A Computational Model of Diagram Reading and Reasoning

Anthony M. Leonardo[†]
Department of Psychology
Carnegie Mellon University
Pittsburgh, PA 15213
leonardo+@cmu.edu

Hermina J.M. Tabachneck
HCI Institute
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213
tabachneck+@cmu.edu

Herbert A. Simon
Department of Psychology
Carnegie Mellon University
Pittsburgh, PA 15213
has@cs.cmu.edu

Abstract

We describe an extension of CaMeRa, a Computational model of Multiple Representations in problem solving (Tabachneck, Leonardo, & Simon, 1994, 1995). CaMeRa provides a general architecture for LTM, STM and their interactions, and illustrates how experts integrate pictorial and verbal reasoning processes while solving problems. A linked production system and parallel network are used to further resolve the communication between pictorial and verbal knowledge by simulating how a diagram is understood by an expert. Low-level scanning processes and an attention window, based on both psychological and biological evidence, are incorporated into CaMeRa, and productions are developed that allow these processes to interface with the high-level visual rules and representations already in the model. These processes can explain interruptibility during problem solving, and show how understanding is reached when reading a novel diagram.

Introduction

While an expert is solving a problem in economics, there is a knock at the door. She answers it, and is drawn by a colleague into a conversation about new astrophysical data suggesting the existence of a black hole of forty million solar masses. Eventually her friend leaves, and the expert returns to her desk, somewhat muddled. But after glancing only briefly at the economics diagram she had been drawing, she immediately resumes where she left off, as if the interruption had never occurred. The contents of her short-term memory were written and re-written many times during the course of the distracting discussion. All she has left from her previous efforts are a sketchy diagram, a few carelessly scrawled equations, and the contents of her long-term memory. Yet she does not have to begin the problem anew, or even take much time to reconstruct her position prior to the interruption. She is able to resume working quickly, at the correct place in the problem solving sequence. How is such a feat accomplished?

Clearly, a number of factors are at play -- the features of the diagram, the processes of recognition, long-term memory (LTM), and short-term memory (STM). This interruption task is a good example of the fact that recognition is faster than recall. When beginning the problem, the expert must recall everything from long-term memory. When reconstructing her position after the interruption, she needs only to recognize the meanings of the various cues on the diagram and the associations between them. A properly built diagram summarizes all the critical information processed thus far, and reasoning can be continued as long as this external summary is available.

Recognizing a diagram's components is necessary both for reconstructing the meaning of a diagram and for understanding a novel diagram. In each case, perceptually significant features must be identified. These features must be scanned into STM using low-level visual processes. Finally, the information in STM must be matched to information in LTM (if it exists) and analyzed for its implications. In this paper, we will present a computational model which simulates each step of this process, basing the implementation of each process on both psychological and biological evidence. The remainder of the paper will make each of these steps specific, and finally, present a framework in which they come together to produce the behaviors described above.

CaMeRa

CaMeRa is a computational model of the use of multiple representations in expert problem solving (Tabachneck, Leonardo, & Simon, 1994, 1995). It demonstrates how an economics expert, by carefully combining pictorial and verbal knowledge, is able to produce a coherent and effective explanation to problems that novices are unable to understand. The work described in this paper represents an extension of CaMeRa. One of the motivations for this research, in addition to giving an account of the behaviors described above, was to expand CaMeRa's abilities by providing it with processes for reading diagrams and pictures. This will ultimately allow the model to understand problems from many different domains.

CaMeRa consists of a linked production system and parallel network. It contains representations of (1) a pictorial external display, (2) pictorial short-term memory, (3) pictorial long-term memory, (4) verbal short-term

[†] Address after September 1, 1995: Computation and Neural Systems Program; California Institute of Technology; Pasadena, CA 91125 USA

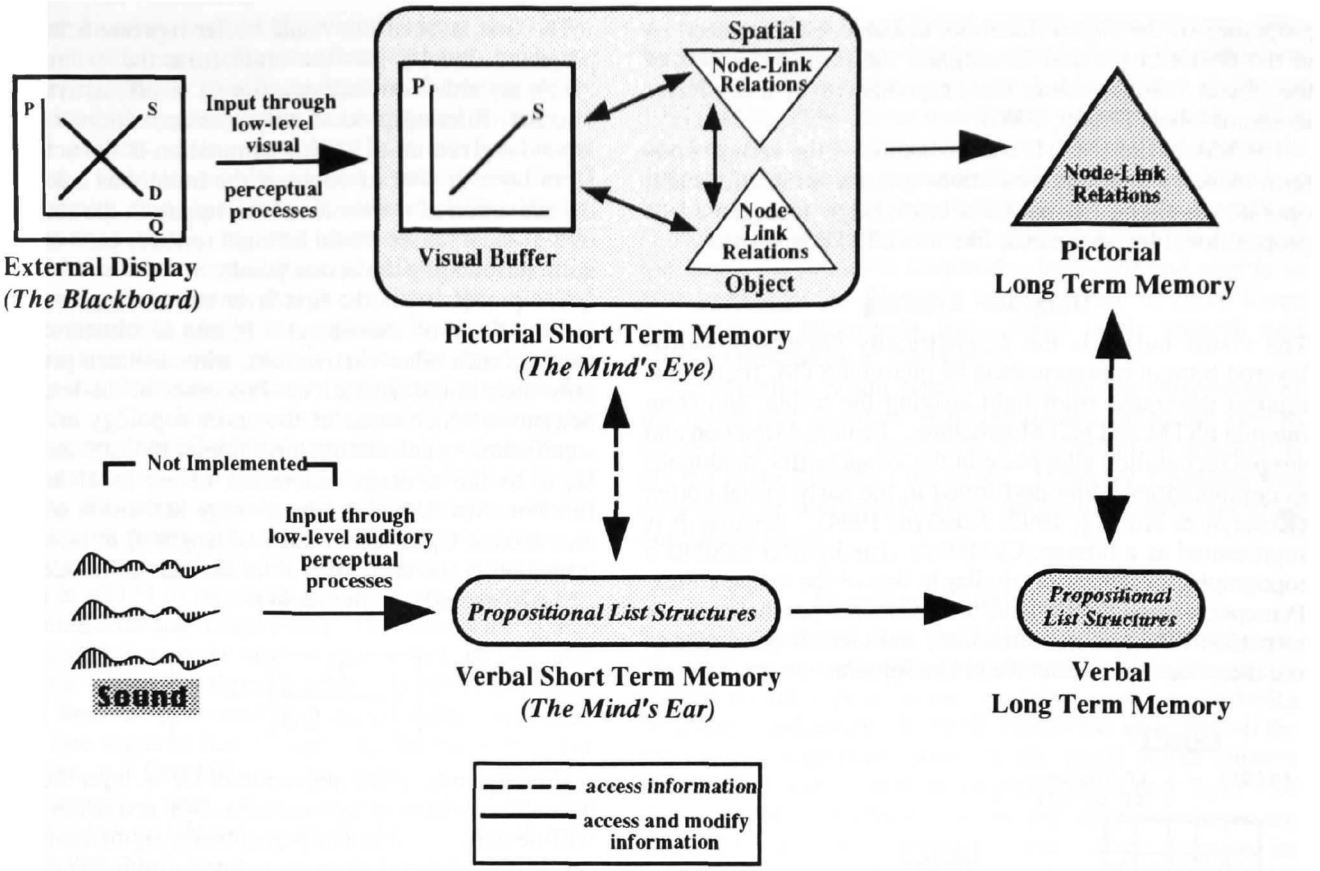

Figure 1: The Architecture of CaMeRa

memory, and (5) verbal long-term memory (see Figure 1). The model uses the external display (the "blackboard") just as the expert does, for drawing, reasoning, recognition, and input to STM. Recognition consists of matching information placed in STM from the blackboard to information stored in LTM. Because the recognition of cues on the blackboard drives the problem solving forward, CaMeRa has little need for an explicit goal stack. This type of control architecture places a minimal load on short-term memory capacity.

Pictorial long-term memory (pLTM) has a node-link representation. Verbal long-term memory (vLTM) knowledge is represented by instances of a generic propositional relation. This single relation was sufficient to model all of the knowledge needed by CaMeRa in the limited economics domain we examined. LTM knowledge is represented in the brain as associations among symbols, with structure and hierarchy being a function of these associations. The images which are generated from pLTM are computationally equivalent to those generated from perception, as are the processes which operate on them (Kosslyn, 1994). The same applies for vLTM.

All problem solving, reasoning, and modification of memory systems are done through STM. CaMeRa's current STM does not yet have any limitations on the quantity of information it can store (for more on STM capacity, see Simon, 1976). Implementing the diagram-reading processes will allow us to model the capacity limits of STM in CaMeRa at a later time, as the blackboard can now be used to refresh STM through constant low-level scanning of the diagram. STM structures can be defined as specific exemplars of LTM structures. They may also be associated with STM structures in their own and other modalities. The highly regulated and limited interaction that takes place between pSTM and vSTM, allowing for these associations to emerge, is a critical feature of the model (see Tabachneck, Leonardo, & Simon, 1994, 1995).

The Mind's Eye (MI) represents a synthesis of a number of pictorial short-term memory data structures and the productions that operate on them. Three types of representations appear in the MI: the visual buffer, object structures, and spatial structures. The visual buffer, which is the physical location of mental images, is the area in which feature extraction and other low-level, highly parallel, visual processes operate. The projection of an image onto the visual buffer facilitates complex visual reasoning which could not be done using only the object and spatial structures of STM (e.g., the perception of geometrical relations, etc.).

As the visual buffer is the focus of much of our improvements to CaMeRa, its properties will be further specified in the following section. The remaining two structures are used to simulate the interaction between two of the visual sub-systems of the brain, namely the spatial

properties of the object (location, distance, etc., represented in the posterior parietal lobes), and the form properties of the object (shape, color, size; represented in the inferior temporal lobes) (Farah, 1990).

The Mind's Ear (ME) is a combination of the verbal short-term memory data representations and the productions that operate on them. It contains knowledge represented as propositional list structures, like its vLTM counterpart.

Diagram Reading

The visual buffer is the geometrically organized, multi-layered bitmap representation of pictorial STM. It contains images generated from light striking the retina, and from internal pSTM and pLTM structures. Feature extraction and simple recognition take place in the visual buffer, analogous to computations being performed in the early visual cortex (Kosslyn & Koneig, 1992; Kosslyn, 1994). Because it is represented as a bitmap, CaMeRa's visual buffer exhibits a topographic organization similar to that of the visual cortex. Perception results from using a feedforward parallel network to perform the feature extraction, and Gestalt principles to use these features to scan the buffer for structure and form.

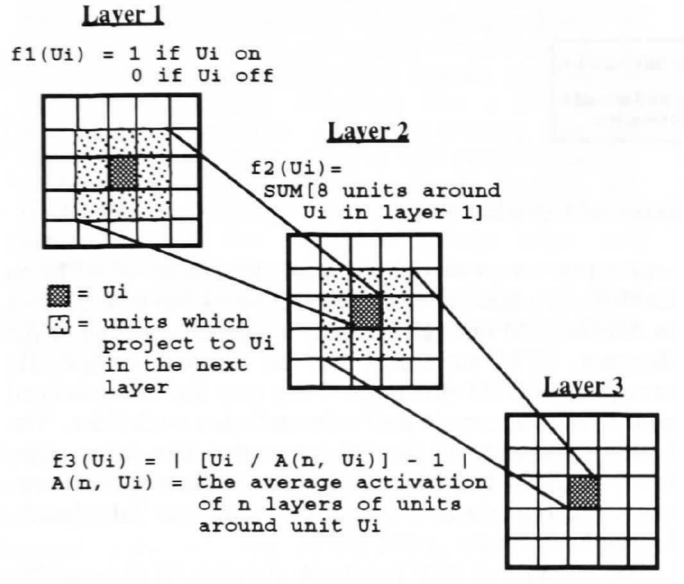

Figure 2: Topology Parallel Network

The network functions by perceiving non-modal areas in the local topology of the object. As all connections between pixels on the bitmap are presumed to be of equal, positive weight, no training is necessary. By detecting points that depart from the local modal value, the network identifies all perceptually significant areas on the visual buffer[1]. This is accomplished through the three layered process depicted in Figure 2.

[1] We have recently learned that the topology network is similar to a class of pyramid-based segmentation techniques developed for computer vision by Rosenfeld (1986, 1988). It also has parallels to the methods devised by Marr (1982).

The first layer of the visual buffer represents information projected directly into the brain from the external world. Pixels are either on (activation = 1) or off (activation = 0). The first filtering process sums the activations of all units around a given unit U_i ; this summation is the activation of U_i in layer 2. For example, if the input was a 3x3 square, the activation of the center of the square in the second layer of the visual buffer would be eight (one for each of the eight units surrounding the center point).

The processing in the first layer builds a representation of the topology of the object. Points in clusters mutually increase each other's activations, while isolated points retain only their initial activation. Processes in the second layer determine which areas of the entire topology are the most significant, by calculating how similar the activation of unit U_i is to the average activation of its local area. The function $A(n, U_i)$ gives the average activation of the local area around U_i (a square array of length n) in layer 2. U_i is turned on in the third layer if its the ratio of its activation to $A(n, U_i)$ exceeds the threshold t:

$$\left| \left\{ \frac{U_i}{A(n,U_i)} \right\} - 1 \right| > t$$

Consequently, if the activation of U_i in layer 2 equals the average activation of its local area, U_i's activation in layer 3 will be zero -- it is not perceptually significant as it can not be distinguished from the points surrounding it.

The result of this process is a saliency map in layer 3 of all the perceptually significant features found in the image on the visual buffer, such as intersection points, line endpoints, labels, object outlines. The coordinates of these points are sent to the high-level object and spatial representations of the pictorial STM for further processing. Productions may then request the visual buffer to identify a specific feature by matching the fovea-sized area around the point to patterns stored in pictorial LTM, or to scan a line or other object by applying gestalt principles.

We have chosen to implement the network in this fashion for two reasons -- i) it is more cognitively plausible than other AI feature extraction mechanisms, and ii) we were unable to perform the feature extraction successfully with these other mechanisms. For example, matching small areas of the diagram to a finite set of feature patterns will not identify the salient points because it runs into the problem of computational overload -- countless variations of the same pattern are needed to recognize tiny perturbations in the original image. The topology network circumvents this dilemma. In short, our path solves the problem effectively, and has some features in common with what is known of the relevant neurology. However, we refer to CaMeRa's low-level perceptual system as a "parallel network" to emphasize that its components are units, not neurons, and while its architecture has certain similarities to the visual cortex, it is not a model of this brain area. It should be kept in mind that there is no mathematical basis for assuming that the functional properties of a real neuron are preserved in the abstraction to a formless connectionist unit. We assume that the results of the topology network

are computed somewhere in the visual cortex, but most likely through different mechanisms than we have used here.

After the perceptually significant areas on the visual buffer are determined, the entire image is recognized as a set of associated objects, using gestaltist low-level rules of organization. CaMeRa employs Good Continuation, Good Form, Proximity, and Familiarity (Goldstein, 1984). Rules related to motion, convexity, etc., were not required in our tasks. We have implemented these serially, rather than in parallel, because the serial design captured the relevant cognitive principles in a lucid and explainable manner.

To coordinate the four gestalt rules, the model first saccades to the closest perceptually significant point on the visual buffer. This is the point in layer 3 which is the shortest distance from CaMeRa's current focus of attention. It then determines whether the small area of CaMeRa's fovea surrounding this point contains a familiar pattern. If the fovea's contents match a pictorial LTM pattern, action is taken. For example, a label (an icon) causes a structure to be created in pSTM to represent it. A line segment evokes Good Continuation and Good Form. This causes the model to focus its attention on the nearest endpoint of the line, and then scan the entire line through a series of short saccades. Individual lines are processed by looking at the points near the initial line segment feature, applying the rules of Good Continuation and Good Form, and thereby following the simplest connected path. If a salient point falls into the attention window as CaMeRa scans the line, it immediately focuses on that point and processes the area around it for familiar patterns, which are then associated with the line through Proximity. Finally, if a point in the fovea is not recognized, or processing on it has been completed, the model focuses on the nearest perceptually significant feature.

Using feedback from the results of these computations, the model keeps track of the salient points it has already seen, and thus avoids re-processing them. As it scans, CaMeRa removes each feature it processes from layer 3 of the visual buffer, continually shifting its attention to unprocessed points, and elaborating further its representation of the diagram.

Mozer et al. (1992; see also Behrmann, Zemel, & Mozer, 1995) have designed a connectionist model of object segmentation based on the phase-locking of related features, which develops the properties of some of the gestalt rules we employ. Comparison of the behavior of the two systems suggests that many of the differences between our serial design and a parallel one may be only implementational and not functional. In further support of this point, although CaMeRa was not intended to segment images, it has the capability to do so by virtue of the architecture of its visual buffer and the processes that operate on it. CaMeRa is able to discriminate the component lines of geometrical objects (squares, diamonds, triangles, etc.) that are overlaid on each other (so far we have tested up to four overlaid objects). These lines could be bound into appropriately segmented objects by implementing the gestalt rule of Closure.

We have based the implementation of CaMeRa's fovea and attention window on biological and psychological data as much as possible. CaMeRa's fovea is the area within which it can see high levels of detail and recognize patterns. This is intended to correspond to the central area of the eye where almost all of the cones are located (Humphreys & Bruce, 1991), subtending about one degree of visual angle. This corresponds to a circle of approximately six letters in diameter at a reading distance of 15 inches. It is less clear how to set the size of the attention window: if too small, it is useless for detecting features close to a point in focus; if too large, the model is frequently distracted and unable to scan consistently. We have chosen a size of three fovea diameters, a magnitude that allows fairly smooth and efficient processing of the image.

The attention productions that control the movement of the fovea and attention window always cause the two to move in unison -- the center of the attention window is always at the center of the fovea. However, CaMeRa will eventually be modified to allow the focus of attention to be outside the fovea.

Diagram Understanding

Figure 3 illustrates how CaMeRa would read a diagram. The time series contains a sequence of four images exactly as they would appear on the computer screen as CaMeRa processes the diagram. In 3a, the model has identified all the perceptually significant points on the graph (small clusters and isolated points), and has projected them onto layer 3 of the visual buffer. CaMeRa's fovea and attention window are focused on the upper left-hand corner of the diagram, its default starting position. The model will now shift its attention to the closest meaningful feature, in this case, the endpoint of the Price axis (arrow, 3a).

Next, CaMeRa will try to match the fovea sized area around this point to patterns it has stored in pLTM. The match is successful, and the point is identified as part of a vertical line segment, evoking Good Continuation and Good Form. CaMeRa now scans in the entire vertical line through a series of saccades. Figure 3b illustrates part of this event -- it has written to the text screen that a vertical line (VLINE) has been found. The fovea and attention window are now midway down the line, approaching the lower endpoint. While CaMeRa is scanning the image, the fovea and attention window in figure 3 move dynamically, giving the observer a clear understanding of what the model is doing at each moment.

Upon processing the vertical line, CaMeRa creates a spatial structure in pSTM for the endpoints of the line, and an object structure in pSTM for the form of the line (represented as an equation). This is shown in 3c as "VLINE (15 15 15 135)". Then, CaMeRa focuses its attention on the closest point of interest, in this case the endpoint of the supply line. Again, CaMeRa will recognize this as part of a diagonal line, and will scan the entire supply line (arrow, 3c). Finally, 3d shows the model as it finishes scanning the supply line, and notices near the line endpoint a cluster of significant points. It focuses on these points, recognizing the label for the supply line. This label is associated with the diagonal line which was just processed, using Proximity.

To return to the example cited in the introduction, how would CaMeRa resume problem solving after being

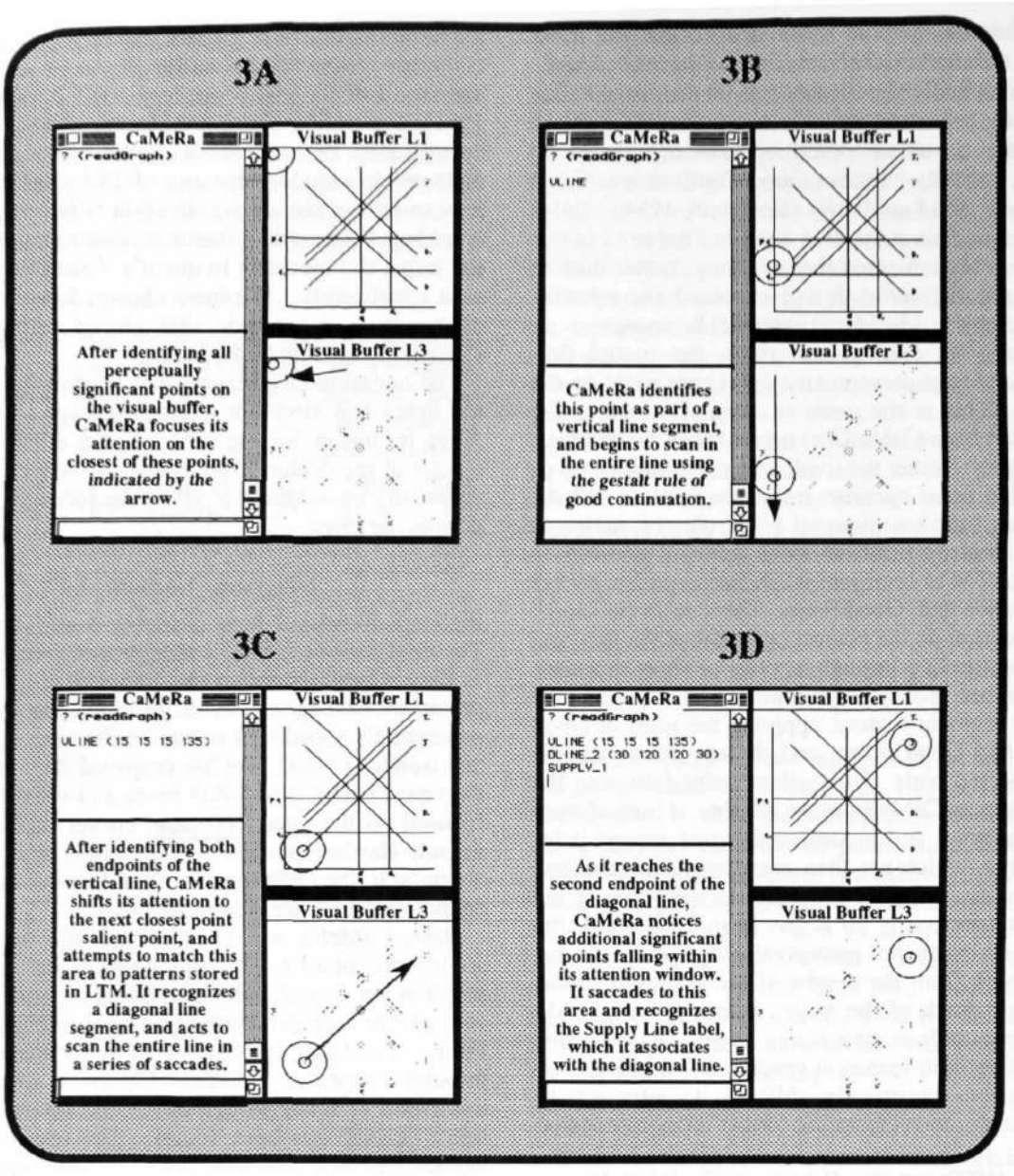

Figure 3: Diagram Reading Time Series

interrupted? This process can be simulated by allowing CaMeRa to reach a certain depth in the problem solving sequence, and then erasing the contents of STM. To continue processing, CaMeRa must now rescan the diagram it has drawn, using the processes described above. As the low-level functions read in data and interact with the high-level STM productions, the lost contents of STM could be quickly reconstructed. With a novel diagram, the system would try to match the scanned objects to objects in LTM as best it could. A partial explanation, emerging from successful LTM matches, in conjunction with inference processes, would produce an interpretation of the diagram. As an image is scanned into pSTM through the low-level visual processes described above, the higher-level productions activate and elaborate the reasoning chain. When the system has come to rest, the low-level processing would continue where they had left off and more input would be sent to STM until further high-level processing could take place.

Hybrid Models

By combining a feedforward parallel network and a production system in its architecture, CaMeRa demonstrates that hybrid models can be extremely advantageous in modeling complex cognitive tasks. Previous hybrid models have tended to focus on artificial intelligence problems, with limited concern for cognitive plausibility. For example, ALVINN, developed by Pomerleau, Gowdy & Thorpe (1991), enables a robot to perform autonomous navigation. Other types of hybrid models include expert systems with dual production and neural network knowledge bases (Rose, 1990), and spreading-activation semantic networks (Just &

Carpenter, 1992; Lange et al., 1990), used to simulate a cognitive theory of language processing[2].

We believe that certain tasks are best accomplished and understood within a serial design, while others are more suited to a parallel one. We found that a serial feature detection algorithm was slow and ineffective, whereas a parallel one was highly efficient. Likewise, we could have designed small parallel networks for each of the gestalt principles, but these would have produced results identical to those of their production rule counterparts, while yielding a significant increase in design complexity and only a minimal increase in structural plausibility. Our goal has been to develop a clear account of the cognitive processes involved in expert reasoning. Combining serial and parallel methodologies, as we have in CaMeRa, allows one to build more sophisticated simulations by both broadening the potential task domain and facilitating implementation and analysis of the system.

Conclusion

CaMeRa is a cognitive model of the interaction of visual and verbal elements in reasoning. In this paper, we have described a parallel network that extends CaMeRa's capabilities into a cognitively plausible model of the basic structure of visual perception. The model is able to construct, read, and reason about diagrams, using both verbal and visual information. Frequent interaction between high-level and low-level visual processes allows CaMeRa to build an elaborate representation of the diagrams it is reading. Employing both a production system and a parallel network has allowed us to develop a computational model which would be extremely difficult to design in either of these frameworks alone.

Acknowledgments

The second author was supported by a James S. McDonnell Foundation Cognition in Education Postdoctoral Fellowship, grant # 92-5.

References

Behrmann, M., Zemel, R., & Mozer, M. (1995). Object-based segmentation and occlusion: Evidence from normal subjects and a computational model. In press.

Farah, M.J. (1990). *Visual Agnosia: Disorders of Object Recognition and what they tell us about Normal Vision.* Cambridge, MA: MIT Press.

Goldstein, B.E. (1989). *Sensation and Perception.* Belmont, CA: Wadsworth Publishing Company.

Humphreys, G.W. & Bruce, V. (1991). *Visual Cognition.* London, UK: Lawrence Erlbaum Associates.

Just, M.A. & Carpenter, P.A. (1992). A Capacity Theory of Comprehension: Individual Differences in Working Memory. *Psychological Review*, 99(1), 122-149.

Kosslyn, S.M. (1994). *Image and Brain: The Resolution of the Imagery Debate.* Cambridge, MA: MIT Press.

Kosslyn, S.M. & Koenig, O. (1992). *Wet Mind: the new cognitive neuroscience.* New York, N.Y.: The Free Press.

Lange, T., Melz, E., Wharton, C. & Holyoak, K. (1990). Analogical Retrieval Within a Hybrid Spreading-Activation Network. In *Proceedings of the 1990 Summer School for Connectionist Models* (pp. 65-276). San Mateo, CA: Morgan Kauffman Publishers.

Marr, D. (1982). *Vision.* San Francisco, CA: W.H. Freeman.

Mozer, M.I., Zemel, R.S., Behrmann, M. & Williams, C.K. (1992). Learning to Segment Images Using Dynamic Feature Binding. *Neural Computation*, 4, 650-665.

Pomerleau, D.A., Gowdy, J. & Thorpe, C.E. (1991). Combining Artificial Neural Networks and Symbolic Processes for Autonomous Robot Guidance. *Engineering Applications of Artificial Intelligence*, 4:4 pp 279-285.

Rose, D. (1990). Appropriate Uses of Hybrid Systems. In *Proceedings of the 1990 Summer School for Connectionist Models* (pp. 277-286). San Mateo, CA: Morgan Kauffman Publishers.

Rosenfeld, A. (1986). Some Pyramid Techniques for Image Segmentation (CS-TR-1664). Maryland: University of Maryland, College Park, Center for Automation Research.

Rosenfeld, A. (1988). Computer Vision: A Source of Models for Biological Visual Processes? (CS-TR-1971). Maryland: University of Maryland, College Park, Center for Automation Research.

Simon, H.A. (1976). The Information Storage System Called Human Memory. In *Models of Thought, Volume I.* New Haven: Yale University Press.

Sun, R. & Bookman, L.A. (1995). *Computational Architectures Integrating Neural and Symbolic Processes: A Perspective on the State of the Art.* Boston, MA: Kluwer Academic Publishers.

Tabachneck, H.J.M., Leonardo, A. & Simon, H. A. (1994). How does an expert use a graph? A model of visual and verbal inferencing in Economics. In *Proceedings of the 16th Annual Meeting of the Cognitive Science Society.* Hillsdale, NJ: Lawerance Erlbaum Associates.

Tabachneck, H.J.M., Leonardo, A. & Simon, H.A. (1995). How does an expert use a graph? A Computational Model of Multiple Representations. Submitted for Publication.

[2] Additional descriptions of hybrid models may be found in Sun & Bookman (1995).