Figure 2: The Visual Input: (a) shows the 15 segments, which may be set/unset/occluded, (b) shows a complete character ‘A’, and a partially occluded character ‘E’.

manner. Thus, we implement the visual input as a set of 15 segments, arranged as shown in Figure 2(a). Each segment may be either set, unset, or occluded. Figure 2(b) shows the settings for one complete and one partially occluded character.

The *verbal input* to the model is a simple string or character object, representing the character’s name. For simplicity, we assume that the verbal input to the model is free of noise.

Learning Patterns and Sequences

Learning occurs after a pattern has been sorted down the test links in the chunking network. A single LTM structure is used, and it is assumed that patterns from different modalities can never match. Hence, when a visual pattern is sorted through LTM, it can only follow test links representing visual patterns; retrieved nodes are always of the same modality as the input pattern.

One of two learning processes can occur, depending upon the match between the image at the node reached and the input pattern. First, if the node image matches the input pattern (i.e. the input pattern is a superset of the current node image), then extra information is added to the node image, in a process known as *familiarisation*. (If the node image *equals* the input pattern, then no extra information will be added.) Second, if the node image mismatches (i.e. the input pattern disagrees with the current node image), then an extra child node is added to the current node in a process known as *discrimination*. This child node is linked with a test for the mismatching feature, and, initially, its image is empty. Further presentation of the same input pattern would lead to the node image being completed through a process of familiarisation. (Further details of these learning mechanisms may be found in the EPAM/CHREST literature, e.g. see Gobet *et al.*, 2001.) Whether learning occurs through familiarisation or discrimination, the trained node is pushed onto the appropriate STM.

Input patterns from both modalities are learnt in the same manner, with only the criteria for matching varying. For the visual input, two patterns will *match* if the collection of set/unset segments is the same. The familiarisation process involves setting the value of any segment whose value is unknown in the node’s image to the value of the input pattern. Discrimination occurs when the node image and the input pattern disagree on the value for one or more segments. One of the mismatching segments is used to create the test leading to a new child node.

A similar process occurs for the verbal patterns, except that two verbal patterns containing different values can never match. The discrimination tree for verbal patterns is thus shallow, with discrimination occurring only at the root node.

Sequences are represented within the chunking network by connecting those nodes whose images occur consecutively. A sequence link is formed between two nodes when: firstly, they are present in the *same* STM at the same time, and secondly, their images represent patterns which were presented *consecutively*. A check for the formation of sequence links is triggered whenever a node is added to STM. During retrieval, CHREST can use the sequence link to *predict* the node most likely to appear next. Sequence links are formed only between nodes which are present in an STM. Any node may have multiple sequence links reflecting, in our example, that any character may be within many words.

Creating Cross-Modal Links

In order to utilise interactions between the two modalities, CHREST must form links between nodes from *different* modalities. This is achieved through a simple extension of the process by which sequence links are formed, leading to the creation of *naming links*. A naming link is formed between two nodes when: firstly, they are present in *different* STMs at the same time, and secondly, their images represent patterns which were presented *simultaneously*. A check for the formation of naming links is made whenever a node is added to either STM. Figure 3 illustrates the process.

Using the Cross-Modal Links

Naming

Cross-modal links can be used to name an input visual pattern. The process is applied after sorting the visual pattern through LTM. If the node retrieved has a naming link, then the associated verbal pattern is output by the model: the model thus ‘names’ the input visual pattern. Using this mechanism, it is possible to train CHREST on a succession of characters, and then request the model to name a succession of new characters; the model’s success rate is its classification accuracy.

Priming

The model can be ‘primed’ to recognise a given visual pattern by presenting its name on the verbal input. Sorting the verbal pattern through LTM, CHREST will locate a node and place this node into its verbal STM. If this node has a naming link, then the linked node is used to prime the model.

The priming mechanism uses this linked node as follows. When a visual pattern is presented, it is first compared to the image in the linked node. If it matches to within a given tolerance, then the primed node is returned

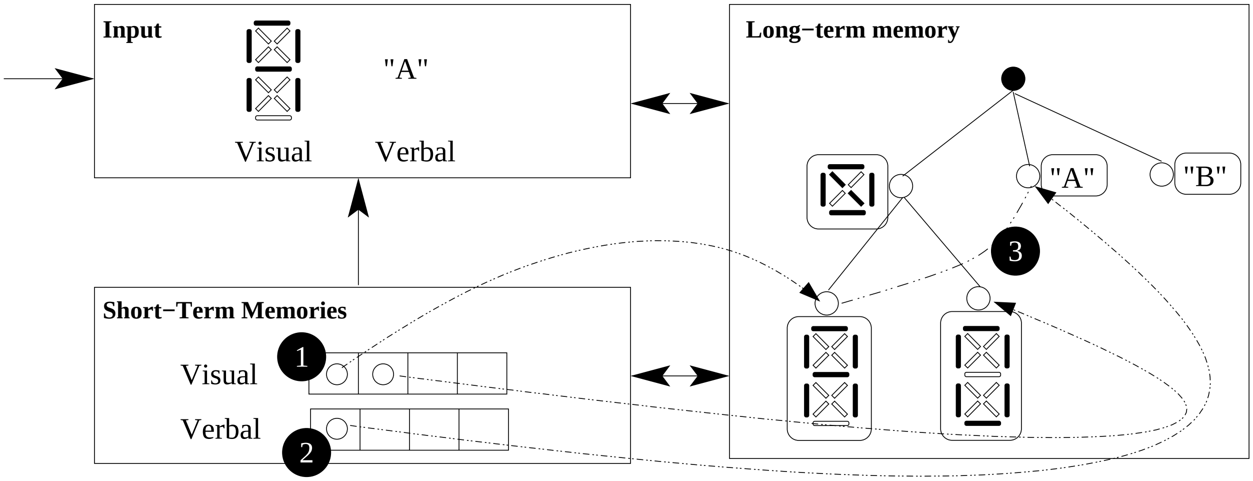

Figure 3: Learning a 'naming link' across two modalities. (1) The visual pattern is sorted through LTM, and a pointer to the node retrieved placed into visual STM. (2) The verbal pattern is sorted through LTM, and a pointer to the node retrieved placed into verbal STM. (3) A 'naming link' is formed between the two nodes at the top of the STMs.

as the matching node. This priming process can allow the model to successfully match input patterns even though they would not have been retrieved during the usual sorting process.

Expecting Sequences

In a similar way to priming with a verbal input, sequence links can be used to prime the model to expect a pattern of the same input modality. Thus, if a node for the visual pattern 'T' is retrieved, and it is linked with a sequence link to a node for the visual pattern 'H', then the model will expect the character 'H' to appear next on the input. The priming mechanism described above applies equally to nodes linked through sequence links.

Simulations

We perform three sets of simulations. First, we explore the accuracy with which CHREST can classify characters with increasing amounts of noise. Second, we consider the speed with which characters are visually recognised, comparing the speed of 'pure' bottom-up recognition with that of top-down, expectation-driven recognition. Third, we consider how the use of two modalities enables CHREST to disentangle very noisy data when attempting to satisfy high-level constraints.

Accuracy of Classification

Our input data consists of the standard 26 characters, each represented both visually and verbally. We first train CHREST fully on this input data, so that it accurately classifies all 26 of the original characters. We then explore the impact of two kinds of noise, in two separate simulations. The first simulation explores the effect of randomly occluding segments within each character, thus making the state indeterminate. The second instead randomly reverses the state of any segment in the character. The likelihood of a segment's state being changed is varied from 0.0 to 1.0 in steps of 0.1.

To remove any bias in the precise ordering of characters presented to the model, we train 10 CHREST models, each with a different random order of the original 26 characters. Also, we create 10 datasets based on randomising the visual patterns with the appropriate type of noise. The classification ability of the models is computed in two forms:

bottom-up Only the visual input is used by the model in searching its LTM. Noise affects classification accuracy by preventing the correct discriminatory tests being made. (We call this 'unprimed' classification.)

top-down The verbal input is used to 'prime' the model with the expected character. A match is made if the primed character matches the input to within a given tolerance: results are given for 6.7% (1/15 segments matching), 20% (3/15) and 46.7% (7/15).

Figure 4 indicates the average performance of the models in the first simulation, where segments may be randomly occluded. As some segment states are unknown, CHREST will fail to sort the visual pattern past certain tests, hence returning an incorrect classification. However, the use of priming with an expected character significantly improves classification ability. When the amount of noise exceeds 50%, then classification accuracy decreases rapidly. A similar advantage is seen in the second simulation; refer to Figure 5.

Speed of Classification

In this simulation, we explore the speed with which classification occurs when characters appear in isolation (unprimed), or instead appear within words (primed). We

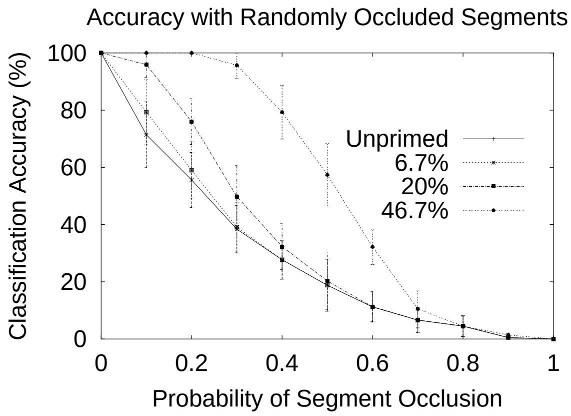

Figure 4: Classification Accuracy vs Noise: Occlusion

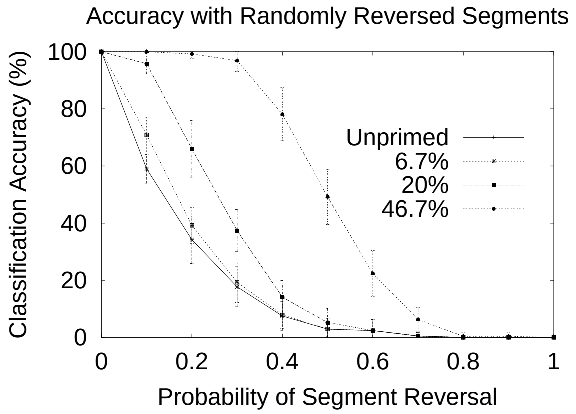

Figure 5: Classification Accuracy vs Noise: Reversal

	μ	σ
unprimed	2.8	1.0
primed	1.4	0.8

Table 1: Average number of pattern matches.

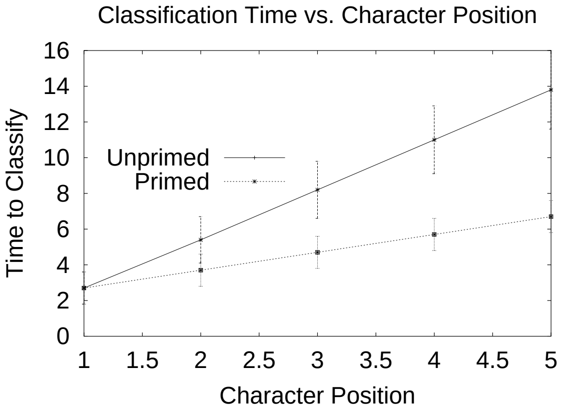

Figure 6: Relative Classification Time by Position

define the *speed* of classification by counting the number of pattern matches made by the model when searching its LTM. For input, we use a set of ten 5-character words. The model is trained so that it forms sequence links between the nodes in its LTM representing the individual characters.

Time to recognise characters First, we compare the number of pattern matches required by the model to recognise each of a sequence of characters. 50 models were trained, each with the word-list sorted in a different random order. Table 1 shows the average time, μ, and standard deviation, σ, required when attempting to recognise the characters in isolation (unprimed) as opposed to recognising them when forming part of a word (primed). There is a significant reduction in the required searching time when the model is primed. The use of an expected schema to predict the characters appearing in the visual input significantly reduces classification time.

Time to find a given character Second, we consider the time required to find a given character within a sequence of characters. The time required depends on the position of the character within the word, as the previous characters must first be searched. Figure 6 shows the amount of time required to identify a given character in each position: priming means that CHREST identifies and uses a word schema to assist in finding the character, the unprimed timings measure when CHREST treats each character distinctly from its neighbours. CHREST is quicker to locate a character when using a schema, and this advantage increases with character position.

Reconstructive Memory

If the visual input given to a model is such that every character is ambiguous, the only way to attempt a classification is to consider potential schemata which match every character in the scene. For instance, Figure 8 shows three characters, each badly occluded: the first character could be ‘A’ or ‘H’, the second ‘R’ or ‘K’, and the third ‘E’ or ‘F’. From prior familiarity with likely sequences of characters, a viewer may be expected to retrieve the word ‘ARE’ as the likely interpretation of the scene.

We explore CHREST’s ability to reconstruct scenes by training CHREST on the standard 26 characters, in isolation. Next, we train CHREST to recognise words using *the verbal input only*. Thus, CHREST’s LTM has sequence links *only* within nodes representing verbal patterns. We then provide sequences of damaged characters, visually. The matching process proceeds as follows: in turn, we prime the visual matching process with the named nodes from the LTM’s verbal knowledge. When a match is made to the visual pattern, the *sequence links* in the verbal nodes are used to further prime the

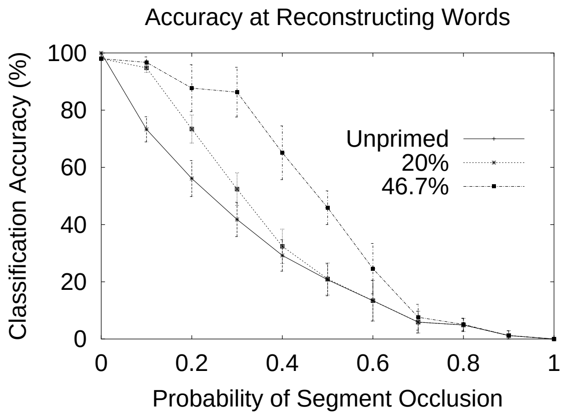

Figure 7: Reconstruction Accuracy

visual matches. Hence, CHREST relies on its familiar schemata to match the noisy data.

Figure 7 plots CHREST's average performance at classifying characters with progressively larger amounts of noise. (We trained 5 CHREST models and used 5 randomly generated sets of test data for each noise level; words were 5 characters in length.) The two 'primed' lines indicate CHREST's performance when it takes advantage of prelearnt words. As is evident, the use of priming significantly increases the accuracy of CHREST's performance.

Discussion and Conclusion

In this paper, we have described an extension of the CHREST architecture enabling it to learn and combine information from multiple input modalities. The simulations have demonstrated the presence of key qualitative phenomena, in the effect that expectations have on low-level perception. First, we have shown an improvement in classification accuracy when particular objects are recognised. Second, we have shown that a familiar schema enables object recognition to proceed faster than without. Third, we have shown that interpreting a set of ambiguous objects is possible with the use of a schema from a different modality.

One limitation of the model at present is that the visual input is still in symbolic form, and cannot handle image data directly. We are currently developing representations of bitmaps which will overcome this limitation. We will then apply our model to more complex applications, taking advantage of the model of eye movements already present in CHREST.

Acknowledgements

We thank the European Office of Aerospace Research and Development for funding this project.

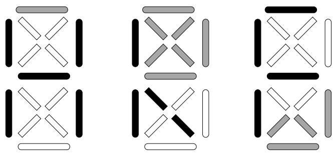

Figure 8: Reconstructive Memory Example.

References

Biederman, I. (1981). On the semantics of a glance at a scene. In Kubovy, M. & Pomerantz, J. R., editors, *Perceptual Organization*, pages 213–254. Hillsdale, NJ: Lawrence Erlbaum.

Byrne, M. D. (2001). ACT-R/PM and menu selection: Applying a cognitive architecture to HCI. *International Journal of Human-Computer Studies*, 55: 41-84.

de Groot, A. D. & Gobet, F. (1996). *Perception and Memory in Chess: Heuristics of the Professional Eye*. Assen: Van Gorcum.

Feigenbaum, E. A. & Simon, H. A. (1984). EPAM-like models of recognition and learning. *Cognitive Science*, 8:305–336.

Freudenthal, D., Pine, J. M., & Gobet, F. (2002). Modelling the development of Dutch optional infinitives in MOSAIC. In *Proceedings of the 24th Meeting of the Cognitive Science Society*, pages 328–333. Mahwah, NJ: Erlbaum.

Gobet, F., Lane, P. C. R., Croker, S. J., Cheng, P. C-H., Jones, G., Oliver, I., & Pine, J. M. (2001). Chunking mechanisms in human learning. *Trends in Cognitive Science*, 5:236–243.

Gobet, F. & Simon, H. A. (2000). Five seconds or sixty? Presentation time in expert memory. *Cognitive Science*, 24:651–82.

Lane, P. C. R., Cheng, P. C-H., & Gobet, F. (2001). Learning perceptual chunks for problem decomposition. In *Proceedings of the Twenty-Third Annual Conference of the Cognitive Science Society*, pages 528–33. Mahwah, NJ: Erlbaum.

Lindsay, P. & Norman, D. (1972). *Human Information Processing*. New York: Academic Press.

Marr, D. (1982). *Vision: A computational investigation into the human representation and processing of visual information*. San Francisco: W. H. Freeman.

Neisser, U. (1966). *Cognitive Psychology*. New York: Appleton-Century-Crofts.

Richman, H. B. & Simon, H. A. (1989). Context effects in letter perception: Comparison of two theories. *Psychological Review*, 3:417-432.