The Living Machine Initiative

John (Juyang) Weng
Department of Computer Science
Michigan State University
East Lansing, MI 48824

Technical Report MSU-CPS-96-60, Dec. 1996

Key words:
Autonomous learning, comprehensive learning, human-machine interface, computer vision, speech recognition, speech synthesis, robotics, autonomous navigation, knowledge bases, language understanding, autonomous agents, artificial intelligence, cognitive sciences, developmental psychology, neuropsychology, living machines.

Abstract

While digital multimedia are entering all walks of life, breakthroughs in machine understanding of multimodal information such as video, images, speech, language, and various forms of hand-written or mix-printed text, can lead to numerous applications that will significantly expand the application base of computer technology, and improve human life, scientific and engineering research, education, and human resource base. However, machine understanding of multimodal information in its general form proves to be an extremely challenging task facing the research community today, despite the fast and sustained advance of computers in their speed, storage capacity, performance-to-price ratio, and installation base. The principal investigator (PI) has been investigating persisting difficulties encountered by the existing basic methodology — manually-modeling-knowledge and spoonfeeding-knowledge (MMKSK). Researchers in each subfield have been manually developing knowledge-level theories and methods, and using them to write programs or build hardware. Then, they manually "spoon feed" knowledge into the systems at the programming level, hoping that these systems will be sophisticated and complete enough one day to understand single-modal or multimodal information. In sharp contrast, however, each human child learns to acquire such capabilities through everyday activities in the real-world environment, from birth to toddler age, to school age, and through his or her entire life. As developmental psychologist Jean Piaget observed and categorized, each human individual undergoes four stages of cognitive development: sensory motor (birth to age 2); preoperational (age 2 to 6); concrete operational (age 6 to 12); formal operational (age 12 and beyond). No knowledge-level manual-modeling seems to be able to handle the complexity of cognitive knowledge that human-level performance requires, and no spoon-fed knowledge can practically handle the richness and amount of multimodal interactions that are essential to the success of human's cognitive development. In order to avoid the fundamental limit of the current MMKSK methodology, the PI proposes a fundamental shift to the new direction termed "living machines". This document describes the PI's three-phase endeavor to develop autonomous learning machines and train them in the human's real-world. The objectives include development of a systematic theory and a practical methodology for machines to learn autonomously while interacting with its environment, on a daily basis, via its sensors and effectors, on-line in real time, under interactive guidance from human teachers. This three-phase endeavor has now entered the second phase. This new direction of research is expected to result in long-lasting and sustained pay-offs in developing machine capability of understanding multimodal information and that of using such a capability.

Contents

1 EXECUTIVE SUMMARY

1.1 The Grand Challenge

Despite the power of modern computers, whose principle was first introduced in 1936 by Alan Turing in his now celebrated paper [83], we have seen a paradoxical picture of artificial intelligence: Computers have done very well in those areas that are typically considered very difficult (by humans), such as playing chess games (e.g., Kasparov vs. Deep Blue [39]); but they have done poorly in areas that are commonly considered easy (by humans), such as vision (see, e.g., a report of discussions on challenges in computer vision during an NSF workshop [56]). It seems a relatively simpler task to write a program to solve a symbolic problem where the problem is well-defined and the input is human-preprocessed symbolic data. However, it is very difficult for a computer to solve a problem that is not definable mathematically while using the raw sensory data in their original form.

On one hand, the computer industry has enjoyed fast and sustained advances in speed, storage capacity, performance-to-price ratio, and installation base. These advances have brought digital media into all walks of life. Now, all forms of digital media, video, audio, images, and text, have become ubiquitous, from living rooms, to classrooms and to corporate floors. These mass-market phenomena of once-expensive digital technology has opened an unprecedented possibility for developing reasonably-priced multimodal human-machine communication systems and multimodal understanding machines. Breakthroughs in machine understanding of multimodal information, such as video, images, speech, language, and various forms of hand-written or mix-printed text, can lead to numerous long-awaited applications.

On the other hand, however, the Grand Challenge that the research community faces is grim. There is no breakthrough in sight toward machine understanding of multimodal information in general settings, including vision, speech and language, and hand-written general text. In fact, each sensing modality has already met tremendous difficulties. For example, although there have been some limited applications in controlled settings [25], the computer vision community has seen a wave of pessimism toward solving the challenging problems of visual understanding in general settings, as indicated by several invited talks at ICPR (International Conference on Pattern Recognition), 1996 presented by some very well-known senior researchers in the field. The difficulties encountered in vision might not be surprising if one realizes that it is the most difficult sensing modality in human. In fact, a half of the cerebral cortex in the human brain is given over to visual processing. As is well known, a huge portion of human knowledge is acquired through vision. The speech recognition field has some limited applications for recognizing words and short sentences in a controlled setting [1] [86]. However, further advances applicable to general settings requires understanding the situation, context, speaker's intention, speaker characteristics, language and the meaning of what is said [34] [113]. In the natural language understanding field, we have seen various low-level applications in language processing from text inputs, such as spell checkers, grammar checkers and natural language search [14], but these low-level applications do not require true understanding of text. It has been well known that language understanding requires not only syntax but also semantics and "common sense" knowledge. Several grandiose language knowledge-bases are being developed, such as the common-sense knowledge-base, CYC [45] [46] and lexical database, WordNet [54]. However, it is an open question whether a machine can really understand anything in a pure text form without its own experience. Is linking from one string of letters to another true "understanding"? Can the system use text properly in, e.g., language translation? Language discourse and language translation are good tests of language understanding. However, tremendous difficulties persist in these subjects. Like speech recognition, hand-written character recognition without understanding the meaning of the context cannot go very far.

Facing the Grand Challenges and with seemingly minimal hope of a significant breakthrough in existing methodology, the research community must seek and encourage fundamental changes in the way we approach these problems.

The conceptualization and development of the PI's theory and methodology on living machines have benefited a lot from related fields such as psychology, neurophysiology, evolutionary biology, ecology, cybernetics, machine intelligence, system sciences, statistics, robotics, and control theory. A huge volume of

incomplete yet very rich and illuminating facts about human cognitive development has demonstrated that this field should adopt a fundamentally different paradigm, a paradigm that is probably not well accepted by the establishment in the field now, but will lead to fundamental breakthroughs toward meeting the Grand Challenge.

1.2 Living Machines

The new paradigm is to develop *general-purpose* living machines (or just living machines for short). In order to distinguish the concept of living machines from other robots or virtual machines, a characterization is necessary. A machine is called a "*living machine*" if it has the following properties.

Sensing and action The machine must use its own sensors and effectors. The sensors may include all those that can sense the world around it (i.e., environment) without human's manual processing, such as cameras, microphones and many other types of sensors. The keyboard is excluded because it requires humans to type. The effectors are things that can change the environment in some way under computer control. They include mobile drive systems, robot manipulators, camera positioners, speakers, printers, displays, etc. The level of performance that a living machine can reach depends very much on the type of sensors and effectors that are used.

Knowledge-free representation Due to the tremendous volume and the vast variety of knowledge that the Grand Challenge requires, the living machine's program-level representation should not be constrained by, or embedded with, handcrafted knowledge-level world models or system behaviors[1]. No manually built model or behavior is general enough to handle either the high complexity of the general real-world or the high complexity of the system behavior required by the Grand Challenge. Thus, those systems that are embedded with human handcrafted world models or programmed-in behaviors (e.g., the series of robots developed by Brooks [8], autonomous agents such as ALIVE [48] and artificial life systems such as artificial graphics fishes [78]) cannot go much beyond what has been modeled.

Low-level physical channels Certain low-level innate functionalities are built into the system, especially those that are innate in human, such as pain avoidance (which can be simulated by a negative feedback from a human teacher) and love for food (which can be simulated by a positive feedback from a human teacher). Thus, human can influence the behavior at this "physical" level. The physical channels also include override channels through which a human teacher can impose certain actions when needed (e.g., for simulating hand-in-hand teaching without using a compliant robot arm).

Reinforcement learning The machine must be able to learn the right thing according to feedback. It avoids actions that are associated with negative feedbacks and chooses actions that are associated with positive ones. Those feedbacks are received either directly from human teachers through the low-level physical feedback channel (mainly for early learning), or indirectly from the machine's association with what it has already learned (mainly for later learning).

Self-organizing High-level knowledge and behaviors are learned and practiced based on rich and vast build-up of low-level knowledge and behaviors. However, there is no static definition to decide which one is low level and which is high. It depends on how each individual living machine learns. Roughly speaking, if a piece is learned based on another piece that has already learned, the former is at a higher level than the latter. However, such a dependency is not always one way.

Autonomous learning During autonomous learning, the machine has a certain degree of autonomy in the learning environment, similar to the way humans learn at home or at school. When doing so, the machine autonomously interacts with the environment around it, including humans. Although a human teacher may show the machine how to do a job, and sometimes may even use a hand-in-hand demonstration, the human does not have a full control of all the aspects of the learning machine, e.g.,

[1] An example of knowledge-level model is a hand-coded program section that determines something like: "If there are two dark areas of a similar size on a horizontal line, they might be human eyes." An example of knowledge-level behavior is a hand-coded program section in a system that does something like: "if the distance sensed from the forward sonar is smaller than a certain number, turn away."

what to see, what to remember, what to do, or how to interact with human. The human may evaluate how well the living machine is doing. In contrast, the other end of the spectrum is spoon-fed learning, in which a human teacher completely controls what the learning system receives as input, what objective function to minimize, and what the expected output is. Both the autonomous and spoonfed learning may include supervised and unsupervised learning [24] [33].

Real-time For practical reasons, the learning machine must *learn while performing* in real-time, on-line, and learn incrementally and directly from its second-to-second activities, on a daily basis.

The PI calls this type of general-purpose autonomous learning machines "living machines" because they "live" in the human environment and interact with the environment (including humans) on a daily basis. The emphasis of the term is not "life", but rather, the daily autonomous learning activities that are associated with a biological living thing, especially human, such as playing, communicating with humans and learning to perform tasks. Such machines are fundamentally different from a nonliving regular machine, such as an automobile or a computer, since they do not operate autonomously.

1.3 Why Living Machines?

This is a question whose answer is very broad. Here, a brief description of the major reasons is given. The history will explain it better in the future.

1.3.1 Each modality must be learned

The cognitive knowledge that is required to communicate with humans in single or multiple modalities in a general setting is too vast in amount and too complicated in nature to be manually modeled adequately and manually spoon-fed sufficiently into a program. Probably few will question the fact that language is learned. Therefore, vision, the modality that many human individuals do not feel requires much learning (i.e., learned subconsciously), is appropriate to demonstrate the importance of learning. How complete is a child's vision system when he or she is born? In fact, as early as the late 19th century, German psychiatrist Paul Emil Flechsig had shown that certain regions of the brain, among them V1, have a mature appearance at birth, whereas other cortical areas including V2, V3, V4, and V5 regions, continue to develop, as though their maturation depended on the acquisition of experience [114]. A lot of studies have been done since then. It is known that learning plays a central role in the development of human's versatile visual capabilities and it takes place over a long period (e.g., Carey [12], Hubel [30], Anderson [4], Martinez & Kessner [32]). Human vision appears to be more a process of learning and recalling than one that relies on understanding of the physical processes of image formation and object-modeling (e.g., the "Thatcher's illusion" [79] and the overhead light source assumption in shape from shading [64]). Neurologist Oliver Sacks' report [68] indicated that a biologically healthy, adult human vision system that has not learned cannot function as we take for granted. A large amount of evidence seems to suggest that except for low-level processing such as edge detection, many middle- and high-level sensory and motor behaviors in humans are learned on a co-occurrent basis from very early days of childhood and they continuously improve through later learning. The biological brain is so much determined by learning that the normal biological visual cortex is reassigned to tactile sensory functions in the case of the blind. Many researchers in the field have realized that the visual knowledge required by the human-level performance is certainly too vast and too complex to be adequately modeled by hand. Letting a machine learn autonomously by itself is probably the only way to meet the Grand Challenge. In other words, we should move from "manual labor" to "automation." Intuitively, manual modeling is hard and costly, while automation is more effective in productivity and less costly than human labor. Furthermore, it is extremely difficult, if not impossible, to build an adult human brain that has learned. It appears more reasonable to build a machine "infant brain" that can simulate, to some degree, brain's learning after the birth.

1.3.2 The machine must sense and act

The question is then how to automate this learning process. In the field of traditional artificial intelligence (AI), the main emphasis of the establishment has been symbolic problem solving (see Minsky's collection of annotated bibliography [55]). Later, due to the need of common sense knowledge in reasoning systems, some grandiose projects have been launched to manually feed reasoning rules with symbolic, common sense knowledge via computer keyboards (e.g., the CYC project [45]). The hope is that these rules and common sense knowledge are complete enough to derive all the needed knowledge. Learning in traditional AI is conducted at a symbolic level, even when autonomy is a goal (e.g., the autonomous learning with the LIVE system [69]). The machine does not have its own sensors and effectors. This has two fundamental problems. First, the machine cannot deal with all the knowledge that is directly related to sensing and action, such as how to recognize a scene and move around it using vision. Second, since a huge amount of human's symbolic knowledge, both low level and high level, is rooted deeply in sensing and action, a sensor-free machine can neither really understand nor properly use it if it is input manually. For example, even for seemingly symbolic problems such as language translation, no reasonable translation is possible without understanding the meaning of what is said (a lot of which is about sensing and action), except for simple cases.

Brooks emphasized the need of embodiment for developing an intelligent machine [8]. In his view, an intelligent machine must have a body to be situated in the world to sense and act. He advocated that intelligence emerges from robot's interaction with the world and from sometimes indirect interactions between its components [8]. A recent book by Hendriks-Jansen provides perspectives for embodiment and situatedness from psychology, ethology, philosophy and artificial intelligence [28]. In fact, embodiment, situatedness, sensing, and action have been common practices in robotics and vision communities for many years [85] [80]. The emphasis on these important points is not only useful for symbolic AI, but also for other fields related to machine intelligence.

1.3.3 The machine's representation must be knowledge-free

The programming-level representation must be free of handcrafted rules. A manually selected set of features cannot be applicable to open-ended learning. Aloimonos [2] and others advocated that modeling 3-D scenes is not always a reasonable thing to do. Each vision problem should be investigated according to the purpose. Brooks 1991 [8] attributed the difficulties in vision and mobile robots to the so called SMPA (*sense-model-plan-act*) framework which started in the late 60's (see Nilsson's account [57] for a collection of the original reports). However, the behavior-based methods of Aloimonos [2] and Brooks [8] avoid a trap but accept another — imposition of handcrafted behaviors on the system. A more recent work of Brooks' group used an explicit, qualitative representation of the sonar-sensed world with a set of explicitly programmed-in behaviors [50]. In fact, the pattern recognition community and the machine learning community have had a long history of recognizing patterns without fully modeling it. Features have been used in these fields to classify patterns (e.g., [24], [23], [33] [61], [7], [53]). The basic difference between a pattern recognition problem and a typical computer vision problem is that the former has a controlled domain with a limited number of classes but the latter is open-ended. The fundamental problem of the existing SMPA approaches is not in reconstructing the world, but rather, it is the *practice of handcrafting knowledge-level rules for one of both enties: the world and the system.* Avoiding modeling 3-D surface or abandoning the SMPA framework is not enough. For open-ended applications with unpredictable inputs, as is the case with the Grand Challenge, handcrafted behaviors embedded into a programming representation must be abandoned because no handcrafted behavior is generally applicable to the open world. Brooks is currently building a human-shaped machine called COG [9]. Facing with tremendous difficulties in handcrafting a huge number of behaviors and world models, his current architecture does not seem to be able to meet the Grand Challenges in vision, speech, language, etc, where the sensing dimensionality and the complexity of the understanding task are much higher than that of sonar sensors [50]. As we explained in Section 1.3.1, handcrafted knowledge models or system behaviors cannot deal with the full complexity of vision, speech, language, etc.

Therefore, the system representation for programming should avoid modeling the knowledge level. Consequently, self-organization is a must, to allow the system to acquire knowledge and behavior at different

levels of abstraction automatically. This point is closely related to our *comprehensive learning* concept, which is explained in Section 2.1.

1.3.4 The machine must learn autonomously

Spoon-fed learning is not practical for the Grand Challenge because (1) a huge amount of knowledge and behavior must be learned, (2) the system must experience an astronomical number of instances, (3) the system must learn continuously while performing (no human being can practically handle this type of spoon-feeding work on a daily basis), (4) high-level decisions are based on so much contextual information that only the machine itself can handle (automatically). During autonomous learning, a human teacher serves very much like a baby sitter (robot sitter in this case), sending occasional feedback signals depending on how the living machine is doing. Later on, once the robot has learned basic communication skills through normal communication channels (such as speech and visual gesture), the robot sitter is replaced by school teachers. It appears that only the relatively low cost associated with autonomous learning is practical for meeting the Grand Challenge.

1.3.5 The machine must perform multimodal learning in order to understand

Studies on humans who are born blind and deaf have demonstrated tremendous difficulties in learning very basic knowledge [112] [51]. Learning basic skills become virtually impossible with those few who are born blind, deaf, and without arms and legs. For example, a system that cannot see cannot really understand concepts related to vision (e.g., pictures and video, film, color, mirror, etc) and those concepts that are understood mainly from visual sensing (e.g., trees, mountains, birds, streets, signs, facial expressions, etc). A system that cannot hear is not able to really understand sentences related to speech and sound (e.g., the sound of music instruments, bird chirps, characteristics of a person's voice). The proverb, "a picture is worth a thousand words" vividly points out the deficiency of text (words) in describing information that are best conveyed visually. Furthermore, understanding any single sensing modality, including vision, speech, language, and text, requires knowledge about other sensing modalities too. A system that does not live and interact with humans cannot really understand the concepts related to human emotions, characteristics and relationships (e.g., angry, happy, sympathy, care, cruel, friends, colleagues, enemy, spies, etc). In fact, a system that cannot see, cannot hear, cannot touch is deprived of the three most important sensing modalities through which a human acquires knowledge. Therefore, such a system has a fundamental limit in understanding any human knowledge and in using such a knowledge even if it is manually fed in. Sensor-free systems like CYC have met tremendous difficulties toward applications that require understanding (such as language translation) and they are also very difficult to use due to the lack of any sensing modality for retrieval. Lack of multimodel sensing and action is a major reason to account for why existing knowledge-base systems do not really understand the knowledge they store.

1.4 The Three-Phase Plan

The task of developing living machines consists of two integral aspects:

1. Developing the physical system, including theory, algorithm, hardware and software, for autonomous multimodal learning. In some sense, our goal is to build a machine counterpart of a human new born, although an exact duplicate is neither possible nor necessary.

2. Teaching the living machines to do things. In a sense, human beings try to "raise" and teach the machine "babies" properly so that every one of them will become successful in an assigned professional field.

The PI has been following a three-phase plan for this endeavor.

1.4.1 Phase 1: Comprehensive learning

In Phase 1, the task is to develop a framework for basic brain functionalities such as memory store, automatic feature derivation[2], self-organization, and fast associative recall. The major goal is *generality and scalability*. The generality means that the framework must be applicable to various domains of sensor-effector tasks. The scalability means that it must have a very low time complexity[3] to allow real-time learning and performance (i.e., scalable to number of learned cases). A wide variety of sensing and action tasks must be performed to verify the generality and scalability. However, learning at this stage is "spoon-fed", meaning that the learning process is not autonomous.

1.4.2 Phase 2: Autonomous Learning by living machines

This phase is to complete the theory and methodology development for autonomous general-purpose learning and build one or more prototypes of living machines. The living machines are trained to perform certain tasks autonomously, such as moving around, grabbing things, saying simple words, and responding to spoken words, all in a fully autonomous mode and in unrestricted general settings. This roughly corresponds to the *sensorimotor* stage of a human child (from birth to age 2), according to the renowned developmental psychologist Jean Piaget [26] [13] [12]. In his theory, human's cognitive development can be roughly divided into four major stages, as summarized in Table 1. There is no doubt that these four stages have a lot to do

Table 1: Piaget's Four Stages in Human Cognitive Development

Stage	Rough ages	Characteristics
Sensorimotor	Birth to age 2	Not capable of symbolic representation
Preoperational	Age 2 to 6	Egocentric, unable to distingiush appearance from reality; incapable of certain types of logical inference
Concrete operational	Age 6 to 12	Capable of the logic of classification and linear ordering
Formal operational	Age 12 & beyond	Capable of formal, deductive, logic reasoning

with neural development in the brain. Furthermore, more recent studies have demonstrated that the progress into each stage depends very much on the learning experience of each individual and thus, biological age is not an absolute measure for cognitive stages. For example, Bryabt and Trabasso [11] showed that given enough drill with the premises, 3- and 4-year old children could do some tasks to construct linear orderings, a deviation from the classical Piagetian theory. During our development, the theory and methodology for developing living machines will be modified depending on how well the living machine can learn. In this phase, communications between human teachers and the living machines during training are partially visual, partially vocal, and partially physical (through the low-level physical channel).

Table 2 lists five integrated task groups as the benchmarks for Phase 2. The current *status quo* in the field is (1) no existing system can complete all these task groups in a controlled setting; (2) no existing system can do any one of the task groups in a general setting; (3) no existing system can do any one of the task groups in a truly autonomous mode (i.e., the system cannot be explicitly preprogrammed at the task level), when instructed via gesture or speech. These tasks to be performed by the living machines must be taught in the autonomous learning mode. The benchmarks must be tested continuously, in a truly autonomous mode and in any order. If the benchmark test is successful, this will be the first time that a machine really

[2] We do not use the term feature selection here because it means to select from several pre-determined feature types, such as edge or area. The term feature extraction has been used for computation of selected feature type from a given image. Feature derivation means automatic derivation of the actual features (e.g., eigenfeatures) to be used based on learning samples.

[3] For the living machines, the time complexity is logarithmic in the number of cases learned. Thus, the exact term should be "logarithmic scalability" — logarithmic to the scale of the problem.

Table 2: Benchmarks of Phase 2 for the Living Machines in *General Settings*

Task group	Benchmark
Visual recognition	Say hello with correct names of 5 human teachers when they enter the scene. Say the name of 5 toys when being asked.
Speech recognition	Understand sentences: Come. Call me. Hello! Goodbye! Wave your hand. Say hello to .... Say goodbye. What's it? Pick this. Put down. Pour water into this. Yes. No. Follow me. Watch this. Stop. Go home. I'm home. I got lost.
Speech synthesis	Respond using sentences: Hello! Goodbye! Yes. No. I'm home. I got lost. Call the names of 5 toys and 5 teachers.
Navigation	Autonomous indoor navigation without running into anything. Outdoor navigation following campus walkways and crossing streets. Follow a teacher to go, via an elevator, from a lab on the third-floor of a building to the parking lot outside the building. Return from the parking lot to the lab alone.
Hand action	Pick correctly one of the 5 toys. Put a toy down. Wave its hand when saying hello or goodbye. Place one toy on top of another. Pour a cup of water into another cup.

understands something[4]. By the time it has passed the test, the living machine actually has learned much more because the setting is unrestricted — much more has been seen, much more has been heard, much more has been tried, and much more has been learned.

1.4.3 Phase 3: Living machine's pre-school learning

At this stage, the living machine enters Piaget's preoperational stage (age 2 to 6). Significant improvements of the living machine software will continue, similar to the way operating systems are improved and upgraded now. Computers move to new levels of storage and speed and their cost continues to fall. The new demands from living machines will stimulate the robotics industry to produce new generations of light weight, reliable, dexterous manipulators and drive systems.

A new emphasis in this phase is to investigate how to teach the living machines to learn things that are taught in human preschools. Since machine computes fast and is never tired of learning, potentially they can learn faster than a human child. At this stage, communications between human teachers and the living machines become mostly visual or vocal, whichever is more convenient. The physical channel is seldom used. Breakthroughs in vision, speech recognition, speech synthesis, language understanding, robotics, intelligent control and artificial intelligence are simultaneous at this stage. The benchmark to measure success is a standard entrance test for human pre-schoolers.

At this time, a new industry will appear. Living machines are manufactured and delivered to research institutions as experimental machines; to federal agencies for special tasks, to schools as educational material, to amusement parks as interactive attractions, to the media industry as machine personalities, and to homes for those who are physically challenged or just need a friend. At the end of this phase, the bright future of living machine is well known to general public.

1.4.4 Brain size and speed of the living machine

A question is naturally raised here: how much space does the living machine need? How fast can it recall from a large brain? These two important questions cannot be clearly discussed until the methods are presented. See Section 4.2 for a discussion about the brain size and Section 4.3 for the speed issue. With the logarithmic time complexity of the living machine and the steady advance of computer storage technology, it is expected

[4] A link from one text string to another, or a mapping from a camera input to a label is probably not understanding, since the machine does not understand the text or label.

Table 3: Major Tasks Tested and the Demonstrated Functionalites

SHOSLIF subproject	SHOSLIF-O (recognition)	SHOSLIF-M (spatiotemporal)	SHOSLIF-N (mobile robot)	SHOSLIF-R (robot arm)	SHOSLIF-S (speech)
Spatial recognition	X	X	X	X	X
Temporal recognition		X	X	X	X
Image segmentation		X		X	
Prediction		X	X	X	
Visual attention		X		X	
Sensorimotor			X	X	
Incremental learning			X		X
On-line learning			X		X
Performance redefined the state-of-the-art	Yes	Yes	Yes	Yes	Not yet

that in a few years, real-time living machines will have a storage size comparable with that of the human brain at a reasonable cost, although in many applications we probably do not need as much space as the human brain.

1.5 The Technology is Ready Now for Phase 2

The PI's Phase 1 work toward the living machine and the resulting SHOSLIF system started in the Fall of 1992. Phase 1 addressed the two conflicting criteria — generality and scalability. The generality requires that we must avoid handcrafted models for the environment or handcrafted models for the system behavior, which is the essence of the *comprehensive learning* concept introduced by the PI in the Fall of 1993 [89] [93]. In the field, there have been many attempts to build autonomous robots, from experimental micro robots (e.g., see Brooks [8]), to software-based virtual autonomous agents (see a survey by Maes *et al.* [48]), to full-size robots (see a survey by Kanade *et al.* [36]). All of them ended up with a special-purpose system with an *ad hoc* solution, because they use handcrafted knowledge-level models or handcrafted behavior rules. However, the PI's SHOSLIF aims at a full generality and scalability. For this goal, it has been tested on a wide variety of projects as summarized in Table 3. The extent of the work in using a single unified framework for such a wide range of challenging tasks is unprecedented in the field. It indicates a significant *breakthrough* toward the living machines. SHOSLIF is the only work that has raised and successfully addressed the generality-and-scalability issue and thus enables us to embark on general-purpose living machines. Without the success in meeting the conflicting criteria of generality and scalability, the living machine is not possible.

In Phase 2, the first living machine SAIL (Self-organizing Autonomous Incremental Learner) is currently being constructed. The result from Phase 2 is expected to bring long-lasting pay-offs to a wide variety of applications, including human-machine multimodal interface systems, image understanding systems, speech recognition systems, language understanding systems, robotics systems, high-performance knowledge-base, expert systems, artificial intelligence, and future entertainment systems, educational systems, and intelligent personal assistant systems.

Although a lot of progress has been made toward autonomous robots and construction of knowledge-bases, general-purpose autonomous learning with real-time sensing and action in general settings has not been possible until now.

1. *Very few frameworks are truly general as SHOSLIF.* Traditional autonomous systems and knowledge-base systems rely on humans to manually model the world and handcraft decision rules for the system, which determines that the systems are not general, since no handcrafted models are general enough either for the world or for a general purpose sensing-and-action learning system. The behavior-based

Table 4: Comparison of Existing Approaches to Learning Machines

Approaches	Major representatives	For the world	For the system	Sensing and action integrated in learning
Symbolic AI	Minsky and others [55], CYC [45]	Hand-crafted knowledge models	Hand-crafted reasoning rules	Text output
Robotics	CMU Navlab [36], DANTE [109], Rap [62],	Hand-crafted world models	Hand-crafted decision rules	Various
Bahavior-based	Brooks [8], Aloimonos [2]	Avoid handcrafted model; respond with reflex	Hand-crafted low-level behaviors	Depth/motion sensing; navigation
Comprehensive learning	Weng [93] and SAIL being constructed	General learning, avoid handcrafted world model	Self-organize from low to high levels, avoid handcrafted-behavior	Vision, hearing, tactile; attention, navigation, manipulation, speaking

Table 5: Comparison of Several General Tools

Approaches	Completeness	Learning speed	Retrieval speed	Incremental growth
HMM	Low	Slow: iterative	$O(N^2)$	No
Neural networks	High	Slow: iterative	$O(N^2)$	Difficult
SHOSLIF	High	Fast: noniterative $O(\log(N))$	$O(\log(N))$	Yes

approaches have abandoned the practice of modeling the world due to the observed difficulties, but they got stuck with a deeper fundamental problem: handcrafted behaviors cannot handle open-ended general-purpose learning. Although they are situated, behavior-based robots are repeating the practice of simple-minded language conversation programs which respond human's questions with a few pre-programmed phrases and the fundamental differences among approaches. See Table 4 for a summary of existing approaches.

2. *The SHOSLIF is the only existing system that is both general and scalable.* It is difficult for a general system to be efficient and the system tends to be slow, because no manually imposed constraints are allowed. However, a slow system cannot be situated and learn in real-time, which is a major challenge to all the vision-based robot systems. SHOSLIF is both general and fast (logarithmically scalable). Artificial neural network is also general as a tool, but it is not scalable as shown in Table 5. A good test for scalability is to use vision, which requires a very high dimensionality in input and is the major sensing modality for humans to acquire knowledge. The scalability requires that the system does not slow down significantly violating the real-time on-line learning criterion even when the number of cases learned increases through time. The logarithmic time complexity of the SHOSLIF accomplishes this goal. Sonar-only systems, although they use low-dimensional inputs, have very fundamental limit in performing nontrivial cognitive tasks, as indicated in blind-deaf cases with humans [112] and more so with the born complete blind-deaf [51].

3. *SAIL is the first system for autonomous learning with multimodal sensing and action without imposing*

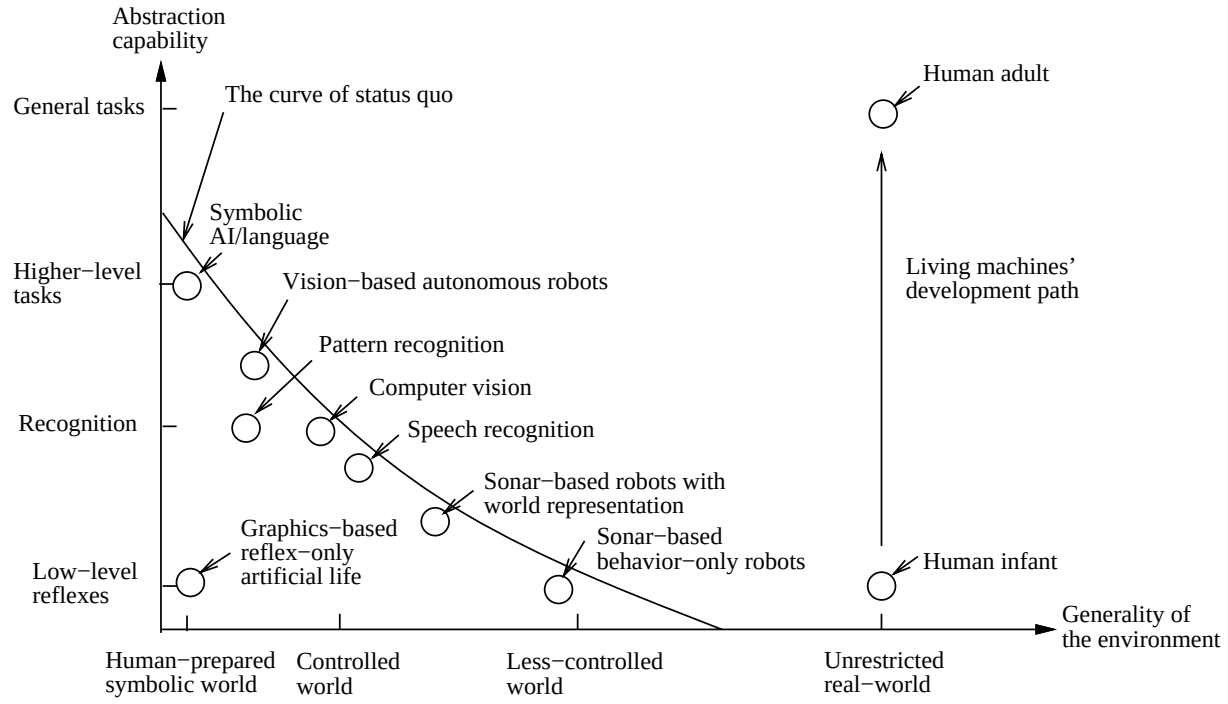


Figure 1: A schematic illustration of the *status quo* in human endeavor toward intelligent machines. The difficulty level is related to the required abstraction capability and the generality of the envrionment to which the system is exposed to. The upper right corner in the diagram is the goal. The mode of development in the past has been gradually pushing the front of the curve of *status quo* in the upper right direction. However, human will never reach the goal this way, due to the reasons explained earlier. The living machine approach outlined here has taken into account the lessons we learned from many related fields. It is a fundamentally different development path, which is in line with that of human cognitive development. The technology base for this strategic path is ready now.

> *handcrafted models for the world or the system.* Hand-crafting knowledge models and system behavior models is one of the major reasons for persisting difficulties toward a general-purpose multimodal understanding systems. The framework for SAIL on how to boot-strap a system that is free of both world-model and behavior-model[5] is unprecedented.

Fig. 1 illustrates the *status quo* of the existing works in various fields and the development path of the living machine. The proposed SAIL scheme, as presented in more detail later in this document, is a systematic way toward the Grand Challenge. It is natural that such a scheme was originated from a group having extensive experience in computer vision. Arguably, vision is the most difficult sensing modality in human.

1.6 The Future of the Living Machines

After completion of Phase 3 in the above plan, continued education for living machines is more or less like teaching a human child.

1.6.1 A revolutionary way of developing intelligent software

A large number of living machines will be made. Each newly made machine is loaded with the Phase 3 brain (software and network as a database) as the starting point. Computer experts work with educational experts

[5]Some low-level human behaviors are innate. With SAIL, the desirable behaviors will be taught to the system during the process of interactive hand-in-hand training, via the physical channel.

to teach professional skills to living machines, very much like the way human students are taught. Each professional field or subject has a living-machine school, in which several living-machine robots are taught to master the professional knowledge. What each school does is to train a generic living machine to become a professional expert. A living machine can be trained to become a space-craft pilot, a fighter plane pilot, a deep-sea diver, a waste-site cleaner, a security-zone monitor, an entertainer, a nursing-home care-taker, a tutor, or a personal assistant. Each living machine is a software factory. This represents a revolutionary way of making intelligent software: Hand programming is no longer a primary mode for developing intelligent software, but rather, school teaching is. Teaching and learning are carried out through visual, auditory, and tactile communications. Two types of products are sold, expert software brains and expert living machines.

1.6.2 Expert software brains as product

The expert brain (software and network) of each specially trained living machine is down-loaded and many copies are sold as software brain. The software-only brain is useful and cheap, but it does not have a full learning capability because of its loss of most sensors and effectors. For example, a software-only brain running on the Internet can only learn from those available from the Internet, based on its knowledge it learned when it had a full array of sensors and effectors. Depending on the richness of the information available from the Internet and how his owner instructs its learning from the Internet, this software brain's knowledge may gradually become obsolete. This is very similar to the case where a human's knowledge becomes obsolete, once he or she does not keep learning. However, software brain buyers can always get upgrades regularly from software brain makers, very much like the way commercial software gets upgraded now.

1.6.3 Expert living machines as product

Each living machine school also sells expert "brains" with a physical body as a full living machine with sensors and effectors. Such complete living machines can continue to learn at their application sites to adapt to the new environment and learn new skills. This type of full living machines are used to extend three types of human capabilities (1) physical capability, such as operating a machine or driving a car, (2) thinking capability, such as finding the most related literature from a library or replying piled-up electronic mails, (3) learning capability, such as learning to use computers for the handicapped or learning to predict economic ups and downs better. Humans can do things that they enjoy doing, leaving living machines to do other things. Further more, the living machines can work round-the-clock without fatigue and loss of concentration, which is something that no human being can match.

1.6.4 The longevity of living machine

The living machines eventually *outlive* human individuals because once their hardware is broken or worn, the brain can be down-loaded and then up-loaded to a new hardware. They can also learn faster and learn more hours everyday than each human individual. This longevity may lead to tremendous creativities. For example, Albert Einstein passed away and human lost his creativity. The creativity and the knowledge that each living machine learned can be preserved for future human generations for more innovations and better service.

1.6.5 Social issues with living machines

Just like the case with human children or pets, each human teacher is responsible for his or her living machine. If a living machine happens to have learned something that we do not want it to learn, we can always replace its brain with the backed-up copy of, say, yesterday, to make it back to the yesterday's status. We do not have such a convenience with a human child or a pet.

1.6.6 A huge new industry

Now, the automobile industry is huge because every household needs at least one automobile. The computer industry is huge now because the computer is general purpose in extending human's computation needs (and all the functionalities that accompany the computation). Unlike all the special-purpose robots that have been built so far, the living machine is general purpose. Therefore, the market for the living machine is huge. With the decreasing cost that accompanies advances in robotic technology and computer technology and the ever increasing volume of mass-market sale, the living machines will not only enter every business and industrial sector but also millions of American households to serve as intelligent personal assistants. Its annual sale number will be eventually on a par with that of personal computers and automobiles. With its strong industrial base and competitive environment, US will become the birth place and the first growing ground for the future living machine industry.

1.6.7 The government's role

The Internet has grown out of ARPANET whose seed was planted in 1969 by a DARPA supported project titled Resource Sharing Computer Networks. The funding from US Federal Government played an important role in the development of ARPANET and Internet. By the time when the last piece of NSF backbone ceased to work in the Internet on April 30, 1995, the Internet has become an outstanding example to show how the government can play its positive role in facilitating the growth of a new industry. Thanks to the popularity of the Internet, the number of computers sold annually in this country is on a par with that of automobiles sold, and the Internet has entered billions of American families. Its international coverage is growing very fast.

Timely Federal funding support is crucial for the new, upcoming living machine technology in its fledging stage. The development of the living machines is of great importance to this country's scientific and technical advances, future national security, and the future US economy in this ever more competitive world market.

2 WORK COMPLETED TOWARDS LIVING MACHINES

Currently, the PI and his students have completed Phase 1 and have already started Phase 2. In the following, the PI briefly summarizes the series of work that has been completed.

2.1 Comprehensive Learning

The major new concept introduced by the PI during Phase 1 was the concept *comprehensive learning* which the PI first presented in an NSF/ARPA sponsored workshop in 1994 [89] [93]. As illustrated in Fig. 2, this concept consists of two basic ideas:

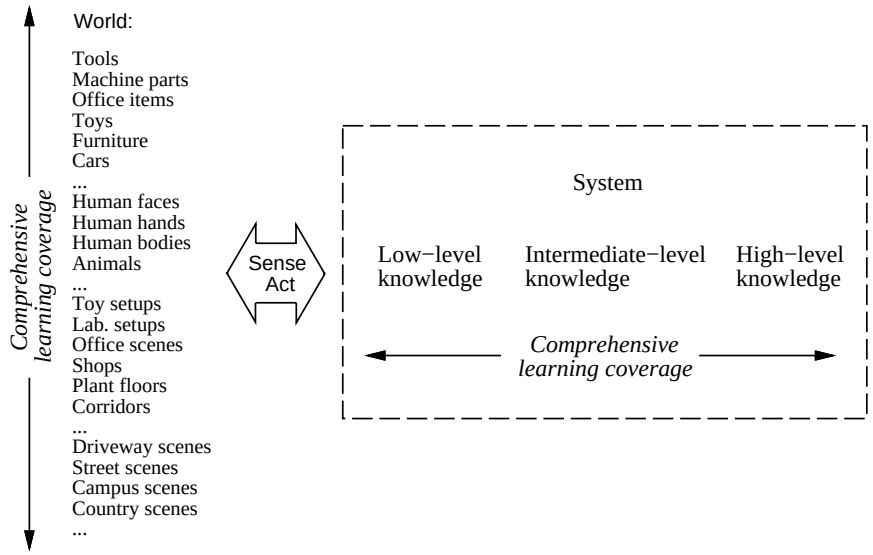


Figure 2: The concept of comprehensive learning implies that learning must comprehensively cover (1) the sensed world and (2) the understanding system. This implies that one needs to avoid handcrafted knowledge-level rules for the world or the system.

1. Learning must comprehensively cover the sensed world (visual, auditory, etc). This implies that we must not manually model the world. No manually built knowledge-level model is general enough for open-ended learning. Thus, the PI's approach is different from all the "knowledge-based" methods in the field of artificial intelligence. Further, the learning method should not assume which type of scene condition is acceptable by the system and which is not. The method must be able to learn from any scene sensed by the sensor, because there exists no automatic condition checker which can tell, given an arbitrary situation, whether the scene condition is acceptable to the method. A system is not able to operate autonomously in the real world, if it assumes conditions that it cannot verify by itself.

2. Learning must comprehensively cover the entire understanding system (visual, auditory, etc). This implies that we must not model system behavior. Therefore, the PI's approach is different from behavior-based methods. In other words, handcrafted knowledge-level rules (such as shape from shading rules, shape from contour rules, edge linking rules, collision-avoidance rules, planning rules, reasoning rules, etc) should be avoided for the programming level as much as possible. These rules result in brittle systems due to their very limited applicability and lack of automatic methods to determine their applicability in the open-ended real world environment.

Two basic entities are involved in the scenario: the environment and the machine agent. The comprehensive learning points out that the human system designer should neither handcraft models which restrict the

environment nor handcraft behaviors which restrict the machine agent. We should only embed the system with self-organizing principles that the system can use to organize sensed information. These principles do not penalize the generality of the system, but only affect the efficiency of the system. Violation of this basic principle of comprehensive learning is the primary reason why there has been no general-purpose living machine so far.

The first comprehensive coverage implies that the comprehensive learning concept is in sharp contrast with the current mainstream approaches in the computer vision and robotics field, where each method requires certain assumptions about the world condition and then a model or rule is derived about the environment to compute a solution using the assumptions. Shape-model-based methods for object recognition, all the SMPA approaches in robotics, and the recent model-based trend [21] in the multimedia community (which repeats the deep-trenched, not very successful approaches in the computer vision community) all belong to this category of modeling-environment approaches. There is no assumption verifier which, given any image, verifies whether the assumptions are satisfied. In fact, such a verifier may be more difficult to develop than solving the original application problem itself[6]. In summary, for understanding an uncontrolled environment, we should not impose, at the programming level, handcrafted models or restrictions on the environment.

The second comprehensive coverage implies that the concept is in sharp contrast with the behavior-based approaches which impose models on the system behavior. Although behavior-based approaches, such as the work of Aloimonos 1990 [2] and Brooks [8], have avoided explicit reconstruction of the scene, they still impose handcrafted rules about the behavior of the system. Although the imposed behavior in a simple situation may turned out to be close to what one needs, such a handcrafted behavior is not applicable to billions of more complex situations in the real world. For example, collision avoidance is needed only in navigation, but is damaging in docking. Hand-crafted rules will never be enough to handle the complex situations that a living machine must handle during its open-ended learning. In summary, for understanding an uncontrolled environment, we should not impose handcrafted behavior on the system at the programming level.

2.2 Generality and Scalability

The generality (i.e., the real-world applicability) is resulted from the comprehensive learning approach, as stated above. The remaining issue is the efficiency. The scalability means that the method must work in real time even when the system has learned a huge number of cases. Generality and scalability are two conflicting goals. A general method does not use pre-imposed special-purpose constraints and thus tends to be less efficient. However, we must achieve both generality and scalability. ' In Phase 1, we applied SHOSLIF to a wide variety of tasks to test its generality and we used a large number of cases for each task to test its scalability.

2.3 SHOSLIF Developed

The basic framework that we developed to achieve the above two conflicting goals is called *S*elf-organizing *H*ierarchical *O*ptimal *S*ubspace *L*earning and *I*nference *F*ramework (SHOSLIF).

2.3.1 The Core-Shell Structure

A level of SHOSLIF consists of a core and a shell, as shown in Fig. 3. The core is task independent. It serves the basic function of memory storage, recall and inference. The shell is task dependent. It is an interface between the generic core and the actual sensors and effectors. For a particular problem with a particular set of sensors and effectors, a shell needs to be designed which converts input data into a vector in space S, to be dealt with by the core. The output of the core is fed into the corresponding effectors by the shell. Mathematically, the SHOSLIF core approximates a high dimensional function $f : S \mapsto C$ that maps from

[6] It is reasonable to assume conditions that are really true in a particular application. However, few conditions are both true and useful in typical multimedia applications.

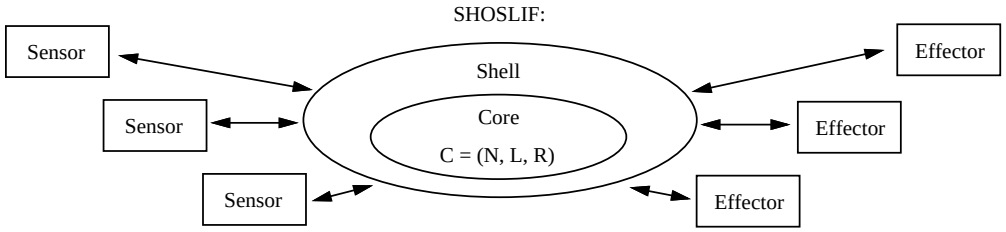


Figure 3: The SHOSLIF's core and shell, with sensors and effectors. Such a core-shell structure can be nested.

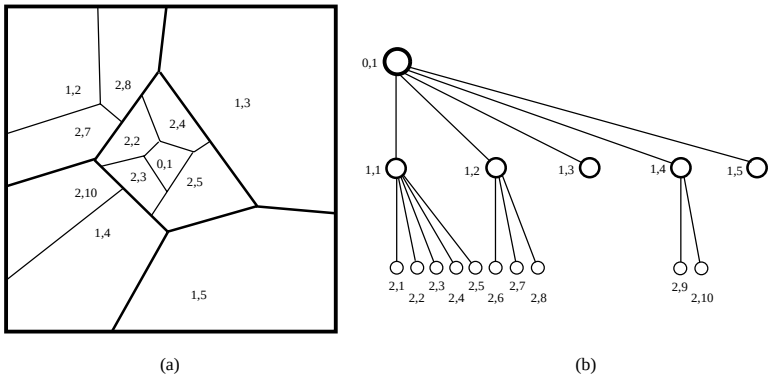


Figure 4: (a) samples in the input space marked as i, j where i is the level at which its position marks as the center of the partition cell and j is the index among the brothers in the SHOSLIF tree in (b). The leaves of the tree represent the finest partition of the space. All the samples in each leaf belong to the same class. A class is typically represented by more than one leaf. Linear boundary segments (i.e., corresponding to linear features) at the finest level are sufficient becase any smooth shape can be approximated to a desired accuracy by piecewise linear boundaries.

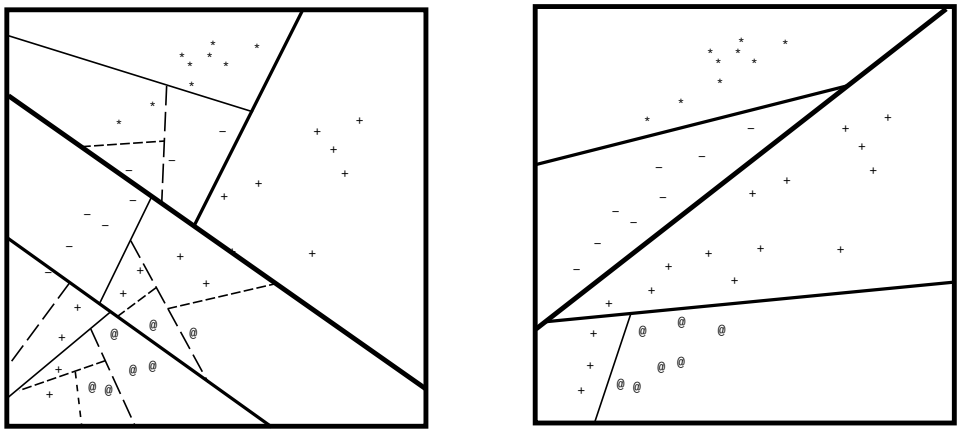
sensor input space S to the desired output space C. This very general representation is the key to enable the core to be problem independent.

2.3.2 The SHOSLIF tree

SHOSLIF is a framework for automatically building a tree that is used to approximate the function $\mathbf{Y} = f(\mathbf{X})$ as indicated by each learning sample $L_i = (\mathbf{X}_i, \mathbf{Y}_i)$, with $\mathbf{Y}_i = f(\mathbf{X}_i)$, in the learning data set. Fig. 4 shows a hierarchical space partition in the input space of $\mathbf{X}$ and its corresponding SHOSLIF tree. The SHOSLIF tree shares many common characteristics with the well known tree classifiers and the regression trees in the mathematics community [7], the hierarchical clustering techniques in the pattern recognition community [23] [33] and the decision trees or induction trees in the machine learning community [61]. The major differences between the SHOSLIF tree and those traditional trees are:

1. The SHOSLIF automatically derives features directly from training images, while all the traditional trees work on a human pre-selected set of features. This point is very crucial for the completeness of our representation.

2. The traditional trees either (a) at each internal node, search for a partition of the corresponding samples to minimize a cost function (e.g., ID3 [61] and clustering trees [33]), or (b) simply select one of the remaining unused features as the splitter (e.g., the k-d tree). Option (a) results in an exponential complexity that is way too computationally expensive for learning from high-dimensional input like images. Option (b) implies selecting each pixel as a feature, which simply does not work



(a) (b)

Figure 5: (a) Hierarchical partition of the MEF binary tree. (b) Hierarchical partition of the MDF binary tree, which corresponds to a smaller tree. The symbols of the same type indicate the samples of the same class.

for image inputs (in the statistics literature, it generates what is called a dishonest tree [7]). The SHOSLIF directly computes the most discriminating features (MDF), using the Fisher's multi-class, multi-dimensional linear discriminant analysis [27] [110] [24], for recursive space partitioning at each internal node.

2.3.3 Automatic derivation of features

In each internal node of the SHOSLIF tree, one or several feature vectors are automatically derived to further partition the training samples. The MDF subspace is such that in that subspace, the ratio of *the between-class scatter* over *the within-class scatter* is maximized. Computationally, the MDF vectors are the eigenvector of $W^{-1}B$ associated with the largest eigenvalues, where W and B are the within- and between-class scatter matrices, respectively. In the case where class information is not available, SHOSLIF uses PCA (principal component analysis) [35] [38] [84] to compute the principal components of the sample population (which we call the most expressive features MEF in order to bring up a contrast with the MDF).

How does the system know the class label? In fact, the MDF is also suited for autonomous learning where, although the exact class is not known, the low-level physical feedback is given which enables a two-class discrimination at each internal node to find desired action and avoid undesired ones. At later autonomous learning, class information will be available from the living machine itself.

In the training phase, as soon as all the samples that come to a node belong to a single class, the node becomes a leaf node. Fig. 5 shows an example of SHOSLIF binary tree, for which only one feature vector is computed, resulting in a binary tree, which is very fast in retrieval since only one project needs to be computed as each internal node. As shown, the MDF gives a much smaller tree than the MEF since it can find good directions to separate classes. In reality, we explore $k > 1$ paths down the tree to get top k matches. Then the confidence is estimated from a distance-based confidence interpolation scheme using the top k matches.

2.4 Some Proved Theoretical Results for SHOSLIF

Here we briefly describe some new theoretical results that we have established and proved for SHOSLIF. We avoid a lot of mathematical equations here. In the following, the first two properties address the *generality* and the later two properties deal with the *scalability*.

Point to point correct convergence: Loosely put, as long as an image $\mathbf{X}$ has a positive probability to occur, the approximating function $\hat{f}$ represented by the SHOSLIF tree approaches the correct $f(\mathbf{X})$ as the number of learning samples increases without bound.

Functionwise correct convergence: Loosely put, if the training samples are drawn according to the real application and the function to be approximated has a bounded derivative, then the approximate function $\hat{f}$ represented by the SHOSLIF tree approaches the correct f *in the mean square sense* [7] as the number of learning samples increases without bound. The above condition does not mean that the samples must be uniformly drawn from all the possible cases, but rather, it means that one just takes the samples roughly in the way the system is used in the actual application. This theoretical result means that the system will never get stuck into a local minima and thus fails to approach the function wanted. This is a property that the artificial neural networks lack.

Rate of convergence: Let R_n be the error risk of the SHOSLIF tree and R^* be the corresponding Bayes risk (which is the smallest possible risk based on a given training set). We have proved the following upper bound on R_n:

$$R_n \leq 2R^* + A(2\beta)^2 k^{2/d} (\frac{1}{n})^{2/d} \tag{1}$$

where A is the upper bound on the Jacobian of the function f to be approximated, β is the radius of the bounded domain in which f is to be approximated, k is the number of neighbors used for interpolation employed in SHOSLIF, d is the dimensionality of the feature space, and n is the number of learning samples. This result is consistent with the intuition that k-sample based interpolation is useful only for a smooth function but it slows down the approximation when the function surface is rough. When n goes to infinity, the inequality gives $\lim_{n\to\infty} R_n \leq 2R^*$, which is a well known result proved by Cover and Hart [16] for the classification problem and later extended to function approximation by Cover [15]. This result gives a theoretical foundation for using the k-nearest neighbor rule since its resulting error rate is not too far from that of the best possible Bayes estimator. The Bayes estimator is impractical in our case because we do not know the actual distribution function and estimation of the distribution function in a high dimensional space is computationally very expensive, even if we impose some artificial distribution models.

Logarithmic complexity: The time complexity for retrieval from the recursive partition tree used by SHOSLIF is $O(\log(n))$, where n is the number of samples stored as leaf nodes in the tree, which is typically smaller than the size of the training set. This result is true not only for a balanced SHOSLIF tree (guaranteed by a binary tree version of the SHOSLIF), but also true for *Bounded Unbalanced Tree* typically generated by a general version of the SHOSLIF.

As we know, the above theoretical results gave only some insights into the nature of the problem. Evaluating actual error rates, some of them were quoted in this proposal, is a more practical way for evaluating actual algorithms.

2.5 Functionalities Tested for the SHOSLIF

The demonstration of generality and scalability is an unprecedented challenging task. It requires us to test SHOSLIF method for a wide array of problems. We selected several domains of challenging vision tasks to test SHOSLIF theory and performance. The selection of the vision tasks was determined in such a way that they cover major functionalities that must be implemented in Phase 1. Since the living machine requires also speech recognition capability, we have also selected the speech recognition domain. Table 3 lists the representative tasks that we have selected as test domains for SHOSLIF and the related functionalities that have been tested if applicable.

[7] Also in probability 1 which is stronger than the mean square convergence.

The SHOSLIF project is unique in that it has applied a new, unified method to a wide variety of very challenging vision/robotics problems (shown in Table 3) and the performance achieved has redefined the state-of-the-art in each of the vision/robotics problems.

Two other widely used general tools are the Hidden Markov Model (HMM) and the artificial neural networks [47] [58]. Table 5 gives a comparison. In the table, N roughly corresponds to the number of classes that need to be recognized. The completeness of HMM is low because of its three assumptions [63] (1) independence of successive observations, (2) the mixture of Gaussian or autoregressive probability density, (3) the Markov assumption: the probability for a particular event to occur at time t depends only on the event at time $t-1$.

In the following, a summary of each SHOSLIF subproject is presented. Due to the unified core-shell structure of SHOSLIF, the programming for each subproject was systematic, virtually without any parameter tuning. Basically an interface needs to be developed as a shell for each subproject.

2.6 SHOSLIF-O: Face and Object Recognition

This project is to use the SHOSLIF method to recognize a large number of objects from their appearance. The SHOSLIF approach is different from other conventional approaches to recognition in that it deals with real-world images without imposing shape rules or shape models on the scene environment. But rather, it uses a general learning method to make the system learn how to recognize a large number of objects under complex variations. It copes with critical issues associated with such a challenging task, including automatic feature derivation, automatic visual information self-organization, generalization for object shape variation (including size, position and orientation), decision optimality, representation efficiency, and efficient indexing into a large database.

Now, the proposed method has been fully implemented. In order to test the system using a database that is as large as possible, we combined several different databases for training and testing: (1) MSU face database (38 individuals); (2) FERET face database (303 individuals); (3) MIT face database (16 individuals); (4) Weizmman face database (29 individuals); (5) MSU general object database (526 classes). Fig. 6 shows some examples of face and object images used for recognition. Table 6 summarizes the result obtained. The training images were drawn at random from the pool of available images, with the remaining images serving as a disjoint set of test images.

In order to deal with variation in the well-framed (foveal) images, the system was trained to handle variation in size, position and 3-D orientation within a certain range. We trained the system using samples generated from the original training samples to randomly vary in (a) 30% of size, (b) positional shift of 20% of size; (c) 3D face orientation by about 45 degrees and testing with 22.5 degrees. The training and test data sizes are similar to that in Table 6. The top 1 and top 10 correct recognition rates were, respectively, (a) 93.3% and 98.9%, (b) 93.1% and 96.6%, (c) 78.9% and 89.4%.

Publications resulting from the NSF award include general theory and framework of SHOSLIF [93] [92] [89] [90] [91], work on recognition of human faces and other objects [77] [76] [73], applications of the method to content-based image database retrieval [74] [72] [75], and searching of human faces in crowds [71].

The closest case for both face and general object recognition is its predecessor Cresceptron by Weng *et al.*[98], but it has been tested with only about 20 classes. The SHOSLIF-O system is the only face recognition

Table 6: Experimental Results for Face and Object Recognition from Well-Framed Views

Type	Training set	Training classes	Test set	Top 1 correct	Top 15 correct
Face	1042 images	384 individuals	246 images	95.5%	97.6%
General	1316 images	526 classes	298 images	95.0%	99.0%

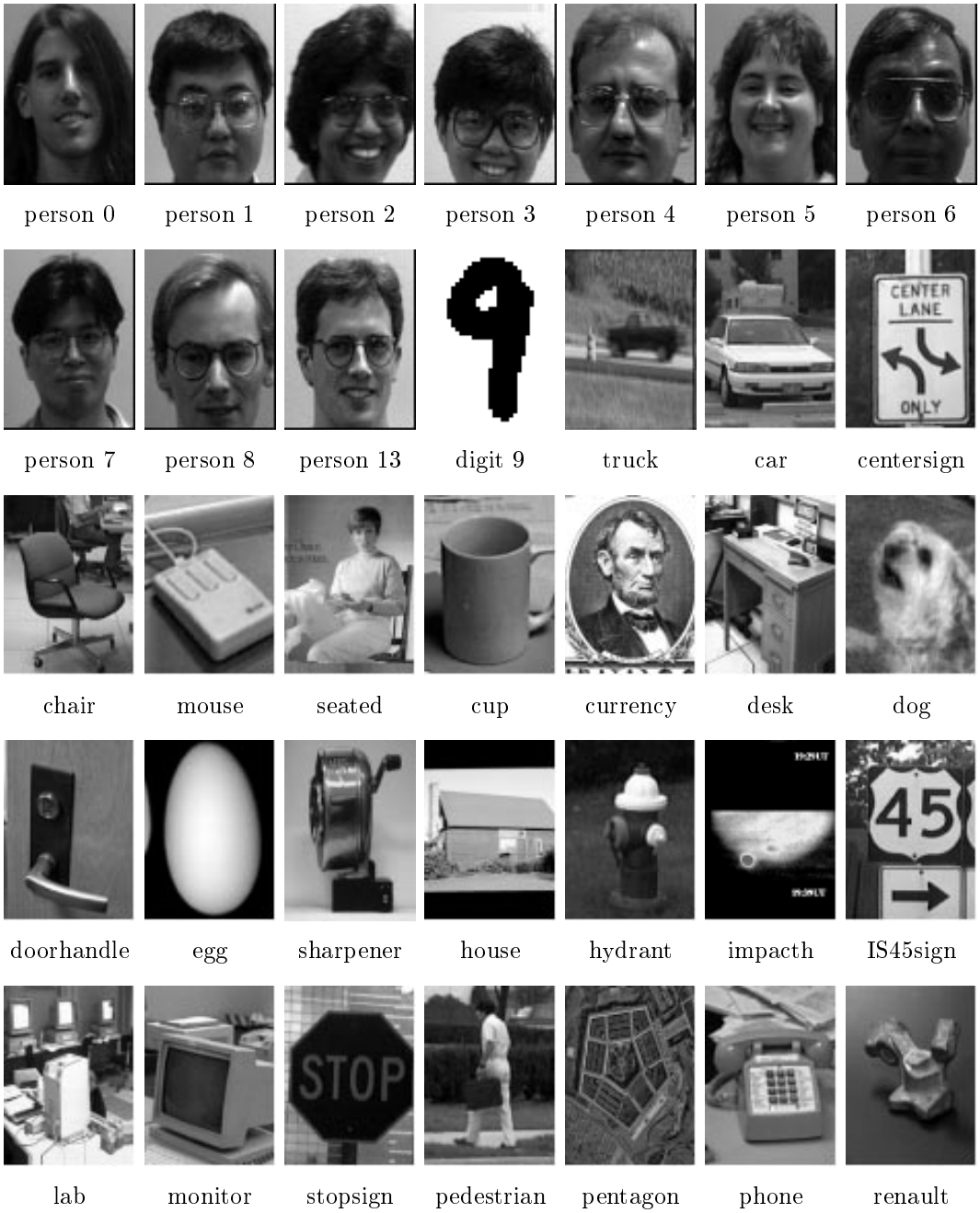


Figure 6: Some examples of face and object images used for recognition.

Table 7: A List of Works for Vision-Based Moving Hand Sign Recognition

Researcher group	Sign type	Handwear	Background	Method	Sign retrieval complexity	No. of sign classes
Darrell & Pentland 93 [20]	ASL words	None	Uniform black	Normalized correlation	$O(n)$	2
Cui-Swets-Weng 95 [18]	ASL words	None	Sparce texture	PCA, MDF, Tree (SHOSLIF)	$O(\log(n))$	28
Bobick & Wilson 95 [6]	ASL word	None	Uniform black	PCA and PCA trajectory	Not addressed	1
Starner & Pentland 95 [70]	ASL words	Color gloves	Without glove colors	HMM, hand as ellipse	Not addressed	40
Lanitis & *et al.* 95 [44]	Special shapes	None	Not mentioned	Hand tracking with parameters	$O(n)$	5
Kjeldsen & Kender 95 [40]	Special shapes	None	No skin tone	Color histogram and thresholding	$O(n)$	5
Triesch & Malsburg [82]	Speical shapes	None	Moderately complex	Labeled graphs	$O(n)$	10
Cui & Weng 96 [19]	ASL words	None	Arbitrarily complex	SHOSLIF with visual attention	$O(\log(n))$	28

algorithm that uses a systematic tree structure. It is the *only* system that has been extensively tested for *both* face recognition and object recognition.

2.7 SHOSLIF-M: Motion Event Recognition

For spatiotemporal recognition, we selected a challenge task: recognition of hand signs from American Sign Language. Recently, there has been a significant amount of research on vision-based hand-sign recognition from images. Table 7 summarizes some major studies. Most of the works listed in the table reported over 90% accuracy, comparable to our 93%. A direct performance comparison is not appropriate because the difference in the training and testing data and the different numbers of classes that have been tested.

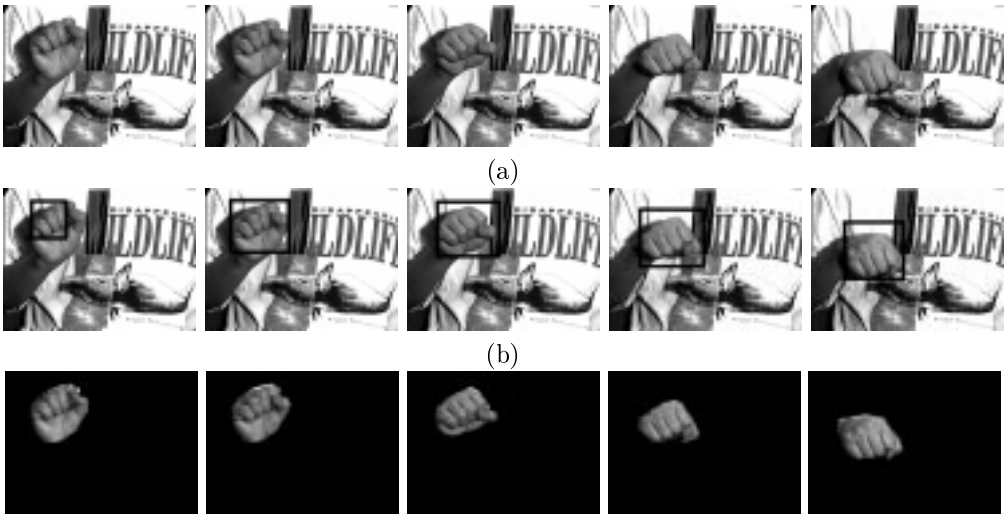
Compared with the existing studies on hand-sign recognition from image sequences, our work [18] [19] is unique in the following sense:

1. The capability to segment a detailed hand (a complex articulated object) from a very complex background as shown in Fig. 7. The Work of Weng *at al.* is the only one whose recognition result is completely independent of the background. (Malsburg *at al.* [82] requires that the background covered by local views do not affect the local matching significantly.) With this capability, we can significantly reduce the constraint on what kind of clothes that the signer can wear.

2. The logarithmic sign-retrieval time complexity $O(\log(n))$, as indicated in Table 7, such a low complexity is very desirable for dealing with a much larger number of hand shapes in real time.

3. The system distinguishes a large number (over 140) of hand shape classes, the largest among the existing works on static hand shape recognition from images (a list can be found from Huang and Pavlovic's recent survey [29] and the workshop proceedings in which that survey was published).

4. The largest number of hand-signs among the existing works on *handwear-free* moving hand-sign recognition (see Table 7 and [29]).



(a)

(b)

(c)

Figure 7: (a) An example of many hand-sign sequences. It means "yes". (b) The results of motion-based attention mask found, shown with a bounding (dark) rectangular window. Notice that motion-based segmentation alone is not sufficient for hand sign recognition. Without a detailed shape information of the hand, reliable hand sign recognition is not possible with a large number of hand-sign classes. (c) The result of final segmentation is shown with the background automatically masked off. Such a detailed and accurate segmentation is crucial to the success of hand sign recognition with a large number of classes and vice versa.

Figure 8: The mobile robot running SHOSLIF navigates autonomously at a walking speed, along hallways, turning at corners and passing through a hallway door. The real-time, on-line, incremental learning and the real-time performance is accomplished by an on-board Sun SPARC-1 workstation and a SunVideo image digitizer, without any other special-purpose image processing hardware.

2.8 SHOSLIF-N: Autonomous Navigation

This is to accomplish vision-based control. SHOSLIF-N is the only autonomous navigation system that is able to perform on-line, incremental, real-time learning. It uses the incremental version of the SHOSLIF core [99]. It is a good example to explain how the living machine may achieve real-time on-line learning without need of special image processing hardware, thanks to the extremely low (logarithmic) time complexity of the SHOSLIF.

During the learning, using a joystick, the human teacher controls the robot on-line to navigate along the desired path by updating the control signals in terms of speed and heading direction. The system digitizes the current image frame and links it with the current control signal to form a training sample, which is used to train the SHOSLIF tree. To construct a lean, condensed tree without overlearning, the SHOSLIF tree rejects a training sample when the current tree outputs a control signal vector that is the same (according to the accuracy required) as the human's control signal vector, without learning the current training sample. When almost all the recent training samples are rejected, the system is almost fully trained and ready to perform. The on-line learning with both tree retrieval and update runs at 5 Hz. The real-time performance, running at 7 Hz, is accomplished by an on-board Sun SPARC-1 workstation and a SunVideo image digitizer,

without any other special-purpose image processing hardware. The trained system successfully navigated along the hallways of our Engineering Building, making turns and going through hallway doors. It was not confused by passers-by in the hallway because it looks at the entire image and uses the most useful features (MEF or MDF), instead of tracking floor edges which can be easily occluded by human traffic. The comparison shown in Fig. 8 indicates that it is the state of the art.

Table 8: Several Autonomous Navigation Systems

System	Method	Scene tested	Feature	Learning	real-time on-line learning
Dickmanns [22]	edge following	outdoor driveway	Hand-select	No.	No.
CMU Navlab [81]	road model	outdoor driveway	Hand-select	No.	No.
Martin Marida ALV [85]	road model	outdoor driveway	Hand-select	No.	No.
Meng & Kak [52]	edge finding	indoor hallway	Hand-select	Partial	No.
CMU ALVINN [59]	MLP network	outdoor driveway	Auto-derive	Yes.	No.
L. Davis *et al.* [65]	RBF network	outdoor driveway	Auto-derive	Yes.	No.
MSU Weng et. al. [99]	SHOSLIF	indoor hallways & outdoor walkways	Auto-derive	Yes.	Yes.

2.8.1 The effect of SHOSLIF tree

To indicate the scalability performance of the SHOSLIF and the effect of the hierarchical tree, Fig. 9 shows the computer times for tree and flat versions. A total of 2850 images from various hallway sections in the Engineering Building of MSU were used for training. From the table, it is clear that the use of MEF tree can greatly speed up the retrieval and that real-time navigation can be achieved. The computation times were recorded on a SUN SPARC-10 computer.

2.8.2 Smaller tree using MDF

For comparison purpose, two types of trees have been experimented with, *MEF RPT* and *MDF RPT*. The former uses MEF and the latter uses MDF in each internal node of the respective tree. Both trees used the same 318 learning images, 210 from the straight hallway and 108 from the corner. As presented in Table 10, the MDF tree has only a total of 69 nodes, with only 35 leaf nodes; while MEF tree has a total of 635 nodes, with 318 leaf nodes. Fig. 5 explained why MDF can give a smaller tree. Note that the timing data shown in Table 9 is for MEF tree, which is larger than the MDF tree. An MDF tree is typically much faster.

2.9 SHOSLIF-R: Vision-Guided Robot Manipulator

This is for sensorimotor coordination and task-sequence learning, SHOSLIF-O was used as the object locator and recognizer. A SHOSLIF-R network was automatically built from training temporal sensor-guided control

Table 9: Computer Time Difference between the Flat and the Tree Versions

Time per retrieval	MEF tree	Flat version	
		in MEF space	in image space
Time per retrieval (in milliseconds)	27.7	738.3	2853.7
Slow down w.r.t MEF tree version	-	26.7 times	103.0 times

Table 10: MDF Results in a Smaller Tree

Tree type	MDF	MEF
Total number of nodes	69	635

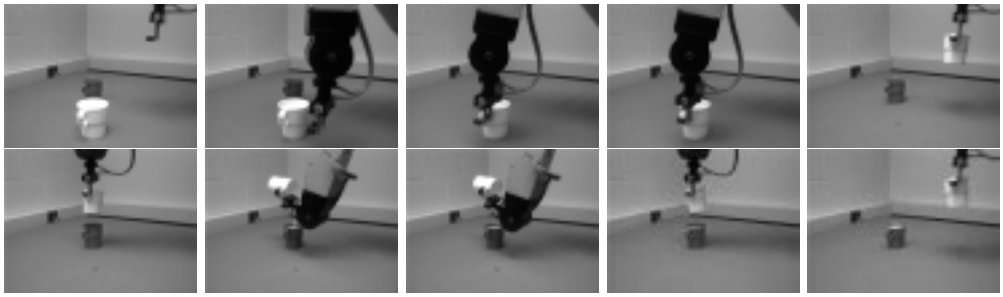


Figure 9: A demonstration of various actions learned: approaching the handle of cup A, picking the cut up, moving to top of cup B, pouring, and putting on table.

sequences [31]. The input to the SHOSLIF-R network is the image position of the objects (from SHOSLIF-o) and the index of the task from the human teacher. The output of the SHOSLIF-R network is the incremental values of the six joint angles of the robot manipulator.

Five actions were learned interactively at several places in the work space. Tests were done randomly in any place in the workspace. The success rate was 100% for all the actions, except that the liquid was spilled partially 20% of the time during the pouring action. More training can improve the pouring accuracy. This is an example of learning by doing, instead of explicit modeling. Modeling the dynamics of poured liquid is not possible because there are too many unknown and unobservable parameters in fluid dynamics. As far as we know, no other published systems exist that can perform the task of pouring water into a cup using vision.

Several robotics groups (e.g., [37] and [43]) have recently published works in which a robot manipulator can repeat the action sequence from human's demonstration using a data glove. Case-specific features and decision rules are written into their programs which the algorithm will use to identify the sequence of actions (such as "when the speed of the hand is smaller than certain number, do the following ... "). SHOSLIF-R is fundamentally different from those works in that SHOSLIF-R does not contain any case-specific rules (handcrafted knowledge-level rules). Specifically, SHOSLIF core does not contain any knowledge-level rules and the SHOSLIF-R shell contains only the arm hardware specification, i.e., the anatomy (e.g., degree of freedom of the hand) instead of knowledge-level rules. Thus, it is, in principle, generally applicable to any robot manipulator task. SHOSLIF-R is the first general-purpose robot manipulator learning system that can learn to perform tasks through interactive learning without handcrafting any knowledge-level rules.

2.10 SHOSLIF-S: Speech Recognition

The objective of this study [10] was to test the feasibility of using SHOSLIF for spoken word recognition. It was performed as a class project in a graduate class "Learning in Computer Vision and Beyond", which covered the human cognition and machine learning subjects for learning in vision, speech and sensorimotor coordination. The experiment was not done thoroughly, due to the limited available time during the single-semester class. In particular, there was not a sufficient number of training samples. The preliminary experiment was performed with 10 spoken words from "zero" to "nine". The speaker-dependent testing reached 90% accuracy among 20 speakers, each word was trained with only one training sample and tested with 4 different instances.

The fully dynamic speech recognition requires the recurrent version of the SHOSLIF: SAIL. The recurrent version will be able to learn to handle time warping, coarticulation, temporal acceleration, and pause that are very common in the real-world speech. Currently, a post doctor researcher who has finished his Ph.D. research in speech recognition is working with the PI to contribute to the speech recognition and speech synthesis functionalities in the current SAIL project.

2.11 Cresceptron: the Predecessor of the SHOSLIF

The PI's work along this line can be traced back to early 1990 when he conceptualized Cresceptron for general, open-ended, sensing-based learning. Cresceptron [93] [98] is the predecessor of the SHOSLIF. Cresceptron is the first work that is capable of learning directly from natural images and performing the task of general recognition *and* segmentation from images of the complex real world, virtually without limiting the type of objects that the system can deal with. It has been tested for recognizing and segmenting human faces and other objects from complex backgrounds. It addressed the issue of self-organizing dynamically by growing the system on-line according to inputs. Although Cresceptron has a very high generality, it does not attempt to solve scalability. Its successor SHOSLIF solved both generality and scalability.

2.12 Other Works

The PI has worked on several vision problems, including camera calibration whose high accuracy has been verified by a system-independent performance evaluation method [100]; an efficient octree representation for an arbitrarily moving object [94]; stereo matching and large motion flow computation with occlusion detection using multiple attributes [96] and using the windowed Fourier phase (WFP) [87]; completeness of WFP representation and signal reconstruction from the WFP [88]. His work on motion and structure estimation includes: an improved closed-form solution [106]; using line correspondences and its uniqueness proof [107]; for planar surfaces and its intrinsic uniqueness condition [95]; the closed-form matrix-weighted approximate solution and near-bound optimal solution for stereo case [101]; theory and methods for optimal motion estimation, its stability analysis and near-bound performance [97]; modeling smooth long sequence motion with a general LCAM model and its model parameter estimation for motion prediction [105]; integration of long image sequence for motion and structure analysis [17]; and the recent introduction of the concept of *transitory* image sequence and its integration [103, 102]. Some of the results on motion and structure estimation have been collected in a research monograph *Motion and Structure from Image Sequences* [108] coauthored by PI and in a chapter in *Handbook of Pattern Recognition and Computer Vision* [104].

3 SAIL: THE FIRST LIVING MACHINE UNDER DEVELOPMENT

Since all the objectives of Phase 1 have been achieved, Phase 2 is well under way. In Phase 2, the PI's group is developing the first living machine: SAIL. Its conceptual development, algorithm design, and hardware design have been completed. The hardware installation and programming are under way.

3.1 Overview

3.1.1 SAIL's system structure

Fig. 10 illustrates the SAIL system structure. It consists of a function f which accepts sensory input vector

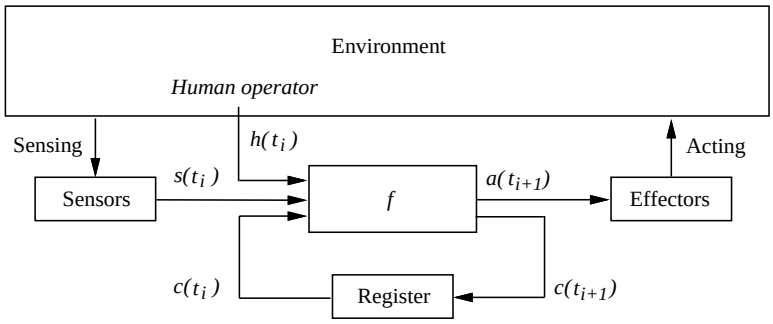


Figure 10: The basic structure of the SAIL system. It consists of a function which accepts sensory input vector $s(t_i)$, context $c(t_i)$ and physical input vector $h(t_i)$ and outputs the corresponding effector control vector $a(t_{i+1})$, and the context vector $c(t_{i+1})$.

$s(t_i)$, context $c(t_i)$ and low-level physical vector $h(t_i)$ and outputs the corresponding effector control vector $a(t_{i+1})$, and the context vector $c(t_{i+1})$. The central part of this is function f, which is implemented by a SHOSLIF tree which dynamically changes through the living machine's interactive autonomous learning in its environment, according to the incremental, real-time version of SHOSLIF developed in Phase 1. In other words, the major change we need to make is to convert a feed-forward SHOSLIF into a recurrent network with a delay register, plus a few system modifications for SHOSLIF. The major modifications include:

- The new SHOSLIF uses internal sparse-matrix representation to handle either sparse or dense, dynamically generated connections.
- Adding a periodic forgetting process, so that the system can collect disk spaces for should-be-forgotten nodes. The should-be-forgotten nodes are determined by a memory trace model that mimic humans' memory curve. The forgetting process is necessary for generalization and efficient use of storage space, as with a human's brain [12, 4].
- At the output end of SHOSLIF, add a process that selects the best matched case (indicated by SHOSLIF matching distance measure) weighted by the low-level physical input signal.

A lot of considerations and criticisms have been examined, resulting in the abandonment of several previous architectures. The context vector is used to keep information about the mental status, e.g., what the living machine wants to do, what has been done, what to watch for. Depending on the current mental status and the environmental stimuli, the living machine can autonomously recall long-term intention and short-term intention. It is not surprising that such a general-purpose living machine is implemented by a systematic approach, due to the living machine's principle that no handcrafted knowledge rules are allowed.

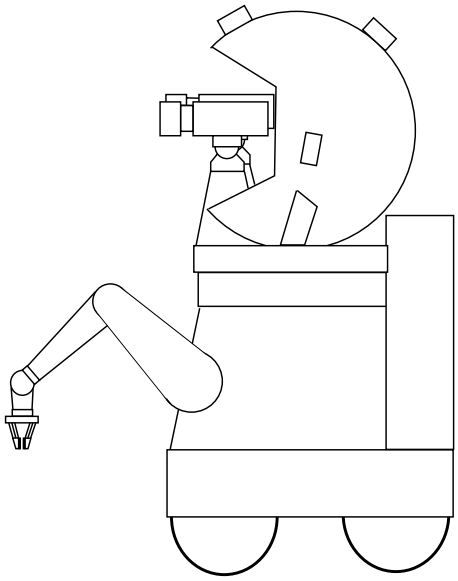


Figure 11: SAIL, a living machine under construction. It has cameras with real-time center-of-focus stereo, microphones, speakers, a robot manipulator, a mobile base, and positioning systems for the eyes and the head. Two on-board Pentium Pros are the computing engines.

Since the scheme architecture is systematic and domain independent, the architecture is applicable to any sensing and actuation modality. For the same reason, we have seen that a generic SHOSLIF approach is applicable to a wide variety of tasks as shown in Table 3.

3.1.2 SAIL's system hardware

SAIL, a living machine, as shown in Fig. 11 is constructed with a Labmate mobile platform and a light weight arm (from Eshed Robotec). Each of the two cameras is mounted on a pan-tilt head (from Directed Perception) for eye motion. The two pan-tilt heads are mounted on a pan-head (from Eshed Robotec) for head motion. The SAIL's sensors include: visual (two video cameras with auto-iris lens), auditory (four microphones for learning sound source localization using 3-D sound stereo), tactile (robot arm gripper figure-distance sensor, arm overload sensors, collision detection sensors, etc). The SAIL's effectors include: arm, speaker, drive system, eye's pan-tilt units, head's pan unit, vertical sliding base (for sliding the arm-and-head assembly vertically for different heights), and one internal attention system for each of the sensors. The computation engine consists of two on-board Micron Pentium Pro 200Mhz computer connected by on-board 100-BaseT fast Ethernet connection. Now, each computer has 4GB dual disk pack which will be extended with external disks when needed. According to the speed of SHOSLIF tested, and the speed of the Pentium Pros, SAIL is expected to run at a refreshing rate above 10Hz.

3.1.3 SAIL's training

After the "birth" of SAIL, it will be in "awake" status for about 8 - 12 hours everyday. The remaining hours will be at "sleeping" status when the forgetting process can run and the batteries be changed. Our graduate students and some graduate students in educational psychology will be the baby robot's trainers. At the early training stage, the instruction to SAIL starts with mostly "hand-in-hand" instructions and demonstrations. These will be done through the living machine's low-level physical channels using graphics user interface (GUI), teach pendant, and a joystick. Autonomy of the living machine will gradually increase after some basic skills have been learned in the "hand-in-hand" mode. It is expected that through the

development of SAIL's visual and language capabilities, the communication modalities between humans and the living machines will gradually move to visual gesture and speech.

The understanding of high-level concepts will be built up through interactions with human teachers, such as the concepts of eye, hand, door, left, right, up, down, fast, slow, good, bad, etc. SAIL will link these concepts (represented by e.g., sound of the word) with many instances that it has experienced. The representation of these concepts are implicit in the system, as a pattern of active nodes in the SHOSLIF tree.

Detailed training log will be kept to record the behavior of SAIL during every training hour. The system-level software design will be modified depending on the observed SAIL's behaviors and cognitive progress.

3.2 Representation

The following detailed scheme is what is currently being implemented. However, just like all the other research projects, the planned scheme is subject to change during the research development. Since this document is meant for funding agencies, the descriptions are kept as brief as possible.

Suppose that the world is sensed at t_i, $i = 0, 1, \cdots$. Each sensing time t_i corresponds to the time at which the sensor's input is taken. Following this, the corresponding computation is performed before the next sensing time t_{i+1}. Therefore, each t_i corresponds to a computational cycle, which is called a mental cycle. Due to variable computational time needed before the next cycle, the length of each mental cycle $t_{i+1} - t_i$ is typically not a constant. For real-time computation, each mental cycle is at most 100ms long. The system consists of the following components.

Sensors A sensor senses a certain aspect of the environment and produces a signal vector. There are several sensors in SAIL. Each sensor senses an aspect of a partial world at each time instant t and gives a sensed signal vector $s(t)$.

Effectors An effector (or actuator) accepts a control signal vector $a(t)$ and acts on the world according to the control signal vector.

System function f The system function f is responsible for computation in each mental cycle. The system function f accepts an input vector $x(t)$ and produces an output $y(t)$. The dimensions of $x(t)$ and $y(t)$ dynamically change with t. The register stores the current context vector computed by the system function f to be used by f in the next mental cycle. The system function f together with the feedback register form a recurrent system, in the sense that $x(t_i)$ contains the output $y(t_i)$ resulted from the previous mental cycle.

The environment The environment contains the physical world that the system senses and acts upon. As a part of the world, the human teacher interacts with the system during operation (learning and performing). The human enforcement signal is fed into the system via the low-level physical channel $h(t_i)$, which can contain the control signals for "hand-in-hand" training and physical evaluation signal. In other words, it contains human's enforcement in terms of what to do or the evaluation of how the system is doing.

3.2.1 Sensors

The types of sensor used by SAIL include:

- Visual. The visual sensors are video cameras and the associated digitizers.
- Auditory. The auditory sensors are microphones and the associated digitizers.
- Tactile. This type of sensors include gripper figure-distance sensor, arm overload sensors, collision detection sensors, etc.

3.2.2 Effectors

An effector acts on the environment according to the given control signal vector. As a special case, it can also act on a part of the robot itself, such as loading a cargo onto the robot's carrier. The types of effector used by SAIL include:

- Attention extractor. It extracts a part of the signal vector and applies some weights to the extracted part. For example, our visual attention mechanism determines where to look at from a scene.
- Pan-tilt unit. It pans and tilts the sensor. For example, the head's neck has a pan unit and each of the two stereo cameras has a pan-tilt unit. With this arrangement, the neck's pan unit is to direct the two eyes (as well as ears) to a given horizontal angle while keeping the relative relationship of the two eyes fixed. The pan-and-tilt units of the two cameras can control the viewing direction and the convergence between the two cameras. This is one of the major mechanisms for human to find an object of interest and achieve certain degree of positional invariance.
- Structured speaker. It consists of a structurer and a speaker. The structurer is used to reduce the dimensionality of the input vector to the speaker. A structured speaker, for example, can only speak using the voice of a particular human individual. An time-varying MEF parameter vector is used to drive the structured speaker through time in order to speak, sing, laugh, whistle, etc.
- Robot arm. The robot arm is used to manipulate objects in the SAIL's environment.
- Vertical slider. The vertical slider is used to raise or lower the head-arm assembly so that they are in a good work distance with the target objects.
- Drive system. The drive system is used to navigate on the ground. The arm, the slider, and the wheels form a redundant system in that there are infinitely many configurations that result in the same 3-D orientation and 3-D position at end effector of the gripper.

3.2.3 The SAIL recurrent system

The recurrent system consists of the system function f and the feedback implemented by the register. The system function f computes output y from input x: $y(t_{i+1}) = f(x(t_i))$. This recurrent system is an extension of the incremental version of the SHOSLIF tree in the following sense.

1. Input $x(t_{i+1})$ contains current input from sensors and the context vector $c(t_i)$. The dimensionality of the sensory input is fixed but that of the context vector increases (during learning) and decreases (during forgetting) over time.
2. It allows the user to directly interact with the system during operation through the physical channel.
3. It conducts reinforcement learning, learning according to human's encouragement and discouragement.
4. It is able to explore the world and thus learn autonomously.

The system function changes with time. Thus, it can be represented by a series of functions: $f_i : \mathcal{R}^{n_i} \mapsto \mathcal{R}^{m_i}$, $i = 0, 1, 2, \cdots$, where i indicates the mental cycle. That is, for each vector $x \in \mathcal{R}^{n_i}$, f_i maps it to $y \in \mathcal{R}^{m_i}$: $y = f_i(x)$. The input and output dimensions n_i and m_i change with the time index i. Typically, f_{i+1} is an improved version of f_i, due to the learning in the mental cycle i. Conceptually, the input and output dimensions, n_i and m_i, are unbounded. In other words, there exists no constant N such that $n_i \leq N$ and $m_i \leq N$ hold true for all $i \geq 0$. In reality, however, n_i and m_i are bounded above by the largest number that a computer can represent. Due to the time delay in the computation of f, the output from f_i, given input $x(t_i)$ at time t_i is available at time t_{i+1}: $y(t_{i+1}) = f(x(t_i))$.

Each input $x(t_i)$ contains the current sensory input vector $s(t_i)$, the current context vector of the system $c(t_i)$, and the override input $e(t_i)$ that is a part of the physical vector $h(t_i)$.

$$x(t_i) = (s(t_i), c(t_i), e(t_i)) \tag{2}$$

The context vector includes system status information that is needed for the next action. The context vector will include, at a later mature stage, (1) a limited past history, (2) the living machine's intention in terms of what to do, (3) a prediction of the consequence.

The output vector $y(t_i)$ contains the control signal vector for the effectors $a(t_i)$ and the system context vector $c(t_i)$:

$$y(t_i) = (a(t_i), c(t_i)) \tag{3}$$

A coarse-level diagram of the system is shown in Fig. 10. The computational unit f has a delay, so that the output $y(t_{i+1})$ due to the input $x(t_i)$ at time t_i is available at t_{i+1}. A register is used to keep the context vector before it is used in the next computational cycle.

The motive of defining the domain and range of the function f as a space of dynamically change dimensionality is to facilitate the following capabilities:

1. Addition of newly learned concepts.
2. Disrememberance of old concepts.
3. Allowing the sensors to be added and deleted during the different life stages of the system.
4. Allowing the effectors to be added and deleted during the different life stages of the system.

The function of f is implemented by a recursive space partition network (SAIL network) which allows the dimension of the space to increase and decrease dynamically. The basic structure of the f network is a tree. However, the output of the tree is fed back as the next-time input. Therefore, the network as a whole is not a tree.

3.2.4 Use of the physical input channel

In this section, we build in some innate mechanisms that may correspond to innate human characteristics, such as pain avoidance and love for food. This is a very critical part through which human can train the living machine according to what human wants.

Consider f as a function that maps the input domain of x to the range of y so that the corresponding effector vector a results in the desired action. The context vector $c(t_i)$ is to keep context information so that the temporal and spatial relationships are fully used.

The function f is realized by two stages. The first stage is a recursive partition tree (RPT) (see [93] for a more detailed description of the RPT used in SHOSLIF). The second stage is the computation of the output vector.

Suppose that the RPT has learned l input samples $x_1, x_2, \cdots, x_l$, where each x_i, $i = 1, 2, ..., l$, contains a sensory input s_i, and the associated context vector c_i. Given an input s and the current context vector c, the best top k matches, $x_{i_1}, x_{i_2}, \cdots, x_{i_k}$ are found from all the learned samples. Each matched item in the RPT has an associated action vector $a_{i_1}, a_{i_2}, \cdots, a_{i_k}$ that has been memorized during the past learning. Each match gives a probability estimate $p_{i_1}, p_{i_2}, \cdots, p_{i_k}$, where p_{i_j}, $j = 1, 2, ..., k$, is the approximate probability $P(s_i, c_i \mid s_{i_j}, c_{i_j})$. The confidence vector is then defined by $p = (p_1, p_2, \cdots, p_l)$, where $p_j = p_{i_j}$ if $i = i_j$ for some $1 \leq j \leq k$, and $p_j = 0$ otherwise. In other words, the confidence vector is such that the positions at the top k matches have the corresponding confidence values and other positions have zeros.

The physical channel signal h has two types of inputs, override input e and physical feedback b. The signal e is to force the living machine to do an action, to simulate a hand-in-hand teaching action. The physical feedback b is a low-level physical feedback used to make up the drawback that machines do not have any physical feedback that are available in humans, such as the punishment of physical pain and the

enjoyment of a suitable food. The value of b ranges from 0 to 1. The b value is stored in each leaf node and is associated with the effector parameter corresponding to that leaf node. The default b value is 0.5, when there is no teacher supervising and the robot is doing something on its own. If the teacher wants to "beat the machine up" (punish it), he or she enters a value smaller than 0.5. If the teacher wants to let the machine feel happy, he or she enters a value larger than 0.5. At an early stage, b is entered on-line during learning. This physical feedback will be used as an essential channel through which human teacher can create human innate characteristics during the early development stage of the robot, such as pain avoidance and love for food. For robot, for example, we will use b to teach it to avoid being hit by wall (a low b value will result) and love to do things that will lead to physical pleasure (a high b value).

Then, the system finds j so that $j = \text{argmax}_j\{p_j b_j\}$. Whether a_j is executed depends on the value of b_j and whether there is an override signal through the physical channel from the human teacher. When the system has developed the visual or auditory understanding capability, the evaluation from the human teacher will be given through gestures or speech.

Why will a mature living machine listen to the human teacher even when physical value b is not used? This is because during the early learning, the trainer has always given a good b value if the living machine follows what is said. Thus, the machine gradually developed a behavior pattern — do what *the teacher* wants. During the later stage of learning, the human teacher does not give a b value every time or even very rarely. The high-level concepts learned by the system allow the system to choose what to do among the possible actions all with the similar b_j values (e.g., $b_j = 0.5$). We know that a human adult does not make a decision just based on physical pleasure. Many high-level things a human being does do not directly link with physical pleasure or pain either. Of course, the living machine may listen to his own teacher, but may not necessarily listen to other human beings. All these behaviors depend on how the living machines are trained.

In summary, the living system will automatically choose actions that are associated with a good physical value b when the living machine is young. When the living machine becomes mature, it makes a decision based on the high-level concepts (active patterns of later added nodes in the network) while the physical feedback b diminishes its role. However, no matter whether the living machine is young or mature, the override input e and physical feedback b from the physical channel can always be used when it is necessary.

3.2.5 Speech synthesis

The goal of speech synthesis here is different from many applications where a fixed way to say a sentence is sufficient. Depending on the situation, the living machine must be able to say the same words or sentences in different ways, including varied speed, volume, tone, pause, etc. This capability is the major difficulties in the field of speech synthesis (see a survey by Alexander *et al.* [1]) but is a must for natural discourse between humans and machines.

We call an effector structured if the space of its control vector is of a low dimensionality and most points in that space are useful in practice. An unstructured effector has a high dimensionality and only a small part of the space is useful. By this definition, a robot arm is a structured effector and so is a mobile drive system. A speaker is not a structured effector because most of the temporal sequences will not produce any sound that is useful (i.e., representing human's voice in this case).

We need to convert an unstructured effector (speaker in this case) into a structured one, by inserting a structurer before the effector, as shown in Fig. 12. The goal of the structurer is to make the dimensionality of the input space small so that a very large portion of the parameter space is useful. In other words, the structurer is to parameterize the effector's control space. The principal component analysis is suited here. We use the MEF space to represent the space of all the possible short human utterances. Various human's utterances are cut into segments of 10ms length. These segments are used to compute a correlation matrix , . A MEF projection matrix M is computed from , whose columns are the eigenvectors of , (i.e., the principal components of the segment distributions). We can keep several structurers available, each corresponds to a unique projection matrix trained using one person's utterances. Different users may choose different structurers to fit their own preference. For example, a machine TV personality may be assigned a

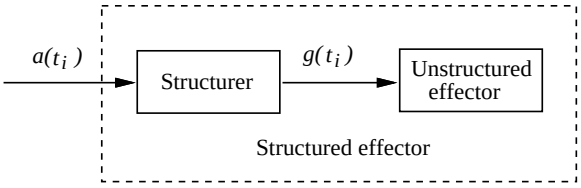


Figure 12: A structurer converts an effector control vector into the corresponding signal sequence (e.g., temporal speech signal) to be played by the unstructured effector (e.g., speaker).

child voice while a machine care-taker may be assigned a female voice.

The structurer for speech synthesis must run in real-time. For segment length of 10ms, it must run at 1/(10ms) = 100Hz. The well-known double buffering method can be used to achieve continuous speaking. For real-time performance with a huge number of utterance units, the structurer will be implemented as a SHOSLIF tree. The input to the tree is the parameter vector (a part of $a(t)$). The output of the tree is the temporal wave signal corresponding to the parameter.

Note that there is no need to define the concepts of syntax, such as word, phrase or sentences, for SAIL. They are learned later in baby SAIL's learning phase. We just build a "mouth" and its "vocal tract" represented by the speaker and its structurer.

We will teach the baby SAIL interactively by letting the program run many hours a day. Once the baby SAIL hears anybody's voice, its computer projects the voice into the structurer's input space. The projected feature parameter vector is the actuation vector $a(t)$ to be learned by the SAIL network. It can imitates other's speaking by sending the learned $a(t)$ vector to the structurer which constructs sound waves using the MEF projection matrix. The sound wave is then sent to the speaker.

3.3 Working of SAIL

This section briefly explains how SAIL can work.

3.3.1 Early learning

Initially, the human teacher uses the override signal e to feed actual control signals to the system via the GUI, joystick and robot-arm teaching pendant. This is to simulate the way a human care-taker teaches a human baby by holding their hands. Due to cost restrictions, SAIL does not use a compliant robot arm by which human can enforce the configuration of the arm by applying force directly on to the arm. The trainer may mostly use good actions with a good b value. After a sufficient hand-in-hand training, the living machine can be set free to try by itself, first in the same environment and then gradually move to slightly different environments. During this stage, some mistakes can be made, depending on how extensive hand-in-hand training was and how different the new environment is from the training one. The human teacher enters appropriate b value in real time, to discourage actions that may potentially lead to a failure and encourage actions that can lead to a good result. At this stage, human teacher may want to speak and use gestures in addition to feeding b value via a joystick so that the system can associate the visual and auditory signal from the human trainer with the corresponding b values.

3.3.2 Concept learning and behavior learning

This is probably the most interesting, most controversial, and probably most exciting part of the endeavor. Our task is to investigate how a machine can develop high-level knowledge and motives from low-level "physical feedback". The PI's ideas presented here were inspired by studies in child developmental psychology.

The SAIL associates each sensory input with the visual and auditory signals from the human to learn the concepts taught by the human. It will learn good behavior from very early stages, such as "do not run

toward a wall", "do not run too fast when there are things nearby", "handle it if you see something like that", etc.

The understanding of concepts will be built up through interactions with human, such as the concepts of left, right, up, down, fast, slow, good, bad, right, wrong. Many experiences and the occasions are associated with the concept that is used. The representation of these concepts are implicit in the system. It represents a pattern of response in the network.

Later, many instances will allow the living machine to associate good physical feedback b with his actions that follow what the teacher wants. It also will gradually accumulate the information about what the human teacher likes and expects from it. It will then link the concepts of "good and bad", "right and wrong" with the many instances it experienced.

Generalization of the concepts occurs naturally. For example, the living machine first associate words with the b value. Some words appear together with a bad b value and some other words appear with a good b. Then, the living machine associates many good words with a particular word such as "happy" and bad words with a particular word such as "sad." Since happy is associated with a good b value and sad is associated with a bad b value, the living machine gradually establishes a behavior habit to choose actions that will make the human teacher happy (good). At that time, the importance of the b value gradually diminish. A default $b = 0.5$ value is often enough during this stage. A more mature system will receive most human feedback from normal sensory input, via speech, gestures, etc.

Further on, the system will link the sense of good and bad to other more complicated activities, such as schooling, the evaluation system in its school, and what to do in order to receive a good evaluation from its school, etc. In summary, as long as the teacher uses the b value correctly, he or she will be able to train the living machine to do what he or she wants, from simple to complex, even during later stages when b is used very rarely.

3.3.3 Cognitive maturity

In an autonomous learning process, the environment provides an endless sequence of stimuli coupled with the actions of the living machines. Certainly, the living machine should not just remember the sequence as, e.g., a single 8-hour sensing-action sequence everyday, because a lot of events do not have close relationships. The machine should only remember things that are important at its current cognitive developmental stage.

The cognitive maturity determines what a machine can learn. For example, a child in the sensorimotor stage will not be able to learn formal reasoning, even if a teacher tries that. Before a sufficient amount of low-level knowledge and skills has been learned, it is not possible to learn higher-level knowledge and skills. As discussed before, our definition of the concept level is basically operational. If a concept that is learned earlier and is used effectively in the learning for a later concept, the former one is a lower-level concept and the latter is a higher one. Therefore, the definition of levels can differ from one living machine to the next, depending on the experience of each machine individual.

In the living machine, the maturity scheduling is realized automatically by the process of automatic feature derivation using MEFs and MDFs in the SHOSLIF together with the forgetting process. Consider the following two cases. The first one is with an immature machine and the other is with a mature machine.

When a set of stimuli representing a high-level concept is sensed by an immature machine, the configuration of the current SHOSLIF tree is not sufficient to learn the new concept associated with the stimuli. This is reflected by the fact that the features derived by the SHOSLIF tree constructed so far is not able to successfully recall a set of learned concepts. Thus, the corresponding stimuli looks meaningless by the living machine and are just simply temporarily memorized. Without extracting recallable features, stimuli containing the same high-level concepts in different occasions look very different. Later on, that temporary memory is quickly forgotten (deleted) by the forgetting process because there is no recall to this memorized stimuli within a certain time period.

If the living machine has learned a sufficient number of low-level concepts and skills, indicated by the corresponding nodes in the SHOSLIF tree, the same stimuli mentioned above will result in a very significant amount of feature recall (MEF or MDF) from the mature SHOSLIF tree. Thus, the corresponding stimuli are

memorized, at a deep location linked from the corresponding lower-level concepts (or features) as ancestors. Within a certain period of time, if the same concept appears in the stimuli in another occasion, the mature SHOSLIF tree can recall a sufficient number of features and retrieve the newly memorized concept. Such a successful retrieval indicates a memory reinforcement. The forgetting process will record this reinforcement at the corresponding node and apply a much slower memory decay curve to this concept. This concept is then learned by the living machine.

In this way, the living machine is able to learn autonomously, and going through various cognitive developmental stages which are probably similar to those characterized by Jean Piaget (see Table 1). During each day, it learns what it can learn and forgets what it must forget. As is the case with humans, entering a new cognitive stage by a living machine is natural and gradual, depending on the learning experience with each machine individual. There is no need for the living machine designer to enforce each development stage into the program.

3.3.4 Thinking

The living machine is able to think autonomously. How does the machine think? This is a question that has fascinated scientists and public alike for many years [111]. A computational process implemented by a computer is not thinking. Otherwise, any program is doing thinking. The thinking process must be autonomous. However, autonomy is not sufficient for qualifying thinking. Otherwise, an autonomous road-following vehicle is doing thinking.

We must not program logic rules into the living machines The fundamental characteristic of thinking is that the contents and the rules of reasoning cannot be pre-arranged — i.e., programmed in. Let's consider an example. A common sense knowledge "all human are mortal" can be represented by compound proposition (tautology) $\forall x[p(x) \rightarrow q(x)]$ where $p(x)$ = "x is a human" and $q(x)$ = "x is mortal". Then, with input $p(\text{Tom}) = \text{True}$, a program can prove $q(\text{Tom}) = \text{True}$ using logic deduction rules. When doing the above reasoning, the program is not thinking because the logic deduction rules are pre-programmed by the human. In order to make a machine think, its representation must be knowledge-free. In our example, the knowledge is formal logic. The representation for $\forall x[p(x) \rightarrow q(x)]$ is not knowledge-free.

It is worth pointing out that symbolic reasoning is not as difficult as the other parts of cognition. In the above case, e.g., it is much more difficult to figure out that Tom is a human from visual sensing than to perform the symbolic reasoning.

What is thinking? A process in a system is a thinking process if (1) the process is autonomous, (2) its representation is free of knowledge, (3) it is conducted according to knowledge that has been autonomously learned. The first two points are meant to distinguish a pre-programmed computation from an autonomous thinking process. The last point is to make sure that thinking is not a useless process or directly directed by humans.

Thinking at mental cycle level With the SAIL living machine, the thinking process is to feed the context vector $c(t_i)$ (mental status) as the input to the network f while the attention selection for external sensors keeps off. A mental cycle of the thinking process is to go through f once as shown in Fig 10. A long thinking process corresponds to many consecutive mental cycles going through f, each with a new context vector from the former mental cycle. Thus, a long thinking process may allow prediction of many chained events. Naturally, all the thinking processes are originated from the stimuli in the environment. During the thinking process, the stimuli input from the sensors may be temporarily turned off to allow prediction and planning to be performed.

3.3.5 Reasoning and planning

Reasoning and planning are special types of thinking process in the living machines. One might think that in order to conduct reasoning and planning, there must be a controller, which controls what to think about and organizes various stages of reasoning and planning. However, no such controller can work.

Must not have a thinking controller There is a fundamental dilemma with this "controller thinking." This controller, as a supervisor, must be smarter than what it controls. In other words, it must know the global situation, understand it, and figure out what to do and how to do. Furthermore, this controller must be general-purpose because the living machine is a general purpose creature. We know that no controller can perform a reasonable control task without understanding what is going on. However, understanding is exactly our original problem. Thus, we have a chicken-and-egg problem — one cannot be solved without first having the other.

Programming behaviors into system cannot work either Another alternative is that we do not use any controller. This is the case with some behavior-based methods (e.g., see Brooks [8]). However, the behavior-based methods cannot go beyond the very limited number behaviors that have explicitly modeled and programmed in. None of the existing behavior-based approaches really try to understand the world. Human beings, on the other hand, can gradually learn and understand more and more complex concepts in the world and their high-level behaviors are based on understanding instead of pure wired-in reflexes.

Consciousness A much deeper issue is consciousness. Each human being has a sense of self, and can consciously control what to think about and what to do at his or her will. Apparently, each human being has only one consciousness. How does the living machine develop the consciousness?

Consciousness and the controller in the living machine With the living machine, the controller is fully distributed, and fully embedded into the function f. At the output end of f, only top k matched leaf nodes are computed. Each leaf node corresponds to a concept-action combination. As explained in Section 3.2.4, the best leaf node j is determined using the physical value b. Thus, this leaf node results in the context vector c, representing the current mental status in terms of what to do or think next. This choosing-the-best-match process tells what to do next and guarantees that the living machine has only one consciousness. Therefore, there is no central controller in the living machine. All the living machine does is to choose the best matched case from many cases learned before, taking into account the physical feedback experience, and then to use it as the context vector for the next mental cycle of sensing and action.

Taking care of multiple objects at a time Since the living machine has only one consciousness, it cannot take care of two things at the same time. For example, in order to line up two marbles along a horizontal line, it needs to learn the combination of the two marbles as a single pattern. When the two marbles are lined up horizontally, the desired pattern is observed. In order words, the combination of objects is considered as a single pattern to be learned. This phenomena of learning multiple things as a unit is very common in human children's cognitive development [5] [13]. With sensorimotor learning, a child is able to move one marble in the right direction until the desired pattern appears. During the formal operational stage, all the combinations have been learned and thus the alignment task becomes virtually effortless.

Formal logic reasoning During the preoperational stage and the concrete operational stage, a large number of concepts are learned, such as

- spatial relationship, such as left and right, up and down.
- temporal relationship, such as before or after a particular event. This includes the skills of planning and replanning.

- more general logical relationships, including all those that are termed "common sense".

A lot of such concepts must be learned before the living machine can enter the formal operational stage.

During the formal operational stage (human age 12 and beyond) of the living machine, formal logic reasoning is performed based on pattern matching. Let's finish the previous example: $\forall x[p(x) \rightarrow q(x)]$ where $p(x)$ = "x is a human" and $q(x)$ = "x is mortal". First, $\forall x[p(x) \rightarrow q(x)]$ may be learned as a pattern, represented by, e.g., the look of the mathematical equation the way it is written on a white board in the robot school. (There is huge amount of evidence that demonstrates that visual pattern helps the memory and understanding of symbolic meanings in human.) The statements $p(x)$ = "x is a human" and $q(x)$ = "x is mortal" create a memory image as, e.g., a visual image as the way it was written on a white board. Learning of the relationship — once $p(x)$ is true, then $q(x)$ is true — is learned as a sensorimotor temporal sequence. Then, once p(Tom) = True is given or understood from visual inspection, q(Tom) = True is derived based on the learned sensorimotor temporal sequence. A key point here is that logic reasoning in the living machine (and humans) is not represented internally as explicit mathematic relationships. But rather, they are represented as a vision-guided sensorimotor operation sequence. Later, after repeated thinking processes, the sensorimotor operation sequence is memorized without the help of visual stimuli. Then, such a logic process can be easily and quickly applied to other cases with new predicates $p(x)$ and $q(x)$.

In summary, the living machine does not have any embedded (programmed-in) rule about formal logic reasoning. It can reason like an illiterate before it goes to school. It will learn the mathematic logic in school. Potentially, it will make some inventions after it has gone through some school learning and has had enough experience with a subject of study.

3.3.6 Time warping and co-articulation

The recurrent architecture shown in Fig. 10 facilitates temporal recognition and temporal action, under time warping and co-articulation. For example, suppose that an input sequence is $x = x_1x_2x_3...x_m$. The temporal learning through recurrent f will remember each subsequence $y_i = x_1x_2...x_i$, for $i \leq m$, as indicated by a corresponding context vector c at each i. The transition from y_i to y_{i+1} can be delayed or moved ahead when time warping occurred, because from y_i to y_{i+1} the system may take more or fewer mental cycles than a normal case. This is very similar to the case of HMM, except that the model here is more general than HMM. Therefore, we need to fully implement the recurrent SAIL version before fully testing speech recognition. The feed-forward SHOSLIF does not take care of time warping explicitly, but the recurrent SAIL does.

Such a temporal mechanism is also very effective for action synthesis, such as vocal discourse and robot arm manipulation. For example, consider a sequence of action $a = a_1a_2a_3...a_m$. When the subsequence $b_j = a_1a_2...a_j$, $j \leq m$, has been executed, the context vector c will contain a_j along with time-warp coded earlier a_i's with $i < j$. This c will be used as a part of input to f in the next mental cycle. According to the learned experience, a_j along with the time-warp coded earlier a_i's with $i < j$ as input to f will lead to a leaf node whose action part is a_{j+1}, which leads to a_{j+1} being executed.

Similarly, co-articulation in speech will be taken care of naturally by the recurrent nature of the architecture. In co-articulation, the segment of input (for recognition) or output (for synthesis) depends on its temporal neighbors. The context vector c contains time-warped context information which will be used by f to recognize or synthesize correctly in each mental cycle.

3.3.7 Modularity

The modularity question concerns whether we should assign a separate f (module) to each sensor. We intend to try both schemes (a) A single f for all the sensors and effectors. (b) A separate f for each sensor (sensor module) and effector (effector module) and a central f for all the modules. Currently, we guess that (a) is more effective, since modularity can be generated automatically by the automatic feature derivation process (subspace determination) in the SHOSLIF, through the learning experience of each individual living

machine. We realize that the human visual cortex is reassigned to hearing in the case of the blind. Thus, there should be a very flexible self-organization scheme among different sensing and action modalities.

3.4 Incremental and Real Time

The main network of f corresponds to a knowledge-base, although it also contains low-level perceptual capabilities and behaviors that typically are too primitive to be considered knowledge. The learning process of the system must be incremental. Input signals are sensed one frame at a time. The action activities are also done one segment at the time. The system updates its network at each mental cycle. The incremental version of the SHOSLIF is well suited here.

As can be expected, the number of leaves in the SHOSLIF tree will grow to a very large number, even if the forgetting process is performed regularly. Due to the ever decreasing price of hard disk, the storage space is not of a major concern. However, the time that it takes to go through the function f must be less than one tenth of a second in order to reach the 10 Hz refreshing rate. The subspace method used in SHOSLIF can effectively control the dimensionality of the input to each internal node of the SHOSLIF tree, since each internal node needs only to determine a hyperplane in a subspace of a limited dimensionality. The time complexity of f is still $O(\log(n))$ and the generality still holds true.

The real-time requirement is also important for speech recognition. If each speech segment is 10ms long, the SAIL network runs at 1/(10ms) = 100Hz, 100 times a second. That means that one cannot use very complicated preprocessing. Some frequency domain features will be computed in the pre-processing stage. The objective of speech preprocessing is a fast speed and good completeness (i.e., do not lose much information). In order to achieve continuous processing, we use the double buffering technique: processing digitized speech segment in A buffer while B buffer is receiving the next segment. Then, the computer processes the segment in buffer B while A is receiving the next segment.

There is no need for detecting the beginning of a sound. Silence is a mental state itself, just like a constant sound "a". The system runs continuously and acts (speaks if needed) continuously. The same idea is also applied to robot arm actions and navigation actions.

4 Further Thoughts

4.1 Generality

Can the system potentially do anything that a human can? This is an open question. The problem here is not just a static function approximation which can be performed by, e.g., a three-layer neural network with back-propagation. The critical issues include the dynamic generation of concepts from sensor-effector-based autonomous learning, concept generalization through experience, the capability to automatically derive feature space, and self-organizing the dynamically changing network. Autonomous learning through real-time sensing and action without handcrafted knowledge-level rules or behaviors is a totally new subject of study. The upcoming study will answer some very important and fundamental questions.

4.2 Space Complexity

The space complexity is directly related to the amount of information learned. The human brain has about 10^{11} neurons, each being connected by roughly 10^3 synapses on average [60] [41] [3]. If each synaptic link is considered a number, the human brain can store about 10^{14} numbers. This amount is now within reach by hard disks as far as the cost is concerned, thanks to the fast advance in computer storage technology[8]. It is expected that in a few years, the absolute storage size of the human brain can be realized with compression by the up-coming rewritable optical DVD disks for about $5,000, although such a high volume may not be needed as explained below.

It is known that a large part of the neurons in the human brain is not activated. Furthermore, the living machine does not need its brain to control heart beating, breathing, and digestion, which are served mainly by the medulla and the pons in the human brain. It does not need much service from the somatocensory system. The taste system and the olfactory system are not needed either except for certain special applications. It probably does not need its brain to serve for sexual drive either, which is taken care of by the amygdala in the human brain. Further, computers have effective high-level computational mechanisms, such as the fast and effective algorithms for computing eigenvector and eigenvalues of a huge matrix, which is a major part of computations in SHOSLIF. The known methods for computing eigenvectors by artificial neural networks are slow, iterative, and not as accurate [49] [66] [67]. Thus, the living machine might be able to reach a good performance with a disk space that is significantly smaller than the absolute storage size of the human brain. Of course, this also depends on the scope of the domain to be learned and the required resolution of the sensors. Since the cost of magnetic hard disks and rewritable optical disks is going down fast and consistently, it is now possible to equip a system with a disk system of 1,000 GB (about 20% of the absolute storage size of the human brain with a moderate compression) at a cost of about $20,000. It is not clear what kind of performance SAIL can reach with a storage size ranging from 10 GB to 1,000 GB. This is one of the major questions to be answered in the project.

If the size of the brain were not an important issue, perhaps monkey could serve as a living machine — a seemingly cheaper one. However, the size is probably just one of the problems with monkey's brain. Although a monkey can perform sensing and action tasks that no machine can do so far, the genetic coding of the monkey brain might not enable it to work up to a high-level comparable to that of humans. We cannot control the self-organization scheme of the monkey's brain, at least not now. With a living machine, we do not have such a limitation.

4.3 Time Complexity

The time complexity should not be addressed in a conventional way. Here the time needed to train the system to reach a certain level of performance is on the order of months or years. It is expected that a

[8] Using a hardware compression/decompression board with a moderate compression rate of 20, 10^{14} bytes of uncompressed data require 5,000 GB storage space. Internal hard drive kits cost about $100 per GB now, which means that 5,000 GB cost about $500,000 if ones buys now on street without volume discount. It is worth noting that the nature of the SHOSLIF tree data allows a moderate compression that is typical in video compression.

machine can learn faster than human beings because it does not feel tired and it computes faster. The speed of learning with the living machines is more controllable than biological systems, such as humans.

The critical type of complexity is the time complexity for each mental cycle when the size of the network becomes very large. Because of our on-line learning and the subspace method, the time complexity for each mental cycle for f is $O(\log(n))$, which is an inverse function of exponential explosion. In other words, when the processing speed increases, the number of cases it can handle in a fixed 100 millisecond time interval grows exponentially. To see how low the logarithmic complexity is, suppose a system whose time complexity is $O(log_b(n))$, regardless of how big the constant coefficient is in the time complexity. If this system can complete a mental cycle in 100 milliseconds with 1000 stored cases (which is about the speed SHOSLIF has reached), the same program running on another computer whose speed is 4.7 times faster can finish a mental cycle in the same time interval for a network that has stored 10^{14} cases (the absolute size of the human brain)! This displays a very high potential for the SAIL approach.

4.4 Knowledge-Base

The subject of knowledge representation and knowledge-base has been studied for many years without serious consideration about sensing and action. A huge amount of human power has been spent to model human knowledge, its representation, and input of data into knowledge-bases. However, the resulting systems are hard to use, hard to maintain, hard to keep up to date, and are brittle for a lack of understanding of what has been stored.

What the field of knowledge representation and knowledge-base has experienced is a natural stage that we humans must go through on our way toward understanding ourselves, machines, our environments, and their relationships. However, it is about time that we tried a fundamentally different approach, an approach that is much closer to the way a human acquires knowledge. Not too surprisingly, the SAIL approach seems to require much less manpower and costs less than many conventional knowledge-base projects, since human is relieved from the tremendous task of building rules for human knowledge and spoon feeding human knowledge. For knowledge-base construction, we want to move from manual labor toward automation.

4.5 Is This a Formidable Project?

To consider whether the proposed living-machine project is formidable, it is helpful to compare the nature of the proposed domain-independent approach with that of domain-specific ones.

Domain-specific approach: With a domain-specific approach, humans manually model knowledge in each domain. Thus, each domain problem is very hard because the knowledge required in each domain is too vast in amount and too complicated in nature. Accustomed with domain-specific approaches, few people in the field believe that anybody can solve the general Grand Challenge problem because each domain problem is already too hard.

The living machine approach: With the proposed living machine approach, humans are relieved from the tasks of developing knowledge-level representation, manually modeling knowledge, and programming knowledge-level rules into programs. Instead, we develop a systematic, unified method to model system's self-organization scheme at the signal level (instead of the knowledge level). Thus, the development task tends to be less labor intensive because the algorithm is very systematic and domain independent. No knowledge needs to be programmed into the program and one program is meant for various sensing and action modalities. It does not take much extra effort to address more modalities because each modality uses basically the same program.

The project for developing living machines is not easy. It is naive to think that the Grand Challenge is easy to meet once we have a framework that probably will eventually work. However, with what we have developed in Phase 1, the project appears more tractable than many domain-specific projects which use domain-specific methods within each area, such as vision, speech, hand-written and mix-printed document recognition, autonomous robots, knowledge base development, and language understanding.

4.6 General Models of Science

The development of computer science fits the general models of science described by Kuhn [42]. The history of science has seen long incremental-work periods where work was performed within the established frameworks. Those periods are broken by drastically different new thoughts which lead to a new age of the science. However, revolutionary new thoughts and theories typically were met with skepticism at the beginning. In some cases, the resistance from the establishment was so strong that the new thoughts were not widely adopted until generations later. In some other cases, new thoughts were quickly adopted which led to a significant increase in the importance of the field and its contribution to society. *The living machines* will generate profound impact on not only computer science and engineering, but also biology, education, social, behavioral and economic sciences. This is the eve of the age of living machines.

Bibliography

[1] I. R.. Alexander, G. H. Alexander, and K. F. Lee. Survey of current speech technology. *Communications of the ACM*, 37(3):52–57, March 1994.

[2] J. Aloimonos. Purposive and qualitative active vision. In *Proc. 10th Int'l Conf. Pattern Recognition*, pages 346–360, Atlantic City, NJ, June 1990.

[3] J. R. Anderson. Cognitive and psychological computation with neural models. *IEEE Trans. Systems, Man and Cybernetics*, 13(5):799–815, 1983.

[4] J. R. Anderson. *Cognitive Psychology and Its Implications*. Freeman, New Worky, third edition, 1990.

[5] M. H. Ashcraft. *Human Memory and Cognition*. Harper Collins College Publishers, New Royk, NY, 1994.

[6] A. Bobick and A. Wilson. A state-based technique for the summarization and recognition of gesture. In *Proc. 5th Int'l Conf. Computer Vision*, pages 382–388, Boston, 1995.

[7] L. Breiman, J. Friedman, R. Olshen, and C. Stone. *Classification and Regression Trees*. Chapman & Hall, New York, 1993.

[8] R. Brooks. Intelligence without reason. In *Proc. Int'l Joint Conf. on Artificial Intelligence*, pages 569–595, Sydney, Australia, August 1991.

[9] R. Brooks and L. A. Stein. Building brains for bodies. Technical Report 1439, MIT AI Lab Memo, Chambridge, MA, August 1993.

[10] J. Brotherton and J. Weng. HEARME: Speaker independent word recognition using SHOSLIF. Technical Report CPS 96-33, Department of Computer Science, Michigan State University, East Lansing, MI, Oct. 1996.

[11] P. E. Bryant and T. Trabasso. Transitive inferences and memory in young children. *Nature*, 232:456–458, 1971.

[12] S. Carey. *Conceptual Change in Childhood*. The MIT Press, Chambridge, MA, 1985.

[13] S. Carey. Cognitive development. In D. N. Osherson and E. E. Smith, editors, *Thinking*, pages 147 – 172. MIT Press, Cambridge, MA, 1990.

[14] K. W. Church and L. F. Rau. Commercial applications of natural language processing. *Communications of the ACM*, 38(11):71–79, 1995.

[15] T. M. Cover. Estimation by the nearest neighbor rule. *IEEE Trans. Information Theory*, 14:50–55, Jan. 1968.

[16] T. M. Cover and P. E. Hart. Nearest neighbor pattern classification. *IEEE Trans. Information Theory*, 13:21–27, Jan. 1967.

[17] N. Cui, J. Weng, and P. Cohen. Extended structure and motion analysis from monocular image sequences. *Computer Vision, Graphics, and Image Processing: Image Understanding*, 59(2):154–170, March 1994.

[18] Y. Cui, D. Swets, and J. Weng. Learning-based hand sign recognition using SHOSLIF-M. In *Proc. IEEE Int'l Conf. Comptuer Vision*, pages 631–636, Cambridge, MA, June 1995.

[19] Y. Cui and J. Weng. Hand segmentation using learning-based prediction and verification for hand-sign recognition. In *Proc. IEEE Conf. Comp. Vision Pattern Recognition*, pages 88–93, 1996.

[20] T. Darrell and Alex Pentland. Space-time gesture. In *Proc. IEEE Conf. Comp. Vision Pattern Recognition*, pages 335–340, New York, NY, June 1993.

[21] G. Davenport. Indexes are "out," models are "in". *IEEE Multimedia*, 3(3):10–15, Fall 1996.

[22] E. D. Dickmanns and A. Zapp. A curvature-based scheme for improving road vehicle guidance by computer vision. In *Proc. SPIE Mobile Robot Conf.*, pages 161–168, Cambridge, MA, Oct. 1986.

[23] R. Duda and P. Hart. *Pattern Classification and Scene Analysis*. Wiley, New York, NY, 1973.

[24] K. Fukunaga. *Introduction to Statistical Pattern Recognition*. Academic Press, New York, NY, second edition, 1990.

[25] W. E. L. Grimson and J. L. Mundy. Computer vision applications. *Communications of the ACM*, 37(3):45–51, 1994.

[26] H. E. Gruber and J. J. Voneche. *The essential Piaget*. Basic Books, New York, 1977.

[27] D. J. Hand. *Discrimination and Classification.* Wiley, Chichester, 1981.
[28] H. Hendriks-Jansen. *Catching ourselves in the act: Situated activity, interactive emergence, evolution and human thought.* MIT Press, Cambridge, MA, 1996.
[29] T.S. Huang and V. I. Pavlovic. Hand gesture modeling, analysis, and synthesis. In *Proc. Int'l Workshop on Automatic Face- and Gesture-Recognition*, pages 73–79, 1995.
[30] D. H. Hubel. *Eye, Brain, and Vision.* Scientific American Library, Distributed by Freeman, New York, 1988.
[31] W. Hwang, S. J. Howden, and J. Weng. Performing temporal action with a hand-eye system using the SHOSLIF approach. In *Proc. Int'l Conference on Pattern Recognition*, Vienna, Austria, Aug. 25-30 1996.
[32] Jr. J. L. Martinez and R. P. Kessner (eds.). *Learning & Memory: A Biological View.* Academic Press, San Diego, CA, 2 edition, 1991.
[33] A. K. Jain and R. C. Dubes. *Algorithms for Clustering Data.* Prentice-Hall, New Jersey, 1988.
[34] F. Jelinek. Self-organized language modeling for speech recognition. In A. Waibel and K. Lee, editors, *Readings in Speech Recognition*, pages 450–506. Morgan Kaufmann, San Mateo, CA, 1990.
[35] I. T. Jolliffe. *Principal Component Analysis.* Springer-Verlag, New York, 1986.
[36] T. Kanade, M. Reed, and L. Weiss. New technologies and applications in robotics. *Communications of the ACM*, 37(3):58–67, 1994.
[37] S. B. Kang and K. Ikeuchi. A robot system that observes and replicates grasping tasks. In *Proc. Int'l Conf. Comp. Vision*, pages 1093–1099, Cambridge MA, 1995.
[38] K. Karhunen. Uber lineare methoden in der wahrscheinlichkeitsrechnung. *Ann. Acad. Sci. Fennicae, ser. A1, Math. Phys.*, 37, 1946.
[39] G. Kasparov. The day that I sense a new kind of intelligence. *Time*, page 55, March 25 1996.
[40] R. Kjeldsen and J. Kender. Visual hand gesture recognition for window system control. In *Proc. Int'l Workshop on Automatic Face- and Gesture-Recognition*, pages 184–188, Zurich, Switzerland, June 1995.
[41] B. Kolb and I. Q. Whishaw. *Fundamentals of Human Neuropsychology.* Freeman, New York, third edition, 1990.
[42] Thomas S. Kuhn. *The Structure of Scientific Revolutioon.* University of Chicago Press, Chicago, IL, second edition, 1970.
[43] T. Kuniyoshi, M. Inaba, and H. Inoue. Learning by watching: Extracting reusable task knowledge from visual observation of human performance. *IEEE Trans. on Robotics and Automation*, 10(6):799–822, Dec. 1994.
[44] A. Lanitis, C. J. Taylor, T. F. Cootes, and T. Ahmed. Automatic interpretation of human faces and hand gestures using flexible models. In *Proc. Int'l Workshop on Automatic Face- and Gesture-Recognition*, pages 98–103, Zurich, Switzerland, June 1995.
[45] D. B. Lenat. Cyc: A large-scale investment in knowledge infrastructure. *Communications of the ACM*, 38(11):33–38, 1995.
[46] D. B. Lenat, G. Miller, and T. T. Yokoi. CYC, WordNet, and EDR: Critiques and responses. *Communications of the ACM*, 38(11):45–48, 1995.
[47] R. P. Lippmann. An introduction to computing with neural nets. *IEEE ASSP Magazine*, 4(2):4–22, April 1987.
[48] P. Maes, T. Darrell, B. Blumberg, and A. Pentland. The ALIVE system: Full-body interaction with autonomous agents. In *Proc. of the Computer Animation '95 Conference*, Geneva, Switzerland, April 1996.
[49] J. Mao and A. K. Jain. Artificial neural networks for feature extraction and multivariate data projection. *IEEE Trans. Neural Networks*, 6(2):296–317, March 1995. Outstanding paper award.
[50] M. J. Mataric. Integration of representation into goal-driven behavior-based robots. *IEEE Trans. on Robotics and Automation*, 8(3):304–312, June 1992.
[51] J. M. McInnes and J. A. Treffry. *Deaf-Blind Infants and Children.* University of Toronto Press, Toronto, Ontario, 1982.

[52] M. Meng and A. C. Kak. Mobile robot navigation using neural networks and nonmetrical environment models. *IEEE Control Systems*, pages 31–42, Aug. 1993.

[53] R. Michalski, I. Mozetic, J. Hong, and N. Lavrac. The multi-purpose incremental learning system AQ15 and its testing application to three medical domains. In *Proc. Fifth Annual National Conf. Artificial Intelligence*, pages 1041–1045, Philadelphia, PA, 1986.

[54] G. A. Miller. Worknet: A lexical database for English. *Communications of the ACM*, 38(11):39–41, 1995.

[55] M. Minsky. A selected descriptor-indexed bibliography to the literature on artificial intelligence. In E. A. Feigenbaum and J. Feldman, editors, *Computers and Thought*, pages 453–523. McGraw-Hill, New York, NY, 1963.

[56] S. Negahdaripour and A. K. Jain, editors. *Final Report of the NSF Workshop on the Challenges in Computer Vision Research: Future Directions of Research*. June 7-8, 1991.

[57] N. Nilsson. SRI A.I. center technical note. Technical Report 323, Stanford Research Institute, April 1984.

[58] T. Poggio and F. Girosi. Networks for approximation and learning. *Proceedings of The IEEE*, 78(9):1481–1497, 1990.

[59] D. A. Pomerleau. Efficient training of artificial neural networks for autonomous navigation. *Neural Computation*, 3(1):88–97, 1991.

[60] M. I. Posner and M. E. Raichle. *Images of Mind*. Scientific American Library, New York, 1994.

[61] J. Quinlan. Introduction of decision trees. *Machine Learning*, 1(1):81–106, 1986.

[62] R. E. Kahn R. J. Firby, P. N. Prokopowicz, and M. J. Swain. Collecting trash: A test of purposive vision. In *Proc. Workshop on Vision for Robots*, pages 18–27, Pittsburgh, PA, August 1995.

[63] L. R. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of IEEE*, 77(2):257–286, 1989.

[64] V. S. Ramachandran. Perceiving shape from shading. In I. Rock, editor, *The Perceptual World*, pages 127–138. Freeman, San Francisco, CA, 1990.

[65] M. Rosenblum and Larry S. Davis. An improved radial basis function network for visual autonomous road following. *IEEE Trans. Neural Networks*, 7(5):1111–1120, 1996.

[66] J. Rubner and K. Schulten. Development of feature detectors by self-organization. *Biological Cybernetics*, 62:193–199, 1990.

[67] J. Rubner and P. Tavan. A self-organizing network for principal component analysis. *Europhysics Letters*, 10:693–698, 1989.

[68] O. Sacks. To see and not see. *The New Yorker*, pages 59–73, May 1993.

[69] Wei-Min Shen. *Autonomous Learning from the Environment*. Computer Science Press, New York, 1994.

[70] T. Starner and A. Pentland. Visual recognition of American sign language using hidden Markov models. In *Proc. Int'l Workshop on Automatic Face- and Gesture-Recognition*, pages 189–194, Zurich, Switzerland, June 1995.

[71] D. Swets, B. Punch, and J. Weng. Genetic algorithm for object recognition in a complex scene. In *Proc. Int'l Conf. on Image Processing*, Washington, D.C., Oct. 22-25 1995.

[72] D. Swets and J. Weng. Efficient content-based image retrieval using automatic feature selection. In *Proc. IEEE Int'l Symposium on Computer Vision*, pages 85–90, Coral Gables, FL, Nov. 1995.

[73] D. Swets and J. Weng. The self-organizing hierarchical optimal subspace learning and inference framework for object recognition. In *Proc. Int'l Conf. Neural Networks and Signal Processing*, Nanjing, China, Dec. 10-13 1995.

[74] D. L. Swets, Y. Pathak, and J. Weng. A system for combining traditional alphanumeric queries with content-based queries by example in image databases. *Multimedia Tools and Application*, 1996.

[75] D. L. Swets and J. Weng. Efficient content-based image retrieval using automatic feature selection. In *Proc. IEEE Int'l Conf. on Neural Networks*, Perth, Australia, Nov. 27-Dec. 1 1995.

[76] D. L. Swets and J. Weng. Image-based recognition using learning for generalizing parameters. In *Proc. 2nd Asian Conf. on Computer Vision*, Singapore, Dec. 5-8 1995.

[77] D. L. Swets and J. Weng. Using discriminant eigenfeatures for image retrieval. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 18(8):831–836, 1996.

[78] D. Terzopoulos, X. Tu, and R. Grzeszczuk. Artificial fishes with autonomous locomotion, perception, behavior and learning in a simulated physical world. In *Artificial Life IV: Proc. Fourth Int'l Workshop on the Synthesis and Simulation of Living Systems*, pages 17–27, Cambridge, MA, July 1994. MIT Press.

[79] P. Thompson. Margaret thatcher: a new illusion. *Perception*, 9:483–484, 1980.

[80] C. Thorpe, M. H. Hebert, T. Kanade, and S. Shafer. Vision and navigation for the Carnegie-Mellon Navlab. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 10(3):362–373, 1988.

[81] C. Thorpe, M. Herbert, T. Kanade, and S. Shafer. Toward autonomous driving: The CMU Navlab. *IEEE Expert*, 6(4):31–42, August 1991.

[82] J. Triesch and C. von der Malsburg. Robust classification of hand posture against complex background. In *Proc. Int'l Conf. on Automatic Face- and Gesture-Recognition*, pages 170–175, Killington, Vermont, Oct. 1996.

[83] A. M. Turing. On computable numbers with an application to the Entscheidungsproblem. *Proc. London Math. Soc.*, 2(42):230–265, 1936.

[84] M. Turk and A. Pentland. Eigenfaces for recognition. *Journal of Cognitive Neuroscience*, 3(1):71–86, 1991.

[85] M. A. Turk, D. G. Morgenthaler, K. D. Gremban, and M. Marra. VITS-a vision system for autonomous land vehicle navigation. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 10(3):342–361, May 1988.

[86] A. Waibel and K. Lee. *Readings in Speech Recognition*. Morgan Kaufmann, San Mateo, CA, 1990.

[87] J. Weng. Image matching using the windowed Fourier phase. *Int'l Journal of Computer Vision*, 11(3):211–236, 1993.

[88] J. Weng. Windowed Fourier phase: completeness and signal reconstruction. *IEEE Trans. on Signal Processing*, 41(2):657–666, Feb. 1993.

[89] J. Weng. On comprehensive visual learning. In *Proc. NSF/ARPA Workshop on Performance vs. Methodology in Computer Vision*, pages 152–166, Seattle, WA, June 1994.

[90] J. Weng. SHOSLIF: A framework for object recognition from images. In *Proc. IEEE International Conference on Neural Networks*, pages 4204–4209, Orlando, FL, June 28-July 2 1994.

[91] J. Weng. SHOSLIF: A learning system for vision and control. In *Proc. IEEE Annual Workshop on Architectures for Intelligent Control Systems*, Columbus, Ohio, August 16 1994.

[92] J. Weng. SHOSLIF: A framework for sensor-based learning for high-dimensional complex systems. In *Proc. IEEE Workshop on Architectures for Semiotic Modeling and situation analysis in Large Complex Systems*, pages 303–313, Monterey, CA, Aug. 1995.

[93] J. Weng. Cresceptron and SHOSLIF: Toward comprehensive visual learning. In S. K. Nayar and T. Poggio, editors, *Early Visual Learning*. Oxford University Press, New York, 1996.

[94] J. Weng and N. Ahuja. Octree of objects in arbitrary motion: representation and efficiency. *Computer Vision, Graphics, and Image Processing*, 39:167–185, 1987.

[95] J. Weng, N. Ahuja, and T. S. Huang. Motion and structure from point correspondences: planar surfaces. *IEEE Trans. on Signal Processing*, 39(12):2691–2717, Dec. 1991.

[96] J. Weng, N. Ahuja, and T. S. Huang. Matching two perspective views. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 14(8):806–825, Aug. 1992.

[97] J. Weng, N. Ahuja, and T. S. Huang. Optimal motion and structure estimation. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 15(9):864–884, Sept. 1993.

[98] J. Weng, N. Ahuja, and T. S. Huang. Learning recognition using the Cresceptron. *Int'l Journal of Computer Vision*, 25(2), Nov. 1997.

[99] J. Weng and S. Chen. Incremental learning for vision-based navigation. In *Proc. Int'l Conf. Pattern Recognition*, volume IV, pages 45–49, Vienna, Austria, Aug. 25-30 1996.

[100] J. Weng, P. Cohen, and M. Herniou. Camera calibration with distortion models and accuracy evaluation. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 14(10):965–980, Oct. 1992.

[101] J. Weng, P. Cohen, and N. Rebibo. Motion and structure estimation from stereo image sequences. *IEEE Trans. on Robotics and Automation*, 8(3):362–382, June 1992.

[102] J. Weng, Y. Cui, and N. Ahuja. Transitory image sequences, asymptotic properties, and estimation of motion and structure. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 1996. accepted and to appear.

[103] J. Weng, Y. Cui, N. Ahuja, and A. Singh. Integration of transitory image sequences. In *IEEE Conf. Computer Vision and Pattern Recognition*, pages 966–969, Seattle, Washington, June 20-23 1994.

[104] J. Weng and T. S. Huang. 3-D motion analysis from image sequences using point correspondences. In C. H. Chen, L. F. Pau, and P. S. Wang, editors, *Handbook of Pattern Recognition and Computer Vision*, pages 395–441. World Scientific Publishing, River Edge, NJ, 1993.

[105] J. Weng, T. S. Huang, and N. Ahuja. 3-D motion estimation, understanding and prediction from noisy image sequences. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 9(3):370–389, May 1987.

[106] J. Weng, T. S. Huang, and N. Ahuja. Motion and structure from two perspective views: algorithm, error analysis and error estimation. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 11(5):451–476, May 1989.

[107] J. Weng, T. S. Huang, and N. Ahuja. Motion and structure from line correspondences: closed-form solution, uniqueness, and optimization. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 14(3):318–336, March 1992.

[108] J. Weng, T. S. Huang, and N. Ahuja. *Motion and Structure from Image Sequences*. Springer-Verlag, New York, 1993.

[109] D. Wettergreen, C. Thorpe, and W. Whittaker. Exploring Mount Erebus by walking robot. In *Proc. Int'l Conf. on Autonomous Intelligent Systems*, pages 172–181, Pittsburgh, PA, 1993.

[110] S. S. Wilks. *Mathematical Statistics*. Wiley, New York, NY, 1963.

[111] R. Wright. Can machines think? *Time*, pages 50–56, March 25 1996.

[112] C. Yoken. *Living with deaf-blindness: nine profiles*. National Academy of Gallaudet College, Washington, 1979.

[113] S. R. Young, A. G. Hauptmann, W. H. Ward, E. T. Smith, and P. Werner. High-level knowledge sources in usable speech recognition systems. *Communications of the ACM*, 32(2):183–194, 1989.

[114] S. Zeki. The visual image in mind and brain. *Scientific American*, 267(3):69–76, 1992.