THE LOCUS MODEL OF SEARCH AND ITS USE IN IMAGE INTERPRETATION

Steven M. Rubin and Raj Reddy
Department of Computer Science
Carnegie-Mellon University, Pittsburgh, Pa. 15213

Abstract

The Locus model of search is a non-backtracking, deterministic search technique in which a beam of near-miss alternatives around the best path are extended in parallel for graph searching problems. In this paper we formulate image interpretation as a giaph searching problem and show bow the Locus model provides a near-optimal minimal effort solution. The structure of the model is illustrated using a detailed example. The relationship of the present approach to earlier attempts at image interpretation are discussed

Introduction

The central problem in image understanding is the representation and use of all the available sources of knowledge in the interpretation and description of an image. The problem of representation is complicated by the diversity of the sources of Knowledge. Converting knowledge into effective algorithms in the presence of error and uncertainty further complicates the issue. In this paper we present a specific framework for representation and use of knowledge which appears to be both sufficient and efficient for a wide variety of image interpretation tasks.

The framework for image interpretation presented here is based on the Locus model successfully used in speed) understanding research (Lowerre and Reddy, 1977). The Locus model is a non-backtracking, non-iterative, deterministic search technique in which a beam of near-miss alternatives around the best path are extended to determine the near-optimal description of the image.

This technique is being applied to several tasks which together exhibit a wide range of image source variability, sensor characteristics, and noise characteristics. The three tasks currently under exploration are: interpretation of uncontrived arbitrary images representing different views of downtown Pittsburgh (3-D world); location of a landmark or identification of an image from satellite and aerial images of the Washington, D.C. area (2-D world); detection of changes in an image using symbolic techniques. The downtown Pittsburgh task involves several interesting subtasks: scene-type identification (indoor, outdoor, office, ...), camera position identification (scale, location and orientation aspects of the image); image structure understanding (relative positions of buildings); and image detail understanding (detect cars, bushes, people walking after the larger context of a "road" is established).

In the following sections we will outline the structure of the model, illustrate its use by a simple but detailed example, and discuss the relationship of the present approach to earlier attempts at image interpretation. A detailed description of the model as applied to the image interpretation task will be given in Rubin (1977). A more complete discussion of the strengths and limitations of the model and its relationship to the other approaches to knowledge representation and search are given in Reddy (1977).

This work was supported by the Defense Advanced Research Projects Agency (F44620-73-C-0074) and is monitored by the Air Force Office of Scientific Research.

Image Understanding as Search

The basic premise underlying the locus model is that the problem of image interpretation can be viewed as a problem of search. Given a specific knowledge representation paradigm and a specific signal-to-symbol transformation paradigm, a highly efficient search can be used to obtain a near optimal global solution satisfying as many of the constraints of the world model as possiblele.

The principal requirement of the locus model is in the area of knowledge representation. Most approaches to image recognition assume the existence *(and* availability) of a world model in terms of some internal symbolic description. The world model usually consists of Knowledge which defines the structure and elationship among objects in all scenes that are interpretable by the world model. *By* iteratively redefining higher level structures in terms of simpler objects one can generate a hierarthieal network (or possibly a relational semantic network).

The particular Knowledge representation paradigm we have adopted in locus is to attempt to represent all images that are admissible by the world model in terms of a graph structure whose nodes are Primitive Picture Elements (PPLs). A PRE is chosen so that all pixels belonging to a given PRE class shaie the same properties in the feature space (or signal space). Thus a *PPL* might sometimes represent an entue object as in the cast[1] of sky, river, or road, or represent a small subpart of an object such as a segment with similar textural properties. As an example, the table below lists the PPLs that have been identified in a typical scene of downtown Pittsburgh. Note that the PPLs are chosen for their structural uniqueness as well as their visual uniqueness.

Hitton Awning roof	Pittsburgh Pror.o plajrn roof
Hilton Awning, side	Pittsburgh Prosy plaza side?;
Hilton Man roof	State Office mam tower roof
Hilton Main structure wallt;	Stat*? Office main tower walls
Hilton Elevator shaft roof	State Office office building roof
Hilton Elevator shaft wells	Style Office office building walls
Hilton Conference area roof	Allegheny River
Hilton Conference area walls	Gateway Towers roof
Gateway 1 man roof	Gateway Towers walls
Gateway 1 walls	Gateway Towers elevator shaft roof
Gateway 1 ©levator shaft walls	Gateway Towers elevator shaft walls
Gateway 2 mam roof	Commonwealth Place
Gateway 2 walls	Liberty Ave Fnr.t
Gateway ? ©levator shaft walls	liberty Ave West
Gateway 3 main roof	Rlvcl of the Allies
Gatoway 3 walls	Point Park Rood A
Gateway 3 elevator shaft walls	Point Park Road B
Jenkins Arcade walls	Point Park A
Homos main arpa roof	Point Pork 8
Homes secondary-area walls	Point Park C
Hornos tertiary-area roof	Point Park D
Homes tertiary-area walls	Liberty Ave Island A
Homes smoke stack	liborty Ave Island B
Pittsburgh Press roof	Mountains to north of city
Pittsburgh Press walls	Sky

The above set of PPEs are primarily intended for use in the detail understanding sub-task of the downtown Pittsburgh imago interpretation task. These are by no means uniqvin and are given here purely to illustrate the type of detail that can be applied. However, at other levels of understanding (such as scene understanding or viewpoint understanding) the numbei of PPL s needed can be substantially smaller, for example, on the scene understanding task, we plan to use only b PPEs: Sky, Mountains, Buildings, River, and Park.

We assume that a set of PPEs are available which can be used to compose any image that is admissible by the world model, further we assume that most, if not all, of the constraints about object structure, size, shape, location, and orientation are expressible in terms of the graph structure containing only the PPEs. It is obvious that this type of knowledge representahon is likely to be expensive in terms of spare fur all except the most trivial problems but it appears to be whal is needed for an efficient solution. Baker (197b) and lowerre (1976) show how diffeient types of knowledge and constraints can be combined into a single gt aph st i i ic l ure.

Let us consider an example. The task is the labeling of a 4X2 scene We will assume that Ibis scene consists of three objects (PPrs; Called A, B, and 0. figure 1 shows the possible relationships that are allowed between A, B, C, and the scene edge. The arrows indicate the adjacency relationships between the objects and the boundaries. Note that either A or B may be adjacent to the left, top, and bottom boundaries and that only C is permitted next to the right boundary.

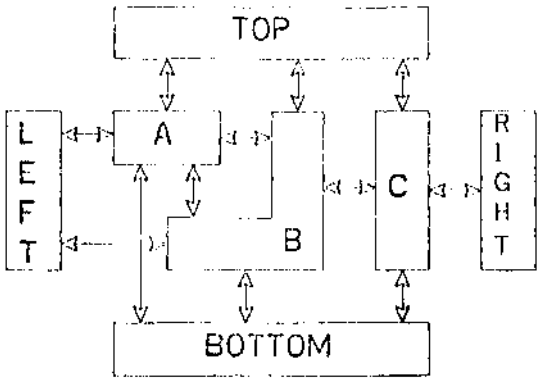


Figure 1 Legal rolationships, between the threo PPEs and the scene edge Note thnt utate A in optional hpcmuc uUle B may border the top and left Bides

The only relationships used in this network are horizontal and vertical. It is possible to employ more contextual information such as diagonal relationships, but this simple network is adequate for our examples

In the absence of any constraints, the optimal assignment of PPEs to pixels can be obtained by selecting the best PPE label in each pixel neighborhood. However, given the semantic, syntactic, structural, and segmental properties of scenes that are acceptable within the world model, one wishes to choose, those assignments of PPEs to pixels that are both globally optimal *and* consistent with the model.

In our example, each point in the 4x2 scene can be labeled arbitrarily from the three PPEs, allowing 3^8 possible interpretations. Given the network constraints, this reduces to only 9 possible labelings, shown below in Figure 2:

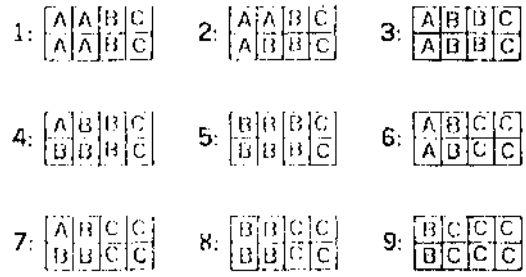


figure 2 lef.nl lahol assi^nmentr* of 3 *PPL*\, to the 8 scene points of the 4x2 im,i(;e URinfi ndwurk constromtr. from figure 1 Note that only 9 of the pm.mhlo 3^8 p.-ithw jirr ipg;.!

These 9 labeling, can be seen graphically in Figure 3 winch shows the possible network paths that can be traversed while searching from the initial state to the final state. This is an unpruned recognition tree. It appears as a graph because some of I lie nodes have boon combined to indicate the independence of the path from the prior context. The nodes at positions (?, ,"), (?, 3), and *{?,* 4) have states arriving from the TOP and EEET. This is the normal case for most PPt. s in a larger image since the average point is not along, the top or left side. Note that for real pictures with real constraints, the complexity of the networks and the number of legal label assignments is significantly higher.

Sinnal-1o-Symbol.Matching

Matching the symbolic elements of the network to the signal requires a signal-to-symbol transformation technique by means of winch one can estimate the likelihood that a given PPE is present at (or around) a pixel location. This basically requires the availability of a pattern template for each PPE and a distance metric for matching the unknown signal with the PPE templates.

Figure 4 shows the values at each position in the sensed image for our example. Our task is to assign a label (A, U, or C) to each pixel. Suppose the expected value (templates) of PPt objects are*:* A-4, B=9, and C=3. Then the question is which of the 9 possible interpretations given in Figures 2 and 3 best represent the signal.

	1	2	3	4
1	3	8	5	4
2	8	6	8	5

Figure 4 Sensed datit for each position in the sample image

For simplicity, let us assume that the distance metric is: Aij = 1 - |I-J|/10, yielding the following statistical matches for As, Bs, and Cs:

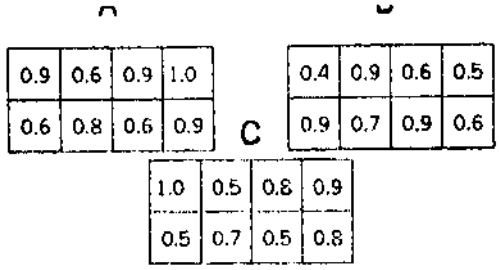


Figure 5 likelihood values of each PPE to the points in the sensod image For example, the point (1, 1) has a 0.4 likplihood of being PPE B because 8 is normatty 9 and point (1, 1) is 3 1 - |9-3j/10 - 0 A

The Locus Model

Given a PPE graph structure representation of the world model and a signal-to-symbol transformation technique, the problem of interpreting an unknown image can be viewed as finding the optimal path through the graph, i.e., finding a sequence of PPEs which best describe each of the pixel neighborhoods of the unknown image, subject to constraints defined by the knowledge sources represented by the graph. In our example, Figures *2* and 3 provide two different representations of the constraint and illustrate the power of knowledge sources in reducing the number of alternatives.

Finding the optimal path through the graph is a classical search problem with many possible alternative search strategies (Nilsson, 1971). In this paper we propose and use a search strategy called Locus which appears to be effective in perceptual problem solving. Locus is a beam search heuristic in which all except a beam of near-miss alternatives around the best path are pruned from the search tree at each pixel (or segmental) decision point. This contains the exponential growth of nodes in the path without requiring backtracking and non-deterministic search.

The Locus search proceeds as follows: 1) a forward pass calculates path likelihoods and inter-node connections, and 2) a backtrace uses the inter-node connections to determine the components of the near-optimal network path. As the forward pass search progresses through the network, unpromising alternatives arc pruned and the interconnections along the beam are saved until the end of the network is reached. At this point, a backtrace of the connections is made to select a path through the network. Note that this path is expected to lie in the beam that was carved out by the forward pass. By delaying the decision making process until ail of the network nodes have been examined, Locus obtains the globally near-optimal path through the network. This is because the calculation of a node's likelihood hinges on all previous nodes that led up to it. Thus, during the backtrace, each node decision is guaranteed against degeneration because its likelihood is supported by all nodes before it. This means that the selection of an object label in one corner of the scene can affect the labeling in the opposite corner. Consideration of all of the near-miss alternatives removes the need for backtracking, and thus removes the problem of whether to search by depth or breadth.

Before we can define the search strategy for finding the optimal path we need to define the term path likelihood in a PPE network. Path likelihood is defined incrementally in terms of the nodes it traverses and uses three pieces of information to calculate a likelihood: the statistical match of the signal to the symbol; the likelihoods of previous network nodes; and the transition likelihoods of arriving from those previous network nodes. Formally:

$$P_{i,j} = A_{i,j} \times \underset{d}{\text{AVERAGE}} \left[\underset{k}{\text{MAX}} \left(P_{k,j+\lambda(d)} \times T_{k,i,d} \right) \right] \qquad (1)$$

where Pij : is the likelihood of being in network state i at position j of the sensed data; Aij is the statistical match of the *PPE* symbol represented by state i to the signal at position j; e)(d) is the state adjacency function which offsets the current state (j) to the previous state (j+d(d)) in direction d; and $T_{K,i,d}$ is the transition likelihood of traveling from state k to state i in direction d. For image processing, the position (j) is an (x, y) vector. The maximum k in the above equation is saved as the best previous state and identifies the best path to take during the backtrace. Note that the likelihood values are not needed during the backtrace: they accumulate on the forward pass only. The back pointers are calculated on the forward pass using the likelihood information, so they reflect all node transitions to that point. The backtrace uses only the best previous node for each state as it quickly steps through the network and selects a path. No search is performed in this pass: it is pure look-up.

Going back to our example, Figure 6 shows the states that were examined in the forward pass. Note that this is simply a pruned version of the recognition tree in Figure 3. The numbers above each node arc the likelihoods calculated for that point on the forward pass. The first number is the f*orward* pass likelihood winch is calculated for each node at a given position. The second number is the normalized forward pass likelihood which is calculated after each node at the position has been examined. The normalization is necessary to keep the likelihoods from degenerating as the network is traversed (discussed below). Note that nodes are pruned from the beam when their un-normalized likelihood drops to 0.5 or below.

Let us look at the likelihood calculations for the point *(2,* 3) in Figure 6 (the two squares in the third column). The bottom point is a C, and it is *on* the path which selects labelings number 6 through 9 (see Figure *2).* There are two paths from the top and left that contribute to the likelihood calculations. Tor simplicity, the transition likelihoods in this example will always be 1.0 when legal. Thus, if a transition appears in Figure 3, it's likelihood is 1.0. Looking at the left context, the point at (?, *2)* has a normalized likelihood value of 1.0. Since this is the only node in that direction, it is the maximum node in that direction (see equation 1) and so the likelihood from the left is 1.0. From the top, the likelihood of being at point (1, 3) is also 1.0. Taking the average gives 1.0 and multiplying this by 0.5 (the statistical from Figure 5) match yields an un-normalized likelihood of 0.5 for this state.

The other *node* at position (2, 3) is a B on the path which includes labelings 1 through 5. The statistical match of that point to PPE B is 0.6. Since there are two previous nodes to the left, calculation of the likelihood from the left involves finding the best previous node in that direction. In this case, the B in position (2, *2)* is stronger with a likelihood of 1.0. Since the likelihood from the top is 0.75, the average likelihood for that state is 0.87. When multiplied by the statistical match for that *PPE,* the un-normalized likelihood of that node becomes 0.79.

Once all node likelihoods are calculated, they are normalized so that the largest likelihood becomes 1 and the others increase proportionately. For this reason, the top node expands to 1.0. The PPE for C at position (2, 3) is pruned because it's un-normalized likelihood is 0.5. Note that the back-pointers which are saved here tell which previous node is the best. The back-pointer for the top node at point (2, 3) selects the B at point (1,3) from the top and the B at position (2, 2) from the left. The selection of the node to the left is done because that node had the greatest previous node likelihood.

The calculation of node likelihoods proceeds in a raster manner. This means that the order of calculation is (1,1), (J,?), (1,3), (1,4), (2,1), (2,2), *(?* 3), (2,4). When all of the likelihoods have been calculated on the forward pass, the backtrace makes a quick pass of the back-pointers to determine the global path. This backtrace proceeds in a reverse raster manner and follows the back-pointers left by the forward pass. Rc?fering to Figure 6 again, you can see that the 0 at point (2, 4) is the first point chosen for the global path. From there, two points are suggested to the top and left: the C at (1, 4) and the B at (2, *3).* The next point in the backtrace is the B at (2, *3)* which was indicated from the C to its right. It indicates the B in (1, 3) and the B in (2, *2).* The backtrace looks *very* simple but occasionally runs into snags. In this example, there is a conflict of back-pointers at position (1, 3). The state to its right is a C and indicates that (1, 3) should be a C. The state below it is a B and indicates that (1,3) should be a 13. Although conflict resolution is very difficult *and* not fully solved, the solution here is obvious. The network does not allow a C at (1, 3) to exist above a D at (2, 3) and the B has already been chosen. Therefore the B al (1, 3) is chosen because it docs not contradict any network constraints. Once the B at (1, 3) is selected, the rest of the backtrace proceeds smoothly and path 4 is identified as the correct labeling.

This example shows an interesting property of the Locus search: error correcting without backtracking. Notice that the strongest path, from the start, is the one which leads to labeling 7. This is based on the weak assumption that the point (1, 3) is a C. In truth, it could be any PPE, but the network rules out A. Of the remaining choices, C is statistically better than B, so the forward pass prefers C. On a global basis, however, B is better because path 4 is more globally consistent with the sensed data. This is because the point (2, 3) is strongly favored to he a B over a C, and it is not until the global pat!) is assembled from the back-pointers that the mistake is corrected. If this data were run through a backtracking system which employed a best-first search, the incorrect path would be selected because of the search technique's inability to bring global context to bear on local labeling. If some way of detecting the error were found, the system would have to back up before continuing. In this example, Locus provides a better solution in the same amount of time while in other cases, we can reduce the search time without sacrificing accuracy.

Discussion

The model presented in this paper has been used to interpret OhlanderV, city scene, demonstrating the initial validity and usefulness of the model. We plan to use the model to interpret arbitrary views of downtown Pittsburgh (a 3-D world), and different satellite views of the Washington, D.C. area (a 2-D world). Representation of the knowledge about 3-D and 2-D world models in terms PPE graph structure requires the development of several preprocessing programs (the PPE graph for Ohlander's city scene was generated manually). In this section we will discuss the relationship of this model to other approaches in image recognition research, and our present views of the strengths and limitations of this approach.

The graph structure representation proposed here is a natural outgrowth of work in languages (Aho and Ullman, 197?), graph representations (Harlow, 1972), and syntax directedpattern recognition (Narasimhan, 1966; Clowes, 1969; *and* Fu, 1976). The approach presented in this paper principally differs from the above in how the network representation is to he used. It rejects the notion that image recognition is best viewed as a problem in parsing. Given the error and uncertainty associated with the decisions, the problem tends to be not one of deciding whether a given pattern is parsahl'e but rather one of search, i.e., deciding which of the many acceptable alternalivc paths represents the near-optimal interpretation.

The view that the problem of image recognition is one of constraint satisfying search has been gaining increasing acceptance (Waltz, 1975; Tenenbaum and Barrow, 1976; Rosenfeld, Hummel and Zucker, 1976). This paper also subscribes to this viewpoint and differs mainly from the other efforts in the representation of constraints and the method of search.

The realization that one needs to introduce some measure of the degree of uncertainty into the interpretation process is reflected in the papers by Fischler and Elschlager (1973), Feldman and Yakimovsky (1975), and the probabilistic relaxation methods under development at SRI and Maryland. The method proposed here is able to handle search in the presence of error and uncertainty in a natural and straightforward manner provided all knowledge and constraints are represented in terms of a PPE graph structure.

Constraint satisfying search in the presence of uncertainty is also a central problem in other areas of AI, in particular in speech understanding systems research. Several techniques developed for use in the speech area such as representation of knowledge sources as cooperating independent processes (Roddy et al, 1973; Lesser et al., 1975; Erman et al., 1977), island driven search (Woods et al., 1977; Erman et al., 1977), and network representations of knowledge (Baker, 1975; l.owerre, 1976) also appear to be relevant to other knowledge based systems research, including vision. The Locus model presented here was first developed for use in the Harpy connected speech recognition system. Though the basic ideas remain the same, the model had to be revised substantially to make it useful in image recognition.

The best-first search given by the A* algorithm (Nilsson, 1971) and the breadth-first graph search of the dynamic programming algorithms (Levine, 1977; Bellman, 1962) provide alternative approaches to optimal graph search problems. The beam search technique of the Locus model provides a minimal effort near-optimal solution and appears to be effective in cases where the evaluation function is a function of an external signal source and where a large number of decisions arc related to each other in that they are all attempting to provide alternative interpretations of the same signal segment.

Occasionally, the heuristics associated with the beam search lead to the elimination of the optimal path. This need not be cause for major concern because a good path will always be chosen. Since the match likelihoods are less than accurate, attempting to find the optimal path at great cost and effort leads to little or no improvement in performance.

The success of beam searching is highly dependent on the choice of the "near miss" threshold that defines the width of the beam. If the threshold is too small, the resulting narrow beam of alternatives could lead to the pruning of the correct path. On the other hand, many unproductive paths will be examined if the threshold is too large. In our present system, the threshold is chosen large enough so as to maKe it

immune to local errors in the assignment of likelihood values to symbols horn the signal.

A significant featuie of the Locus model of search is its linearity. Because Locus prunes all but a narrow beam of alternatives, its search time is linear with respect to the size of the input signal *and* is essentially independent of the symbol space size. Thus, Locus searching controls the combinatorial explosion that occurs in most graph searching techniques. Note, however, that the size of the beam expands and contracts during the search as the connectivity between symbols in the graph increases and with the degree of uncertainty of the decisions.

The order of search in Locus is a subtle issue that appears to be a problem but upon closer inspection turns out to be unimportant. When using Locus in speech understanding, there is one independent dimension of time which can be used to order the search. In image processing with static pictures, there are two dimensions, so a raster scan is used. This might appear to cause continuity problems especially at the end of a scan line. However, Locus requires only the local context for a point and it propagates the global context regardless of search ordor. Thus, any search pattern can be used as long as it is reversed on the backtrace. Note also that the raster scan has the advantage of allowing the use of context for horizontally, vertically, and diagonally adjacent states.

A main concern with the finite state networks is that not all relational constraints are easily representable within that framework. We have not found this to be a problem in the 3-1) and *? D* worlds we have considered so far, although the representations tend to be expensive in terms of the space (memory) required. We expect to use a post-pass to apply constraints that are not easily incorporated into the network.

Knowledge such as shape, size, orientation, and location of objects is different from spectral properties of objects such as color and texture information. This type of knowledge cannot be included directly as part of the signal-to-symbol match. Such supra-segmental knowledge can however be incorporated into the network in several different ways (Rubin, 1977).

The example given in this paper shows how the Locus model can be used in the interpretation of the image on a pixel by pixel basis. The technique is equally useful with pre-segmented data. In fact, segmentation improves accuracy and leads to substantial speedup in matching and search. While extra segments do not cause any problem, missing segments cannot be accomodated within the present scheme without additional'computational effort. This is because the graph representation permits self-transitions in the PPE network but does not (at present) permit skipping PPE nodes.

Conclusion

This paper provides a framework for knowledge representation and search for image recognition tasks, leading to an easily implementable total systems framework within which one can explore the relative merits of different types of knowledge. One still has to decide what knowledge is available, how to acquire and define it, how to select an adequate set of primitive picture elements (PPE) for a given task, and how to match symbols (PPEs) to the signal. However, each of these subtasks look much more manageable to us than the original image interpretation task.

Acknowledgcments

The authors would like to thank Larry Davis, David McKeown, and John Kender whose comments were helpful in the preparation of this paper.

Appendix: .A Roal Example

This appendix shows the Locus model as it has actually been programmed at Carnegie-Mellon University. The program which uses Locus is called ARGOS and this example is direct output from the system. The scene that we are using is the downtown Pittsburgh image shown in Figure 7: the most complex image used by Ohlandcr (1975). The image was broken down into 15 PPEs shown below:

Sky	A
Mountains	B
Gateway Towers	C
Hilton	D
State Office	E
Pittsburgh Press	F
Jenkins Arcade	G
Hornes	H
Hornes Extension	I
Gateway 3	J
Gateway 2	K
Gateway 1	L
River	M
Park	N
Road	O

Figure 8 shows the labelings that were generated for the picture in Figure 7. the letters correspond to the PPE letters in the above list. Although this is not a perfect labeling by any means, it is an improvement over un-guided labeling and shows that Locus is effective for image interpretation.

References

Aho, A. V. and Ullman, J. D. (197/!). The Theory of, Parsing Translation, and Compiling, Prentice-Hall, Englewood Cliffs, N.J.

Baker, J. K. (1975). "1 he DRAGON system—An overview", IEFE Teans Acoust., Spcech, Signal Processing, vol. ASSP-23, 24-29, Feb."

Bellman, P. *and* Dreyfus, S. (1962). Applied Dynamic Peogeamming, Princeton Liniv. Press, Princeton, N.J.

Clowes, M. C. (1976). "Pictorial relationships - A Syntactic Approach", in Machine Intelligence IV (Meltzer and Michie, eds.), American Elsevier, New York.

Erman, L. D., et al. (1977). The Hcarsay-II System, (in preparation).

Feldman, J. A. and Yakimovsky, Y. (1975). "Decision Theory and Artificial Intelligence: I. A Semantics Based Region Analyzer," Artificial Intelligence; vol 5, pp. 349-371.

Fischler, M. A. and Eschlager, R. A. (1973). "The Representation and Matching of Pictorial Structures," IEEE Transactions on Computers, January.

Fu, K. S. (1976). "Syntactic Pattern Recognition," in Digital Pattern Recognition (Fu, eel.), Springer Verlag, New York.

Harlow, C. A. (1972). "Image Analysis and Graphs," TR IAL-TR 17-72, Image Analysis Laboratory, University of Missouri-Columbia, September.

Levine, M. D. (1977). "A Knowledge-Based Computer Vision System," Workshop on Computer Vision Systems, Vol. II, University of Massachusetts, Amherst, Mass., June, 1977.

Lessor, V. R., Fennell, R. D., Erman, L. D., and Reddy, D. R.(1975). "Organization of the Hearsay-II Speech Understanding System," IEFE Trans. Hum. Factors Peclroa, vol. ASSP-?3, J 1-23.

Loweire, B. T. (1976). "I he HARPY Speech Recognition System," (Ph.D. Thesis, Carnegie-Mellon University), Tech. Report, Computer Science Department, Carnegie-Mellon University, Pittsburgh, PA.

Lowerro, 13. T., and Roddy, D. R. 0 977). "Representation and Search in the Harpy Connected Speech Recognition System," paper to bo published.

Narasimhan, R (1966). "Syntax-Directed Interpretation of Classes of Pictures," Communications of the ACM, vol. 9, no. 3, March.

Nilsson, N. (1971). Problem Sqjvinn Methods in Artificial InjrIJi gence, Mc Graw Hill.

Ohtander, R. *B. (197b).* "Analysis of Natural Scenes," (Ph.D. Thesis, Carnegie-Mellon University), Tech. Report, Computer Science Department, Carnegie-Mellon University,

Roddy, D. R. (1977). "Aspects of Representation and Search in Perceptual Problem Solving," (in preparation).

Redely, D. R„ Errnan, L D., and Neely, R. B. (1973). "A model and a system for machine recognition of speech," IEFE Trans Audio Electioacoust., vol. AU-21, 229-238.

Rosenfcld, A., Hummel, R. A. and Zucker, S. W. (1976). "Scene labelling by relaxation operations," IEEE Trans. Syst., Man, Cybcrn,. SMC-6, 420-433.

Rubin, S. M. (1977). "The ARGOS Image Understanding System," (Ph.D. Thesis, Dept. of Computer Science, Carnegie-Mellon University), (in preparation).

Tenenbaum, J. M. and Barrow, H. G. (1976). "Experiments in Interpretation-Guided Segmentation," Technical Note 123, Stanford Research Institute, Menlo Park, CA.

Waltz, D. (1975). Understanding Line Drawings with Shadows, in The Psychology of Computer Vision, (P. Winston, Ed.), MIT Press, Cambridge, MA.

Woods, W. W., et al. (1977). "Final Report on Speech Understanding Systems," Bolt, Beranek and Newman Inc., Cambridge, MA.

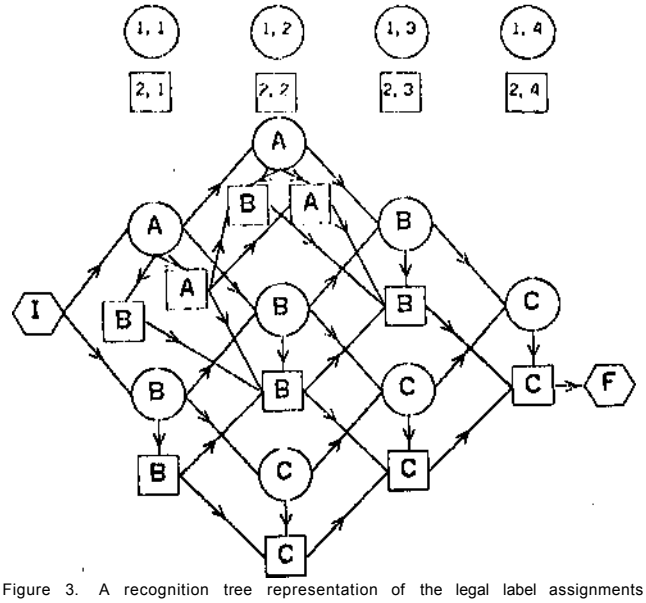
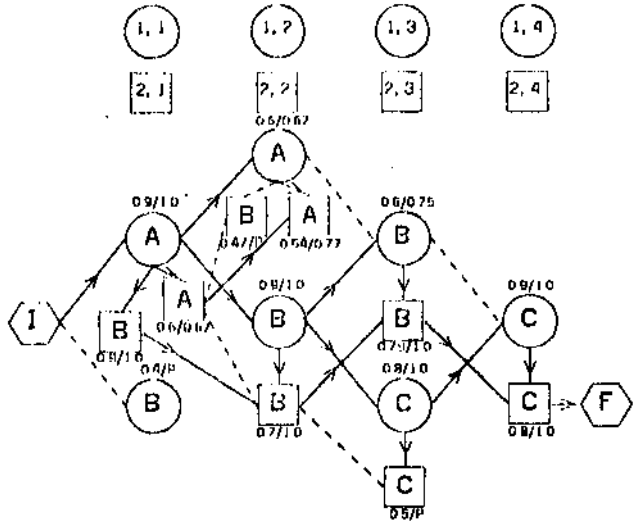


Figure 3. A recognition tree representation of the legal label assignments shown in Figuro 2 Each nodo in the graph represents one of the contextual situations possible for that pixel The graph is arranged in columns corresponding to the columns of the original image The shape of the node indicates the row number (see key at the top) Arrows indicate the possible transitions into and out of each state The hexagonal nodes are the Initial and



Figuro 6 Path of Locus search on reconmtion tree shown in Figure 3 Numbers separated by a slash are the un-normnlized and normalized nodo likelihoods, respectively The arrown indicate pathr. that were saved for global path re-construction and the dotted lines indicate alternative paths that were examined in the forward pess but not selected for the final recognition tree because of low likelihood values See text for more explanation.

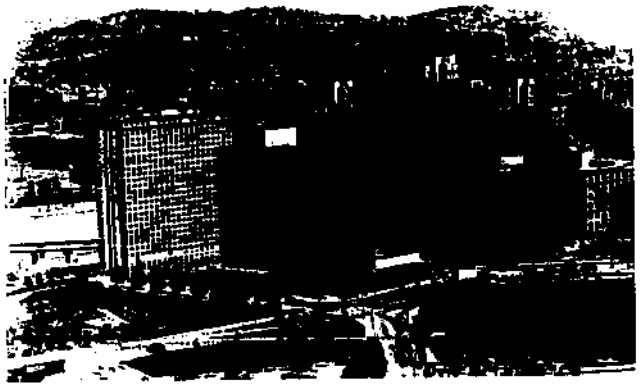


Figure 7 Downtown Pittsburgh Scene

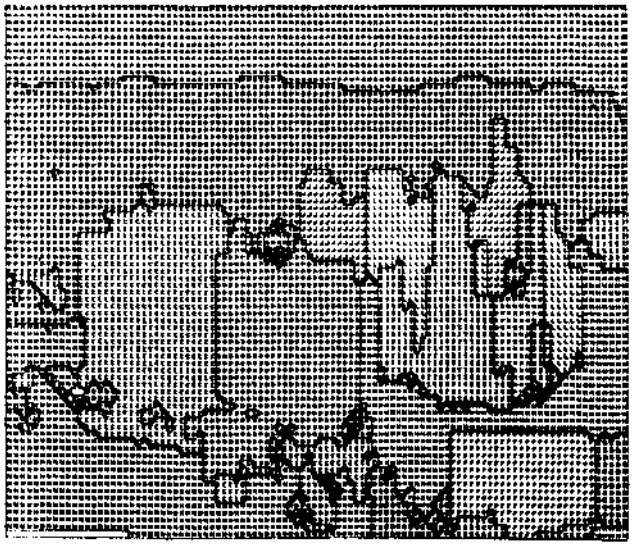


Figure 8. Labeling of Downtown Pittsburgh Scene shown in Figure 7. See Appendix for the label names of each letter.

STEPS TOWARDS THE REPRESENTATION OF COMPLEX THREE-DIMENSIONAL OBJECTS

Ruzena K. Bajcsy and Barry I. Soroka
Dept of Computer S Information Science
Moore School of Electrical Engineering
University of Pennsylvania / D2
Philadelphia PA 19174

INTRODUCTION

We are investigating the description of complex three-dimensional objects in terms of three-dimensional primitives and composition relations between them. Our input is a description of an arbitrary three-space in the form of gray-levels assigned to the points of a cubic lattice. Sequential tomograms are one source of input data, as are serial light and * electron micrographs.

Our approach is motivated by the observation that a primitive can give rise to only a small number of region types on a plane which cuts it. By inverting the process we can reason from a region found on a slice back to the primitives from which it might have been cut. When the sliced object is composed from several primitives, the regions cut by planes will reflect the joints and containment relations between the primitives involved.

AN EXAMPLE

Let us briefly consider right circular cylinders, which are sliced into circles, ellipses (possibly truncated), and rectangles. An ellipse suggests two possible cylinders. The next step in completing the description of space will be to determine which of the suggested cylinders is present (both may be!) and the length of the cylinder axis. This would involve looking at another slice some moderate distance away from the current one.

DESIGN CONSIDERATIONS

We are constructing (in LISP and FORTRAN) a processor with three useful features:

CD Primitives can be added or changed by altering the function which maps regions into possible primitives. Thus, to add "blocks" to the system, we would add the rules which tell us, for instance, that a triangular region is cut from a corner of a block.

(2) The strategy for obtaining a complete description of space may be systematically varied. What should the processor do when it discovers two disjoint regions on a single slice? Should it pursue the hypothesised primitives sequentially or in parallel? Furthermore, we would like to incorporate a mechanism which would mimic human abilities to perceive a Gestalt.

(3) The system should be interruptible so that at any moment we may stop the processor and ask questions like: What do you know at this point? What are you going to do next? What will it tell you?

Our interest in cognitive issues like (2) and (3) has led us to reject holistic approaches (such as the symmetric axis transform) in favor of a more heuristic approach.

INFORMATION FLOW.

If the system hypothesises the existence of a cylinder, then it may request a slice somewhere out along the axis in order to binary-search for the length of the axis. This request may be performed immediately or it may be enqueued with other requests, depending on the search strategy selected. In either case, the outcome of the request must be returned to the requesting hypothesis for evaluation. We store a deductive pattern (if-then-else) on the property-list of the requestor so that the deduction may be performed as soon as the requested information becomes available.

Some slices are a priori more informative than others. The most reasonable initial slice to examine is one through the center of space. Similarly, examining one slice devalues the slices immediately adjacent to it.

A region may affect dormant hypotheses (those waiting for unanswered requests). Thus, regions must be able to index into the set of current hypotheses and activate those potentially affected. A hypothesis may be both dormant and active!

CRITERIA FOR EVALUATING PERFORMANCE

We have been unable to find evidence of experiments into human competence and performance on the task our system will handle. Our system should be able to mimic human behavior in interpreting slices of three-space. Further, our model of hypothesis formation should account for interpolation given two slices, moderately separated, what is happening in the space between them?

CONCLSION

We have raised here some questions appropriate to the task of developing descriptions of three-spaces from slices. We have set forth design considerations for a processor currently under development. Further answers and details will be presented at the Conference.

REFERENCES

General references for this work are given in: B.I.Soroka and R.K.Bajcsy, Generalized cylinders from serial sections. Proc 3rd Intl Jt Conf PatternRecognition, Coronado CA, November 1976.

This research was supported by grant MCS 76-19465 of the National Science Foundation.

A "RECOGNITION CONE" PERCEPTUAL SYSTEM: BRIEF TEST RESULTS*

Leonard Uhr, Robert Douglass,
University of Wisconsin

This paper presents some preliminary test results of a "recognition cone" vision system. (See 1-3 for fuller descriptions). A recognition cone uses a parallel-serial layered architecture that successively reduces the image as each layer processes it. EAch layer contains a set of probabilistic "transforms" that combine diverse sources of contextually related information. Each transform looks at specified regions of the array and, if the combined weights of information found exceed the transform's threshold, then the transform implies a set of names and transforms to apply.

A number of variant systems have been coded in Snobol, to explore perception of binocular images, motion over time, and learning by discovery and induction. The long-term goal is to develop a relatively general model of living perceptual systems, one that perceives a variety of different kinds of scenes. The present results were obtained with a Simula-Fortran program that made possible tests with large input arrays.

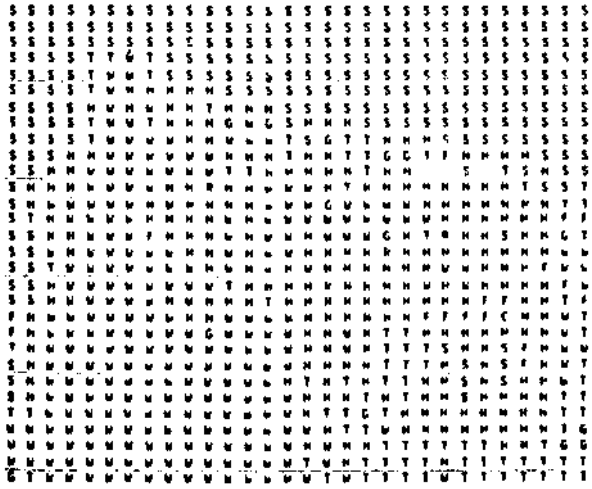
Test Results (see 3)

A 10-layer cone was used to process *a* 600 by 800 color TV picture (a house screened by trees, see 4 for the original).

Transforms were iterated throughout each layer, to assess the following information: Layer 1: averages; 2: hue, saturation, intensity; 3: gradients; 4: short edges, texture; 5: long edges, compounds; 6: higher-level compounds; 7: still-higher-level compounds; 8-10 averages.

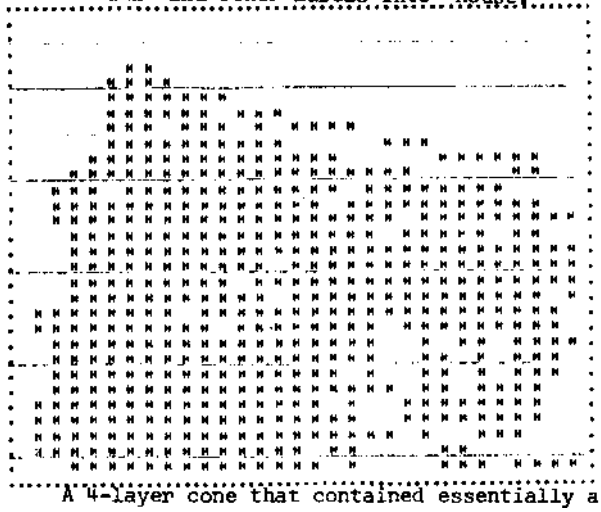
Figures 1-2 indicate some of the descriptive labels achieved.

Figure 1 shows the single most highly implied thing output to each cell by Layer 6 (F = Window, G = Grass, H = House, S = Sky, T = Tree, W = Wall).



*This research was partially supported by the National Science Foundation (grant MCS76-07333) and the University of Wisconsin Graduate School.

Figure 2 shows those cells into which House was most highly implied by Layer 7. Note how higher-level compounds have turned many of the local "Wall" and other labels into "House."



A 4-layer cone that contained essentially a sub-set of the transforms in layers 3 and 4 was used to describe 20 by 48 line drawings of place-settings of forks, knives, spoons and plates, where objects sometimes touched and overlapped. The system was also used to recognize letters and symbols that were severely distorted,by rotations, rubber sheet stretchings and large gaps.

Discussion

This kind of system can be evaluated only by far more extensive tests. But the use of many configurational transforms, each sensitive to contextually interrelated information,seems appropriate for handling a variety of unanticipated types of scenes. The use of weights means that parts can be gapped, distorted or missing. The layering gives speed and efficiency. It allows hierarchies of transforms to be built from common simpler transforms, as when edges build to angles to windows to walls to houses.

These transforms can easily be added, replaced, reweighted, adjusted and learned, without hurting the larger program. So it should be possible to make continuing improvements. The transforms are reminiscent of synapsing neurons, and the serial layering of many parallel processes reflects the overall structure of living visual systems.

References

1. Uhr, L., Layered 'recognition cone' networks that preprocess, classify and describe. IEEE Trans. Computers, 1972, 21, pp. 758-768.
2. Uhr, L., A model of form perception and scene description. Comp. Sci. Dept. Tech. Rept. 231, Univ. of Wisconsin 1974.
3. Uhr,L., and R. Douglass, A parallel-serial "recognition cone" system for perception; SOME test results. Comp. Sci.Dept. Tech. Rept. 292 , University of Wisconsin, 1977.
4. Ohlander, R. B., Analysis of Natural Scenes, Unpubl. PH.D. Diss., Carnegie-Mellon Univ., Pittsburgh, 1975.