—— WALTER REITMAN, ROBERT NADO, AND BRUCE WILCOX ——

Machine Perception: What Makes It So Hard for Computers to See?

Introduction

Go is a board game with an intellectual role in the Orient comparable to that of chess in the West. Several years ago we began work on a program we hoped would be capable of playing interesting Go. Though not intended as a strict simulation, insofar as possible the program was to be modeled after what we could learn of how a highly skilled human player plays the game.

Go is played with black and white tokens (called stones) on the 361 points of a 19 x 19 grid. Black and white take turns, each placing one stone at a time upon the board. Once played, the stones generally remain on the board until the end of the game. Figure 1 illustrates what a Go board looks like somewhere in mid-game, in this case after about 100 moves. As is evident, play typically results in highly intricate patterns of stones. Figure 2 (p. 76) shows an earlier stage of the same game, after approximately 25 moves. There are only a few locally complex patterns, but even at this stage there are quite complex global interrelations among friendly and hostile stones all around the board.

As these figures may suggest, the perception of local patterns and global interrelations among stones is a major factor in skilled play. Since our program was to function as an intelligent human player does, it became obvious that we would have to design perceptual

Note: We would like to thank I. Cantrall for his vivid account of web processing in nature, and M. Alpern, D. Krantz, R. Pachella, J. Reitman, L. Springer, S. Stich, and W. Utall for their helpful comments on an earlier draft.

Support from NSF Grants Nos. DCR71-02038 and MCS77-00880 is gratefully acknowledged.

components for the program. Furthermore, since the system was to be not only a game-playing program, but an instantiation of a more general model of human intelligence, we wanted to develop these perceptual components in ways that would be consistent with a reasonable conception of human perceptual activity.

The first part of this chapter is a brief summary and critique of one approach to machine vision that is now attracting a great deal of attention among those interested in endowing machines with something analogous to human perceptual capabilities. We present it here to suggest some of the main current issues and to motivate what we believe to be a more appropriate set of propositions. The second

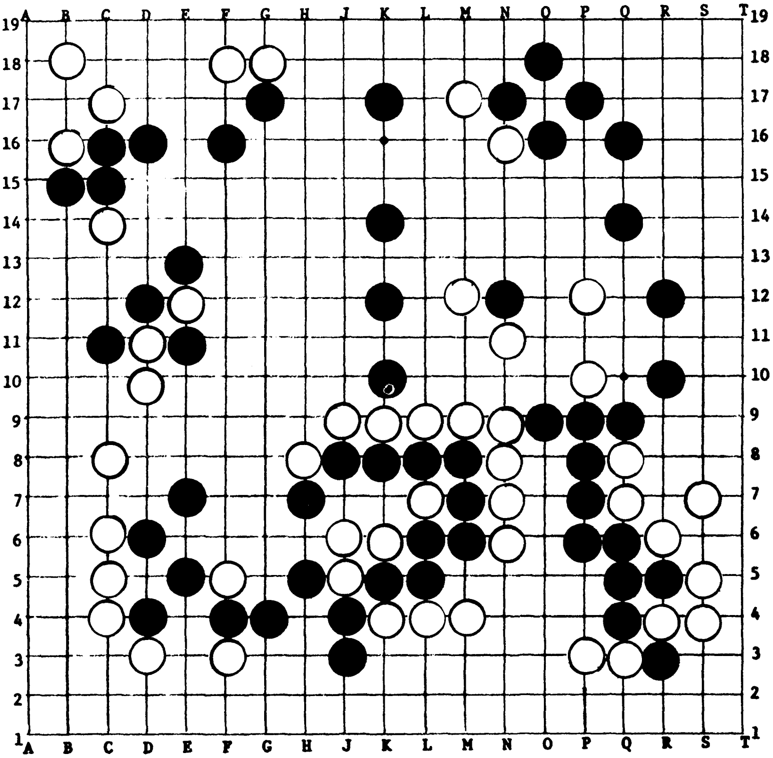


Figure 1. Five-stone handicap game after move 96. From Kerwin and Reitman (1973).

part of the paper describes in some detail one of the major components of the Go program's perception system. This component is certainly not a general theory, either of machine perception or of human perception. The specifics make sense only in the context of Go. But the basic ideas underlying the component and its relations with the rest of the program are, we believe, a reasonable view of the perceptual process, and thus the program illustrates how one can begin to incorporate such ideas in a machine implementation.

Minsky's Frame System Theory of Vision

Artificial intelligence approaches to perception have passed through at least two major phases over the course of their 20-year history. Until quite recently, most work aimed at trying to build perception from the bottom up. One built systems that tried first, for example, to organize points into lines, then to organize lines into regions and simple objects, and finally to describe the interrelations of the simple objects in the scene. Even though almost all of this work has been restricted to static complexes of relatively simple polyhedral objects (pyramids, cubes, etc.), a great deal of work turned out to be needed to get computers to carry out such scene description successfully.

More recently, Minsky (1975) has proposed an approach to vision that emphasizes the perceptual role of what we already know about objects and their interrelations. According to Minsky, the essence of frame system theory is this. When one encounters a new situation, one selects from memory a structure called a frame. This is a remembered data structure that can be adapted to fit the present reality by changing details as necessary. A frame consists of a network of nodes and relations. The top levels of the frame are fixed. They represent things that are always true about the supposed situation. The lower levels have many terminals or slots that must be filled by specific instances or data. Each terminal can specify conditions its assignments must meet. Attached to a frame are several kinds of information—for example, about what one can expect to happen next.

The effects of important actions are mirrored by transformations from one frame to another. As applied to visual scene analysis, a system of interrelated frames might be thought of as describing the scene from different viewpoints. The transformations from one frame of such a system to another represent the effects of moving from place to place.

According to Minsky (p. 213), the power of the theory hinges on the way in which frames account for expectations and other kinds of presumptions. A frame's terminals normally are already filled with "default" assignments. Thus, when used to organize a visual scene, a frame may contain many details not actually warranted by the situation. These default assignments "are attached loosely to their terminals, so that they can be easily displaced by new items that better fit the current situation."

Frame systems are linked in memory by an information retrieval network. When a proposed frame cannot be made to fit reality in the sense that we cannot find terminal assignments that suitably match the terminal conditions, the network provides a replacement frame. Once a frame is proposed to represent a situation, a matching process tries to assign each frame's terminals values consistent with the constraints associated with the terminals.

When we try to take frame theory as an explanatory account of visual perception, a number of objections arise. How is it possible to account for the perception of three-dimensional structure and for the apparent continuity of visual experience in terms of a model whose elements are a list-structured system of nodes and relations? Minsky acknowledges both of these questions, but his attempts to deal with them hardly seem satisfying.

With respect to the problem of the experience of three-dimensional space, for example, Minsky asserts (p. 220), "surely everyone would agree that at some level vision is essentially symbolic," and then argues that at the symbolic level the issue of dimensionality evaporates and the very concept of dimension becomes inappropriate. But the fact that vision is at some level symbolic does not mean that it must be so at every level, particularly if "symbolic" refers to whatever we can represent in structures of nodes and relations. How, for example,do we use such a structure to account for our perceptual experience when we look at the walls of our offices? We might be able to represent our ideas of rooms and blue walls with such structures, but how can they represent our experience as we observe that spatially extended expanse of blue on the other side of the room?

As for the experience of visual continuity, Minsky suggests (p. 221) this is an illusion due to the persistence of assignments to terminals common to the different viewpoints. "Continuity depends on the confirmation of expectations which in turn depends on rapid access to remembered knowledge about the visual world." For example, "just before you enter a new room, you usually know enough to 'expect' a room rather than, say, a landscape . . . and you can often select in advance a frame for the new room. Very often, one expects a certain particular room. Then many assignments are already filled in."

Such an approach may be useful as far as it goes, but it does not seem to go very far. Suppose we toss a pack of matches in the air and watch it carefully as it rises and falls, twisting and turning as it goes. We may not know exactly what it means to say that our perception of the object's path is temporally continuous, but still less is it clear how we account for our visual experience in terms of a sequence of static frames imposed from within.

Much of the inadequacy of Minsky's account arises from his unwillingness to distinguish sharply between experiences with eyes opened and eyes closed. Thus, discussing whether vision is symbolic, Minsky observes (p. 220) that "people have great difficulty keeping track of the faces of the six colored cube if one makes them roll it around in their mind." So they may, but what has that to do with the symbolic or nonsymbolic character of *vision*, as opposed to *imagination*? Perhaps a frame system may be useful for thinking about what happens when we imagine objects in space, or objects changing over time. But it hardly seems to account for the perceptual experience we have when we actually watch and see things going on in our environment.

Minsky's account also fails to distinguish adequately between what we can perceive and what we can recall. Minsky notes our inability to remember large amounts of perceptual detail after a perceptual experience. But that does not really speak to the role of that detail at the time of the experience. I may very well chuck out my intermediate calculations once I have arrived at my income tax. But my inability to retrieve those intermediate calculations *now* is no evidence at all for the view that I never made them, or that they were not absolutely necessary for arriving at the final result in the first place.

As we noted above, Minsky recognizes and attempts to deal with the two objections to his theory we have just discussed. There are other problems he does not deal with that should also be pointed out. The first is the problem of perceptual autonomy. Engrossed in a conversation or a chain of thought, you can walk for blocks or drive for miles with apparently only minimal awareness of your surroundings. But although you may have walked or driven in the wrong direction in such circumstances more times than you would care to count, rarely did you bump up against a wall or another car. Such observations suggest that whatever you were thinking about, your perceptual-motor system was doing a nice job all on its own of maintaining object separation and keeping you posted at least on the significant physical masses you were encountering in the course of your excursions. It is easy to think of such experiences, at a general sort of level, in terms of a semi-autonomous perceptual system operating largely independently of frame structures and processes. It seems difficult to account for such experiences in terms of a fundamentally serial frame-based system of the sort Minsky describes.

Even the basic paradigm for recognition that Minsky uses is suspect. Sitting in a dark room, you can see and identify an unanticipated object under a single flash of stroboscopic illumination even though the light falling on your retina lasts less than a thousandth of a second. It is difficult to see how such an experimental fact is to be accounted for in terms of a model that talks of pulling frames from memory on the basis of cues, checking out the various candidate frames against incoming data, etc.

To be sure, as any experimental psychologist will tell you, the information conveyed to the retina by a flash of light does not disappear instantaneously. A detailed record of the experience at something close to the level of the original sense data (Neisser's [1967] term is *icon*) endures for at least a tenth of a second, and some trace may be present for as much as half a second or even longer. Furthermore, the lowest levels of the perceptual system feed upward, and we do not know how long the "intermediate calculations" made at higher levels remain available. Note, however, that we now are talking about information and perceptual components that are neither part of the environment (the room is dark again) nor part of frame system theory (long-term memory has not yet even been accessed).

In our view, these lower and intermediate level perceptual components are fundamentally important in perception. They are, we suggest, the physiological bases of the perceptual manifold, the rich, detailed experience we have of the environment around us. No doubt we have symbolic representations of faces, which lead us to expect to see two eyes, a nose, and a mouth together, with the nose positioned roughly between the eyes and the mouth. We also may have frames embodying previous experience with the different kinds of scratches one can have on a face. But when your daughter comes running in to you after the cat has been at her cheek, you *see* where the scratch is, its color, how it runs, what its orientation is, and so on. Minsky never suggests how such visual experience (everything we actually see of the scratch and its relations to the face) is to be represented in terms of interconnections among the terminals of frames, and frame systems appear to us an inherently unsatisfactory way to account for the perceptual manifold.

Minsky no doubt would agree that the lower-level sensory systems are important. He might also grant that they are not well represented by symbolic structures of nodes and relations. Our disagreement with Minsky concerns the intermediate levels. For Minsky the organization imposed by the frame systems is primary, and the low-level sensory information is used only to suggest and confirm frames and to furnish values to be assigned to frame terminals. For us the lower and intermediate level perceptual-motor components are semi-autonomous. As the driving and walking examples suggest, they lead a life of their own. They are essential to the experience of spatial extent, exact spatial locations and relations, spatial and temporal continuity, and the perceptual manifold in general. What we know affects that perceptual experience, not by providing a structure of symbolic slots, but by sculpting and interpreting the shapes, masses, and colors directly represented within the intermediate perceptual components.

We hope the foregoing discussion has indicated what we see as some of the main problems of machine perception, and suggest why we see frame theory in its present form as inadequate in the light of these problems. In what follows, we list some general propositions we would want to include in a more adequate theory of perception, and then show how one may begin to incorporate such ideas in a machine implementation for the game of Go.

Some Propositions for a Theory of Perception

The propositions we wish to consider all have to do with how humans perceive a collection of objects. We are not concerned here with specific sensory mechanisms involved; we wish to assert certain general functional propositions about the overall perceptual process itself.

(1) Perception of the external world is inherently spatial. The human perceiver does not use numerical coodinates to compute the spatial interrelations among objects. He has a unitary overall sense of the general positions of objects in relation to one another and to himself. He can tell directly which are close together (locally connected) and which are far apart. He has a sense of angle and direction. If one object is between two others in his visual field, he sees it directly. The inherent spatial quality of perception is most readily apparent when we consider such phenomena as the perception of symmetries. Complex symmetric relations requiring detailed computation on a point-by-point basis the human perceives immediately.

(2) The example of symmetry perception also suggests, at least as far as the distal senses, vision and audition, are concerned, that perception handles space in volume, not point by point. A slow processor, required to keep up to date on a more or less continuous basis, and often at a very fine level of detail, could hardly afford to process point by point, like a blind man with a cane, or like Shrdlu the simulated robot (Winograd, 1972) trying to find space for a block on a table. When we look up and see clouds, we are simultaneously seeing the absence of any sizeable opaque objects in the volume of space between us and the clouds. The "points" of the intervening space are handled simultaneously, in parallel.

(3) Those concerned with the analysis of static scenes tend toview perception as a matter of describing, symbolizing, and recording. Once these operations have been carried out, no further use is usually made of the information in the scene itself. But if we consider everyday perception over time, attempts at detailed, once-and-for-all recording for its own sake are very infrequent. The primary function of perception is to keep our internal framework in good registration with that vast external memory, the external environment itself. With the exception of eidetic imagery, an odd phenomenon seen mostly in children, and even then not very often (Neisser, 1967),

there is no evidence to suggest that man constantly takes in and stores in long-term memory great quantities of information from the environment. Why should he? When he wants the information he can look. Thus much of the describing, symbolizing, and recording that goes on is instrumental, sustaining the one critical condition the perceptual system must satisfy: that it maintain good alignment between the real world and the internal spatial model so that when we do want to use the external memory to find something out, we know both what we are looking for and what we are looking at. On this point, incidentally, our view and Minsky's appear to coincide.

(4) As was suggested earlier, however, the fact that we do not regularly store quantities of environmental detail permanently should not be taken to mean that we cannot see it, or that the perceptual system somehow operates without taking that detailed information into account. We can and often do ignore such information, and we generally cannot remember or reproduce much of it. But seeing at a glance is a fact, not an illusion, and can hardly be accounted for without reference to the rich detail the senses provide.

(5) One great advantage of the environment as an external memory is that it updates itself automatically. Viewed as a representation of itself, the information in the environment is never incorrect, never obsolete. For example, when the position of an object changes, its spatial relations to all other objects in the visual field change simultaneously, with absolutely no rescanning and recalculation necessary on the part of the perceiver. Whenever the perceiver wants to know the current state of some aspect of his immediate world, he can always be sure the information at his sensorium is up to date.

(6) At its most fundamental level, the human perceptual system is built sensitive to change, and in particular to movement. In the visual system, for example, change detection is built in right at the retinal level. The perceptual system thus need not waste time and effort in constantly scanning the environment. When significant change occurs, it is detected directly.

(7) When the person we are talking with frowns, we are aware of the global change of expression, the face frowning. Only with deliberate effort do we move from the overall expression to focus upon particular details. At one end, the perceptual system is tied into the current state of the sensorium. But at the other, as Minsky

emphasizes, it has access to a vast complex body of knowledge that enters, by means we are only beginning to examine in detail, into the definition of the current internal model of the environment. At some level we are of course simultaneously seeing at least some of the low-level sensory detail. What the perceptual component puts out to the rest of the system, however, are not aggregates of sense data but percepts, meaningful collections of objects meaningfully organized in space.

(8) Observe, finally, that most of us have no trouble conversing while driving, and we can solve complex problems walking across campus from classroom to office. It is hard to say anything about the intrinsic economy and efficiency of human perception in and of itself. But with respect to the control overhead it requires of the rest of the system, it is economical and efficient indeed. The perceptual component alerts us to significant environmental changes, and it provides the percepts we need, all with a minimum of deliberate interference and control from the problem-solving component.

To summarize, we suggest that the human perceptual system includes an intrinsically spatial component. This component processes space in bulk, focusing upon things that are there rather than things that are not. It keeps our internal perceptual representation well aligned with a self-updating environment, alerts us to significant changes, and provides meaningfully organized percepts to the other components of human intelligence with a minimum of extrinsic direction and control. We do not regard these assertions as necessarily self-evident; they certainly are not provable in any sense at the present time. But they seem at least plausible, and thus may serve as useful in thinking about the design of perceptual components for an artificial intelligence system. In this light, what follows may be viewed as an exploration of these principles as applied to the problem of designing an intelligent Go playing program.

Go hardly seems a very rich or dynamic perceptual situation. The processing that goes on involves only a few of the many capabilities involved in perception generally. There are no textures, no shading, and no third spatial dimension to worry about. The perception of spatial continuity reduces to a matter of recognizing local connectivity among neighboring discrete points separated by unit distances in the grid. With the exception of captures (relatively infrequent events involving removal of one or more stones from the board), stones once played remain where they are until the game ends. Thus the changes that occur generally come slowly, in discrete increments, as black and white each in turn place one stone at a time upon the board. But although the perceptual component for a Go program will be far simpler than a general perception system, the design problem is not trivial, especially if the design is constrained to satisfy propositions about human perceptual processes in general; and the overall design of the component may be useful as we assess the prospects for future work on more general machine perception systems.

The Elements of Go

Go is a contest for control of *territory* (the vacant intersections of the 19 x 19 grid). For detailed rules, see any good introduction to the game, for example Iwamoto (1972). In addition to the individual white and black stones, the Go program recognizes a variety of higher order units. A *string* consists of a single isolated stone, or of two or more stones of the same color located on immediately adjacent grid points. In Figure 2, for example, the two white stones at Q7 and Q8 form a string. Two strings of the same color, in close proximity, with no intervening enemy stones, are considered by the program to be *linked*. In Figure 2, for example, the white stones at J5 and M4 are considered to be joined by a "large knight's move" link. Strings in close proximity to the edge of the board, with no intervening enemy stones, are considered to be linked to the edge.

The single black stone at J3 is *enclosed* by an uninterrupted set of links running from the edge point F1 through the white stones F3, F5, J5, M4 and back to the edge at M1. Links and enclosures may under some conditions be broken by the interposition of enemy stones, but if conditions are unfavorable, this can lead to fighting that will be disadvantageous to the player attempting to break through.

A *group* consists of a single isolated string, or of two or more strings of the same color in close proximity, with no intervening enemy stones or links. The seven white stones on the lower right side in the figure form a group. So do the four white stones at the bottom of the figure, since there are uninterrupted links connecting each stone in the group to at least one other and therefore, by transitivity, to all of the others. The group is the primary unit of interest for our purposes in this paper, since what we are after is a fast, effective procedure for perceiving the strategic implications of moves, and this entails noticing and taking account of the significant spatial relations between a given new move and the existing groups on the board.

The extent and degree of prospective territorial control exercised by each side at any time during the game is a function of the relative security of each side's groups and the interacting dispositions of the two sets of groups over the board. For example, in Figure 2, black has some measure of control over the upper right side, and white is strong around the middle of the bottom.

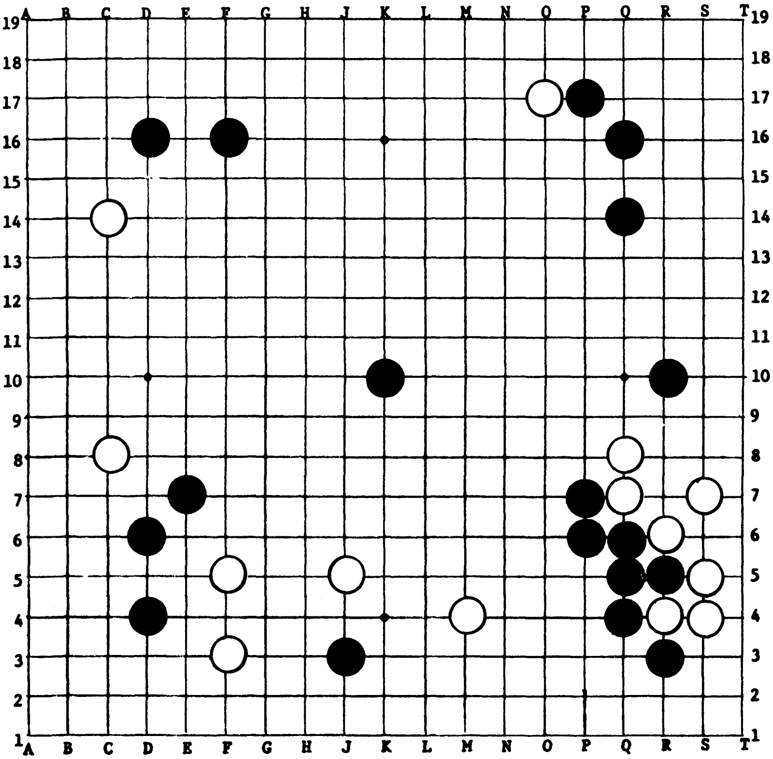


Figure 2. Early stage in a five-stone handicap game, i.e., a game beginning with black stones at D4, D16, Q16, Q4, and K10. From Kerwin and Reitman (1973).

The Go program associates with each group a set of *options* (see Reitman, Kerwin, Nado, Reitman & Wilcox, 1974). These are general ways in which that particular group may be developed, protected, and used. The white group at C8 in Figure 2, for example, may develop in subsequent moves along the left side, or out towards the center. Similarly the white group at O17 may develop along the upper side, or escape out into the center, or serve as a sacrifice stone to minimize black's territorial profits along the upper side or in the upper right corner.

Plays at one point on the board may augment, modify, or reduce the options of groups elsewhere on the board. Just as a car emerging from a side street may significantly affect the options of several other cars at some distance from it on the main road, so the play of a white stone at one point may affect the development of a remote black group—for example, by obstructing a potential escape route of that group. In particular, a group that earlier had several options for development and protection may, as a result of several remote plays by the opponent, be left with only one. Consequently, protection of that group becomes urgent. Otherwise, with a single additional move, the opponent may be able to disrupt the group entirely. In this sense, changes in option sets significantly affect the *focus* of the game.

Note that Go is a resource-limited problem. Not only do the two players compete with one another, but each side's groups compete for resources among themselves. Each player wants to develop each of his groups, to make them more secure. He also wants to establish new groups, to exert influence over a greater prospective territorial area. But he may only put down one token per turn. If he focuses too singlemindedly upon securing a small number of existing groups, his opponent will achieve broader influence than he does. If he spreads himself too thinly, setting up many insecure groups around the board, he risks undergoing attacks that may wipe out a large part of the influence and control he is trying to achieve. As we have seen, however, a single move by a player can have more than a single local effect. It also may broaden his options elsewhere, or restrict those of his opponent. This interactive resource-bound character of Go places a premium on multipurpose moves.

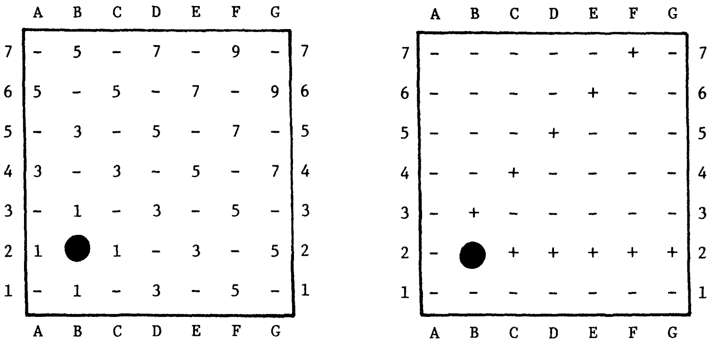
With resources limited and many things to be done, it is important that the perception component of the Go program include a means for determining, quickly and effectively, the multiple strategic implications of a single move. Our goal was to design such a system in accordance with the perceptual principles outlined earlier in this paper. For reasons that will become apparent in what follows, we call our result the web perception process.

The Web Perception System

The general operation of this system is most easily understood in terms of a pair of metaphors. First, to define the maximum scope of the system, imagine each group on the Go board as having its own two-dimensional radar, which operates in the plane of the board. Radar waves from the group pass through vacant points, but are reflected back by stones, links, and the edges of the board. Thus in Figure 2, for example, the white group at C8 can "see" from left to right: the left edge of the board; white C14 and its link to the left edge; black D16, F16, and the link between them; the upper edge of the board from about G19 to O19; white O17; and so on. Since the radar waves do not penetrate past stones or links, C8 cannot see such other stones on the board as white F3 or F5.

Now imagine that each group on the board also has its own spider. Each group's spider spins a web in all directions, over exactly those points of the board passed through or reached by that group's radar. The web terminates wherever it runs into a stone, a link, an edge of the board, or a radar shadow cast by some other stone (once we are clear about the scope of the web, we can forget about the radar). Thus the scope of the web is the entire area contained within the bounds defined by the set of stones, links, and edges visible from that particular group.

The overall process responsible for producing a web is SPINWEB. To understand how the web for a group is constructed, consider first the creation of a web for a one stone group, black B2, on an otherwise empty board (the truncated 7 x 7 grid shown in Figure 3a). The first circumferential strand of the web is spun around the *hub*, B2. It consists of nodes for the four points (A2, B3, C2, B1) directly connected to B2 by the horizontal and vertical grid lines passing through the stone. This strand forms a complete ring. To



	A	B	C	D	E	F	G	
7	-	5	-	7	-	9	-	7
6	5	-	5	-	7	-	9	6
5	-	3	-	5	-	7	-	5
4	3	-	3	-	5	-	7	4
3	-	1	-	3	-	5	-	3
2	1	●	1	-	3	-	5	2
1	-	1	-	3	-	5	-	1
	A	B	C	D	E	F	G	

(a)

	A	B	C	D	E	F	G	
7	-	-	-	-	-	+	-	7
6	-	-	-	-	+	-	-	6
5	-	-	-	+	-	-	-	5
4	-	-	+	-	-	-	-	4
3	-	+	-	-	-	-	-	3
2	-	●	+	+	+	+	+	2
1	-	-	-	-	-	-	-	1
	A	B	C	D	E	F	G	

(b)

Figure 3. Circumferential web strands (a) and radial predecessor-successor strands (b) generated for a one-stone group on an otherwise empty board (for easier reading, only odd-numbered circumferential strands and two of the radial strands are shown).

generate the second level circumferential strand from the first, or more generally, the n + 1th level strand from the nth level *basis* strand, SPINWEB calls SPINSTRAND. SPINSTRAND determines each node in the n + 1th strand from three contiguous nth strand nodes. Of these, the two side nodes may correspond to vacant, occupied, link, or shadow points on the board. The center node, however, must correspond to a vacant point. If the center node corresponds to an occupied point, or an edge, link, or shadow point, it forms part or all of a segment boundary. and this segment of the web ends there.

The three contiguous level n nodes define a local directional sense, which may be thought of as a segment of a straight line from the hub out through the center node. Whatever the specific pattern of the corresponding board points, NEXT uses the X, Y displacement from the first point to the second, and from the second to the third, to generate by table lookup 0, 1, or 2 nodes on the n + 1th level circumferential strand. Each of these new nodes is connected into this strand. At the same time, it also is connected into a radial strand emerging from the hub by means of a bidirectional connection back to the point of generation in the predecessor strand, the basis strand. SPINSTRAND now moves one node clockwise around the nth strand, and generation of the nodes of the *n* + 1th strand continues. To illustrate with respect to Figure 3, using the level 1 nodes corresponding to the configuration formed by A2, B3, and C2, SPINSTRAND generates level 2 nodes corresponding to B4 and C3. It now moves one node clockwise, and since B3, C2, and B1 form an identical configuration, SPINSTRAND adds nodes for D2 and C1 to the second level strand.

Generation of nodes on the *n* + 1th strand halts when SPINSTRAND either reaches its starting node in the basis strand or else come across a basis node corresponding to a nonvacant board point. If all of the nodes of a basis strand are vacant, all of the nodes of the next level strand are generated in one uninterrupted sequence by the process just described. In this case, the nodes just generated form a complete *n* + 1th level ring. If SPINSTRAND encounters in the basis strand a node corresponding to a stone, an edge point, or a link or shadow point, however, it treats that node as a boundary node and generation of further *n* + 1th level nodes ceases for the time being. In this case, the nodes just generated form not a complete ring but a ring arc. In either case, the ring or arc just created is now taken as the new basis, and generation of nodes on the next strand out begins.

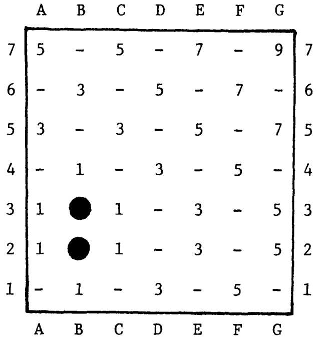
This means that the web is spun segment by segment. Only when the outermost strand of the present web segment has been finished —that is, when no further strands can be generated because the current strand consists entirely of one or more boundary nodes—does SPINWEB move back in one strand, to the right-hand boundary node of the immediately preceding strand. If that right-hand boundary node has an unprocessed vacant node anywhere after it in the strand, then the spinning of another segment begins there. Otherwise SPINWEB again moves inward one level and proceeds in the manner just described. At some point SPINWEB either finds some portion of an inner circumferential strand that can be spun out further, in which case the process continues as before, or it reaches its starting point on the innermost level strand, and the web for the group is finished.

For every board point visible from the hub, there is now a corresponding web node. Every nonboundary web node is connected in its circumferential strand to its immediate left and right neighbors, and it also is tied into a radial strand formed by the chain of bidirectional connections running out through it from the hub. Figure 3, a visualization of the resulting web for the example just discussed, shows separately (3a) the odd-numbered circumferential strands and (3b) two of the radial strands formed by the bidirectional connections between adjacent circumferential strands.

WEB ALTERATION

The web for a group will be updated whenever a friendly or enemy stone is played within its scope. Consider first what happens when a friendly stone is added to the hub. To be specific, assume black adds a stone at B3 to the group shown in Figure 3a, with the result shown in Figure 4. Note that the new stone must always be played at a point corresponding to a node in the level 1 strand of the old web. This node already is tied to a level 1 node immediately to its left, in this case the node corresponding to the board point A2. Now, beginning from that node, REWALK adds to the strand new nodes for A3, B4, and C3. When it reaches C2, it recognizes the node for C2 in the existing strand and reconnects the two ends of the strand at that point. Each successive n + 1th strand of the web is then augmented in its turn, using the newly added level n nodes as the basis.

A similar web modification process is initiated when a friendly stone is played within linking distance of an existing group. The details of this process are more complex because a number of pos-

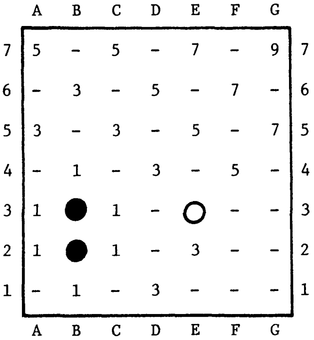


	A	B	C	D	E	F	G	
7	5	-	5	-	7	-	9	7
6	-	3	-	5	-	7	-	6
5	3	-	3	-	5	-	7	5
4	-	1	-	3	-	5	-	4
3	1	●	1	-	3	-	5	3
2	1	●	1	-	3	-	5	2
1	-	1	-	3	-	5	-	1
	A	B	C	D	E	F	G	

Figure 4. Web modification following the addition of a friendly stone to the hub.

sibilities are involved. Consider, for example, a white play at D11 in Figure 2. That stone forms links to both white C8 and white C14. Thus two webs have to be modified and, in effect, merged in order to come up with an appropriate web structure for the new three-string group. Conceptually, however, the end result is identical to that just discussed. The scope of the new web structure includes all those board points in direct line of sight from any string in the group.

When a friendly stone is played beyond linking distance to a group, or when an enemy stone is played anywhere within the scope of a web, a new process, DEWEB, becomes involved. DEWEB rips out from the existing web for the group all points lying in the shadow created by the play of the new stone (this includes points falling in the shadow of any links the play of that stone may have created). In the modified web, the new stone and the points at the border of the shadow now are boundary points, as are any new link points not already removed from the web. Consider, for example, Figure 5. This is the same as Figure 4 except that white has now played a stone at E3. This creates a link from E3 through E2 to the edge at E1. The nodes for these three points now form a new boundary for the redefined black group web, and they occlude the three bottom points on lines F and G. Since the six former web points can no longer be seen from B2 and B3, they are no longer in the scope of the black group's web.



	A	B	C	D	E	F	G	
7	5	-	5	-	7	-	9	7
6	-	3	-	5	-	7	-	6
5	3	-	3	-	5	-	7	5
4	-	1	-	3	-	5	-	4
3	1	●	1	-	○	-	-	3
2	1	●	1	-	3	-	-	2
1	-	1	-	3	-	-	-	1
	A	B	C	D	E	F	G	

Figure 5. Web modification after a white play at E3.

WEB PERCEPTION IN OPERATION

Two brief examples will help to show how the web perception system works in a game context. Figure 6 shows the web for the white stone at C8 in the game situation originally introduced in Figure 2. As in Figure 3, only odd-numbered circumferential strands and a few of the radial strands are shown. Circumferential strands are represesented by unit digits throughout (e.g., 13th-level nodes are represented by 3s, and so on).

Suppose for the first example that the next black play is at D9. This stone links to black E7, black D6, and the edge at A9. Associated with D9 and the link points are lists of web nodes, one node in each list for each group having an unobstructed view of that point. The web for each such group now "vibrates" at those nodes, and

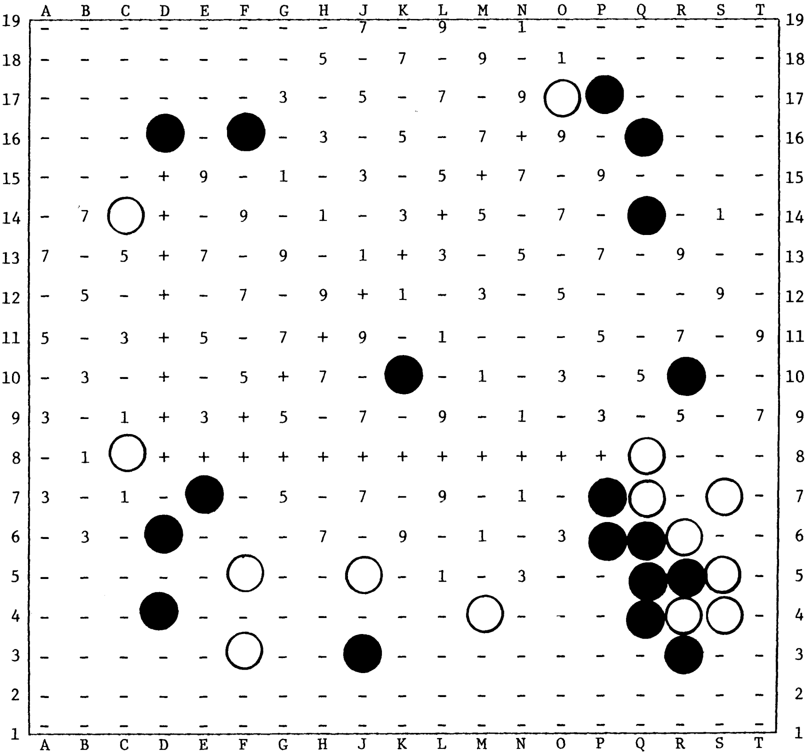


Figure 6. Actual game situation, with web for white C8 shown.

each group for which the play may have strategic significance is immediately alerted. Every such group knows not only that the play has occurred, but also the relative spatial position of the points involved, as given by their radial distances and directions from the hub and their circumferential relations to other nearby groups. In addition, once the web modification process is complete for all affected webs, each group also can see exactly what the immediate strategic effects upon its options are. To appreciate this, consider Figure 7, which shows the web for white C8 after black D9 is played. As a comparison of the two webs in Figure 6 and 7 will show, black's one move effectively encloses the white C8 stone, wiping out its previous developmental options along the left side and out to the center.

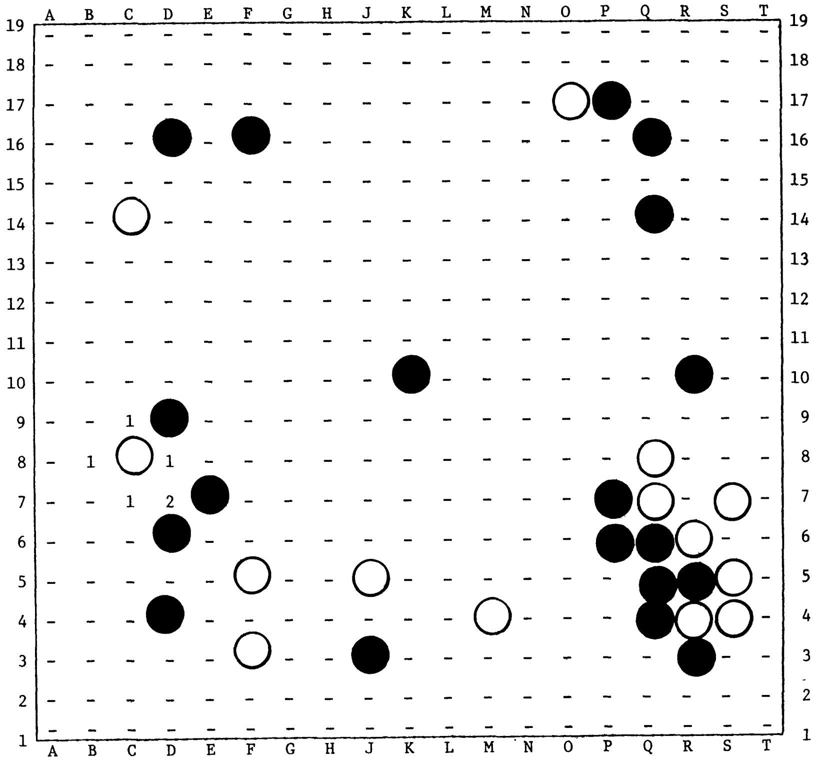


Figure 7. Web for white C8 after black play at D9.

Consider now as a second example a white play at R17. Though premature at this stage of the game because it lacks strategic significance, later on such a move would give white a chance to steal black's corner. The interesting point to notice is that since R17 is enclosed by black stones and links, only one web vibrates, the web for the black group in the upper right corner. In other words, the *absence* everywhere else on the board of any immediate strategic effect is "noticed" directly, with a minimum of computation. It simply falls out of the structure of the webs and the way the web processor works. In the early stages of the game, web processing requires considerable computational and storage overhead. But as this second example suggests, once there are a substantial number of stones on the board and the webs of most groups are, accordingly, quite closely restricted, web processing is not only a conceptually attractive means of handling the resulting complex spatial interrelations, but computationally a rather efficient means as well.

WEB PERCEPTION IN A GO PROGRAM

The system just described is part of a general Go playing program being written in LISP/MTS (Hafner & Wilcox, 1974). An interim version of the system is running on the University of Michigan's Amdahl/470 system. For details of the system and an assessment of its performance, see Reitman and Wilcox (in press). The web processor is only one of several interrelated components responsible for all aspects of spatial perception and representation in the program. Among the others completed or under development are a sectors processor and a patterns processor. The sectors processor complements the web system in maintaining global information about the arrangement of all groups on the board, but with none of the detail provided by the webs. It thus corresponds to what the program *knows* in general about the overall spatial situation, while the webs correspond to exactly what the program can (and cannot) *see* from the vantage point of each group. The patterns processor recognizes configurations of stones the program is acquainted with, and organizes the analysis of all other patterns that fail to match some stored representation.

The web processor serves a number of other functions in addition to determining the immediate strategic implications of plays. In conjunction with the program's lookahead system, and with web scope restricted to link distance, it contributes to tactical lookahead. In this case, the webs correspond to the spatial perception of a player focusing upon the delimited board area involved in a given tactical problem. The web processor also is used in working out the remote strategic consequences of a move, i.e., those consequences that follow not so much from the move itself as from the tactical sequence of plays the move initiates.

The Web Perception System in Perspective

Now that we have considered the details of the web perception system, it may be useful to look at it on a more general level. Interpretation is risky; it is easy to get carried away. But if we do not take the enterprise too seriously, thinking about the system in more general contexts may be suggestive.

(1) Any information-processing system must make choices between storage costs and computing costs. In the case of the web perception system, and perception generally, however, the tradeoffs involve real time. Just as the orb spider invests considerable time and energy in creating its tactile amplifier beforehand, thus ensuring a fast and appropriate response when its prey appears, so the web perception system invests in its webs, to respond quickly and effectively when a new stone is played.

(2) One of the basic bottlenecks for machine perception and problem-solving is the cost and difficulty of effective search. But as the web perception system suggests, if we are willing to invest in building up the appropriate information structures, we can cut down radically on the amount of search required for effective performance. The webs function as a dynamic indexing scheme. They guarantee the Go program that it will be able to relate environmental events (the play of new stones) to exactly those existing structures (the groups) strategically affected by them, and under conditions in which extended search might entail prohibitive time and computational costs.

(3) The webs may also be thought of as a solution in the context of Go to the problem of interfacing general purpose knowledge with ongoing events. Note that although the web-weaving and detection algorithms are fixed and autonomous, the particular webs resulting

from the weaving process are dependent upon and exquisitely sensitive to subtle differences in the configurations on the board.

(4) The web perception system's work results in a significant reduction in the data that have to be considered by higher-level components of the program. Thus web perception may be thought of as a procedure for answering general strategic questions ("which groups are affected by that move?") for higher-order units in the system. In this sense, the role of the web component within the Go program as a whole is consistent with the general "question-answering" organizational scheme for complex artificial intelligence proposed by Bobrow and Brown (1975).

(5) Finally, as we indicated above, the web perception system may be thought of as a first exploration of a set of general principles for dealing with problems of natural and artificial perception. We do not want to argue this point too strongly; what seems to us a reasonable instantiation of a principle may appear strained and farfetched to someone else. But we do feel the general principles outlined earlier have had substantial heuristic significance for us, and we believe they also may prove useful to others.

References

Bobrow, R. J., & Brown, J. S. Systematic understanding: Synthesis, analysis, and contingent knowledge in specialized understanding systems. In D. G. Bobrow & A. Collins (Eds.), *Representation and understanding. Studies in cognitive science.* New York: Academic Press, 1975.

Hafner, C., & Wilcox, B. *LISP/MTS programmer's manual.* The University of Michigan, Computing Center Memo M251, 1974.

Iwamoto, K. *Go for beginners.* Tokyo: Ishi, 1972.

Kerwin, J., & Reitman, W. *Video game #3. A Go protocol with comments.* Unpublished paper, The University of Michigan, Mental Health Research Institute, 1973.

Minsky, M. A framework for representing knowledge. In P. H. Winston (Ed.), *The Psychology of computer vision.* Princeton, N.J.: McGraw-Hill, 1975.

Nado, R., & Reitman, W. *Natural and artificial intelligence.* Hillsdale, N.J.: Lawrence Earlbaum Assoc., in preparation.

Neisser, U. *Cognitive psychology.* New York: Appleton-Century-Crofts, 1967.

Reitman, W., Kerwin, J., Nado, R., Reitman, J., & Wilcox, B. Goals and plans in a program for playing Go. *Proceedings of the 29th National Conference of the Association for Computing Machinery,* 1974, 123-127.

Reitman, W., & Wilcox, B. Pattern recognition and pattern-directed inference in a program for playing Go. In D. Waterman & F. A. Hayes-Roth (Eds.), *Pattern-directed inference systems.* New York: Academic Press, in press.

Winograd, T. *Understanding natural language.* New York: Academic Press, 1972.