Foundations and Grand Challenges of Artificial Intelligence

Raj Reddy

Last November, I got a call from Bob Simpson and Allen Sears, program managers for Artificial Intelligence at DARPA, which has been the major sponsor of AI research for the past twenty-five years. It was a call for help. "We are being asked some tough questions by the front office," they said. "What are the major accomplishments of the field? How can we measure progress? How can we tell whether you are succeeding or failing? What breakthroughs might be possible over the next decade? How much money will it take? What impact will it have? How can you effect technology transfer of promising results rapidly to industry?" They needed the answers in a hurry.

There was a time when we could say, "Trust us, we will accomplish great things." After twenty-five years of sustained support, it is not unreasonable to ask some hard questions. AI has become a big science, demanding a significant chunk of research dollars and accounting for over half a billion dollars of the GNP. The AI research budgets of DARPA, NSF, NASA and NIH alone account for over $100 million. In this age of huge budget deficits and many exciting ideas competing for constant research funds, it is only natural for a research manager to want to invest the research dollar wisely in activities with the potential for highest payoff. As the size of investment in AI rises above the noise level, we can no longer expect people to fund us on blind faith. We are entering an era of accountability. Rather than being concerned, I think we should view this as a challenge and lay out our vision for the future.

My first reaction to Simpson's request was, "Hey, we have nothing to worry about." Our track record is terrific. We have created almost a billion dollar-a-year enterprise around the globe. Previously unsolvable problems are being solved. People who have never used a computer before are solving problems with one. Vision and robotics are beginning to transform manufacturing. Planning and scheduling systems are in routine use in industry. Speech and language interfaces are becoming preferred modes of communication. OK, maybe there are a lot of applications, but what about scientific progress?

AAAI is a society devoted to supporting the progress in science, technology and applications of AI. I thought I would use this occasion to share with you some of my thoughts on the recent advances in AI, the insights and theoretical foundations that have emerged out of the past thirty years of stable, sustained, systematic explorations in our field, and the grand challenges motivating the research in our field. First, let us look at some recent advances.

Advances in Artificial Intelligence

What can we do today that we could not do thirty years ago? It is fortunate that AI has several areas in which there has been sustained research over the past twenty to thirty years. These areas are chess, natural language, speech, vision, robotics and expert systems. I would like to illustrate the progress by providing a historical perspective on some of these areas.

Chess

Let us look at the problem of computers that play chess. Chess is an AI problem par excellence. It plays the same role in artificial intelligence that the studies of E. Coli play in biology. It illustrates and illuminates many problems that arise in AI and leads to techniques that can be generalized to work on other problems. Chess is perhaps one area which has been studied continuously since the birth of artificial intelligence in 1956. Funding agencies were afraid to support research on fun and games lest they receive a Golden Fleece award from Proxmire. In spite of lack of support, the situation is not too bleak. A $100,000 Fredkin prize awaits the system that beats the world champion.

In the mid-fifties, we had several attempts to build chess playing programs. The most notable of these was the Newell, Shaw, and Simon chess program (1958). The first published method for game tree pruning (a precursor to the alpha-beta algorithm) appeared in the NSS chess program. It was also the first system to attempt to incorporate "a practical body of knowledge about chess."

The next major landmark in chess occurred when Greenblatt's chess program (1967) won a game from a class C player in a 1966 tournament. This win revived interest in chess playing programs, and led to the annual ACM computer chess tournaments. The seventies were dominated by the Northwestern chess program (Slate and Atkin 1977). Ken Thompson of Bell Laboratories (1982) developed an application specific computer architecture for chess in the early eighties and illustrated the powerful idea that a $20,000 piece of hardware, if structured properly, could out-perform a $10 million dollar general purpose computer. Thompson's Belle program received the $5,000 Fredkin award for being the first system to achieve a master's level rating in tournament play in 1982. Hitech, a VLSI based parallel architecture created by Hans Berliner and Carl Ebeling (1986) has dominated the scene since 1985; it currently has a senior master's rating. During the last year, Hitech played 48 tournament games. It won 100% of all games played against expert-level players, 70% of all games played against master and senior-master level players, but only 15% of the games against grand masters. Just last week, a new system based on a custom VLSI design developed by two students at Carnegie Mellon University, C.B. Hsu and Thomas Anantharaman (Hsu, 1986; Anantharaman et al. 1988), called "Deep Thought," defeated the highest rated player ever in the US open chess tournament.

What did we learn from chess research? In addition to a number of insights into the nature of intelligence (which we will discuss a little later), chess research has also led to the development of a number of techniques such as the alpha-beta pruning algorithm, B* search, singular-extension-search, and hash table representation for keeping previously examined moves. The proceedings of the 1988 Spring symposium of AAAI on game playing contain a number of interesting papers which are illustrative of current research on the topic.

Speech

Speech recognition has a long history of being one of the difficult problems in Artificial Intelligence and Computer Science. As one goes from problem solving tasks to perceptual tasks, the problem characteristics change dramatically: knowledge poor to knowledge rich; low data rates to high data rates; slow response time (minutes to hours) to instantaneous response time. These characteristics taken together increase the computational complexity of the problem by several orders of magnitude. Further, speech provides a challenging task domain which embodies many of the requirements of intelligent behavior: operate in real time; exploit vast amounts of knowledge; tolerate errorful data; use language and learn from the environment.

Voice input to computers offers a number of advantages. It provides a natural, hands-free, eyes-free, location-free input medium. However, there are many as yet unsolved problems that prevent routine use of speech as an input device by non-experts. These include cost, real-time response, speaker independence, robustness to variations such as noise, microphone, speech rate and loudness, and the ability to handle spontaneous speech phenomena such as repetitions and restarts. Satisfactory solutions to each of these problems can be expected within the next decade. Recognition of unrestricted spontaneous continuous speech appears unsolvable at the present. However, by the addition of simple constraints, such as a clarification dialogue to resolve ambiguity, we believe it will be possible to develop systems capable of accepting very large vocabulary and continuous speech dictation before the turn of the century.

Work in speech recognition predates the invention of computers. However, serious work in speech recognition started in the late fifties with the availability of digital computers equipped with A/D converters. The problems of segmentation, classification, and pattern matching were explored in the sixties and a small vocabulary connected speech robot control task was demonstrated. In the early seventies, the role of syntax and semantics in connected speech recognition was explored and demonstrated as part of the speech understanding program (Erman et al., 1980; Lowerre et al., 1980; Woods, 1980). The seventies also witnessed the development of a number of basic techniques such as blackboard models (Reddy et al., 1973; Nii, 1986), dynamic time warping [Itakura], network representation (Baker 1975), Hidden Markov models (Baker, 1975; Rabiner et al., 1986), beam search (Lowerre and Reddy 1980), and the forward-backward algorithm (Bahl et al.. 1986) . The early eighties witnessed a trend toward practical systems with very large vocabularies (Jelinek et al. 1985) but computational and accuracy limitations made it necessary to pause between words.

The recent speaker-independent speech recognition system called Sphinx best illustrates the current state of the art (Lee 1988). This system is capable of recognizing continuous speech without training the system for each speaker. The system operates in near real time using a 1000 word resource management vocabulary. The system achieves 94% word accuracy in speaker independent mode on a task with a grammar perplexity of 60. The system derives its high performance by careful modeling of speech knowledge, by using an automatic unsupervised learning algorithm, and by fully utilizing a large amount of training data. These improvements permit Sphinx to overcome the many limitations of speaker-dependent systems resulting in a high performance system.

The surprising thing to me was that a simple learning technique, such as learning transition probabilities in a fixed network structure, proved to be much more powerful than attempting to codify the knowledge of speech experts who can read spectrograms. The latter process of knowledge engineering proved to be too slow, too ad hoc, and too error-prone.

What did we learn from speech research? Results from speech research have led to a number of insights about how one can successfully use incomplete, inaccurate, and partial knowledge within a problem solving framework. Speech was one of the first task domains to use a number of new concepts such as Blackboard models, Beam search, and reasoning in the presence of uncertainty, which are now used widely within AI.

Vision

Research in image processing started in the fifties with optical character recognition. What we now call computer vision started with the seminal work of Larry Roberts (1965). Roberts was the first to work on 3-D vision using grey scale images of the blocks world. Roberts used a strictly bottom up approach to vision starting with edge detection, linear feature detection, 3-D model matching, and object recognition.

Minsky and McCarthy initiated the work on robotic vision beginning with the hand-eye projects at Stanford and M.I.T. Minsky proposed the concept of heterarchical architectures for vision in which various knowledge sources interacted with one another in an opportunistic way to solve the image interpretation problem. Successful demonstrations in vision from 1965 to 1975 were limited to problems in which 3-D phenomena such as shadows, highlights, and occlusions were not too troublesome. The work of Guzman (1968), Huffman (1971), Clowes (1971), Waltz (1975), and Kanade (1981) on line drawings, Fischler et. al.'s (1973) work on image matching, Ohlander's work (1978) on natural scenes, and Binford and Agin's work (1973) on generalized cylinders are representative of the best results of that period.

The limitations and problems of the earlier approaches led Marr (1979, 1982) to propose a general framework for vision involving the primal sketch and the 2-1/2-D sketch. This accelerated the "back-to-basics" movement started by Horn (1975) where mathematical models reflecting constraints of physics, optics, and geometry were used to derive 3-D shape properties from 2-D images of a scene.

Barrow and Tenenbaum (1978, 1981) proposed that photometric and geometric properties are confounded in the intensity values available at each pixel, and the central problem for vision was therefore how to simultaneously recover them. Two recent results highlight the dramatic progress that has been made over the last decade. Klinker, Shafer and Kanade (1988) have been able to remove highlights from a color image to produce an intrinsic body reflection image using Shafer's dichromatic reflection model (1985). Matthies, Szeliski and Kanade (1988) have developed an algorithm for estimating depth from a sequence of N images that is N times better than stereo depth estimation.

We are beginning to see the cumulative results of the past twenty five years being applied and improved within realistic task frameworks. One of the best examples of this research is "Navlab," a navigation laboratory used for obstacle detection and navigation research at Carnegie Mellon (Thorpe et al., 1987).

The Navlab is a self-contained laboratory vehicle for research in autonomous outdoor navigation. It is based on a commercial Chevy van modified to permit an onboard computer to steer and drive the van by electric and hydraulic servos. The sensors onboard Navlab include color stereo vision, laser ranging and sonar. Onboard computers include four Sun workstations and a Warp supercomputer.

The Navlab uses a hierarchical system structure. At the lowest level, the system can sense motion and follow steering commands. The next level performs tasks such as road following, terrain mapping, obstacle detection and path planning. The high level tasks include object recognition and landmark identification, map-based prediction, and long-range route selection. The system uses a blackboard model with extensions to deal with geometry and time for communication, knowledge fusion, and spatial reasoning. The system is currently being expanded to produce better predictions through the increased use of map and model information.

Expert Systems

In 1966, when I was at the Stanford AI labs, there used to be a young Ph.D. sitting in front of a graphics display working with Feigenbaum, Lederberg, and members from the Chemistry department attempting to discover molecular structures from mass spectral data. I used to wonder, what on earth are these people doing? Why not work on a real AI problem like chess or language or robotics? What does chemistry have to do with AI? That young Ph.D. was Bruce Buchanan and the system he was helping to develop was Dendral (Lindsay et al. 1980).

Now we know better. Dendral and its successor, Mycin (Shortliffe 1976), gave birth to the field of expert systems. Dendral is perhaps the single

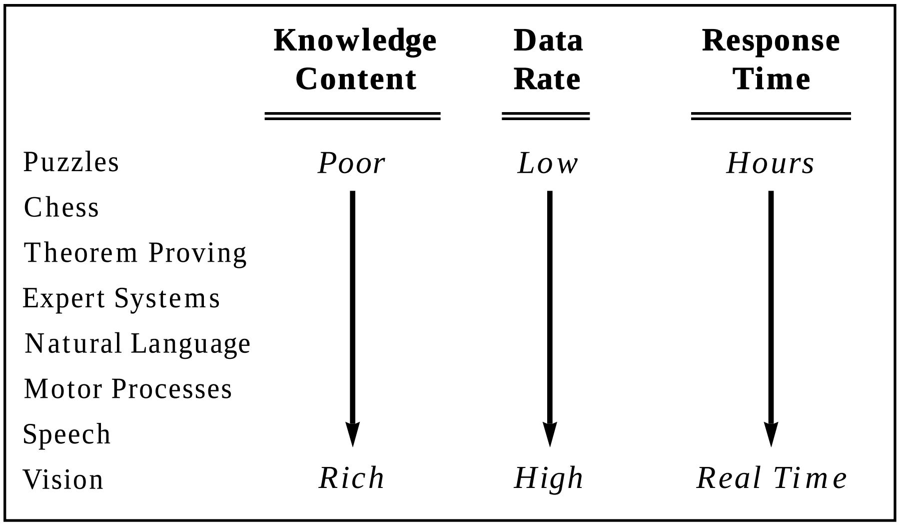

Figure 1: AI Problem Domains and their Attributes

longest-lived continuous set of experiments in the history of AI from 1965-83.

The current growth rate of the expert system field is too explosive for me to do justice in a short time. Let me refer you to the book by Feigenbaum, McCorduck, and Nii on the Rise of the Expert Company (1988) and the recent review paper by Buchanan and Smith on expert systems (1988).

I would like to provide a few examples to illustrate the widespread use of expert systems in many innovative applications:

• Schlumberger is using a Dipmeter Advisor to interpret down-hole data from oil wells to assist in prospecting for oil.

• Kodak is using an Injection Molding Advisor to diagnose faults and suggest repairs for a plastic injection molding mechanisms.

• Hewlett-Packard is using a Trouble Shooting Advisor to diagnose problems in photolithography steps in semiconductor manufacturing.

• NASA is using a system to monitor data during the liquid oxygen tanking process.

• NCR is using a system for preventive maintenance of computers.

• American Express uses a credit authorization system to authorize and screen credit requests.

• AT&T uses a system to act as a software user consultant to advise users of a large statistical package.

• Federal Express uses an Inventory Control Expert to decide whether or not to store spares.

• DEC and IBM routinely use systems for configuring customer orders into shipping orders.

• Xerox uses the PRIDE system to design paper handling systems.

• Hazeltine uses the OPGEN system for planning assembly instructions for PC boards.

• IBM uses a scheduling expert system to plan the factory operations in semiconductor manufacturing. The list goes on and on.

Here are some other interesting facts:

• IBM has deployed over 100 expert systems internally.

• The return on investment at IBM on a series of expert systems costing $2.5 million was estimated at $37.5 million.

• DEC employs over 700 people in its AI division.

• A Gartner Study, released in June 1988, shows that the number of deployed expert systems increased from 50 in 1987 to 1400 in 1988, and the number of expert systems under development increased from 2500 to 8500 over the same period.

I want to digress for a moment to pay tribute to the person most responsible for widespread use of expert systems. We all know that Feigenbaum sometimes tends to overstate the case. But sometimes it takes that kind of passion, conviction, and missionary zeal to make believers out of the masses. Ed, we thank you.

Time does not permit me to review advances in many other areas of AI such as natural language, robotics, learning, discovery, planning, and problem solving. Believe me! We know a lot more about each of these areas than we did thirty years ago. I refer you to papers on these and other subjects in the AAAI proceedings of the past ten years.

Foundations of AI

"...fine examples of bricklaying for a cathedral that is beginning to take shape,"

–Simon (1980).

In an era of accountability, it is appropriate for an outsider to ask what are the intellectual concerns driving the field. What are the fundamental principles, the self evident truths so to speak, that govern our field?

As an empirical science, AI follows the classical hypothesis-and-test research paradigm. The computer is the laboratory where AI experiments are conducted. Design, construction, testing, and measurement of computer programs are the process by which AI researchers validate the models and mechanisms of intelligent action.

It is not enough to say that AI attempts to study mechanisms of intelligent action. What are the attributes of an intelligent system? Minsky (1985) in his book *The Society of Mind* provides a long and formidable list of attributes of intelligence. Newell provides us with a shorter checklist (Newell 1988). An intelligent system must:

• operate in real-time;
• exploit vast amounts of knowledge;
• tolerate errorful, unexpected and possibly unknown input;
• use symbols and abstractions;
• communicate using natural language;
• learn from the environment; and
• exhibit adaptive goal-oriented behavior.

Where is AI today? What do we know now that we didn't know thirty years ago? At any given point, we can attempt to measure and evaluate the state of the art and accomplishments by assessing the capabilities of computers over the same range of tasks over which humans exhibit intelligence. These tasks range from puzzles and games at one extreme to perceptual and motor tasks at the other.

These tasks exhibit a wide variation in characteristics, such as knowledge content, data rates and amount of data, and expected response time (Figure 1). As you look at tasks ranging from puzzles, chess, and theorem proving to expert systems, languages, and perceptual and motor processes, tasks go from knowledge poor to knowledge rich, low data rates to high data rates, and slow response time to instantaneous response time. The diversity of requirements in tasks involving intelligent action places significant demands on architectures for AI with respect to memory capacity, bandwidth, access times, and processor speed.

Lessons from AI Experiments

The last thirty years of stable, sustained and systematic explorations have not only yielded performance systems that we saw earlier, but they have also provided us with a number of fundamental insights into the nature of intelligent action. Let us look at a few.

Bounded Rationality Implies Opportunistic Search. This is the fundamental result of Simon's research on decision making in organizations leading to the Nobel Prize. It says that "computational constraints on human thinking" leads people to be satisfied with "good enough" solutions. Feigenbaum (1988) calls it the founding paradigm of AI research. Prior to Simon's work on human decision making, it was assumed that, given all the relevant facts, the human being is capable of rational choice weighing all the facts. While studying the problem of resource allocation between maintenance and salaries in the administration of public parks in the city of Milwaukee in 1935, Simon observed that the decisions made had no relationship to the expected economic utility. That, and subsequent studies of Simon (1947, 1955) on "human decision making," led to the principle of "bounded rationality." When people have to make decisions under conditions which overload human thinking capabilities, they don't give up, saying the problem is NP-complete. They use strategies and tactics of "optimal-least-computation search" and not those of "optimal-shortest-path search." By "optimal-least-computation search" here, Simon means the study of approximate algorithms for search which can find the best possible solution given certain constraints on the computation, such as limited memory capacity, limited time, or limited bandwidth. Much of the current literature on search theory completely ignores this problem. This leads Simon to observe that many of the results in search theory may be of questionable value to the real world. I believe we should keep an open mind about what search results will be relevant in the future. Silicon-based intelligence, once achieved and given its differences in memory access time and bandwidth, may indeed have different computational constraints than human intelligence.

A Physical Symbol System Is Necessary and Sufficient for Intelligent Action. The second principle of AI is the physical symbol system hypothesis, i.e., a physical symbol system is necessary and sufficient for intelligent action. Newell and Simon articulated this hypothesis in their Turing Award paper (1976). I believe they were aware of this principle as early as 1960. What does it mean? First, we must define a physical symbol system. According to Newell and Simon, it is a system with the following properties: Physical symbols are symbols that are realizable by engineered components. A physical symbol system is a set of these entities. A symbolic structure is an expression whose components are symbols. Operations on the expression include creation, modification, reproduction, and destruction of symbols. Expressions can be interpreted as plans of action. Sounds familiar! Lisp is a physical symbol system! In fact, Lisp, Prolog, the Turing machine, Post Productions (for that matter any computer system) have all the mechanisms required of a physical symbol system.

Let us reexamine the hypothesis: a physical symbol system is necessary and sufficient for intelligent action. We may be willing to concede that any physical symbol system such as Prolog may be sufficient to achieve intelligent action given the past thirty years of experience in creating intelligent systems. But is it necessary? Does the human brain, of necessity, at some level have the mechanisms and properties of a physical symbol system? We in AI believe it is so but we cannot prove it. That is why it is a hypothesis. Not everyone believes it. Many philosophers and biologists have competing hypotheses. Nobel Prize winning biologist Edelman (1987) thinks that the brain is organized in a manner analogous to population dynamics. This view is clearly different from the information processing models of cognition and intelligent action that we in AI have come to love and cherish.

A Gartner Study, released in June 1988, shows that the number of deployed expert systems increased from 50 in 1987 to 1400 in 1988, and the number of expert systems under development increased from 2500 to 8500 over the same period.

The Magic Number 70,000 ± 20,000. The third principle of AI is that an expert knows 70,000 "chunks" of information give or take a binary order of magnitude. This is a surprising statement. Why is it 70,000? Why not a million? Why not one thousand? Several converging lines of evidence lead towards 70,000 as a rough measure of the size of an expert's knowledge base. For example, it appears that Chess Masters recognize about 50,000 chunks. Chase and Simon (1973) were able to quantify this number by constructing an experiment based on the ability of Master-level, Expert-level, and Novice-level players to recreate chess positions they have seen only for a few seconds. Based on these protocols, they were able to create an information processing model and use it to estimate the size of the knowledge base of a Masters-level player in chess. We have no comparable number for Grand Master-level play (rating of 2600 versus 2200 for a Master). It is safe to assume that 50,000 is at the low end of the size of this knowledge base as we go from Master to Grand Master to World Champion.

But is it true for experts in all domains? Indeed the evidence appears to point to the magic number 70,000 $\pm$ 20,000. Everyone of us is an expert in speech and language. We know that vocabularies of college graduates are about that order of magnitude. As we gain more and more experience in building expert systems, we find that a the number of productions begins to grow towards tens of thousands if the system is to perform more than a very narrow range of tasks in a given domain.

It is interesting that, in addition to George Miller's magic number 7 $\pm$ 2 which is an estimate of the size of short term memory (1956), we now have a hypothesis about the size of an expert's long term memory for a given domain of expertise. It has been observed that no human being reaches world class expert status without at least a decade of intense full-time study and practice in the domain (Hayes, 1985 and 1987). Even the most talented don't reach expert levels of performance without immense effort. Everyone of us is an expert in speech, vision, motion, and language. That only leaves enough time to be an expert in two or three other areas in one's life time. Indeed, most of us never get past one area of expertise.

Search Compensates for Lack of Knowledge. The fourth principle of AI is that search compensates for lack of knowledge. When faced with a puzzle we have never seen before, we don't give up: we engage in trial-and-error behavior, usually until a solution is found. Over the past thirty years of AI research, this statement has come to mean much more than the fact that we use search to solve problems. Let me illustrate the point by some examples.

During the sixties and seventies, it was believed that Masters-level performance in chess could not be achieved except by codifying and using knowledge of expert human chess players. We now know, from the example we saw earlier, that Hitech (which is able to examine over 20 million board positions in 3 minutes) is able to play at Senior Master-level, even though its knowledge is nowhere comparable to a chess master. The key lesson here is that there may be more than one way to achieve expert behavior in a domain such as chess.

This leads to the conjecture that this principle may be true for problem solving tasks such as puzzles and games but can't be true for perceptual tasks such as speech. Even here we are in for a surprise. Earlier we saw several examples in speech where search compensates for incomplete and inaccurate knowledge. The principle is also used in language processing where words can have many meanings. For example the verb 'take' can mean many things: take a book, take a shower, take a bus, take a deep breath, take a measurement, and so on. This inherent ambiguity that you carry a book and get into a bus is often clarified by the context. Usually, uncertainty can be resolved by exploring all the alternatives until the meaning is unambiguous. When in doubt, sprout! What this principle tells us about the role of search is that we need not give up hope when faced with a situation in which all the known knowledge is yet to be acquired and codified. Often it may be possible to find an acceptable solution by engaging in a "generate-and-test" process of exploration of the problem space.

Knowledge Compensates for Lack of Search. We now come to an important insight which was not clearly understood even as late as 1970, i.e., knowledge reduces uncertainty and helps us constrain the exponential growth leading to the solution of the many otherwise unsolvable problems. Knowledge is indeed power. In the extreme case "recognition," knowledge can eliminate the need for search altogether. This principle is essentially the converse of the previous principle; search compensates for lack of knowledge and knowledge compensates for the lack of search.

We have a number of experiments that illustrate the role of knowledge. You have probably seen the Rubik's cube. The first time around, it is not uncommon for most people to take half an hour or more to solve this puzzle. With practice however, the situation improves dramatically. There have been cases where the solution was completed in less than 20 seconds. This simple example clearly illustrates the role of knowledge in eliminating the trial-and-error search behavior. The interesting unsolved question in the case of Rubik's cube is: What is the knowledge, and how is it acquired and represented?

The speech task we saw earlier also provides some quantitative data about the importance of knowledge. The Sphinx system can be run with various knowledge sources turned off. Consider the situation where one removes the syntactic knowledge source. In this case, sentences of the form "Sleep roses dangerously young colorless" would be legal. Removing the syntactic knowledge source increases the error rate of Sphinx from 4% to 30%, i.e., on the average, one out of three words would be incorrect. Removing the probabilistic knowledge about the frequency of occurrence of the words increases the error rate from 4% to 6%.

The Knowledge-Search Tradeoff. Figure 2, created by Hans Berliner and Carl Ebeling (1986), graphically illustrates the knowledge-search trade off. A human Chess Master rarely examines more than 200 positions but is able to recognize over 50,000 chess patterns. On the other hand, Hitech, which has a Senior Master rating, explores over 20 million board positions but has less than 200 rules.

This graph also shows the characteristics of various AI systems. The early AI systems had little knowledge (10 to 20 rules) and engaged in modest search. Expert systems generally have more knowledge and do little search.

Can you really trade knowledge for search and search for knowledge? Within bounds, it seems. Newell (1988) observes that Soar, which is able to learn from experience, uses more knowledge and less search with each new round of solving problems in a task domain. For example, each time Soar solves a configuration task, it discovers and adds new rules to its knowledge base which it is then able to invoke in subsequent attempts while solving other configuration problems. Conversely, even experts seem to resort to search when faced with a previously unseen problem within the domain of their expertise—for example in scientific research. Simon (1988) observes that this produces a kind of paradox—that the most "creative" problem solving may have to use "the most primitive" techniques, i.e., the weak methods such as generate-and-test, hill climbing, and means-ends analysis.

This one chart appears to provide the entire space of architectures for intelligence. When the Fredkin Prize for the World Chess Championship is won, it will probably be by a system that has neither the abilities nor the constraints of a human expert; neither the knowledge nor the limitations of bounded rationality. There are many paths to Nirvana.

In this section, we have been looking at lessons from AI research: the insights and the principles that govern our enterprise, i.e., the task of creating systems of intelligent action; the insights that we did not have as recently as thirty years ago. If there is one thing I want you to remember about this talk, it is the five fundamental principles of AI we have just discussed: bounded rationality implies opportunistic search; physical symbol systems are necessary and sufficient for intelligent action; an expert knows 70,000 things give or take a binary order of magnitude; search compensates for lack of knowledge; and knowledge eliminates the need for search. Although these may sound obvious to you now, they didn't thirty years ago! But then, Newton's Laws of Motion and the Fundamental Theorem of Calculus also appear obvious, after the fact.

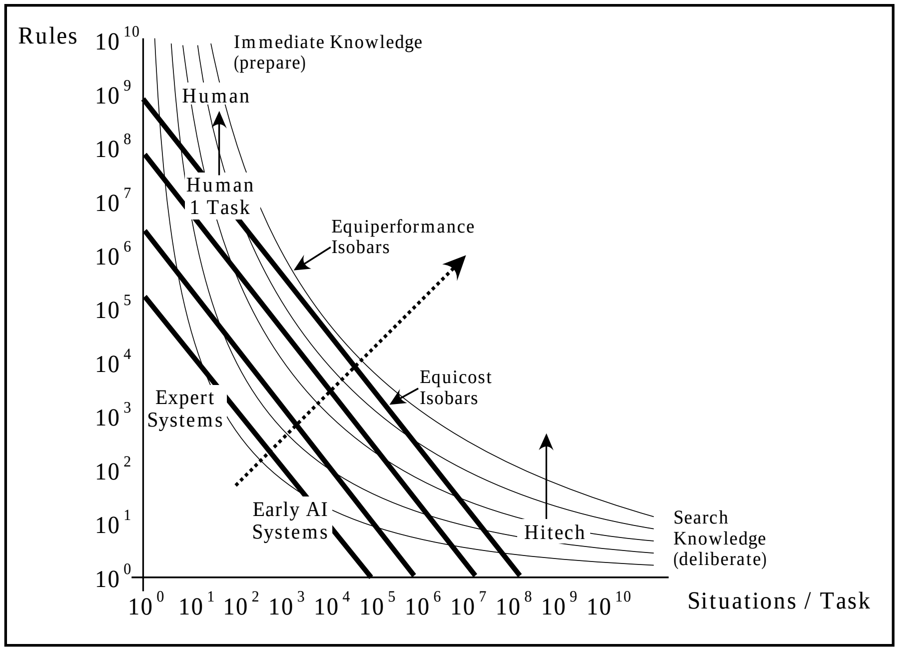

Figure 2: The Knowledge-Search Tradeoff

Lessons from Algorithm Analysis

Let us now look at lessons from other disciplines that have proved to be important to AI such as Algorithm Analysis and Complexity Theory.

Of necessity, almost every problem we look at in AI is NP-complete, i.e., exponential growth and combinatorial explosion are the law of the land. Initial results from the theorists were disappointing. There was a spate of results showing that almost every interesting problem is NP-complete. It does not do any good to the "traveling salesman" to know that the "traveling salesman problem" is NP-complete. He still has to travel to do his job and find a satisfactory travel route. NP-completeness is too weak a result to provide any guidance on the choice of algorithms for real world problems.

My favorite result from the Algorithm Analysis area is Knuth and Moore's analysis (1975) of the Alpha-Beta pruning algorithm. This is the only result I know that gives deeper, crisper and a more concise understanding of one of the major AI problems: Chess. The alpha-beta algorithm was originally articulated by John McCarthy and has been used in all chess playing programs since the early sixties. The key result here is that the use of alpha-beta reduces the exponential growth to the square root of the exponent. McCarthy's student, Mike Levin, conjectured this result as early as 1961. Slagle and Dixon (1969) proved the result for the optimal case in 1969. However, it was left up to Knuth to provide a detailed analysis of the power of alpha-beta. Let us examine the power of this result.

The selection of possible moves in chess is based on a mini-max game tree. Each ply increases the number of moves exponentially as a function of the branching factor. It has been estimated that one would have to explore over 10^{120} board positions in an exhaustive search of the chess game tree. You may know that 10^{120} is larger than the number of atoms in the universe.

On the average, in the mid-game, a player usually has the option of making any one of 35 possible moves. At a branching factor of 35, an 8-ply deep search usually requires examination of over 4 trillion board positions (Figure 3). This quickly increases to six million trillion positions for a 12-ply search, well beyond the capability of any current and projected computer system of the future. If one were to have a supercomputer with nanosecond cycle time, specialized to examine a node at each cycle, it would take

Search Type	Branching Factor	8-ply	Nodes Visited 10-ply	12-ply
Normal	35	4.1×10^{12}	5.0×10^{15}	6.2×10^{18}
Alpha-beta	6	1.8×10^{7}	6.4×10^{8}	2.3×10^{10}
Hash Table	5	5.0×10^{6}	1.3×10^{8}	3.1×10^{9}

Figure 3: Search in Chess

over 200 years to make a move for a 12-ply system. The alpha-beta pruning technique effectively reduces the branching factor by the square-root leading to a branching factor of about 6. 12-ply search still requires evaluation of over 23 billion board positions or only 23 seconds on our supercomputer. A hash table which keeps previously examined moves, further reduces the branching factor to five, resulting in 12-ply search requiring the evaluation of over three billion board positions or just 3 seconds on our supercomputer. Since a human player usually has 3 minutes to make a move, we will only need a system that can examine about 16 million board positions to achieve Grand-Master level play! The power of alphabeta is just awesome.

Systems such as Cray-Blitz that play chess approaching Master-level competence can examine about 25,000 board positions per second or about five million board positions in three minutes using a 25 mips equivalent scalar processor. Hitech is currently able to examine 175,000 board positions per second. Deep Thought, developed by Hsu and Anantharaman, is able to examine almost a million board positions per second. We are only a factor of 16 away from potential Grand Master-level play! With use of additional knowledge, it could even be sooner!

Coming back to Knuth's analysis, the result provides us with the ability to make concise statements about when and under what conditions we might expect to reach Grand Master-level performance. As we noted earlier, Levin, one of McCarthy's students discovered the same result in 1961, but it was left up to Knuth to provide the formal analysis and proof in 1975. There is a lesson in this to all our young scientists. If you happen to create a proof, an interesting algorithm, or an innovative data structure, don't leave it up to some future complexity theorist to re-invent it.

There have been other important results from algorithm analysis that are relevant to AI such as Karp's work on approximate algorithms, Tarjan's analysis of self-adjusting search trees, and Hopcroft and Natarajan's analysis of complexity of robot assembly tasks. While each of these provide a fundamental understanding, they have not yet been as important to AI as the analysis of alpha-beta.

Lessons from Applied Physics

Let us now look at some interesting insights from what I shall call "lessons from Applied Physics." Acoustics and optics have long been studied within Physics. AI speech researchers benefited from several decades of research in models of speech and acoustics at Bell Labs and other centers. Surprisingly, there has not been equivalent formal work in vision processes. I am happy to say that rather than leaving the formal studies to some later day physicist, AI scientists in the late 70's started the "back-to-basics" movement which now provides a firm theoretical foundation for vision. I will highlight several key lessons, leading to theoretically sound computational models of vision processes based on constraints of physics, optics, and geometry:

• Marr's general framework for vision,
• Barrow and Tenenbaum's representation of intrinsic images,
• Horn, Woodham, Ikeuchi and Shafer's work on inferring 3-D shape from photometric properties such as shading and color,
• Huffman, Clowes, Waltz, Kanade, and Kender's results on inferring 3-D shape from geometrical properties and constraints, and
• Ullman, Hildreth, and Waxman's results on inferring shape from motion such as optical flow.

Recently, Witkin and Tenenbaum (1983) questioned the desirability of preoccupation with recovering local surface attributes. They believe the resulting algorithms are frail and will not work well unless appropriate global constraints are simultaneously imposed on the image data. Bledsoe (1985) points out that intelligent agents of the future must be capable of reasoning at many levels. Most of the time, they may reason (like humans) at shallow qualitative levels, moving to deeper causal levels ("reasoning from basic principles of physics") only as needed. The lesson in the case of vision and applied physics is that the huge computational cost and too much precision could limit the usefulness of physics-based models even if they are superior to human vision.

Lessons from Connectionism

Recent blossoming connectionist research is a result of a better understanding of computational aspects of human intelligence. There are roughly a hundred billion neural cells in the human brain. The brain operates with a cycle time of 5 ms and computes a scalar product of binary input and weight vector.

One of the intriguing aspects of this computing element is that the fan-in and fan-out is of the order of 1,000. Most of the brain's volume is occupied by wires, even though the wires themselves are of submicron cross section. A hundred billion processing elements with 1,000 connections each represent a hundred trillion connections. As many as 1% of these wires are active in a 5 ms iteration involving a trillion computations every 5 ms. Thus, one might speculate that the brain appears to perform 200 trillion operations per second give or take a few orders of magnitude. Figure 4 shows a scanning electron micrograph of a neuronal circuit grown in tissue culture on a M68000 microprocessor by Trogadis and Stevens of the University of Toronto (1983). Note that axon and dendritic structures are much finer than the micron dimensions of the 68000.

The human brain possesses an interesting property. For tasks such as vision, language and motor control, a brain is more powerful than 1000 supercomputers. And yet, for simple tasks such as multiplication it is less powerful than a 4 bit microprocessor (Hinton, 1985). This leads us to speculate that silicon-based intelligence, when it is achieved, may have different attributes. We need not build airplanes with flapping wings.

When looking at a familiar photograph such as the Washington Monument, the brain seems to process and identify the scene in a few hundred milliseconds. This has led the connectionists to observe that whatever processing is occurring within the human brain, must be happening in less than 100 clock cycles. AI scientists such as Jerry Feldman (1982, 1985) and Geoff Hinton (1986) want to find answers to questions about what is the nature of representation and the nature of computation taking place within the human brain? For the first time, someone is asking questions about optimal-least-computation-search. I look forward to many exciting new results from connectionist research. Some researchers worry that connectionist architectures do not provide mechanisms for understanding what the system knows, how it reasons and what it believes. Let us first get a system that works, then I am confident that someone will figure out what it knows!

Lessons from Architectures

The Sphinx speech recognition system that you saw earlier achieves its real-time performance not just through the use of AI techniques such as representation, search, and learning but also through the use of efficient data structures and application specific hardware architectures. Roberto Bisiani (1988) was able to reduce the time for recognition from about 10 minutes to under 5 seconds. Any time you can make two orders of magnitude improvement, you have to sit up and take notice. How did Bisiani do this? From October 1987 through May 1988, he achieved the following :

• a speed-up of 1.5 by replacing sparse arrays with link lists

• a speed-up of 3.0 by redesigning the data structures to eliminate pointer chasing

• a speed-up of 2.0 by redesigning the beam search algorithm to use dynamic thresholds and inserting best state at the top of the list

• a speed-up of 2.5 using faster processors

• a speed-up of 1.6 using a multiple memory architecture

• a speed-up of 2.1 by using a multiprocessor architecture for parallel search execution.

Note that all these speed-ups are small, independent and multiplicative (as conjectured by Reddy and Newell, 1977), resulting in a whopping speed-up by a factor of 75, i.e., 7500%. Not bad for six months of research!

The main lesson here is that serious AI scientists of the future must also be competent in other fields of computer science, i.e., algorithm analysis, data structures and software, and computer architecture.

Lessons from Logic

Logic has long been used in AI as a theoretical description language for describing what the program knows and believes; used from outside for theorizing about the logical relations between the data structures before some thinking process has taken place and the data structures after the thinking process. Nils Nilsson's book (Nilsson 1971) on AI is the best example of this use of logic.

I am reminded of the time, twenty five years ago, when John McCarthy gave us an assignment in class to axiomatize the missionaries and cannibals problem. I went through the motions and I believe he gave me an "A" but I was never sure whether what I did was correct or not. There was no way to validate my axiomatization of the problem. For logic to become widely used as a theoretical description of the language of AI, we need tools and systems that help us think, formulate, and validate what we know in a concise form.

Recently, with logic programming tools, there has been an increasing use of logic as an internal representation language for AI programs for representing what it knows. This use may be good for some applications and bad for others. I don't believe a uniform representation language to solve all AI problems is either necessary or desirable at this juncture.

With advances in non-monotonic reasoning and other recent results, formal methods are enjoying a revival. From the beginning, John McCarthy has been a proponent and a major contributor to the formal theory of artificial intelligence. He is responsible for many key ideas in AI and Computer Science: the Lisp programming language, time sharing, common sense reasoning, alpha-beta pruning algorithm, circumscription in non-monotonic reasoning and so on. As my advisor at Stanford, he helped to launch my AI career as he did for many others. It gives me great pleasure to share with you the recent announcement that he has been selected to receive the prestigious $350,000 Kyoto Prize for 1988. McCarthy used to say that "To succeed, AI needs 1.7 Einsteins, 2 Maxwells, 5 Faradays and .3 Manhattan projects." Well, with Simon's Nobel prize and McCarthy's Kyoto prize, our field is making a solid beginning.

The Grand Challenges of AI

In an era of accountability, we cannot rest on our past accomplishments for very long. We must create a vision for the future which is exciting and challenging. Fortunately for us, any significant demonstration of intelligent systems is exciting. But we must go one step further. Whenever possible, we must identify and work on problems of relevance to the nation—bold national initiatives that capture the imagination of the public.

Let me illustrate what I mean by two examples from biology and physics: the Decoding of the Human Genome, and the Superconducting Super Collider Project. These grand challenges of science are expected to require several billion dollars of investment each. However, the expected benefits of these projects to the nation are also very high. AI is in a unique position to undertake and deliver on such nationally relevant initiatives.

What are the grand challenges of AI? Fortunately, we have several seemingly reasonable problems which are currently unsolvable, and which require major new insights and fundamental advances in various subfields of AI, and the criteria for success or failure can be clearly stated. The scope and size of these problems vary greatly—give or take an order of magnitude relative to, say, the Decoding of the Human Genome project. I would like to present a few of my favorite grand challenges here.

As the size of investment in AI rises above the noise level, we can no longer expect people to fund us on blind faith. We are entering an era of accountability.

World Champion Chess Machine. In the early eighties, with a grant from the Fredkin Foundation, we have established a $100,000 prize for the development of a computer program which would beat the reigning world champion chess player. We saw earlier that we are already at the senior masters level. Hans Berliner says that if you plot the progress in computer chess over the past twenty years, the ratings have been growing steadily at about 45 points per year. At that rate, we should have a chess champion computer by about 1998, almost forty years after Simon's predictions. We didn't quite do it in ten years. But in the cosmic scale of time, as Carl Sagan points out, forty or even a hundred years is but a fleeting moment. We have waited billions of years for nature to evolve natural intelligence! We can certainly wait a hundred or even a thousand years to realize a human-created intelligence.

Mathematical Discovery. The second Fredkin prize is for the discovery of a major mathematical result by a computer. The criteria for success in this case are not as crisp as with chess. Woody Bledsoe, Chairman of the mathematics prize committee, is working with a panel of eminent mathematicians to establish the criteria for awarding the prize. For a while, we thought a successor to Doug Lenat's AM program would claim this prize. But Lenat had other plans. He is after an even greater grand challenge—i.e. to create a system called CYC which will have an encyclopedic knowledge of the world.

The Translating Telephone. Japan recently initiated a seven year $120 million project as the first phase towards developing a phone system in which a Japanese speaker can converse with, say, an English speaker in real time. This requires solutions to a number of currently unsolved problems: a speech recognition system capable of recognizing a large (possibly unlimited) vocabulary and spontaneous, unrehearsed, continuous speech; a natural sounding speech synthesis preserving speaker characteristics; and a natural language translation system capable of dealing with ambiguity, non-grammaticality, and incomplete phrases.

Accident Avoiding Car. In the U.S., over 40,000 people die annually in automobile accidents. It appears that a new generation automobile equipped with an intelligent cruise control using sonar, laser, and vision sensors could eliminate 80% to 90% of the fatal accidents and cost less than 10% of the total cost of the automobile. Such a device would require research in vision, sensor fusion, obstacle detection and avoidance, low cost/high speed (over a billion operations per second) digital signal processor chips, and the underlying software and algorithm design.

Self-Organizing Systems. There has been a long and continuing interest in systems that learn and discover from examples, from observations, and from books. Currently, there is a lot of interest in neural networks that can learn from signals and symbols through an evolutionary process. Two long-term grand challenges for systems that acquire capability through development are: read a chapter in a college freshman text (say physics or accounting) and answer the questions at the end of the chapter; and learn to assemble an appliance (such as a food processor) from observing a person doing the same task. Both are extremely hard problems requiring advances in vision, language, problem solving techniques, and learning theory. Both are essential to the demonstration of a self-organizing system that acquires capability through (possibly unsupervised) development.

Self-Replicating Systems. There have been several theoretical studies in this area since the 1950's. The problem is of some practical interest in areas such as space manufacturing. Rather than uplifting a whole factory, is it possible to have a small set of machine tools which can produce, say, 95% of the parts needed for the factory using locally available raw materials and assemble it in situ? The solution to this problem involves many different disciplines including materials and energy technologies. Research problems in AI include: knowledge capture for reverse engineering and replication, design for manufacturability, and robotics.

Each of the above seemingly reasonable problems would require significant breakthroughs and fundamental advances in AI and all other subfields of Computer Science and Technology. Unlike other vague statements of accomplishments, success or failure in these cases can be clearly established and appreciated by non-experts. Each of these tasks requires long-term, stable funding at significant levels. Success is by no means guaranteed and each problem represents a high-risk high-payoff investment. However, even partial success can have spinoffs to industry and have a major impact on the competitiveness of our nation.

I have tried to present several grand challenges in AI which are worthy of long term research focus. Next, I would like to share a few thoughts on the social implications of our research.

Social Implications of AI

Like any other science, AI has the potential to do a lot of good and some bad. In this talk, I would like to accent the positive and implore all of you to devote some of your time to AI applications that might help the poor, the illiterate, and the disadvantaged of our nation and the world.

By the turn of the century, it appears possible that a low cost (e.g. costing less than $1000) super computer could be accessible to every man, woman and child in the world. Using such a system, AI researchers should be able to create a personalized, intelligent assistant which would use voice and vision for man-machine communication, tolerate error and ambiguity in human interaction with machines, provide education and entertainment on a personalized basis, provide expert advice on day-to-day problems, make vast amounts of knowledge available in active form, and make ordinary mortals perform superhuman tasks leading to new discoveries and inventions at an unheard of rate. Believe it or not, such a system would help the illiterate farmer in Ethiopia as much as the scientist in U.S.A. or Japan. Let me see if I can be moreconvincing!

The proposal to share the wealth between north and south advocated by the Brandt Commission and the Cancun Conference never got off the ground. Share the wealth! Who are we kidding! Shipping tons of wheat and corn to feed the hungry is not a solution either. Creating mechanisms for sharing of knowledge, know-how, and literacy might be the only answer. Knowledge has an important property. When you give it away, you don't lose it..*

The great Chinese philosopher Kuan-Tzu once said: "If you give a fish to a man, you will feed him for a day. If you give him a fishing rod, you will feed him for life." We can go one step further: If we can provide him with the knowledge and the know-how for making that fishing rod, we can feed the whole village.

Figure 4. Neuronal Circuit vs. VLSI Circuit

It is my belief that the most important wealth we can share with the disadvantaged is the wealth of knowledge. If we can provide a gift of knowledge to village communities that would make them expert at what they need to know to be self-sufficient, we would then witness a true revolution.

Sharing the knowledge and know-how in the form of information products is surely the only way to reduce this ever-widening gap between the have and have-nots. The current technological revolution provides a new hope and new understanding. The computer and communication technologies will make it possible for a rapid and inexpensive sharing of knowledge.

You can make a difference in achieving this compassionate world of the future. My friends, you can have meetings, publish papers, carry placards and believe you have done your duty about social responsibility and feel good about it. My request is to take the extra step. Get involved in national and international projects, and provide your expertise to help solve problems facing people who cannot help themselves.

Conclusion

Let me conclude by first saying that the field is more exciting than ever before. Our recent advances are significant and substantial. And the mythical AI winter may have turned into an AI spring. I see many flowers blooming. There are so many successes that I never cease to be amazed with wonderment at these new and creative uses of AI.

Second, success in AI depends on advances in all of computer science. We are not, and never have been an island unto ourselves. Finally, all parts of AI belong together. Success in AI requires advances in all of its disparate parts including chess, cognitive science, logic, and connectionism. Each of these experiments yield new insights that are crucial to the ultimate success of the whole enterprise.

What can you do? I believe the time has come for each of us to become a responsible spokesman for the entire field. This requires some additional effort on our part to be articulate and be convincing about the progress and prospects of AI. Finally, choose your favorite grand challenge relevant to the nation, and work on it.

Acknowledgments

I am grateful to Hans Berliner, Bruce Buchanan, Ed Feigenbaum, Takeo Kanade, John McCarthy, Allen Newell, Herbert Simon, and Marty Tenenbaum for the fellowship, and many discussions and comments that led to this paper.

Note

Feigenbaum believes knowledge is wealth and most nations and individuals may be unwilling to share it. This is especially true of secret chemical formulae, etc. This poses a dilemma. Clearly it is necessary to create a market for knowledge to be traded as any other commodity in the world markets. At the same time, the poor nations of the world cannot afford to pay for it. There appear to be several solutions. One is to create a Knowledge Bank which pays for and acquires the rights to knowledge for distribution to third world countries. Second is to create a Free Knowledge Foundation (analogous to Free Software Foundation). Over 90% of the knowledge needed by the poor is already available in the public domain. It is just that they are not even aware of the existence and availability of knowledge relevant to their specific problem. The FKF would have to set up mechanisms analogous to the agricultural extension blocks set up by USDA for creation and communication of knowledge and know-how through expert systems technology.

References

Anantharaman, T., Campbell, M., and Hsu, F. 1988. Singular Extensions: Adding Selectivity to Brute-Force Searching. *AAAI Spring Symposium*. Stanford, CA.

Bahl, L. R., Jelinek, F., Mercer, R. 1983. A Maximum Likelihood Approach to Continuous Speech Recognition. *IEEE Transactions on PAMI* 5(2), 179-190.

Baker, J. K. 1975. The Dragon System-An Overview. *IEEE Transactions on ASSP* 23(2), 24-29.

Barrow, H. R. and Tenenbaum, J. M. 1978. Recovering Intrinsic Scene Characteristics from Images. In *Computer Vision System*, eds. Hanson, A. R. and Riseman E. M. New York: Academic Press.

Barrow, H. R. and Tenenbaum, J. M. 1981. Computational Vision. *Proceedings of the IEEE*, 69: 572-595.

Berliner, H. J., and Ebeling C. 1986. The SUPREM Architecture. *Artificial Intelligence* 28(1).

Binford, T. and Agin, G. 1973. Computer Descriptions of Curved Objects. *Proceedings of the International Joint Conference on Artificial Intelligence.*

Bisiani, R., et. al. 1988. BEAM: A Beam Search Accelerator (Tech. Rep.). Computer Science Department, Carnegie Mellon University.

Bledsoe, W. 1986. I Had a Dream: AAAI Presidential Address, 19 August 1985. *AI Magazine* 7(1): 57-61.

Buchanan, B. and Smith, R. 1988. Fundamentals of Expert Systems. In *Handbook of Artificial Intelligence*, eds. Cohen and Feigenbaum. Forthcoming.

Chase, W. G. and Simon, H. A. 1973. The Mind's Eye in Chess. In *Visual Information Processing*, ed. Chase, W. G. New York: Academic Press.

Clowes, M. B. 1971. On Seeing Things. *Artificial Intelligence* 2: 79-116.

Ebeling, C. 1986. All the Right Moves: A VLSI Architecture for Chess. Ph.D. diss., Department of Computer Science, Carnegie Mellon University.

Edelman, G. M. 1987. *Neural Darwinism: The Theory of Neuronal Group Selection.* New York: Basic Books.

Erman, L. D., Hayes-Roth, F., Lesser, V. R., Reddy, D. R. 1980. The Hearsay-II Speech Understanding System: Integrating Knowledge to Resolve Uncertainty. *Computer Surveys* 12(2): 213-253.

Feigenbaum, E. A. 1988. What Hath Simon Wrought? In *Complex Information Processing: The Impact of Herbert A. Simon*, eds. Klahr, D. and Kotovsky, K. Hillsdale, NJ: Lawrence Erlbaum.

Feigenbaum, E. A., McCorduck, P., Nii, H. P. 1988. *The Rise of the Expert Company.* Times Books.

Feldman, J. A. 1985. Connectionists Models and Parallelism in High Level Vision. *Computer Vision, Graphics, and Image Processing* 31: 178-200.

Feldman, J. A. and Ballard, D. H. 1982. Connectionist Models and their Properties. *Cognitive Science* 6: 205-254.

Fischler, M. A. and Bolles, R. C. 1981. Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. *CACM* 24(6): 381-395.

Fischler, M. A. and Elschlager, R. A. 1973. The Representation and Matching of Pictorial Structures. *IEEE Transactions of Comp.* 22(1): 67-92.

Greenblatt, R. D., et. al. 1967. The Greenblatt Chess Program. *Proceedings of the Fall Joint Computer Conference.* ACM.

Guzman, A. 1968. Decomposition of a Visual Scene into Three-Dimensional Bodies. *Proceedings of the Fall Joint Computer Conference.*

Hayes, J. R. 1985. Three Problems in Teaching General Skills. In *Thinking and Learning*, eds. Segal, J., Chipman S., and Glaser, R. Hillsdale, NJ: Lawrence Erlbaum.

Hayes, J. R. 1987. Memory Organization and World-Class Performance. *Proceedings of the Twenty-First Carnegie Mellon Symposium on Cognition.* Psychology Department, Carnegie Mellon University.

Hebert M. and Kanade T. 1986. Outdoor Scene Analysis Using Range Data. *IEEE International Conference on Robotics and Automation* 3: 1426-1432.

Hildreth, E. C. 1984. Computations Underlying the Measurement of Visual Motion. *Artificial Intelligence* 23(3): 309-354.

Hinton, G. E. 1985. Personal communication.

Hinton, G. E., McClelland, J. L., and Rumelhart, D. E. (1986). Distributed Representations. In *Parallel Distributed Processing* eds. Rumelhart et al. Cambridge, MA: Bradford Books.

Hopcroft, J. E. 1987. Computer Science: The Emergence of a Discipline. *Communications of the ACM* 30(3): 198-202.

Horn, B. 1977. Obtaining Shape from Shading Information. In *The Psychology of Computer Vision*, ed.Winston, P. H. New York: McGraw-Hill.

Horn, B. 1977. Understanding Image Intensities. *Artificial Intelligence* 8(2): 201-231.

Hsu, F. 1986. Two Designs of Functional Units for VLSI Based Chess Machines (Tech. Rep.). Computer Science Department, Carnegie Mellon University.

Huffman, D.A. 1971. Impossible Objects as Nonsense Sentences. In *Machine Intelligence* 6, eds. Meltzer, B. and Michie, D. Edinburgh, Scotland: Edinburgh University Press.

Ikeuchi, K. and Horn, B. 1981. Numerical Shape from Shading and Occluding Boundaries. *Artificial Intelligence* 17: 141-184.

Itakura, F. 1975. Minimum Prediction Residual Principle Applied to Speech Recognition. *IEEE Transactions on ASSP* 23(2): 67-72.

Jelinek, F. 1976. Continuous Speech Recognition by Statistical Methods. *Proceedings of the IEEE* 64: 532-556.

Jelinek, F., et. al. 1985. A Real Time, Isolated Word, Speech Recognition System for Dictation Transcription. *Proceedings of the IEEE* ASSP.

Kanade, T. 1981. Recovery of the Three-Dimensional Shape of an Object from a Single View. *Artificial Intelligence* 17: 409-460.

Kanade, T. and Kender, J. R. 1983. Mapping Image Properties into Shape Constraints:

Skewed Symmetry, Affine-Transformable Patterns, and the Shape-from-Texture Paradigm. In *Human and Machine Vision*, eds. Beck, J., Hope, B., and Rosenfeld, A. New York: Academic.

Kanade, T., Thorpe C., and Whittaker, W. 1986. Autonomous Land Vehicle Project at CMU. *Proceedings of the 1986 ACM Computer Science Conference*, 71-80.

Kender, J. R. 1980. Shape from Texture. Ph.D. diss., Computer Science Department, Carnegie Mellon University.

Klinker, G., Shafer, S. A., and Kanade, T. 1988. Using a Color Reflection Model to Separate Highlights from Object Color. *International Journal of Computer Vision* 2(1): 7-32.

Knuth, D. E. and Moore, R. W. 1975. An Analysis of Alpha-Beta Pruning. *Artificial Intelligence* 6: 293-326.

Lamdan, Y., Schwartz, J. T., Wolfson, H. J. 1988. Object Recognition by Affine Invariant Matching. *Proceedings of Computer Vision and Pattern Recognition.*

Langley, P., Simon, H. A., Bradshaw, G. L., and Zytkow, J. M. 1987. *Scientific Discovery: Computational Explorations of the Creative Processes.* Cambridge, Mass: MIT Press.

Lee, K. F. 1988. Large Vocabulary Speaker Independent Continuous Speech Recognition: The SPHINX System. Ph.D. diss., Computer Science Department, Carnegie Mellon University.

Lee, K .F., Hon, H. W., and Reddy, R. 1988. An Overview of the SPHINX Speech Recognition System (Tech. Rep.). Computer Science Department, Carnegie Mellon University.

Lindsay, R. K., Buchanan, B. G., Feigenbaum, E. A., and Lederberg, J. 1980. *Applications of Artificial Intelligence for Organic Chemistry: The Dendral Project.* New York: McGraw Hill.

Lowerre, B. T. and Reddy, D. R. 1980. The Harpy Speech Understanding System. In *Trends in Speech Recognition,* ed. Lea, W.A. Englewood Cliffs, NJ: Prentice-Hall.

Mackworth, A. K. 1973. Interpreting Pictures of Polyhedral Scenes. *Artificial Intelligence* 4: 121-137.

Marr, D. 1979. Visual Information Processing: The Structure and Creation of Visual Representations. *Proceedings of the Sixth International Joint Conference on Artificial Intelligence.* Los Altos, CA: Morgan Kaufman.

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

Matthies, L., Szeliski, R., and Kanade, T. 1988. Kalman Filter-based Algorithms for Estimating Depth from Image Sequences (Tech. Rep.). Robotics Institute, Carnegie Mellon University.

Miller, G. A. 1956. The Magical Number Seven, Plus or Minus Two: Some Limits on Our Capacity for Processing Information. *Psychological Review* 63: 81-97.

Minsky, M. 1985. *The Society of Mind.* New York: Simon and Schuster.

Newell, A., Shaw, J., and Simon, H.A. 1958. Chess Playing Programs and the Problem of Complexity. *IBM Journal of Research and Development* 2: 320-335.

Newell, A. 1981. The Knowledge Level. Presidential Address, AAAI, 1980. *AI Magazine* 2(2): 1-20.

Newell, A. 1987. Unified Theories of Cognition. The William James Lectures. Psychology Department, Harvard University.

Newell, A. 1988. Putting It All Together. In *Complex Information Processing: The Impact of Herbert A. Simon,* eds. Klahr, D. and Kotovsky, K. Hillsdale, NJ: Lawrence Erlbaum.

Newell, A. and Simon, H. A. 1976. Computer Science as Empirical Inquiry: Symbols and Search. *Communications of the ACM* 19(3).

Nii, P. 1986. The Blackboard Model of Problem Solving and Blackboard Application Systems. *AI Magazine* 7(2&3): 38-53, 82-106.

Nilsson, N. J. 1971. *Problem Solving Methods in AI.* New York: McGraw Hill.

Ohlander, R., Price, K., and Reddy, D. R. 1978. Picture Segmentation Using a Recursive Region Splitting Method. *Computer Graphics Image Process* 8: 313-333.

Poggio, T., Torre, V. and Koch, C. 1985. Computional Vision and Regularization Theory. *Nature* 317(26): 314-319.

Rabiner, L. R. and Juang, B. H. 1986. An Introduction to Hidden Markov Models. *IEEE ASSP Magazine* 3(1): 4-16.

Reddy, D. R., and Newell A. 1977. Multiplicative Speedup of Systems. In *Perspectives on Computer Science,* ed. Jones, Anita K. New York: Academic Press.

Reddy, D. R., Erman, L. D., and Neely, R. B. 1973. A Model and a System for Machine Recognition of Speech. *IEEE Transactions on Audio and Electroacoustics* AU-21(3).

Roberts, L. G. 1965. Machine Perception of Three-Dimensional Solids. In *Optical and Electro-Optical Information Processing,* ed. Tippett, J. T. Cambridge, Mass.: MIT Press.

Rosenbloom, P. S., Laird, J. E., and Newell A. 1987. *Knowledge-level Learning in Soar.* Menlo Park, Calif: AAAI Press.

Rosenfeld, A. and Smith, R. C., 1981. Thresholding using Relaxation. *IEEE Trans. Pattern Anal. Mach. Intelligence* PAM1-3: 598-606.

Sagan, C. 1977. *The Dragons of Eden: Speculations on the Evolution of Human Intelligence.* New York: Random House.

Shafer, S. A. 1985. Using Color to Separate Reflection Components. *Color Research and Application* 10(4): 210-218.

Shortliffe, E. 1976. *MYCIN: Computer-Based Medical Consultations.* New York: American Elsevier.

Simon, H. A. 1947. *Administrative Behavior.* New York: Macmillan.

Simon, H. A. 1955. A Behavioral Model of Rational Choice. *Quarterly Journal of Economics* 69: 99-118.

Simon, H. A. 1988. The Scientist at Problem Solver. In *Complex Information Processing: The Impact of Herbert A. Simon,* eds. Klahr, D. and Kotovsky, K. Hillsdale, NJ: Lawrence Erlbaum.

Simon, H. A. 1988. Accomplishments of Artificial Intelligence. Unpublished notes.

Slagle, J. R., and Dixon, J. K. 1969. Experiments with Some Programs that Search Game Trees. *JACM* 16: 189-207.

Slate, D. J. and Atkin, L. R. 1977. Chess 4.5—The Northwestern University Chess Program. In *Chess Skill in Man and Machine,* ed. Frey, P. W. Berlin, W. Germany: Springer.

Tarjan, R. E. 1987. Algorithm Design. *Communications of the ACM* 30(3): 205-212.

Thompson, K. 1982. Computer Chess Strength. In *Advances in Computer Chess III.* Pergamon Press.

Thorpe, C., Hebert, M., Kanade, T., and Shafer, S. 1987. Vision and Navigation for Carnegie Mellon Navlab. In *Annual Review of Computer Science.* Palo Alto, Calif: Annual Reviews Inc.

Trogadis, J. and Stevens, J. K. 1983. Scanning Electron Micrograph. Playfair Neuroscience Catalog, Toronto Western Hospital. Photograph.

Ullman, S. 1981. Analysis of Visual Motion by Biological and Computer Systems. *IEEE Computer* 14(8): 57-69.

Waltz, D. 1975. Understanding Line Drawings of Scenes with Shadows. In *The Psychology of Computer Vision,* ed. Winston, P. H. New York: McGraw-Hill.

Waxman, A. M. 1984. An Image Flow Paradigm. *Proceedings of the IEEE Workshop on Computer Vision: Representation and Control.* New York: IEEE Press.

Witkin, A. P. and Tenenbaum, J. M. 1983. On the Role of Structure in Vision. In *Human and Machine Vision* eds. Beck, Hope, and Rosenfeld. San Diego, Calif.: Academic Press.

Woodham, R. J. 1981. Analyzing Images of Curved Surfaces. *Artificial Intelligence* 17: 117-140.

Woods, W. A. and Wolf, J. A. 1980. The HWIM Speech Understanding System. In *Trends in Speech Recognition,* ed. Lea, W.A. Englewood Cliffs, NJ: Prentice-Hall.