T-KDE/5/1//06447D

Report on Workshop on High Performance Computing and Communications for Grand Challenge Applications: Computer Vision, Speech and Natural Language Processing, and Artificial Intelligence

B. W. Wah
T. S. Huang
A. K. Joshi
D. Moldovan
J. Aloimonos
R. K. Bajcsy
D. Ballard
D. DeGroot
K. DeJong
C. R. Dyer
S. E. Fahlman
R. Grishman
L. Hirschman
R. E. Korf
S. E. Levinson
D. P. Miranker
N. H. Morgan
S. Nirenburg
T. Poggio
E. M. Riseman
C. Stanfill
S. J. Stolfo
S. L. Tanimoto
C. Weems

Reprinted from
IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING
Vol. 5, No. 1, February 1993

Report on Workshop on High Performance Computing and Communications for Grand Challenge Applications: Computer Vision, Speech and Natural Language Processing, and Artificial Intelligence

B. W. Wah, *Fellow, IEEE,*, T. S. Huang, *Fellow, IEEE*, A. K. Joshi, *Fellow, IEEE,*, D. Moldovan, *Senior Member, IEEE,*, J. Aloimonos, *Member, IEEE*, R. K. Bajcsy, *Fellow, IEEE,* D. Ballard, D. DeGroot, *Member, IEEE*, K. DeJong, C. R. Dyer, *Senior Member, IEEE*, S. E. Fahlman, R. Grishman, L. Hirschman, *Member, IEEE*, R. E. Korf, S. E. Levinson, *Fellow, IEEE*, D. P. Miranker, N. H. Morgan, *Senior Member, IEEE*, S. Nirenburg, T. Poggio, *Member, IEEE*, E. M. Riseman, *Member, IEEE*, C. Stanfill, S. J. Stolfo, S. L. Tanimoto, *Senior Memzber, IEEE*, and C. Weems, *Member, IEEE*

Invited Paper

Abstract—**This paper reports the findings of the Workshop on High Performance Computing and Communications (HPCC) for Grand Challenge Applications: Computer Vision, Speech and Natural Language Processing (SNLP), and Artificial Intelligence (AI). The goals of the workshop are to identify applications, research problems, and designs of HPCC systems for supporting applications in these areas.**

In computer vision, we have identified the main scientific issues as machine learning, surface reconstruction, inverse optics and integration, model acquisition, and perception and action. Since vision algorithms operate in different levels of granularity, computers for supporting these algorithms need to be heterogeneous and modular. Advances in technology, new architectural concepts, and software design methods are essential for this area.

In SNLP, we have identified issues in statistical analysis in corpus-based speech and language understanding, search strategies for language analysis, auditory and vocal-tract modeling, integration of multiple levels of speech and language analyses, and connectionist systems. Similar to algorithms in computer vision, algorithms in SNLP require high computational power, ranging from general purpose supercomputing to special purpose VLSI systems. As processing has various requirements, a hybrid heterogeneous computer system is the most desirable.

In AI, important issues that need immediate attention include the development of efficient machine learning and heuristic search methods that can adapt to different architectural configurations, and the design and construction of scalable and verifiable knowledge bases, active memories, and artificial neural networks. A computer system for supporting AI applications is heterogeneous, requiring research in high-speed computer networks, mass storage and efficient retrieval methods, computer languages, and hardware and software design tools.

Research in these areas is inherently multidisciplinary and will require active participation of researchers in device and networking technologies, signal processing, computer architec-

Manuscript received September 21, 1992; revised November 24, 1992. This work was supported by the National Science Foundation under Grant IRI-9212592, which was jointly funded by the Knowledge Models and Cognitive Systems, Robotics and Machine Intelligence, and Interactive Systems Programs of the Division of Information, Robotics, and Intelligent Systems, the Microelectronic Systems Architecture Program of the Division of Microelectronic Information Processing Systems, and the Computer Systems Program of the Division of Computer and Computation Research. Any opinions, findings, conclusions, and recommendations expressed in this paper are those of the authors and do not necessarily reflect the views of the National Science Foundation.

B. W. Wah and T. S. Huang are with Coordinated Science Laboratory, University of Illinois, Urbana, IL 61801.

A. K. Joshi is with the Computer and Information Science Department, Moore School, University of Pennsylvania, Philadelphia, PA 19404.

D. Moldovan is with the Department of Electrical Engineering-Systems, University of Southern California, Los Angeles, CA 90089-0781.

J. Aloimonos is with the Center of Automation Research, University of Maryland, College Park, MD 20742-3411.

R. K. Bajcsy is with the Department of Computer and Information Science, University of Pennsylvania, Philadelphia, PA 19104.

D. Ballard is with the Department of Computer Science, University of Rochester, Rochester, NY 14627.

D. DeGroot is with Texas Instruments, Plano, TX 75086.

K. DeJong is with the Computer Science Department, George Mason University, Fairfax, VA 22030.

C. R. Dyer is with the Department of Computer Science, University of Wisconsin, Madison, WI 53706.

S. D. Fahlman is with School of Computer Science, Carnegie Mellon University, Pittsburgh, PA 15213.

R. Grishman is with the Computer Science Department, New York University, New York, NY 10003.

L. Hirschman is with the Spoken Language Group, Massachusetts Institute of Technology, Cambridge, MA 02139.

R. E. Korf is with the Computer Science Department, University of California, Los Angeles, CA 90024-1596.

S. E. Levinson is with the Speech Research Department, AT&T Bell Laboratories, Murray Hill, NJ 07974.

D. P. Miranker is with the Department of Computer Sciences, University of Texas, Austin, TX 78712-1188.

N. H. Morgan is with the International Computer Science Institute, Berkeley, CA 94704.

S. Nirenburg is with the Center for Machine Translation, Carnegie Mellon University, Pittsburgh, PA 15213-3890.

T. Poggio is with Department of Brain and Cognitive Sciences, Massachusetts Institute of Technology, Cambridge, MA 02319.

E. M. Riseman and C. Weems are with the Computer and Information Science Department, University of Massachusetts, Amherst, MA 01003.

C. Stanfill is with Advanced Information Systems, Thinking Machines Corporation, Cambridge, MA 02142-1214.

S. J. Stolfo is with the Computer Science Department, Columbia University, New York, NY 10027.

S. L. Tanimoto is with the Department of Computer Science, University of Washington, Seattle, WA 98195.

IEEE Log Number 9206447.

ture, software engineering, and knowledge engineering. Besides extending current frontiers in research, an important aspect to be emphasized is the integration of existing components and results into working systems.

I. Introduction

A. Origin of the Workshop

THE idea of this workshop stemmed from the "Blue Book"[1] which identifies computing and communication technologies as essentials "to satisfy national needs from a variety of perspectives, including technology, science applications, human resources, and technology transition." Computer vision and SNLP are two of these grand challenge applications which "often cut across various agencies and missions" and "are related to solving very intensive large scale computing problems."

The workshop was held on February 21 and 22, 1992 in Arlington, VA, with 23 experts from academia and industry attending and 12 program directors from NSF serving as observers. Participants in the workshop were divided into three areas, with a vice-chair identified for each. Before the workshop, each vice-chair had solicited position statements from members of his area and coordinated discussions of issues through electronic mail. Based on comments received during the workshop, the vice-chair, in consultation with members of the area, prepared a summary report. After the workshop, the summary report was refined and was posted in early March on many electronic bulletin boards with areas related to the focus of this workshop.

This paper was prepared on the basis of the preliminary report distributed in March and further discussions among the participants through electronic mail. In preparing this report, each vice-chair first assembled ideas and material from the workshop participants in his area and prepared an initial draft of a section. Based on the three sections supplied by the vice-chairs, the workshop chair integrated the material, refining the discussion so that there is coherent flow and balance between the sections.

This paper is a collection of ideas expressed by the participants. It does not necessarily represent a consensus among all the participants. Further, ideas expressed in this paper do not reflect the official position of the sponsoring agency.

B. The Workshop Charter

The goal of the workshop was to identify near-term (within five years) and long-term (beyond five years) problems and potential approaches/research directions in grand challenge applications in computer vision, SNLP, and AI Attendees focused on answering the following questions.

1) What grand challenge applications in computer vision, SNLP, and AI can benefit by the availability of HPCC systems?
2) What research problems need to be solved in these grand challenge application areas?
3) How should HPCC systems be designed so that they can better support solutions in these areas?

We have chosen to cover the three areas in this workshop in the order of computer vision, SNLP, and AI. All three areas are important grand challenge application areas. Moreover, they are closely related, since vision, speech, and natural language are three primary modes of perception and communication in humans, and knowledge acquisition and intelligent reasoning is needed for augmenting deficiencies and missing information. We do not attempt to cover all the aspects in these three areas because they are too broad to be discussed in a single workshop, and we will not do justice even if we try to focus on a small portion. Rather, we concentrate on issues related to how high performance computing can help provide new solutions and insights into solving problems in these areas. We also examine new applications where AI, vision, speech, and natural language can be integrated.

C. Organization of This Paper

This paper is divided into three major sections, covering the areas of computer vision, SNLP, and AI. Each section is further subdivided into four subsections: grand challenge applications, fundamental science and enabling technologies, implications for system architectures, and infrastructure support. The subsection on grand challenge applications illustrates applications in the area and relates them to applications in the other areas covered in this paper. The subsection on fundamental science and enabling technologies discusses fundamental research problems that need to be solved. We discuss issues that are application specific, as well as common issues; examples of the latter are machine learning and heuristic search. The subsection on implications for system architectures highlights requirements of each application area and discusses how computers under the HPCC Initiative should be designed to better serve the application. The last subsection on infrastructure support presents what advanced infrastructure tools (hardware and software) are needed to support research in each area.

II. Computer Vision

Computer vision has two goals. From the engineering viewpoint, the goal is to build autonomous systems which can perform some tasks that the human visual system can, and even go beyond the capabilities of the human visual system in multimodality, speed, and reliability. From the scientific viewpoint, the goal is to develop computational theories of vision, and by so doing, gain insights into human visual perception.

Computer vision is related to other grand challenge areas because 1) many applications, such as video compression and human-machine interface, involve both vision and speech; and 2) AI techniques, such as knowledge-based reasoning, are needed in vision systems which must operate in real-world domains.

[1] "Grand Challenges: High Performance Computing and Communications," The FY 1992 U.S. Research and Development Program, Committee on Physical, Mathematical, and Engineering Sciences, Federal Coordinating Council for Science, Engineering, and Technology, Office of Science and Technology Policy, Washington, DC, 1992.

A. Grand Challenge Applications

Grand challenge applications in computer vision fall in two classes. First, there are many important applications in autonomous vision systems, most of which involve the interaction of the vision system with the environment and humans. Examples of these applications include flexible manufacturing, intelligent vehicle highway systems, environmental monitoring, visual prosthetics and rehabilitation robotics, multimedia and model-based compression, and education. Second, there are many basic scientific application problems that can be studied using computer vision techniques as invaluable tools. A prominent example is the visual understanding of turbulence in fluid flow. In the following, we briefly describe some of these applications.

1) Flexible Manufacturing: The manufacturing base is an important element to keep the United States competitive at the turn of the 21st century. The next 20 years of manufacturing will be characterized by ever faster changing products, with success determined by variability, customized usage, and low cost. The computer technologies that are now available indicate the feasibility of achieving integrated and yet flexible manufacturing.

Computer vision is a critical enabling technology that allows cooperating robots to be mobile and dexterous. This is important for effective flexible manufacturing of small batches of customized parts, which may require functionalities and capabilities that are distributed across multiple robot agents. This poses challenging issues in multi-agent cooperation, many of which will require real-time perception.

Distributed active vision for specialized manufacturing tasks is another important element in flexible manufacturing. This allows software configured logical connections between factory-based sensors and a diverse collection of (remote) supercomputers that perform specialized vision tasks as required by the particular manufacturing process. This would certainly take advantage of high bandwidth networks as well as computer vision technologies.

2) Intelligent Vehicle Highway System (IVHS): Computer vision is a key technology for developing better highways to improve mobility, safety, capacity, and efficiency of our surface transportation system. The strategic plan developed by the Intelligent Vehicle Highway Society of America calls for advanced technologies that provide traffic management as well as driver-assist and autonomous capabilities on vehicles. Autonomous vision-based systems for high-speed vehicle control have already been tested in laboratories. These systems can provide driver warning to help detect and avoid collisions for unanticipated emergencies in poor driving conditions. An advanced concept considered is the design of high-speed convoys in special automated lanes that function safely and efficiently in various weather conditions.

3) Environmental Monitoring: An important part of the grand challenge problem in computer vision is the monitoring of earth resources and points of interest via satellite and aerial reconnaissance. Changing resources and their effects on earth, such as the loss of rain forests leading to global warming effects, need to be monitored closely. Others, such as the monitoring of points of military interest, are important for our national defense. Integrated space-based sensing systems would perform complex resource allocation, balancing the capabilities of distributed monitors such as satellites with the needs of individual problems. The advent of computing resources, with computing power of at least 10 000 times better than existing systems, promises to make important parts of this essential task feasible.

4) Visual Prosthetics and Rehabilitation Robotics: Intelligent machines and "assistive robots" capable of perception can tremendously benefit the blind and severely disabled individuals. Although the need remains clear, their widespread use has been overshadowed by cost and complexity. Ideas considered include: 1) a vision system that can monitor human whereabouts; 2) a vision system that can keep track of the users' environment (objects, freeways, and tasks); 3) a computer vision initiated assistant that can be invoked "intelligently" without user intervention, and 4) a vision system that can guide a blind user to objects and landmarks of interest in the environment.

An "assistive workstation" capable of accepting requests to interpret a scene will be invaluable for patients beyond rehabilitation. Such a system can be designed around a computer vision subsystem (with an intelligent user interface to the vision system) and a high-speed network connecting computational resources across the system. Effective load balancing would allow this system to be economically viable.

5) Multimedia and Model-Based Compression: Computer vision can have a profound impact in developing new image compression methods, which are critical for developing multimedia and image dissemination techniques. The multimedia industry within 10 years is likely to integrate two worlds that today are quite distinct: the world of communication and the world of computers. It is expected to have significant impact in many areas, including education, visualization for engineering, and entertainment.

The key technology in multimedia systems is signal compression and especially image compression. From the technical point of view, the main challenge is to exploit computer vision technologies of model-based recognition to achieve *hypercompression*, which is compression at very high rates obtained by encoding only the parameters of models of objects identified in the image (such as cultural objects). Well developed, "classical" image compression technologies, especially in the area of model-based vision, can be significantly enhanced by the focused use of computer vision algorithms.

6) Education: Computer vision technology will have an important role in education. Image analysis techniques are part of several key tools to improve the widespread use of information technology in the delivery of education. They have great potential to help author–animators of visual materials integrate live action and animation. However, the full potential of vision technologies in education will only be realized if major efforts are made to transfer the technology of computer vision to the domain of digital-movie authoring tools. This requires the development of many advanced technologies employing high performance computing that are beyond existing image oriented packages for desktop publishing.

A more ambitious direction is the development of intelligent tutoring systems that can process speech and image data from a student, reason about the student's understanding of material in the subject domain, use stimulating and sensitive pedagogic strategies in the manner of a master teacher, and utilize computer vision so that the system could visually perceive the human with whom the system is interacting.

Initially, one or more pilot studios should be assembled and connected to the National Research and Educational Network (NREN). Eventually, an array of tools should be made available so that anyone on NREN can use digital television authoring tools and large well-organized distributed libraries of source material. Very large funds have already gone into computer vision technology; transferring some of this technology to the educational arena is an important way to capitalize on this existing investment.

7) Turbulent Flow Analysis: Understanding turbulence is important in applications such as weather forecasting, designing aircrafts and ships, and noise control. Turbulence research creates a number of challenging problems for computer vision and image processing. Researchers in this area, however, are overwhelmed by the huge amount of three-dimensional (3-D) time varying vectors of data for modeling turbulent flow. Techniques in computer vision and AI can be of tremendous help in understanding and in reducing the complexity of algorithms in fluid flow. For instance, computer vision can aid in tracking coherent patterns in fluid flow, and unsupervised clustering and learning methods can be applied to classify coherent patterns in turbulent flows. In order for these to be possible, extensions must be made in existing image and signal processing techniques that were designed for scalar data, and in conventional computer vision methods that work mainly on surface data.

B. Fundamental Science and Enabling Technologies

Some of the main scientific issues underlying the applications are: i) machine learning, ii) surface reconstruction, inverse optics, and integration, iii) model acquisition, and iv) perception and action.

1) Learning: Learning has long been a central problem in understanding intelligence and developing intelligent machines. For many years the "Turing test" has represented an operational definition of intelligence against which most workers in AI have implicitly measured their own goals and achievements: if a computer behaves in a way indistinguishable from a human person, then it can be called intelligent, even if its "intelligence" has been painstakingly programmed by a very skilled human being.

Consistent with Turing's implicit definition, intelligence was perceived thirty years ago as mainly reasoning, problem solving, proving theorems, or playing chess. In contrast, we realize today how "intelligent" lower animals are and how complex are the problems that our senses routinely solve. We also realize how intractable the problem of developing hardware and software is and how much of it would be needed in order to replicate even a small part of the simplest aspects of intelligence. From this perspective, a somewhat different definition of intelligence seems needed to better capture today's view of the underlying problems. This new "Turing test" should emphasize the development of perceptual, motor, and language competence. In particular, many would agree that a system that can acquire visual, motor, and language skills should be called intelligent. This revised test introduces, in an explicit way, the problem of learning as the core of any attempt at understanding intelligence.

Within computer vision itself, the next frontier is represented by the problem of learning. Bringing together the two separate disciplines of machine perception and machine learning is an exciting and promising goal. It allows more robust practical systems and flexible devices for visual inspection, and may also have deep implications for understanding biological vision. There are different levels of learning in vision: from the problem of adapting parameters in preprogrammed vision algorithms, to the more challenging problem of learning to synthesize specific vision algorithms (such as stereo) from examples of desired performance (depth maps). The ultimate challenge is to develop a machine that learns to see, somewhat like a child learning visual skills.

Learning in vision is especially challenging from the computational point of view. Algorithms for learning can be computationally expensive, since the "learning" or estimation stage is usually a nonconvex optimization problem over many hundreds or thousands of parameters. These problems are grand challenge problems themselves: they may require computational power that are 100 times faster than real-time.

2) Integration of Vision Modules: A vision system should have the ability to create representations of its visible environment if it is designed to autonomously interact with its environment and perform various tasks related to navigation, manipulation and recognition. Such representations are needed for several properties of the environment like shape and motion, as well as color and other material properties. Extraction of these geometric and physical properties of the environment on the basis of image information is an inverse problem that requires modeling of the image formation process as well as the development of geometric and physical constraints relating 3-D information to image properties or cues.

Research during the decade of the 1980's demonstrated that these inverse problems are undetermined or ill-posed, i.e., their solution does not exist, or is not unique and does not depend continuously on the data. For example, recovering 3-D shape using cues, such as shading, texture or contour, are ill-posed problems. The same holds true for recovering properties such as image motion and discontinuities.

It is becoming clear from work of the past few years that if a vision system integrates information from several cues, such as motion, stereo, texture, and contour, then several inverse problems become well posed and stable, simply because more information is taken into account. A grand challenge for computer vision research in the 1990's is, therefore, the development of a sound framework for integrating vision modules, with the goal of creating robust environmental descriptions. Although several alternatives could be outlined, it is not yet obvious how one should proceed toward the solution of this problem. However, it is known that a solution requires a vast amount of computational power so that multiple processes can cooperatively utilize various image cues and exchange information in high-speed links.

3) Model Acquisition: In each of the applications described previously, an important component of a complete vision system is the use of models describing surfaces and objects of interest for model-based scene analysis and for robust and efficient scene interpretation. Examples include models of manufacturing parts, roads, and vehicles in the IVHS domain, and buildings, rivers, and clouds in environmental monitoring. Consequently, an important open research problem is the automation of the model acquisition process. Successful solutions to this problem will foster rapid prototyping of vision systems and benefit other areas such as computer graphics, robotics, and geometric-model based problems.

Constructing appropriate object models is, at best, a very time-consuming task. Currently, programmers handcraft 3-D models of objects, surfaces and features that can subsequently be used in tasks such as object recognition, navigation, and surveillance. To successfully scale up prototype vision systems so that they can handle applications with a large number of complex objects, will require better methods for automatically acquiring these models. This encompasses several critical problems.

a) How can more general classes of real objects be modeled?
b) How can procedures be defined for automatically acquiring shape models?
c) How can robust object and scene recognition strategies be automatically acquired for utilizing context and the many possible feature cues that may be available in a scene?

The first problem focuses on the generality of geometric modeling. We must extend current static shape modeling techniques to more explicitly describe properties that are detectable by a given sensor. This modeling-for-detectability means taking into account the imaging process in order to identify physically meaningful and measurable features. It also means that dynamic modeling of nonrigid surfaces and objects must be based on physically accurate constraints (such as mass, friction, forces, and torques). Moreover, it needs models based on semantic or functional descriptions of the use of objects.

The second problem posed above is related to the automation of the model acquisition process. Current model construction techniques in computer vision (and computer graphics) require painstaking manual effort to add new models. New algorithms are needed for supporting several issues related to this problem, which will necessarily overlap with other basic research problems in learning and in combining perception and action. Examples include a) incremental shape learning from multiple views, multiple resolutions, and multiple focuses of attention; b) shape learning by active surface exploration; c) compiling characteristic views based on physical constraints on geometric and kinematic shape, lighting and imaging processes; and d) constructing large-model databases using intermodel and intramodel characteristics to produce efficient storage and access methods.

The third problem posed above involves the automatic construction of robust recognition strategies. This is necessary to acquire and optimize a range of information that is difficult for a human to catalogue and integrate. Although the number and types of cues available for object and scene recognition is large, only a small amount is present in a particular image. Thus there are many paths for semantic inference, including data-directed construction of 3-D information (such as points, lines, and surfaces), and model-based recognition using distinctive features (such as color, texture, and shape) and scene context (such as spatial and functional relationships). The strategies developed should be capable of using only the available information.

In short, model acquisition in its most general form is a computationally demanding problem that is at the heart of learning about the physical environment in which we live. Sensors provide measurements about that environment, and the challenge is to incrementally integrate a vast amount of information over time to compactly and explicitly represent what we know, how it can appear, and how it can be applied for object and scene interpretation.

4) Perception and Action: Perception is more than just vision and is highly enhanced by directed "actions." It should be viewed as a composition of processes that use sensory information from several different modalities and/or different cues. This view implies the following research agenda: a) modeling of individual sensory capabilities and limitations; b) modeling of individual cue capabilities and limitations; c) determining a common representation space for integrating the information obtained from different modalities; d) designing integration rules and methods for combining information; e) developing control strategies for data acquisition; and f) studying the amount perceptual information needed for a given task.

The last question points us to the issue of "task" modeling. We believe that there is no such thing as perception without purpose. Hence, the task drives what sensory information and how much of it will be processed for a given task or subtask.

The "activity" is apparent in the agenda described previously. If the task requires accuracy or view that the first measurement did not provide, then one must "act," i.e., take more and different measurements. Many problems such as ambiguities that arise from spatial alignments can be disambiguated by simple actions of moving, grasping, or pushing. The performance of the task can then be used to evaluate the performance of the perceptual system.

C. Implications for System Architectures

The computational requirement of vision algorithms is high and is predicted to be at least tens of trillions of operations per second. Consider an autonomous vehicle moving through an unstructured environment (e.g., a mobile robot operating as a hazardous waste cleanup machine). Assuming color stereo sensors operating at 30 frames per second, the input data rate will be roughly 50 Mbytes/s using standard resolution images (512 pixels by 512 pixels), or 200 Mbytes/s at higher resolution (1024 pixels by 1024 pixels). From these data it is necessary to extract multiple features (such as two-dimensional lines, regions and texture patches, and 3-D lines, surfaces and volumes). Each feature extraction process can take from tens of operations (simple edge detection) to many thousands of operations (texture discrimination). Thus low level processing alone could easily consume anywhere from 5 GOPS[2] (one simple process for a medium-resolution stereo image stream) to 10 TOPS[3] (ten processes with an average complexity of 5000 operations on a high resolution stereo image stream).

The above example illustrates that there are different levels of processing in a vision task. In general, tasks in a vision system fall into roughly three categories: low level, intermediate level, and high level, each of which has different processing requirements and needs different architectural supports.

In low level processing, only a small percentage of the operations involve floating point calculations (generally 32-b); most are 1-, 8-, and 16-b integer arithmetic and require communications that exploit locality. The development of computers that support these features is not the primary focus of many current architecture efforts, which instead target machines mainly used by the scientific and engineering communities.

Beyond low level processing, there usually are one or more intermediate levels that emphasize symbolic processing of the feature information extracted from sensory data. This hierarchy allows the information to be organized so that matches with high level structures in the knowledge base can occur efficiently. Intermediate level processing may require many diverse operations for organizing many local features into structures and for matching them to projected model data. These diverse operations involve computations such as spatial geometric relations, linking symbolic representations of features into graph structures, transforming graph representations of models (via shifting, rotating, scaling, and warping operations), and determining subgraph isomorphisms and confidence measures of matches.

Most intermediate level processing involves 16- and 32-b integer arithmetic, some 32-b floating point arithmetic, and lots of graph traversal and database search operations. Again, these are not typical of general-purpose computers. More importantly, the computation, communication, and control differ significantly from that of low level processing. In low level vision, the control mechanisms involve one or a small number of threads (in SIMD[4] mode), while intermediate level processing is better characterized by a large number of highly orchestrated threads (in SPMD[5] mode). Communication in low level processing can also take advantage of a great deal of locality inherent in image structures, while at the intermediate level the communication involves parallel global searches and reduction operations.

At the high level, there are further differences: computations are coarse grained, have independent control threads and communication, and may need spatially disparate processing. At this level, computation may have to be carried out at several different locations in a distributed fashion. One aspect of using distributed computing in this level is the transmission and management of huge amounts of image data.

[2]GOPS stands for giga-operations per second.

[3]TOPS stands for tera-operations per second.

[4]SIMD stands for single-instruction-multiple-data-stream.

[5]SPMD stands for single-program-multiple-data-stream.

It is, therefore, clear that vision cannot be mapped efficiently onto a homogeneous parallel processor. Yet efficiency is a key concern in almost all vision applications, most of which require small, lightweight, low power, low cost, real-time embedded processors that serve as powerful research tools as well as engines driving vision applications in the commercial and the military markets. These issues, and especially that of real-time processing, are generally neglected by the traditional architecture community, which focuses on large homogeneous parallel systems with a strong emphasis on double precision floating-point and vector performance.

High performance systems for computer vision lie at two ends of a wide spectrum of parallel architectures where research needs to be focused. The first one is the class of parallel supercomputers with distributed memory and a large number of coarse grain processing elements. Examples include Thinking Machines Corporation's CM5 and Intel's Paragon. These systems are useful tools in research as well as in production applications that are extremely data intensive, but do not require real-time responses (such as processing satellite and aerial imagery). The second type consists of special-purpose (digital and possibly analog) VLSI chips, such as focal plane processors, for very specific, real-time vision tasks. This would allow computer vision to enter the commercial market in areas such as consumer electronics, multimedia systems, optical character recognition, surveillance, and driver assistant. It will also have military applications for smart weapons, automatic target recognition (ATR), and autonomous surveillance vehicles. Between the two technologies described earlier, there are a variety of intermediate general-purpose and special-purpose computers, such as less expensive fine grain image processing boards and processors, that are suitable as experimental tools in research and in less time critical applications.

A research program on architectures for computer vision will involve three components: basic technology, computer architecture, and software support.

At the basic technology level, research needs to be carried out on integrating existing as well as emerging technologies in the design of real-time, small, inexpensive, and general-purpose vision systems. Examples of these technologies include 3-D VLSI and/or photonics, digital (and possibly analog) hybrid CMOS-CCD VLSI, application specific dynamic random access memory (DRAM) based processors, large hybrid substrate technology, and high density interconnect packages. Although some of these technologies may not be fully available today, it is essential that design tools be developed to incorporate them in vision applications when they become viable.

At the architecture level, there remain many issues to be resolved in order to integrate heterogeneous peripheral preprocessing modules (such as optical processors, analog neural nets, sensors with integrated analog preprocessing, and focal plane VLSI chips) with powerful parallel processing systems. Immediate issues that need to be addressed include:

a) development of a consistent set of interfaces for connecting parallel processors into a heterogeneous ensemble; b) development of a standard set of abstract machine models that can be mapped onto different parallel hardware platforms; and c) research on heterogeneous parallel and distributed architectures that permit different parallel systems to be tightly or loosely coupled in a flexible range of configurations, where the processing element at each level can be matched to the varying computational characteristics and data structures of vision algorithms. Tight coupling (i.e., high bandwidth, low latency communication, and coordinated control) is essential for parallel systems to work together efficiently; flexibility in configuring the system, on the other hand, is necessary to address the requirements of different vision applications. Of particular interest is the modularity and scalability of such architectures, especially the question of how different components can be easily "glued" together, and the communication and control pathways between different homogeneous parallel processors.

Last, but far from least, are the software issues. Software must be as easy to port from one heterogeneous configuration to another as it is to move from one workstation to another. It ought to be easy to program, taking advantage of the algorithmic simplifications that result from a more direct mapping of the vision problem onto a suitably structured hardware platform. This research will involve studies on languages and compilers, reusable and extensible libraries for supporting multiple abstract machine models, real-time operating systems, and tools for supporting the use of heterogeneous parallel machines. It will also involve the development and tests of vision software and algorithms on teraflop and scalable parallel supercomputers now becoming available and on other HPCC systems to be developed. One important challenge is to develop easy-to-use software that can adapt to different architectures and application requirements.

D. Infrastructure Support

The infrastructure needs of computer vision and its related applications are several: a) abundant computational cycles, tools, and displays for the design and development of application specific vision systems; b) a variety of building blocks that can be configured for the various applications requirements; and c) software infrastructure, image databases, and standards of image communication to enable the community to move forward coherently.

To design a high performance computer vision system, it is necessary to narrow the gap between what traditional supercomputers can offer and what is needed in computer vision applications to process pixel level, intermediate level, and symbolic level tasks. It is important that there be more joint efforts among members of the computer vision community and computer architects. It is also critical that the cost of visual processing systems be reduced to the range of \$1K–\$10K rather than \$100K–\$10 million. Such an engineering challenge may require a high technology production infrastructure quite different from that used to produce current day supercomputers. It is also essential that tools be available for configuring substrates and facilities (such as MOSIS) for fabricating them in sample lots at low cost. Finally, computer vision, as part of AI, is a field where experimental algorithm development tends to be much more the critical bottleneck than computational power, in spite of the tremendous need for cycles. Therefore, support for developing new vision-oriented software tools, as well as support for generic software infrastructure, is extremely important. Part of such software development efforts need to be directed at achieving at least a modicum of community consensus (a lá DARPA Image Understanding benchmarks) in order to have their desired impact across the community.

III. Speech and Natural Language Processing

The goal of SNLP is to process speech and natural language in real time in order to achieve natural, fluent human/machine communication. HPCC related research in this area will lead directly to developments that will revolutionize the modes of our communication with each other as well as with machines, and our ways of using and sharing vast amounts of information either in a monolingual or in a multilingual environment.

SNLP is strongly related to computer vision. Vision, proprioception, and controlled motion help understand the model of the world and extract the meaning from spoken messages. For instance, computer vision can be considered as a front-end for identifying characters in text, if presented in written form, before they are processed. This in itself is a computationally demanding task similar to the acoustic/phonetic recognition stage in speech recognition, especially when the text is handwritten in cursive script. At present, the technology for this task is no more advanced than that of recognizing colloquial discourse. Computer vision is also a crucial front-end for scanning and for storing bit maps when information is presented in printed form. In this case, semantic analysis is needed to analyze the images in order to extract the underlying structure and to connect the scattered columns of text that make up the printed articles (such as newspaper and magazine).

Spoken language (speech) also shares many common aspects with written language in terms of the lexicon, syntax, and semantics. Since all these features are necessary to accomplish automatic recognition of handwritten or printed language, studies of speech processing are applicable to those of the written language. Moreover, the computational complexities of the linguistic analysis algorithms for both areas are similar.

Speech is an important component when integrated with other communication media — sound, still images, moving images, and text — in multimedia systems. It is expected that information in such systems will be stored predominantly in digital form in electronic libraries by the end of the decade. Such databases can quickly grow to very large sizes, presenting challenges for designers of information storage, transmission, and presentation systems.

In the rest of this section, we describe a few applications to illustrate the importance of this area, discuss the fundamental science and enabling technologies, and highlight various implications for computer architects in designing new computers for this area.

A. Grand Challenge Applications

1) Electronic Library and Librarian: An electronic library represents one of the most exciting and potentially important grand challenges of the decade. We can foresee, by the year 2000, developing the technology needed to electronically store and distribute quantities of information large enough to rival a major national library such as the Library of Congress. This capability can fundamentally change the way scientists, engineers, scholars, teachers, and students work by making a wealth of information available from every desktop computer in the country. Full scale projects, such as putting the Library of Congress on line, await the appropriation of large amounts of money as well as further reduction in the cost of mass storage. However, over the next three years, it is possible to engage in scale-up experiments in which specialized libraries of intermediate size can be built and made available to the public.

To manage information in such an electronic library, an electronic librarian is needed to organize and catalog text as well as to assist users in locating information. Such systems will go beyond current automatic classification and full-text retrieval systems, not only providing access to individual files but also to entire collections of full text files, catalogs, reference material, encyclopedias, indexes, abstract files, and collections of images and sounds. One of the critical issues is the user interface of which speech and machine translation are important components. The issue related to information retrieval is discussed in Section IV-A.

a) Spoken language interfaces: A spoken language interface is a natural way to access information. A real-time spoken language interface can carry on a conversation with a range of users: from the casual or first-time user who may need guidance on how to use the system, to the expert who may choose to talk in a focused, concise manner. This will require a system that supports speaker-independent real-time spontaneous speech recognition and understanding (on a large vocabulary, and possibly even for multiple languages, if the library is multilingual). The system must also support dialogue modeling, language generation, and speech synthesis (all in real-time) in order to interact with the user.

b) Machine translation (MT): MT capabilities will be crucial for using documents in different languages as well for translating documents into other languages for their wider distribution. The major challenge for MT of texts is to bring the quality of text-oriented systems up to a level where everyday use is economically viable. The complexity and productiveness of human language, together with the difficulty of developing language understanding and generation capabilities, constitute the major obstacles for designing MT systems. It is widely perceived that in practice these obstacles translate into difficulties of accumulating and manipulating very large quantities of knowledge about the language and about the world. In particular, techniques for automatic acquisition of lexical, grammatical, and discourse knowledge are essential for significant progress in this area.

2) Spoken Language Translation: This capability will revolutionize human–human communication, involving applications in telephone-based as well as face-to-face modes of communication. It will improve international trade, communications among government and other national organizations, as well as activities in these organizations. Together with MT of written language, it will allow translation using facsimile machines, and will resolve the worldwide bottleneck in the quantity of documents that cannot be translated at the present speed and cost.

A central issue in the development of spoke language systems is the speed of processing. In this environment, the complexity of speech recognition is exacerbated by the complexity of natural language processing, since the amount of knowledge to be manipulated by the systems is significantly larger than in text translation. Moreover, there is a need to improve computer systems for speech-to-speech translation, with the aim of increasing the integration of the various component modules — speech recognition, language analysis (including morphology, syntax, semantics, pragmatics, and discourse), message content manipulation, message planning, language generation, and speech synthesis.

B. Fundamental Science and Enabling Technologies

1) Statistical Analysis in Corpus-Based Speech and Language Processing: The availability of large corpora, including some corpora annotated in various ways, has already led to the development of techniques that allow automatic acquisition of linguistic structure. We illustrate in the following examples the importance of statistical techniques for work with large corpora.

a) Semantic constraints: Semantic constraints determine what words can meaningfully appear as the subject or object of a particular verb, for example — these are critical for analyzing natural language, be it in text or spoken form. Building up such constraints by hand, however, is a very time-consuming and error-prone task. An alternative to the manual approach involves the automatic analysis of co-occurrence patterns in a text corpus. In this approach, a text is analyzed syntactically and various patterns (such as subject—verb pairs) are gathered. Word similarities are computed based on the frequency with which words occur in the same context (such as the subject of the same verb). Based on these similarities, words are gathered into classes, and the word level co-occurrence patterns are generalized into patterns stated in terms of word classes.

This approach has been tried on small corpora with some limited success. However, since we are looking for co-occurrence patterns involving very large numbers of word pairs, it is clear that proper evaluations will require large corpora, and accordingly expanded computing resources to handle these corpora.

b) Syntactic constraints: In developing robust SNLP systems, handcrafted grammars are inadequate because they fail to provide a wide coverage and are very brittle, rendering them not useful for processing speech and free texts. Techniques have been designed for automatic acquisition of linguistic structure from large corpora, some of which are annotated in various ways. These techniques have to be lexically sensitive; hence, they have to be integrated with techniques for discovering semantic constraints. To support these acquisition techniques, high performance computing and storage resources are needed.

Text collections of several gigabytes are available; however, with current computers, these would require months or even years to analyze. Computing power at the levels projected for HPCC would allow such analysis to be done in days, permitting experimentation with different analysis methods. Moreover, with iterative procedures, partial syntactic and semantic information from an initial text analysis can be used to produce improved parses, thereby improving syntactic as well as semantic information.

2) Search Strategies for Language Analysis: Language analysis is typically viewed as a large search problem, with various constraints and heuristics introduced to make the search tractable. Often we are obliged to introduce restricted constraints to limit the search, although more flexible or complex constraints would be preferred. For instance, syntactic and semantic constraints need to be relaxed in order to handle a number of types of ill-formedness. It is desirable to replace simple semantic constraints in metonymic and some other nonliteral usages by more complex reasoning procedures. Such approaches will be feasible only when computing systems developed under HPCC are available.

3) Auditory and Vocal Tract Modeling: The mathematical models of the speech signal in use today were proposed over fifty years ago and have remained essentially unaltered since that time. This model, called the source filter model, assumes that there are sound sources as well as an independent vocal tract that is a time-varying linear filter driven by the sound sources. Although there are many computational realizations of this model, the underlying mathematics for all of them is the wave equation. This model has been remarkably successful in speech processing applications partly because it is well matched to early forms of digital computing.

The source filter model is inadequate today because it has many limitations with respect to speech generation and recognition. For instance, speech recognizers based on this model are incapable of making fine phonetic distinctions reliably when synthetic voices are used and bandwidth is limited. These limitations can partly be overcome by developing new signal processing models that can exploit high performance computing systems developed under the HPCC Initiative. A particularly intriguing approach is that of computational fluid dynamics (CFD's) that views the speech signal as the acoustic consequence of fluid flow in the human vocal apparatus. CFD is also appropriate for studying the human auditory periphery whose principal organ is a fluid-filled chamber. Such studies may well help explain how the human listener can accurately classify speech sounds under adverse conditions.

4) Integration of Multiple Levels of Speech and Language Analyses: A speech recognition system typically operates in a very large search space, producing many candidate sentence hypotheses for each spoken utterance. The most obvious role of the language understanding component is to provide a meaningful representation for an utterance, which can be translated into some appropriate actions (such as database queries, function calls). This component needs to be designed so that it constrains the recognizer's search space and selects the most "meaningful" candidate word sequence.

Current recognition systems are generally "loosely coupled," where the recognition component delivers a set of candidate sentences to the language understanding component after completing its search. A tightly coupled system can potentially provide more accurate recognition, with the language understanding component working in locked steps with the recognition component in order to select the most promising partial hypotheses. Such an integration, however, is very computationally intensive, due to the size of the search space traversed during recognition. More research needs to be done in this area, especially in developing real-time tightly coupled systems for incorporating multiple levels of linguistic knowledge (syntax, semantics, discourse structure) and in providing constraints in the overall recognition process.

5) Connectionist Approach for Speech and Language Processing: Recent research has found that the connectionist approach can provide a cost effective yet specialized hardware solution to a number of problems in SNLP (such as continuous speech recognition). Further research needs to be carried out on connectionist algorithms and representation methods in the context of realistic tasks, such as the recognition and understanding of large vocabulary speech. This will allow the integration of knowledge in spoken language understanding at the acoustic, syntactic, semantic, and pragmatic levels, as well as data-driven learning over all of these levels.

A major issue of using the connectionist approach is that the amount of computational power and communication bandwidth required is large (well into the tera-operations and tera-bytes-per-second range), especially when practical and large training sets and neural networks are concerned. New research needs to be conducted in developing efficient yet general experimental hardware and software solutions.

C. Implications for System Architectures

Algorithms in SNLP generally require an abundant amount of computing power and memory space. This need arises partially from the desire to do real-time analysis. A somewhat less obvious reason is the need to analyze huge databases in experiments; these experiments will simply not be done if analysis takes a very long time.

To illustrate the computing power required, consider sentence analyzers available today that typically operate at 1–100 sentences per minute (on a 10 MIPS[6] workstation), with the range reflecting the degree to which they explore alternative hypotheses and make use of complex semantic constraints. At this rate, a 10^7 sentence corpus (such as is now becoming available) would require 10^5–10^7 min (2–200 mo) to parse. For meaningful iterations in parse/semantic analysis, the iteration cycle needs less than 10^3 min, implying a speedup of 10^2–10^4 (1 to 100 GIPS[7]).

In addition to the need for higher computational power, the architectural needs also differ for different basic research topics addressed. These requirements may be better served by

[6] MIPS stands for million instructions per second.
[7] GIPS stands for giga-instructions per second.

a hybrid system containing general-purpose as well as special-purpose computers. In the rest of this section, we discuss these requirements.

1) General-Purpose Supercomputing — More Cycles, More Memory, More Bandwidth: Many symbolic computations required for SNLP do not have the specialized regular structure that would justify special-purpose architectures. These computations simply require a substantial amount of computing power and will best be served by general-purpose parallel computer systems. For instance, many aspects of language analysis can be formulated as search tasks and can be carried out in parallel computers. In addition, for analyses of large corpora, much of the processing can be done independently on individual sentences and can be executed with minimal communication overhead in a general-purpose parallel system.

2) Homogeneous Architectures: While large problems in SNLP are inherently heterogeneous, homogeneity within large subtasks can be exploited. For instance, many tasks can be expressed as matrix and vector operations that can be executed efficiently on parallel and vector architectures. Attempts have also been made to unify the multiple aspects of SNLP using homogeneous architectures. Connectionist models are good examples for providing a consistent representation at all levels, where activation variables may be viewed as a particularly simple form of message (a single low-precision number). Although actual implementations may require elements that are not of this form, to the extent that they are connectionist, both software and hardware can be greatly simplified and still be quite flexible.

3) Special-Purpose Architectures: For a large part of SNLP, capabilities provided by general-purpose supercomputers (such as floating-point arithmetic and sophisticated memory-management schemes) are not required. For instance, many common paradigms — such as dynamic time warp, table-driven Viterbi decoding for vector-quantized input, and connectionist processing — use fixed-point arithmetic with moderate word widths (8–16 b). Reduced precision arithmetic helps reduce the silicon area required for implementation, permits high bandwidth communication in pin-limited circuits, and simplifies interprocessor communication requirements. These factors are particularly important for tasks common in SNLP, where data movements commonly overwhelm computations required. In addition, virtual memory and elaborate operating systems are not needed, thereby further reducing the complexity and cost of such systems.

The principal restriction of such a specialized system is its inflexibility. While some researchers argue that a completely special-purpose machine is adequate, substantial changes on the algorithms may be required for these machines. The challenge for architects in the speech and natural language area is, therefore, to design machines that are sufficiently flexible and programmable, and yet have a high performance gain over the rapidly advancing general-purpose machines. Such machines have been built and used successfully for limited subdomains in speech processing (multilayer perceptron training and time-synchronous Viterbi decoding). More research needs to be conducted in developing architecture independent computational models (such as object-oriented models) and algorithms, and methods for mapping these algorithms on general-purpose as well as special-purpose computers.

4) Heterogeneous Architectures and Hybrid Systems: Spoken language systems generally require a heterogeneous architecture, involving signal processing, numeric processing for the search algorithms during recognition, and symbolic processing to support the knowledge-based discourse and reasoning components of the system. In addition, in order to carry on real-time conversations with users, a real-time system is needed for processing the input as it is received, performing incremental signal processing, recognition, understanding (and even answer generation and speech synthesis). This creates an interesting challenge for high performance architectures, requiring heterogeneous, parallel, and pipelined processing. Moreover, a suitable architecture would need high bandwidth local networks and input/output devices for conducting on-line experiments in real-time at audio bandwidths.

There are many architectures that can be used for SNLP, ranging from heterogeneous subsystems that are loosely coupled (say, a general-purpose network of workstations), to a completely homogeneous SIMD array, and a whole spectrum of approaches in between. In general, programmable systems require some homogeneity so that software can be developed. Furthermore, nearly every algorithm requires some amount of general processing capability that is not obvious from the guiding equations. This "incidental" computing can easily dominate resources when the design of an architecture and specialized hardware has been targeted for a special algorithm. One possible solution is to provide a common computing element, such as a reduced-instruction-set-computer (RISC) core, to every node of the architecture, with specialized accelerators provided by (possibly heterogeneous) coprocessors. Such a system can be a good platform for developing and executing parallel software, while permitting specialization for subtasks (such as Viterbi search and feature extraction).

D. Infrastructure Support

The infrastructure needed for SNLP includes the development and technical support by high quality staff of sharable text and speech databases, and open, easy-to-access, and easy-to-program parallel systems.

1) Sharable Text and Speech Databases: An important role for a shared electronic library is to provide not only one place where information can be shared and transmitted, but to provide a place where the information itself can be studied. As we accumulate large quantities of text, speech, and images in sharable databases in parallel supercomputers, we have a valuable resource for research into SNLP and machine vision.

DARPA is in the process of establishing an industry-university consortium called Linguistic Data Consortium (LDC) for generating very large text and speech corpora. Initial plans call for the collection of texts up to 5 to 10 billion words, speech data for interactive tasks up to 400 h, and read speech up to 1000 h. HPCC environment is crucial for the successful exploitation of these resources, which are soon to be available.

2) Open Parallel Systems: One way to close the gap between special-purpose architectures and general-purpose computers is to develop compilers that aid designers in mapping programs on general-purpose systems. This will allow designers to develop prototypes on general-purpose systems that can be migrated later to special-purpose chips.

3) Easy Access to High Performance Computing Systems: Research in this area will be enhanced significantly when researchers have easy access to high performance computing centers through high performance wide area networks. This not only allows the sharing of expensive computing resources, but allows common on-line experiments to be conducted in real time at multiple geographically distributed laboratories.

IV. Artificial Intelligence

In the last two sections, we have discussed grand challenge issues for two important forms of communicating information — computer vision and SNLP. In general, communication takes place in a shared context that extends well beyond vision or language itself; we see objects and understand language because we share certain knowledge about how things behave in the world. Hence, any vision or language interface that involves understanding requires that the system have information about the "world" that it operates in. For limited domains, this information may be static and highly restricted. However, many domains will require substantial knowledge about the real world, an ability to reason about objects in this larger context, and an ability to maintain an updated view of the world based on actions that either the system or the user has performed. In addition, in complex problem solving situations, the system may plan how to interact with a user or how to solve a problem. This means that computer vision and language understanding are intimately tied to knowledge representation, reasoning, common-sense models of the world, and planning. Without this information, a vision or language understanding system may need to model sizable chunks of world knowledge.

Most AI systems are built from knowledge and search, and high performance hardware can make a big difference in how much knowledge can be collected and how much search can be carried out. The exact nature of the hardware varies from one AI paradigm to another. There is room for very general hardware architectures (for experimenting with new ideas) as well as very specialized architectures, once the performance and cost requirements are defined.

A. Grand Challenge Applications

In this subsection, we present eight specific applications requiring substantial innovation and breakthroughs in AI and in systems research. These applications were what the workshop participants thought would be of national significance and potentially achievable in a decade's time. Although many others were discussed, these were identified here to give a sense for what might be accomplished in the future with significant advances in basic research and system developments in AI.

1) Electronic Librarian: One of the grand challenge applications in AI is to design a computerized electronic librarian that knows how to navigate through the vast amount of knowledge and information in an electronic library and provide useful information. Besides communicating with users in a form close to natural language (Section III-A), the system must be able to access databases in a variety of query languages, update the databases automatically when new knowledge and retrieval methods are available, and assist clients in locating and utilizing information within an extremely large collection. Such a system will be immensely valuable to scientists and researchers. It could, for example, provide a natural language interface to users, scan the vast amount of knowledge and information, propose tests to run, find the right software and computer systems, run the programs, and report the results.

In designing the electronic librarian, information retrieval is an important problem to be addressed. The librarian would employ a mixture of AI, statistical inference, and linguistic analysis to locate information which is of interest to patrons of the library. Intelligence is also required in presenting the results of the search to the user and guiding the user as he or she browses the contents of the database. The challenge to this problem is two-fold. First, current techniques have been developed on relatively small databases and are only now being used and evaluated on databases with a few gigabytes of data. An electronic library will require that these methods be used on files containing tens, hundreds, and eventually thousands of gigabytes. Additional research is needed to ensure that full text search is as easy to use on these large files as it is on the more modest files of today. Second, most full text retrieval methods to date are concerned with reference searches in which the goal is to locate all documents pertaining to a given topic. Future full text retrieval techniques will need to adapt to account for differing needs of users who wish to find tutorial material or to browse, and to organize searches efficiently on thousands or tens of thousands of items on popular topics.

2) Nation Wide Job Bank: To help ensure that resources in the United States can be best matched to the greatest needs, a national job and training database system would be very important. Users with no computer skills can enter their work and skills profiles, scan job listings, have their profiles automatically matched against employer postings, learn what training is required to enter other fields, find sources of such training, and control the distribution and matching of their profiles. Required technologies include SNLP, distributed computing, distributed database technology, constraint satisfaction, agent modeling, and possibly intelligent and interactive animation.

3) Tutoring System: To bolster our education system, we need to exploit recent technological developments and develop intelligent and truly individualized tutoring systems for providing education on standard subjects. In addition to the computer vision technologies discussed in Section II-A, a real-time interactive tutoring system will need to integrate new technologies such as interactive multimedia systems, virtual reality, knowledge-based authoring systems, hypertext, and parallel processing.

4) Electronic Market Place: Providing a nation-wide electronic market place to bring suppliers and consumers (or investors and offerers) together electronically offers enormous potential for squeezing out inefficiencies in the retail, wholesale, and financial marketplaces. Reducing these inefficiencies means enhanced competitiveness, and removing intermediaries or the "middle man" from the marketplace. The huge costs in doing business in this country can, therefore, be greatly reduced, offering a relatively easy way for driving up productivity and enhancing the gross national product. While this is not a new idea, it has not been implemented at a large scale.

In such a system, both buyers and sellers of goods and services are represented by AI programs in the network, which find each other and negotiate prices and other details in order to conclude the transactions. For instance, a customer wishing to purchase an airline ticket can input into the system his or her utility function, encoding how much flexibility he or she has and what he or she is willing to pay. The system then interacts directly with the corresponding sales programs for airlines in order to negotiate the best times and fares for the traveler.

The grand technical challenge here is to provide a nationwide network that allows anyone (hundreds of millions of users, businesses, institutions, and individuals) to post requests for some commodity (or financial investment) and to be matched with an optimal counterparty. Thus at each moment, many hundreds of millions of outstanding requests have to be matched with counterparty offerers. One can list a multitude of enabling technologies needed to carry this out, many impinging upon AI techniques. Essentially, much knowledge needs to be brought to bare on a "nationwide optimization problem" that is inherently distributed and very large.

5) Semantic Integration and Knowledge Discovery: Besides collecting enormous amount of data (from satellites, various sensors, laboratory data, etc.) and applying "complex numerical modeling" and analysis techniques to that data, it is very important to consider the general desideratum of "knowledge discovery." An important and necessary grand challenge application is the problem of merging multiple data/knowledge sources and of discovering new knowledge from the merged sources. This involves much of what AI researchers have been studying for many years: learning, reasoning, and knowledge representation. The differences here, and consequently the grand challenges for AI, are: a) the scale of the problem is much larger than anything attempted before in AI; b) means for integrating multiple knowledge representations are important; c) methods for *efficiently* integrating multiple knowledge bases of very large size are essential; and d) symbolic techniques need to be integrated effectively with traditional numerical approaches.

6) Automatic Knowledge Acquisition: High performance AI systems will undoubtedly require very large knowledge bases. Today, the construction of even small to medium knowledge bases is a very time-consuming process and often prevents the application of AI to real-world problems. To overcome this deficiency, automation of knowledge-base construction is needed. Knowledge may be acquired from the vast amount of information stored as texts. Patterns of concepts and their semantic properties may then be extracted from text via natural language parsing and learning techniques. Knowledge-classification and knowledge-base management techniques will need to be developed to support the incorporation of information coming from diverse sources.

7) Intelligent Planning and Scheduling: This is a very large and economically important area, especially for problems such as scheduling production lines to minimize bottlenecks and delays, and planning flights for an airline to reduce overbooking and empty aircrafts. The problem here is similar to the problem in the electronic market place discussed previously: find the best match between demands and supplies. Some of these problems have been addressed before; however, efficient solutions have not been found due to the facts that optimizations are often beyond the limit of current supercomputers, and that experiments often involve perturbations of existing systems that may not be acceptable in a working environment.

8) Medical Diagnosis and Information: This is an important application that requires the integration of computer vision for analyzing medical images and the on-line storage and retrieval of patient records, medical history, and expert knowledge. Solutions will require the development of large knowledge bases interconnected by high bandwidth interconnection networks; many of the design issues are similar to problems encountered in the development of an electronic librarian.

B. Fundamental Science and Enabling Technologies

In this section, we describe six research issues. We start by presenting issues related to machine learning and its implementations on high performance computing systems. In addition to its prominence in AI, we have pointed out earlier the importance of machine learning in computer vision and SNLP. This is followed by discussions on heuristic search and four important enabling technologies that would aid in achieving success in basic research and application of these results to realistically large problems. The key idea of these research issues is to embed AI into "mainstream" systems in such a way that we can easily integrate AI with new systems and allow it to co-exist seamlessly with other HPCC applications.

1) Machine Learning: Machine learning is an essential method for acquiring knowledge in AI. Besides the learning mechanisms discussed earlier in computer vision and SNLP, much recent work has focused on sequential problems in decision, control, optimization, robotics, and planning. There are numerous types of learning methods that learn by instruction, explanation, induction, deduction, and analogy. Although many prototypes have been developed over the last forty years to test different learning methods, there remain many open issues related to machine learning and computer architecture. We list a few important issues related to the implementation of machine learning methods in high performance computing systems in the following.

a) Scaling and generalization across tasks: Scaling is related to the generalization of the knowledge learned to more general tasks beyond those used in the process of knowledge acquisition. Many current methods are restricted in their ability to generalize across the same problem of different sizes and different problems. A particularly difficult problem is the verification and validation of the knowledge generalized.

b) Quality efficiency trade-off: This issue is related to the cost of using the knowledge learned and the effectiveness of the knowledge acquired. Their trade-off is often not well defined; hence, the learning system may have to be designed so it can propose alternative forms of the knowledge for different application conditions.

c) Learning in knowledge-rich and knowledge-lean environments: When the learning environment has access to domain dependent and domain independent knowledge, it is necessary to design a learning system that can utilize this knowledge to the best extent. On the other hand, when the domain is knowledge lean, the issue is to design a robust learning method that can acquire the best knowledge in limited time. The learning system must be able to trade between generation (sampling) and testing (confidence building).

d) Resource constraints: Resource constraints are almost always important in any learning situation. These constraints include real-time constraints, constraints on storage and computational power, and constraints on communication bandwidth. Recent studies have addressed these issues but not many powerful methods have been found. Since the search space for learning methods may be unbounded, the designer of the learning system must trade between the amount of high performance computing resources available for learning and the quality of the knowledge learned.

e) Instrumentation of application environment for learning: Many learning situations may involve physical instrumentation of the learning environment and possibly modification of strategies used in the application. This may not be acceptable, especially in an operational system. A possible approach is to build a model or a prototype, learn new knowledge and strategies based on the model, and verify the knowledge on the physical system. In many cases, building the prototype can be as hard as building the original system itself. More efforts are needed on applying learning methods in realistic environments.

2) Heuristic Search: Heuristic search through a problem space is an important problem solving paradigm in computer vision, SNLP, and AI applications in general. It is a powerful model of high level reasoning and problem solving. Heuristic search algorithms fall into three main categories: single agent problem, two-player games, and constraint satisfaction problems. Since heuristic search techniques may require an unbounded amount of time, space, and other resources, it is critical to develop efficient search techniques that use limited resources, and automated learning methods for acquiring heuristic knowledge used in the search.

3) Scalable and Verifiable Knowledge-Based Systems: Scalability and verifiability are two important properties in symbolic systems. On one hand, knowledge may be acquired incrementally and expert systems may need to be refined over time. It is, therefore, important for such systems to be scalable when more knowledge is available. On the other hand, expert systems applied in life-critical systems, such as process control and sensor fusion, may need to be verified before they can be applied. Studies in developing verification methods have been carried out in the past, but none have been successful to the extent of software engineering. More work on knowledge engineering — the engineering approach to acquire, verify, and maintain knowledge — needs to be done.

4) Construction and Utilization of Very Large Knowledge Bases: Knowledge is an important element in AI systems. A desideratum is to build large knowledge bases by the end of the decade with orders of magnitude increases in size and scope. These knowledge bases may be used for storing information in the applications discussed earlier. Issues on knowledge representation, knowledge access methods, handling multiple views of the same knowledge, user interface, and the relationship between knowledge-base management and database management should be addressed.

5) Active Memories: Some of the main problems with current AI systems are related to the effective use and access of knowledge. Since knowledge accessed in a particular context may be scattered in a large knowledge base, the efficiency is low when a large knowledge base is implemented in an existing system designed on the basis of locality of access. One solution to this problem is to use active memories, which can support marker passing and value passing, and reasoning techniques such as memory-based and case-based reasoning. This concept is consistent with the desire in computer architecture to decrease the so-called Von Neumann bottleneck created by the separation of memory from processors. The development of effective architectures for supporting operations in active memories is an important issue to be addressed.

6) Artificial Neural Networks: In the next few years it is expected that artificial neural networks can contribute in very important ways to grand challenge applications that require subtasks such as pattern classification. Although neural network technology in its modern incarnation is only 6 or 7 years old, and most successes to date have been on simple classification problems, it may eventually be able to handle high level recognition and planning tasks. Besides the applications involving low level operations in computer vision and speech processing discussed earlier, it is recognized that neural networks can complement learning and classification tasks studied in traditional AI. An important research project is, therefore, to develop techniques that can exploit both neural networks and traditional machine learning methods in grand challenge applications.

C. Implications for System Architectures

Prior to designing HPCC systems, there is need for understanding the system's requirements imposed by the AI computation paradigm. The more we understand about the nature of AI processing, the more efficient HPCC systems we can build. Although most of the current research in AI is based on sequential reasoning, we expect that many future AI systems will use massively parallel processing techniques. This is straightforward in some AI paradigms which can take advantage of the natural parallelism in applications. However, the design of large scale parallel systems for supporting a large spectrum of AI techniques may impose a difficult task on computer architects and programmers. It is clear that this research could benefit from interdisciplinary teamwork among AI researchers, knowledge engineers, computer architects, and application designers. The following identifies issues in designing computers for AI processing.

1) Understanding the Requirements: More work is needed for understanding the computational requirements for each AI processing paradigm used in high performance systems. It is important to determine the processor structure for each paradigm. Architectures suited to supporting AI will almost certainly differ significantly from conventional, numeric-oriented parallel processing architectures. The computation models deemed most suited for AI computations tend to exhibit large degrees of dynamical parallelism, dictating architectural support for dynamic load balancing and dynamic extraction of parallelism. The grain sizes in AI computations will most likely have far greater variances than those of numeric computations, requiring processor structures oriented more toward latency toleration than latency hiding. Memory and I/O subsystems will offer the opportunity to off-load much of the repetitive computations from the CPU. The interconnection network will need to support large amounts of nondeterministic accesses and queries in knowledge processing and learning.

The design and evaluation of such systems may have to rely on expensive software simulations in order to arrive at a proper trade-off between a software solution that maps the AI computations on existing parallel machines, and a hardware solution that requires the design of new dedicated architectures.

2) Supporting Large Knowledge Bases: As we expect that future AI applications will involve large knowledge bases, hardware and software supports for maintaining these knowledge bases are very important. In these knowledge bases, reasoning techniques using the knowledge stored may access information that is not localized in a particular storage media or system. Moreover, the knowledge stored will evolve over time and may be revised by frequent updates. Hence, important issues to be considered include the design of techniques for knowledge-base partitioning and knowledge-base maintenance (such as update policies), and the development of high-speed distributed computing systems, computer networks, and input/output peripherals for supporting knowledge accesses.

3) Homogeneity Versus Heterogeneity: In many applications such as image and speech understanding, symbolic AI processing and reasoning is done after intensive numeric computations take place. For these application areas one approach to high performance systems is to design heterogeneous architectures that combine low level numeric processing with high level reasoning. For other applications, it is seldom that one form of computer architecture is effective for the entire spectrum of algorithms. This phenomenon has been discussed earlier with respect to grand challenge issues in computer vision and speech and natural language understanding. In such cases, it is desirable to have a heterogeneous architecture that has custom-designed hardware for processing certain specialized tasks and general-purpose hardware for carrying out the high-level reasoning and management tasks.

An important decision the systems designer must make, therefore, is the use of general-purpose computers versus specialized accelerators. The system designer must understand the computational requirements of AI processing and balance between the performance gains from the specialized hardware and the cost of such implementations.

4) Language and Compiler Support: The choice of the language in AI processing is important but it is not likely to solve the problem in the near future. Currently, there is a mismatch between the high level languages suitable for developing AI software (such as Lisp and other traditional AI languages) and an architecture that can execute the resulting software for real-time applications.

More work is needed in the development of good compilers, software engineering techniques, software development environments, debugging tools for large scale parallel systems, performance visualization tools, and methods for decomposing AI software between general-purpose and special-purpose hardware.

D. Infrastructure Support

AI researchers working on HPCC projects will benefit by having access to on-line knowledge bases, benchmarks, corpora, (hardware and software) systems developed by other researchers, and design and evaluation tools. Sharing such information will avoid spending unnecessary efforts to develop similar information, and will help calibrate new results developed in the future. Such information sharing practices are emerging, as is in the development of common benchmarks in computer vision and SNLP.

However, the community at large will need more coherent and coordinated efforts in collecting such information and in making it easily accessible.

The AI community also needs access to fast and large scale parallel computing resources with fast I/O, and toolkits embodying traditional numerical modeling codes (such as Monte Carlo simulations, ordinary differential equations, and various statistical analysis packages).

To integrate AI systems with traditional numerical systems implies that the AI systems must be "embeddable" into "mainstream" computing environments. This also means that AI symbolic computing at a minimum must be compiled into the native environments hosting all of the components. These sharable resources allow AI researchers to scale their small scale prototypes and simulations on workstations to a more realistic environment. Using these resources, researchers can evaluate performance of large scale experiments, process large data-sets in a reasonable amount of time, integrate with numerical systems, and find means of efficiently assimilating diverse sources of information.

V. Final Remarks

We have identified in this paper many challenging applications and numerous issues that need further study. These grand challenge application problems are probably among the hardest that any researchers have attempted before. Some of these applications may be within our reach by the year 2000, while others may take longer for solutions to mature. Progress in these areas will require advances in device and networking technologies, signal processing techniques, AI, algorithms, and new ideas in integrating many existing results and components into working systems.

In this paper we have only scratched the surface of the issues related to grand challenge applications in the areas discussed. Although we do not offer any solutions for these problems, we have succeeded in bringing together many experts in these diversified areas to discuss common issues, solutions, and techniques. We feel that there is much to gain by holding frequent workshops such as the one that this paper is based on. These workshops allow people working in different areas — researchers, application designers, AI engineers, tools builders, and experts working on specific grand challenge application problems — to get together and reexamine their work and attempt to find common solutions. We hope that this paper will motivate further excitement in these areas.

ACKNOWLEDGMENT

The authors of this paper wish to thank Dr. Su-Shing Chen and Dr. Y. T. Chien of the NSF for suggesting this workshop. The authors also very much appreciate the guidance provided by Dr. Chen in focusing our attention on important issues. The authors are also grateful for the comments provided by the following Division and Program Directors of NSF who attended the workshop: Syed Kamal Abdali (Numeric, Symbolic, and Geometric Computations), Paul G. Chapin (Linguistics), Su-Shing Chen (Knowledge Models and Cognitive Systems), Bernard Chern (Microelectronic Information Processing Systems), Y. T. Chien (Information, Robotics, and Intelligent Systems), John H. Cozzens (Circuits and Signal Processing), John D. Hestenes (Interactive Systems), Richard Hirsch (Supercomputer Center), John Lehmann (Microelectronic Information Processing Systems), Howard Moraff (Robotics and Machine Intelligence), Pen-Chung Yew (Microelectronic Systems Architecture), and Zeke Zalcstein (Computer Systems Architecture).

B. W. Wah (S'78–M'79–SM'85–F'91) is a Professor of Electrical and Computer Engineering and a Research Professor at the Coordinated Science Laboratory and the Beckman Institute, University of Illinois at Urbana-Champaign. His current research interests are in areas relating to parallel and distributed processing and AI. His major focus is on the development of efficient resource management schemes for machine learning and search, with applications in operating systems, neural networks, computer vision, and genetic algorithms.

Dr. Wah is Editor-in-Chief of IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, and serves as a member of the IEEE Computer Society Board of Governors.

T. S. Huang(S'61–M'63–SM'76–F'79) is a Professor of Electrical and Computer Engineering, and a Research Professor at the Coordinated Science Laboratory and the Beckman Institute, University of Illinois at Urbana-Champaign. His current research interests are in image processing and computer vision, especially time-varying image processing and dynamic scene analysis.

Dr. Huang is a Fellow of the Optical Society of America; Area Editor for *CVGIP: Image Understanding*, and an Editor of the Springer Series in Information Sciences. He served as the Chairman of the IEEE Computer Society Technical Committee on Pattern Analysis and Machine Intelligence, 1989–1991, and received the Technical Achievement Award and the Society Award of the IEEE Signal Processing Society, in 1987 and 1991, respectively.

A. K. Joshi (S'54–M'56–SM'68–F'77) is the Henry Salvatori Professor of Computer and Cognitive Science at the University of Pennsylvania, Philadelphia, PA. His research is in the areas of computational linguistics, mathematical and processing models of language, AI, and cognitive science.

Dr. Joshi is a Past President of the Association for Computational Linguistics, and a Founding Fellow of the AAAI. At present he is a Co-Director of the Institute for Research in Cognitive Science, a NSF Science and Technology Center, at the University of Pennsylvania.

D. Moldovan (S'76–M'78–SM'91) received the M.S. and Ph.D. degrees in electrical Engineering and computer science from Columbia University, New York, NY, in 1974 and 1978, respectively.

He is currently an Associate Professor of Computer Engineering at the University of Southern California. His primary research interests are in the fields of parallel processing and AI. He is also the Director of the Parallel Knowledge Processing Laboratory, where a parallel computer for AI applications is being built.

J. Aloimonos (M'91) studied mathematics in Athens, Greece and received the Ph.D. degree in computer science from the University of Rochester, NY in 1986.

He is currently an Associate Professor at the Department of Computer Science of the University of Maryland, College Park, and is affiliated with the Institute for Advanced Computer Studies. He is also the Head of the Computer Vision Laboratory of the Center for Automation Research of the same University. His research interests are in the area of computer vision and robotics.

R. K. Bajcsy (M'81–SM'88–F'92) received the Ph.D. degree in electrical engineering from the Slovak Technical University in Bratislava, Czechoslovakia in 1967 and the Ph.D. degree in computer science from Stanford University in 1972.

Since 1972, she has been on the faculty of the Computer and Information Science Department at the University of Pennsylvania, where she served as its chairperson between 1984 and 1989. She has worked in the general area of machine perception, covering the whole spectrum of problems (segmentation, 3-D shape recognition, multiresolution problems, etc.) that the field of computer vision involves, and has extended it to other modalities, in particular to touch. She has built the GRASP Laboratory.

Dr. Bajcsy is a Founding Fellow of the AAAI.

D. Ballard received the Ph.D. degree from the University of California at Irvine in 1974. During 1974–1975 he had a Post-Doctoral appointment at the Laboratorio Tecnologie Biomediche in Rome, Italy.

Since 1975 he has been at the University of Rochester in the Computer Science Department where he is currently a Professor. He is the coauthor with Professor Brown of *Computer Vision*, a standard text. His current research focus is in computational theories of vision that account for real-time performance. In 1985 with Chris Brown, he led a team that designed and built a high-speed binocular camera control system capable of simulating human eye movements.

D. DeGroot (M'81) received the Ph.D. degree in computer science from the University of Texas at Austin in 1981.

He then joined IBM's T.J. Watson Research Center where he worked on the design team for an internally parallel, numeric, and vector processor based on the IBM 801 processor architecture. He then became VP of Research and Development at Quintus Computer Systems, developers of Quintus Prolog. In 1987, he joined the Computer Science Center of Texas Instruments in Dallas, TX, as Manager of the Parallel and System Architectures Branch.

Dr. DeGroot is the ex-Chairman of ACM's SIGARCH. He is on (or has served on) the Editorial Boards of *IEEE Expert*, the *Journal of Parallel and Distributed Computing*, *New Generation Computing*, and the IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS.

K. A. DeJong received the Ph.D. degree in computer science from the University of Michigan in 1975.

He is currently an Associate Professor of Computer Science and Associate Director of the AI Center at George Mason University in Fairfax, VA. His research interests include adaptive systems, machine learning, expert systems, and knowledge representation. He is active in the Genetic Algorithms research community with a variety of papers, Ph.D. students, and presentations.

Dr. DeJong is currently Chair of the executive council of the Society of Genetic Algorithms and Editor-in-Chief of a new journal, *Evolutionary Computation*.

C. R. Dyer (S'75–M'79–SM'85) received the Ph.D. degree in computer science from the University of Maryland, College Park, in 1979.

He is currently Professor and Chair of the Computer Sciences Department at the University of Wisconsin, Madison. His research interests in computer vision include model-based image understanding, active and purposive vision, three-dimensional object representations, spatio-temporal image analysis, and parallel algorithms and architectures for vision. He has written over 80 publications in these and other related areas of computer vision.

Dr. Dyer was an Associate Editor of the IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE from 1987 to 1991, and is currently on the Editorial Board of the *Journal of Machine Vision and Applications*. He is Program Co-Chair of the 1992 IEEE Workshop on Applications of Computer Vision.

S. E. Fahlman received the B.S. M.S. and Ph.D. degrees from the Massachusetts Institute of Technology.

He is currently a Senior Research Scientist with Carnegie Mellon University's School of Computer Science, where he has been on the faculty since 1978. His research interests include AI and artificial neural networks. He has developed several new learning algorithms for neural nets, including Quickprop and Cascade-Correlation. He has also been active in the development of hardware and software tools for AI, including the Common Lisp language.

R. Grishman is Professor of Computer Science at New York University. He has been involved in research in natural language processing since 1969. He served for a year (1982–1983) as project leader in natural language processing at the Navy Center for Applied Research in Artificial Intelligence. Since 1985, he has directed the Proteus Project, funded by the Defense Advanced Research Projects Agency and the National Science Foundation, which has conducted research in natural language text analysis, including information extraction, document retrieval, and machine translation.

Dr. Grishman is a Past President of the Association for Computational Linguistics, and the author of the text *Computational Linguistics: An Introduction*.

L. Hirschman (M'82) received the Ph.D. degree from the University of Pennsylvania in 1972.

She is currently a Principal Research Scientist in the Spoken Language Systems Group at Masschusetts Insititute of Technology Laboratory for Computer Science. She has done research on language understanding since 1972. Her early research included work on sublanguage grammars and discovery procedures for sublanguage semantics, as well as logic grammar-based text understanding systems. Her recent work focuses on spoken language systems, particularly integration of speech and natural language processing and dialog modeling. She chairs the DARPA Spoken Language Multi-Site Data Collection Working Group, coordinating data collection and performance evaluation of interactive spoken language systems.

R. E. Korf received the B.S. degree in computer science from the Massachusetts Institute in 1977, and the M.S. and Ph.D. degrees in computer science from Carnegie-Mellon University in 1980 and 1983, respectively.

He is currently an Associate Professor of Computer Science at the University of California, Los Angeles. From 1983 to 1985, he served as Herbert M. Singer Assistant Professor of Computer Science at Columbia University. His research is in the areas of problem-solving, heuristic search, and planning in AI. He is the author of *Learning to Solve Problems by Searching for Macro-Operators* (Pitman, 1985).

Dr. Korf received the 1985 IBM Faculty Development Award, a 1986 NSF Presidential Young Investigator Award, and the first UCLA Computer Science Department Distinguished Teaching Award in 1989.

S. E. Levinson (S'72–M'74–SM'82–F'86) received the Ph.D. degree in electrical engineering from the University of Rhode Island in 1974.

He has served as a design engineer at General Dynamics between 1966 and 1969, and was a J. Willard Gibbs Instructor in Computer Science at Yale University between 1974 and 1976. In 1976, he joined Bell Laboratories in Murray Hill, NJ. He is currently head of the Linguistics Research Department at AT&T Bell Laboratories, where he directs research in speech synthesis, speech recognition and spoken language translation.

Dr. Levinson is a fellow of the Acoustical Society of America. He is on the editorial board of *Speech Technology*, a founding editor of *Computer Speech and Language*, and a member of the IEEE Signal Processing Society Technical Directions Committee.

D. P. Miranker received the B.S. degree in mathematics from the Massachusetts Institute of Technology in 1979 and the Ph.D. degree in computer science from Columbia University in 1987.

He is an currently an Associate Professor of Computer Science at the University of Texas at Austin, where his work encompasses compilation of rule-based programs for high performance execution, targeting parallel supercomputers, database systems and real-time environments. He has done research on the parallel execution of expert-system programs on DADO, a 1023 processor multicomputer and is the co-patent holder on the DADO architecture.

N. H. Morgan (S'76–M'78–SM'87) is currently the group leader for the Computer Engineering Department (called the Realization Group) at the International Computer Science Institute (ICSI), a nonprofit research laboratory closely associated with the EECS Department at University of California, Berkeley, where he is also an Adjunct Professor. In the areas of speech and brain wave processing he has developed connectionist approaches to signal analysis, as well as hardware and software systems to speed research. His current interests include the design of algorithms, architectures, and systems for parallel signal processing and pattern recognition, particularly using connectionist paradigms.

S. Nirenburg received the Ph.D. degree in linguistics from the Hebrew University of Jerusalem, Israel.

He is currently a Senior Research Computer Scientist at the Center for Machine Translation, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA. He has written or edited four books and has published over 60 articles in various areas of computational linguistics and AI, including machine translation, natural language understanding and generation, knowledge acquisition, intelligent interfaces, planning, and cognitive modeling. He has directed many large research and development projects on machine translation and natural language processing. Since 1987 he has served as Editor-in-Chief of *Machine Translation*.

Dr. Nirenburg has served as First Vice President of the American Machine Translation Association since 1991.

T. Poggio (M'86) is the Uncas and Helen Whitaker Professor in the Department of Brain and Cognitive Sciences at the Massachusetts Institute of Technology. He is known for his pioneering work on stereo vision, for his work on the application of regularization theory to the problem of early vision, and more recently, for the development of a new theory of learning and networks. He is Co-Director of the Center for Biological Information Processing at M.I.T., where he pursues the understanding of information processing in the brain.

Dr. Poggio is a Corporate Fellow of Thinking Machines Corporation and has won many awards. He has written over 200 papers in the areas of AI, computer vision, biological information processing, learning, and nonlinear systems theory. He is a Fellow of the AAAI, and an Associate of the Neuroscience Research Program at Rockefeller University.

E. M. Riseman (S'65–M'68) received the Ph.D. degree in electrical engineering from Cornell University in 1969.

He is currently a Professor of Computer Science at the University of Massachusetts, where he has conducted research in computer vision, AI, learning, and pattern recognition. Since 1976, he has been the Director of the Computer Vision Laboratory with current projects in knowledge-based scene interpretation, motion analysis, mobile robot navigation, and parallel architectures for real-time vision. He has applied the analysis of static and dynamic images to a variety of domains, including vehicle navigation in outdoor terrain, industrial manufacturing environments, and satellite photointerpretation. He is on the Editorial Boards of *CVGIP: Image Understanding* and the *International Journal of Computer Vision*.

Dr. Riseman is a Fellow Member of the AAAI.

C. Stanfill received the Ph.D. degree from the University of Maryland in 1983 for work on qualitative reasoning for Artificial Intelligence.

Since 1983 he has worked for Thinking Machines Corporation, studying the use of massively parallel computers for a variety of tasks including memory-based reasoning, statistical computing, database mining, information retrieval, and relational databases.

Dr. Stanfill is currently serving as Secretary of ACM SIGIR.

S. J. Stolfo received the Ph.D. degree in computer science from NYU's Courant Institute of Mathematical Sciences.

He is currently an Associate Professor of Computer Science at Columbia University, where he has been on the faculty since 1979. He He has conducted research in the area of AI knowledge-based systems and parallel processing. In 1982 he co-designed and developed ACE, an expert system that integrated AI inference techniques with conventional database management system. He also led the design and development of the DADO parallel computer specialized to run knowledge-based systems and signal interpretation tasks at high speeds. In 1983, he established the Massively Parallel Computing Focal Project of the Center for Advanced Technology at Columbia University.

S. L. Tanimoto (S'75–M'76–SM'86) is a Professor of Computer Science and Engineering and Adjunct Professor of Electrical Engineering at the University of Washington in Seattle, where he has been a member of the faculty since 1977. His research has concentrated on pyramidal data structures and the languages, parallel architectures, and algorithms that work with them. Other projects are or have been concerned with the use of pictures and diagrams to program and use computers and with the use of image processing in mathematics education. He served as Editor-in-Chief of the IEEE Transactions on Pattern Analysis and Machine Intelligence from 1986 to 1990. He is the author of *The Elements of Artificial Intelligence Using Common Lisp* (W.H. Freeman, 1990).

C. Weems (S'76–M'84) received the B.S. and M.A. degrees in computer sciences from Oregon State University in 1977 and 1979, respectively, and the Ph.D. degree in computer science from the University of Massachusetts at Amherst in 1984.

Since 1984 he has been Director of the Parallel Image Understanding Architectures Research Group at the University of Massachusetts, where he is an Associate Professor. His research interests include parallel architectures to support low, intermediate, and high level computer vision, benchmarks for vision, parallel programming languages, and parallel vision algorithms. He is also the coauthor of a widely used introductory computer science text.