UNITED NATIONS EDUCATIONAL, SCIENTIFIC AND CULTURAL ORGANIZATION
INTERNATIONAL CENTRE FOR THEORETICAL PHYSICS
I.C.T.P., P.O. BOX 586, 34100 TRIESTE, ITALY, CABLE: CENTRATOM TRIESTE

H4.SMR/473-10

COLLEGE ON NEUROPHYSICS

"Neural correlates of behaviour, development, plasticity and memory"

1-19 October 1990

Computer Science:

Achievements and Opportunities

Drew McDermott and Tomaso Poggio

M.I.T. Cambridge, USA

Computer Science: Achievements and Opportunities

Report of the NSF Advisory Committee for Computer Research
John E. Hopcroft and Kenneth W. Kennedy, Chairs

This volume is dedicated to Kent K. Curtis,
whose leadership and encouragement led to the
intellectual development of computer research.

Published by the
Society for Industrial and Applied Mathematics
Philadelphia
1989

Artificial Intelligence

Drew McDermott Tomaso Poggio

Prior to the invention of the computer, information was something that was transmitted and stored, but it could be processed only when examined by a human being. The human was what made it information, by giving it *meaning*. However, the computer has made it possible to build autonomous, formal agents that process and condense information while respecting its meaning. For example, a payroll program performs a long series of uninspired actions that produce the *right answer*. The *objectification* of information made it conceivable that our original picture of the relationship between humans and information processing could be turned around: instead of humans making information processing possible, perhaps information processing makes humans possible. This hypothesis, that the operation of the mind is to be understood in terms of many small acts of computing going on in the brain, has captured the imagination of an entire generation of researchers since it was first proposed by Alan Turing. Not everyone agrees, however, that all mental activity can be explained in terms of computation, but it is obvious by now that large parts of what the brain does (say, in vision and natural-language processing) can be analyzed as symbolic or numerical computing. It seems pointless to draw boundaries around other parts of the mind where computing must not trespass.

Artificial intelligence (AI) can thus be defined as the science that studies mental faculties with computational models. How much of the mind can ultimately be accounted for this way is as yet unknown. It is important to note that *computational model* does not refer exclusively to a Turing machine or a programming language. Parallel computers, analog networks, and cellular automata represent acceptable models of certain computations. A new branch of psychology, called *cognitive science*, came into being in the late 1950s, inspired by the work of Newell and Simon in AI. The work of the cognitive scientists has helped replace behaviorism with less simplistic models of humans. For example, work on visual imagery, which had all but died out, is now alive and well. Philosophy has been influenced by AI as well.

One of the most debated questions in the philosophy of mind is the status of "functionalism," which explains mental states as analogs of the states of computers. Many philosophers believe that this kind of model explains much of psychology; the debate is whether it is compatible with the facts of consciousness and intentionality. This debate could not be held without the production of actual models of mind.

It is not easy to describe AI tidily. At this early stage, it is not clear whether AI is based on a few fundamental principles or is a loose affiliation of several different subfields, each concentrating on a different part of the mind or on different applications. This report discusses some of the most active areas of AI as examples of the kind of work being done.

Knowledge Representation and Reasoning

One candidate for a unifying principle in AI is the idea of *knowledge representation.* Although in some sense any computer program embodies knowledge (if only of what to do next), AI programs are unique in that they often make *inferences* from complex pieces of knowledge expressed in general notations. The knowledge implicit in a procedure is made manifest only by executing that procedure, whereas knowledge represented *declaratively* as a set of facts is explicit from the start and is accessible in more than one way. Indeed, such knowledge can be assembled, analyzed, and corrected before we have decided upon *any* particular way of using it. We can take as an analogy the laws of Newtonian mechanics, which can be expressed in abstract mathematical equations, and then later applied for many different purposes.

An important requirement for a useful representation language is that the meaning of its sentences should depend *compositionally* only on the meanings of the constituent structures of the sentence and not on the meanings of other sentences or on other surrounding context. As a way of ensuring this property, we are naturally led to the use of *logic-based notations*, in particular, various versions of the *first-order predicate calculus.* Starting in the late 1960s, AI researchers came to realize that computers could not achieve sophistication in various reasoning tasks unless they had formal encodings of large quantities of information about their problem domains that could be processed efficiently. This realization soon led to the elaboration of new problems:

1. What kind of facts can be expressed in formal languages?

2. What is the best method to embody knowledge in data structures?

3. What reasoning algorithms can be brought to bear?
4. How is new knowledge acquired?

Many of these problems now have at least partial solutions, which we discuss briefly.

Concerning the most basic question of what can be expressed in formal languages, experience has been encouraging. There are by now several detailed frameworks for representing general facts about time, physics, and the mental states of agents, as well as more specific facts about medicine, business, geology and other subjects (particularly in the territory of expert systems). It remains to be seen, however, whether these pieces can be put together into a whole that covers a large chunk of human knowledge in a unified way.

It is worthwhile studying formal languages in isolation, but for the computer to make use of them, the formal assertions must be both connected to information stored in data bases and themselves embodied in data structures that summarize relations between assertions. There are now several known ways of doing this, depending on the application. Many of them involve translating logical assertions into systems of nodes, with links between them that can be followed by computer programs to perform inferences efficiently. Such *semantic nets* are also very useful for storing information about hierarchical classifications of objects.

The study of reasoning algorithms has a somewhat different flavor. The initial focus of work in this area was on making deduction more efficient. The result was the discovery of elegant algorithms, based on Robinson's *resolution* principle, and employing the *unification* algorithm. Our understanding of how to carry out deduction has been revolutionized by discoveries like this. But for any given application there are many nondeductive components. Hence, there has been a blossoming of several different reasoning algorithms and an undermining of the notion of a general foundational principle for AI. In practice, each reasoning algorithm follows its own domain-dependent strategies and tends to demand somewhat different knowledge representation techniques. On the other hand, since it would be very useful to have a general theory of reasoning, this state of affairs has exerted a pull on AI theory to come up with broader reasoning algorithms.

One result of the study of reasoning in AI has been the invention of *nonmonotonic logics*, or pseudodeductive systems in which conclusions are revocable given more information. There are several paradigms for accomplishing this extension of traditional deduction. Most of the results are in what is known as a *circumscription* framework. Circumscribing a predicate P in a theory means adding an axiom schema or second-order axiom to the effect that "Any predicate P' that satisfies the same laws as P and is as strong as P is no stronger." This new axiom allows us to conclude not P in more circumstances than we can from laws for P alone. In semantic terms, it rules out all models of the original facts about P except the *minimal models*. Different versions of the circumscription axiom yield different kinds of minimality. Circumscription is nonmonotonic because adding more laws about P and recircumscribing can eliminate conclusions.

Many of the new reasoning patterns discovered by AI researchers have not been reduced to deduction, and it is not clear whether deduction can be extended to capture them; hence, they must be taken on their own terms. One example is work in *qualitative physics*. Quantitative simulation is in the domain of scientific computing, but human engineers can often predict or explain the behavior of a system without needing detailed numbers describing its components. Elegant methods now exist for predicting, as specifically as possible, the behavior of a system starting from a qualitative description of how its parts interact. One can think of these descriptions in a certain sense as *qualitative differential equations*, which specify the directions in which state variables influence each other but without specifying the magnitudes. The prediction algorithms note the directions in which quantities are changing and the interesting thresholds towards which they are heading. If just one quantity can reach its threshold next, that tells the program unambiguously what the next *qualitative state* of the system will be. Qualitative states can be defined technically as regions in state space in which all quantity-influence relations remain the same. In many cases, the behavior of the system is underspecified, and more than one qualitative state is a possible successor to the current one. The system pursues all possibilities. The ultimate result is a finite graph of qualitative states showing all possible behaviors of the system.

An investigation of the AI literature reveals a multiplicity of knowledge representation reasoning methods. It is not yet clear what unifying principles underlie them, if indeed any do.

Machine Learning

If knowledge, its representations, and reasoning algorithms to manipulate it are indeed central to AI, then the problem of machine learning (the automatic acquisition of new facts and reasoning methods) is crucial. Here, too, various powerful techniques have been discovered but no unifying principles. In fact, the absence of unifying principles may be counted as a major discovery of AI. The idea that all mental activity might be explainable in terms of learning, in an organism that starts as a *tabula rasa*, has been discredited by the discovery that certain apparently plausible unifying mechanisms are in fact meaningless. For hundreds of years, psychologists and philosophers have thought that the basic mechanism of learning was the transference of successful behavior in a situation to novel but similar situations. When we attempts to realize this idea on computers, we discover that there is no such thing as *intrinsic similarity*. Two situations are similar if some algorithm says they are, and any algorithm must neglect some differences; hence, for any two situations some algorithm will say they are similar, and we are left with the problem of devising algorithms for particular domains. It is now clear that an algorithm for, say, learning cognitive maps will have little to do with one for learning language. There is no choice but to study such problems on their own terms. As a result, in learning as elsewhere, we now know a little bit about a profusion of different learning tasks.

Some general principles have emerged, however. We can make a distinction between *internal* and *external* learning. The former is learning consequences of what we already know, as when we improve our skill at applying methods of symbolic integration. External learning is acquisition of genuinely new facts, as when we learn physical laws through observation. The former can profit from the powerful technique known as *explanation-based learning*. This method consists of extracting from a particular problem-solving session a general principle that will allow similar problems to be solved faster later by skipping over intermediate steps.

For external learning, guaranteed explanations are not obtainable. When learning a new law, a learning program must search through the space of possible versions of the law, trying experiments or making observations to rule incorrect versions out. When the language of the law is simple enough that all possible versions can be expressed as a lattice of more general and less general candidates, then we can keep track of exactly which versions are still viable by keeping track of the upper and lower bounds in the lattice within which the correct version lies. As more observations come in, they can be used to narrow the bounds. When applicable, this idea allows a "binary search" through the set of candidate laws.

Computer Vision

Not all subfields of AI are oriented directly around knowledge manipulation. Computer vision, the attempt to understand how information can be extracted from the light bouncing off objects, is a good example. As we will discuss later, vision algorithms must embody a lot of knowledge aoout optics, but they do not need to represent it declaratively. This distinction has not prevented the field of computer vision from developing some of the most satisfying results in AI. Before computational methods were brought to bear, vision theory had progressed little beyond optics. Electrodes could be stuck into cells in the visual cortex, but their signals were generally a mystery. Since approximately 1970, vision researchers have produced a plethora of detailed models of different aspects of vision. Many workers believe that the job of the visual system is to build a symbolic description of what it is looking at, and the role of computer science is to tell us what a symbolic description is. We may not know where to locate it in the brain, but we know we are looking for it.

Problems in vision are usually classified as part of *low-level* (or "early") *vision* or part of *high-level vision*. Early vision performs the first steps in processing images through the operation of a set of visual modules such as edge detection, motion, shape-from-contours, shape-from-texture, shape-from-shading, binocular stereo, surface reconstruction, and surface color. Its goal is to yield a map of the physical surfaces around the viewer. High-level vision can be identified with the "later" problems of *object recognition* and *shape representation*. Here, questions of knowledge representation will enter in an essential way.

The problem of vision begins with a large array of numbers recording an intensity value for each pixel (picture element) in the image. The precise value at each pixel depends not only on the color and texture of the three-dimensional (3-D) surface that is reflecting the light but also on the orientation and distance of the surface with respect to the viewer; on the intensity, color, and geometry of the illumination; on the shadows cast by other objects; and so on. The goal of early vision is to unscramble the information about the physical properties of the surfaces from the image data. In a sense early vision is the science of inverse optics. In classical optics (or computer graphics) the basic problem is to determine the two-dimensional (2-D) images of 3-D objects, whereas vision (whether biological or artificial) is confronted with the inverse problem of recovering 3-D surface from 2-D images. In color sensing, for instance, the goal of vision is to decode the measured lights in terms of the reflectance of the surfaces and the spectral power distribution of the illuminant.

The problems of inverse optics are very difficult to solve, despite the apparent ease and reliability with which our visual system gives meaningful descriptions of the world around us. The difficulty is at least twofold. First, the amount of information to be processed is staggering: a high-resolution television frame is equivalent to 1 million pixels, each containing eight bits of information about light intensity, making a total of 8×10^6 bits. The image captured by the human eye is even more densely sampled, since in the human eye there are in excess of 100 million photoreceptors. Real-time visual processing must be able to deal with many such frames *per second*. It is therefore not surprising that even the simplest operations on the flow of images (such as filtering) require billions of multiplications and additions per second. Second and more important, the images are highly ambiguous: despite the huge number of bits in a frame, it turns out that they do not contain *enough* information about the 3-D world. During the imaging step that projects 3-D surfaces into 2-D images, much information is lost. The inverse transformation (from the 2-D image to the 3-D object that produced it) is badly underdetermined.

The natural way to approach this problem is to exploit a priori knowledge about our 3-D world to remove the ambiguities of the inverse mapping. One of the major achievements of computer vision work in the last decade is the demonstration that *generic* natural constraints (the term generic is used here in the same sense as in the mathematical theory of dynamical systems) that is, general assumptions about the physical world that are correct in almost all situations are sufficient to solve the problems of early vision; and very specific, high-level, domain-dependent knowledge is not needed. Two main themes are therefore intertwined at the heart of the main achievement of early vision research: the identification and characterization of generic constraints for each problem and their use in an algorithm to solve the problem.

Some of the most powerful constraints reflect generic properties of 3-D surfaces. One of the best examples is the recovery of structure from motion. Perceptual studies show that a temporal sequence of images of an object in motion yields information about its 3-D structure. Consider for instance a rotating cylinder with a textured surface: its 3-D shape becomes immediately evident as soon as rotation begins. It has been proved that a 3-D shape can be computed from a small number of identified points across a small number of frames — if one assumes that the surface is *rigid*. Various theorems characterize almost completely the minimum number of points and frames that are required. *Continuity* of surfaces is another useful assumption: surfaces are typically regions of coherent aggregates of matter, do not consist of scattered points at different spatial locations, and are usually smooth. These constraints are very powerful for solving the correspondence problem in stereo and motion and for reconstructing surfaces from sparse depth points. Of course, constraints of this type are occasionally violated, and in these cases algorithms that strictly enforce them will suffer from "visual illusions."

It is natural to ask whether a general method exists for formalizing constraints in each specific case and translating them into algorithms. An interesting answer to this question has emerged in the last two years. We will describe it from a representative point of view, though by no means the only possible one. This unifying theoretical framework is based on the recognition that most early vision problems are mathematically ill posed problems. A problem is well posed when its solution exists, is unique, and depends continuously on the initial data. Ill posed problems fail to satisfy one or more of these criteria. In vision, edge detection (the detection and localization of sharp intensity changes) is ill posed when considered as a problem of numerical differentiation, because the result does not depend continuously on the data. Another example is the reconstruction of 3-D surfaces from sparse data points, which is ill posed for a different reason: the data alone, without further constraints, allow an infinite number of solutions, so that uniqueness is not guaranteed without further assumptions. The main idea in mathematics for "solving" ill posed problems (i.e., for restoring them to well posed problems) is to restrict the space of admissible solutions by introducing suitable a priori knowledge. In vision, this is identical to exploiting the natural constraints described earlier. Mathematicians have developed several formal techniques for achieving this goal that go under the name of *regularization theory*.

In standard regularization the solution is found as the function that minimizes a certain convex functional. This functional can be regarded as an "energy" or a "cost" that measures how close the solution is to the data and how well it respects the a priori knowledge about its properties. Consider the direct problem of finding y, given z and the mapping A:

$$Az = y$$

The inverse and usually ill-posed problem is to find z from y. Standard regularization suggests transforming the equation into a variational problem by writing a cost functional consisting of two terms. The first term measures the distance between the data and the desired solution z; the second term measures the cost associated with a functional of the solution $||Pz||$ that embeds the a priori information on z. In summary, the problem is reduced to finding z that minimizes the quantity

$$||Az - y||^2 + \lambda||Pz||$$

where λ, the regularization parameter, controls the compromise between the degree of regularization of the solution and its closeness to the data. Mathematical results characterize various properties of this method such as uniqueness and behavior of the solution. Solutions of this type have been obtained for several early vision problems: edge detection, optical flow, surface reconstruction, spatiotemporal approximation, color, shape from shading, and stereo.

Computer vision has always had a special two-way relationship with brain sciences: suggestions from visual physiology and psychophysics have played a role in many developments of computer vision. For instance, discoveries of neurons that seem to behave as edge detectors in the visual cortex of primates had a significant influence in the development of early computer vision programs. In turn, computational theories of vision are now influencing the psychophysics and the physiology of vision. It is very likely that this trend will grow more important for both fields. Mainly because of the theoretical advances of the last decade, it seems that early vision is now on its way to a systematic solution. Much less has been accomplished in high-level vision, however. At the level of object recognition and scene description, the vision system begins to blend with the rest of the mind, about which elegant unifying theories do not yet exist.

Concluding Remarks

In summary, AI is in a way the branch of computer science that is most nearly a classical empirical science. It studies the world at the computational level, in much the same way that chemistry studies the world at the chemical level. It is not a priori obvious that there *is* a chemical level; in principle, everything is just physics. But in many situations it is possible—and necessary—to ignore the details of elementary-particle interactions and focus on interactions in terms of *molecular bonds*, *valence*, *stoichiometry*, *reaction rates*, and so on. Similarly, in principle, the brain's functioning can be explained in terms of the behavior of its neurons and their membranes. Attention should be focused not only on these details, but also on the *information* that the neurons are transmitting and the *computations* they are doing. If this is approximately correct, then it may be just as necessary to focus on this higher level in understanding the brain as it is to focus on the chemical level in understanding chemical systems.

The nervous system is not the only place in the universe where nature has exploited computation. Another good candidate is the operation of the cells of organisms. Although in principle the behavior of DNA is describable chemically, the important thing about a particular DNA molecule is the message encoded in its nucleotides; this message is completely *arbitrary* from a chemical point of view. In many cases the only reasonable way to describe the operation of a cell is at a computational level in which genes are thought of as switching each other on and off, so that the set of active genes behaves like the state of a computing device, the next state and the outputs (proteins) being functions of the current inputs and the previous state. The study of such molecular computers—if they really do exist—might or might not be assimilated to AI. Indeed, it is not clear whether in the long run AI will be stable as a single discipline or split up along mental-module boundaries that are yet to be discovered. The point to absorb is that computation appears to exist in nature as well as in artifacts; its study is now emerging as a new empirical science.