Autonomous Mental Development

ICDL 2004 Tutorial

Juyang Weng

Embodied Intelligence Laboratory
Department of Computer Science
Michigan State University
East Lansing MI 48824 USA
http://www.cse.msu.edu/~weng/

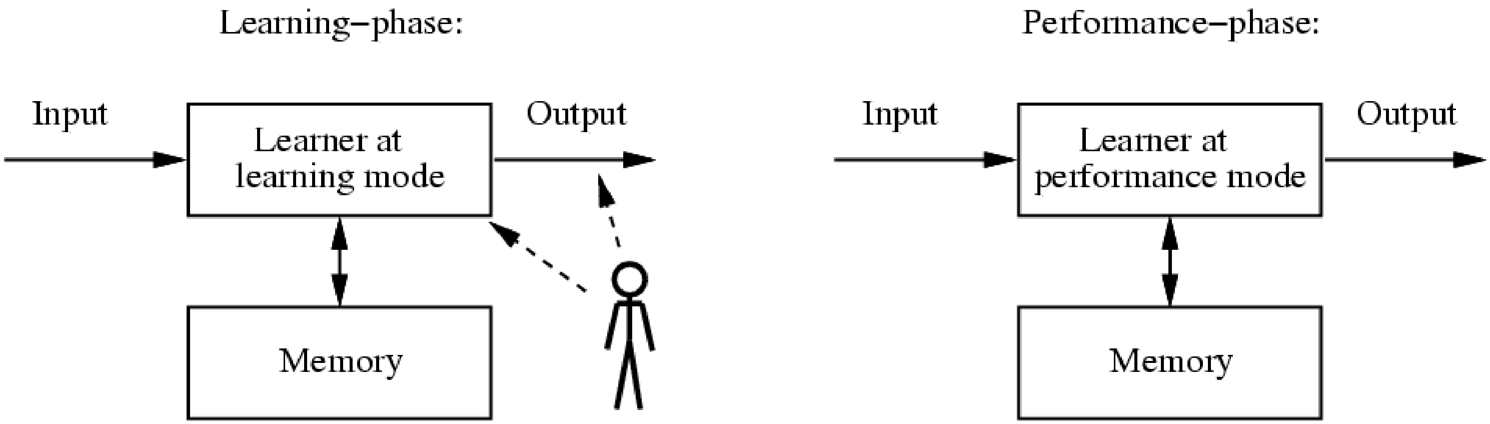
1. Overview of Approaches

How We Came Along?

- Turing test and dreams of AI
- Knowledge based approach
  - Book *“Computers and Thought”*
  - CYC, WordNet
  - Marr’s primal sketch
  - Expert systems
  - Work in vision, such as stereo and motion

Learning-Based Approaches

- Turing's imaginary "child machine"
- Pre-designed representation for a given task
- Some undetermined parameters in the representation



Neural Networks

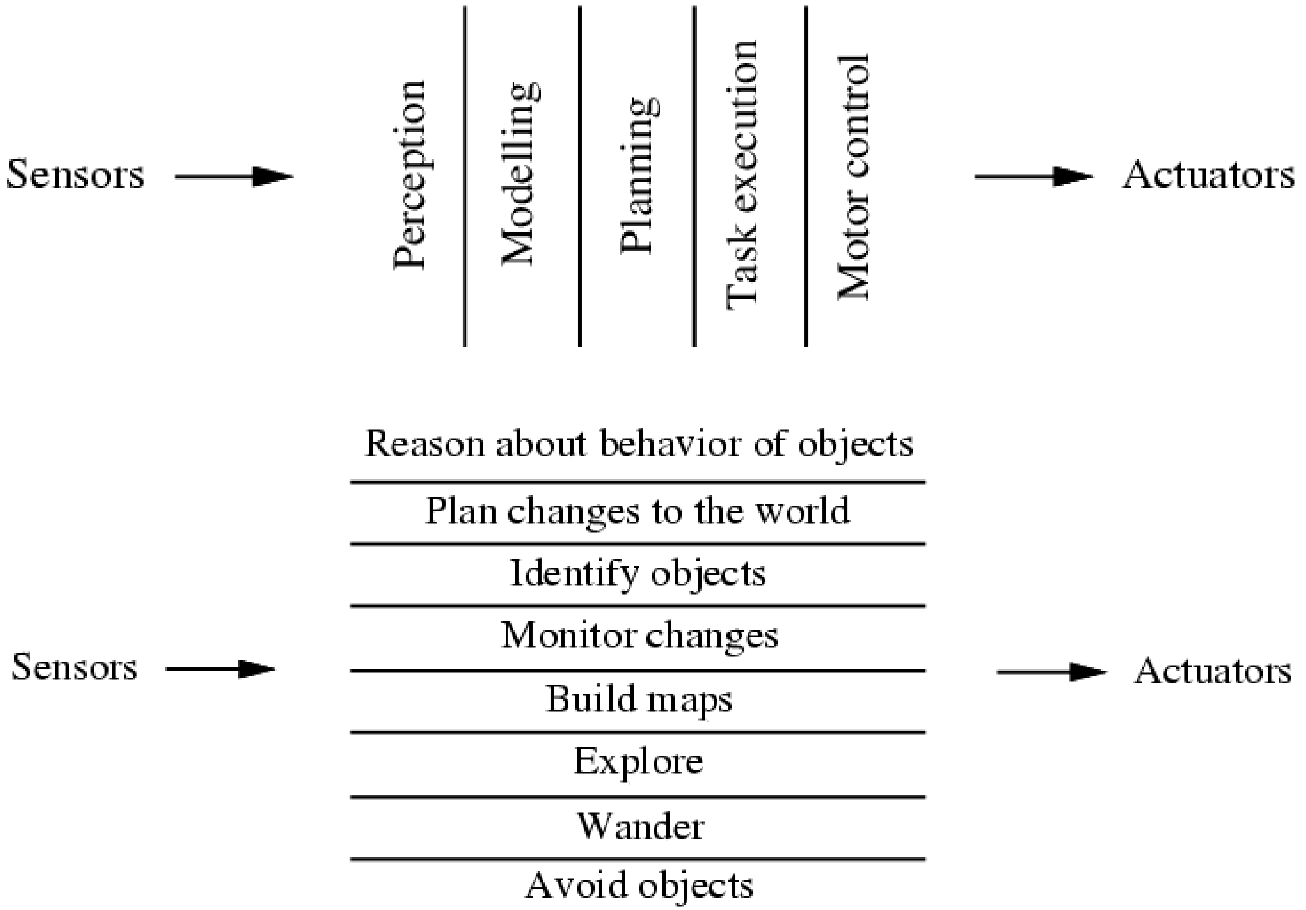
- Numeric representation
- Learning as a regression problem
- Feed forward network: state less
- Recurrent network: with state
- Supervised learning and reinforcement learning
- Most incremental
- System example: ALVINN by Dean Pomerleau

A Model by James Albus

(James Albus)

- Symbolic representation
- Belongs to behavior-based approach
- Four-element module:
  Sensory processing (SP), world modeling (WM), value judgement (VG) and behavior generation (BG)
- Hierarchy in sensory space and behavior space
- Not meant for automatic development
- An architecture outline, missing some crucial detail
- Has not implemented yet

Behavior-Based Framework by Brooks



Sensors → Perception | Modelling | Planning | Task execution | Motor control → Actuators

Sensors →

Reason about behavior of objects

Plan changes to the world

Identify objects

Monitor changes

Build maps

Explore

Wander

Avoid objects

→ Actuators

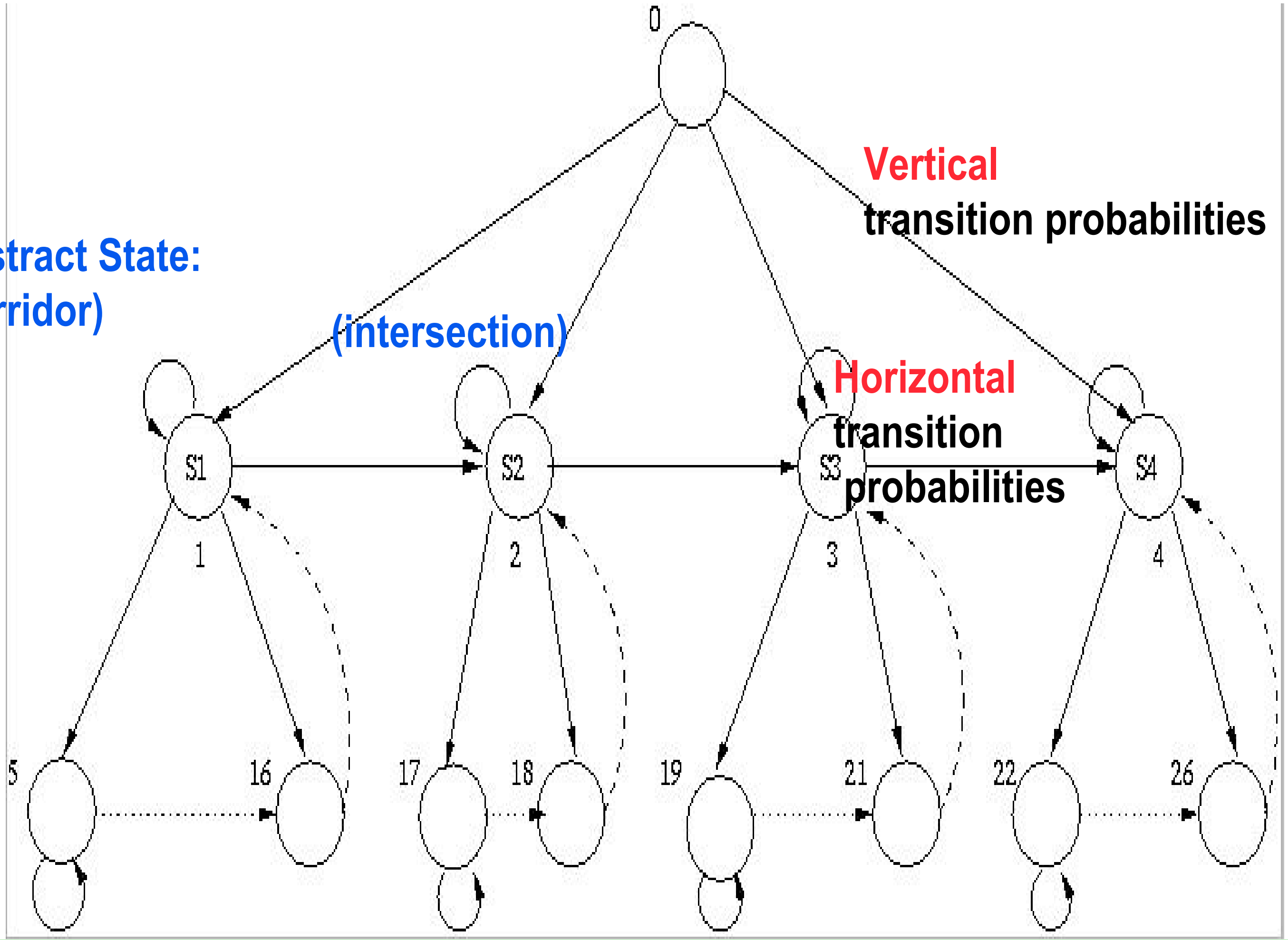
Behavior-based Methods

- Do not require explicit world representation
- Symbolic states of the environment
- Emphasis on generation of behaviors
- Hand-programmed, supervised learning and reinforcement learning
- Use of probabilistic models to improve system reliability
- System example: Pavlov by S. Mahadevan

Markov Decision Process

- One of the most general frameworks for learning
- State is about the world, not internal state
- Partially observable version: state of world is not totally observable.
- Programmer typically imposes task-specific internal representation:
  - The number of states
  - The distribution restriction (e.g., left to right model)
  - Initial estimate of probability (transition and observation)
- Difficult to grow (develop)
- Hard to scale up

Hierarchical HMM:
Multi-level Stochastic Processes (Mahadevan)



Abstract State:
(corridor)

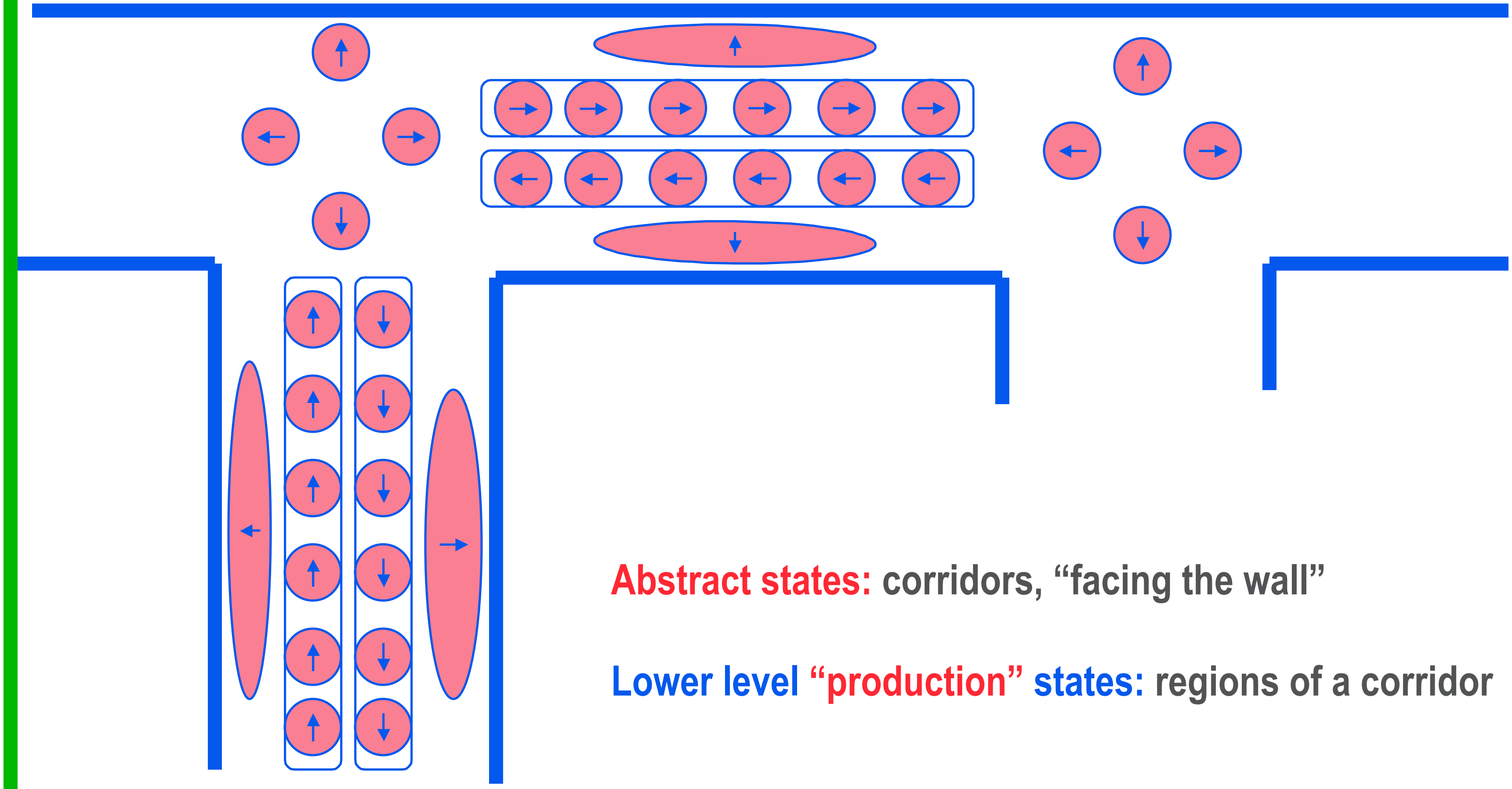
(intersection)

Vertical
transition probabilities

Horizontal
transition
probabilities

Production
State:
(segment)

Partially Observable Markov Decision Process (Mahadevan)



Abstract states: corridors, “facing the wall”

Lower level “production” states: regions of a corridor

Evolutional Approach

- Steps:
  - Task definition
  - Problem formulation
  - Chromosome representation
  - Population generation
  - Fitness computation
  - Mate and reproduction
  - Generation replacement
  - Repeat above steps

Evaluation of Evolutional Approach

- Pros:
  - Can perform high dimensional search
  - Simpler programming
  - For highly complex fitness functions
- Cons:
  - Extremely slow
  - Computationally expensive
  - For a given task
  - Human designed task-specific chromosome representation

Lessons We Learned

- AI fragmentation: Task-specific
- Humans design task-specific representation
- Learning is not autonomous
- Learning is off-line
- Learning and performance are separate phases
- Machines cannot acquire tasks autonomously
- Task-specific representation cannot scale up to more other tasks

2. AMD Approach

Change of Engineering Paradigm

- Traditional paradigm:
  - Start with a task and the environment. Humans understand the task, not the machine
  - Humans design task-specific representation
  - Task-specific programming plus task specific learning
  - Run the program to perform
- New paradigm:
  - Given rough ecological conditions of muddy environment, design a robot body
  - Design developmental program
  - Birth: run the developmental program
  - Develop mind: robot autonomously interacts with the world

Alan Turing's Child Machine

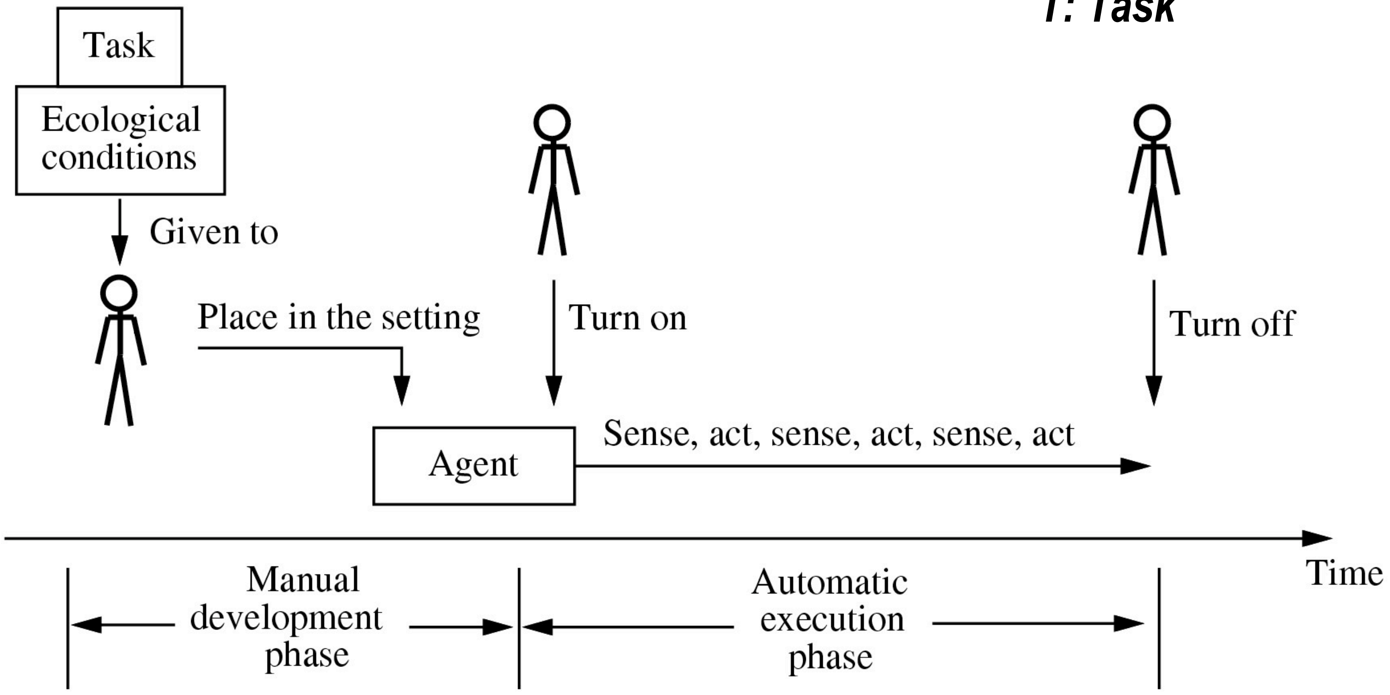
Our hope is that there is so little mechanism in the child brain that something like it can be easily programmed. The amount of work in the education … to be much the same as for the human child

- Alan Turing, 1950

Traditional Manual Development

$$A = H(E_c\ ,\ T)$$

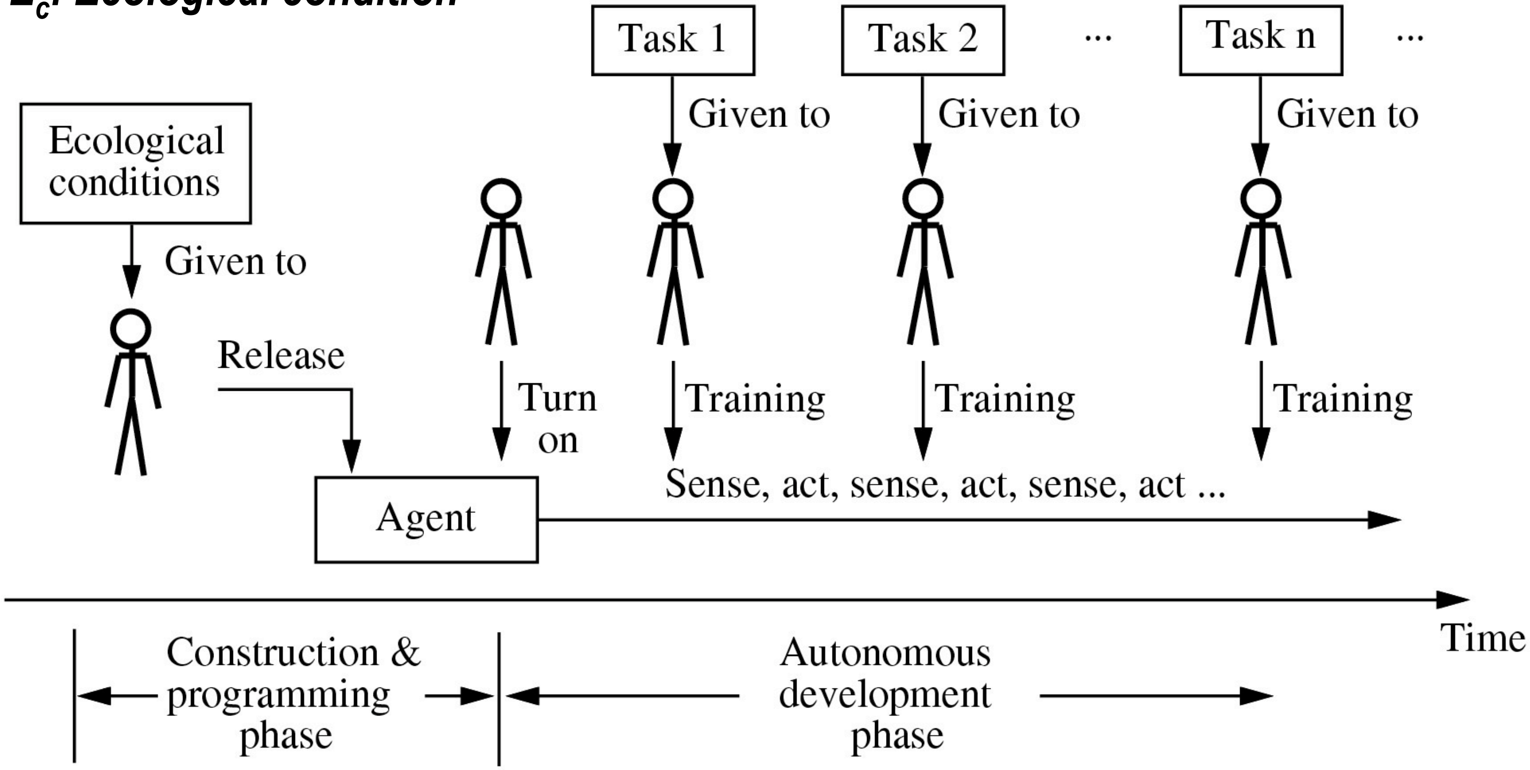
A: agent
H: human
E_c: Ecological condition
T: Task



New Autonomous Development

A*: *agent
H*: *human
E_c*: *Ecological condition

$$A = H(E_c)$$



Task Nonspecificity

- A program is not task specific means:
  1. Open to muddy environment
  2. Tasks are unknown at programming time
  3. “The brain” is closed after the birth
  4. Learn an open number of muddy tasks after birth
- Avoid trivial cases:
  - A thermostat
  - A robot that does task A when temperature is high and does task B when temperature is low
  - A robot that does simple reinforcement learning

Two Types of Work on Robot Development

- Emulate some phenomena in the developmental process: About development but not AMD
  - Useful as a component study
  - Not AMD by itself
  - Setting specific
  - Task specific
- AMD: what we mean by developmental approach
  - Eight operational requirements
  - Aim at muddy environments and muddy tasks

Comparison of Approaches

ApproachesSpeciesArchitectureWorld KnowledgeSystem behaviorTask-specific**Knowledge-based**Prograr

8 Requirements for Practical AMD

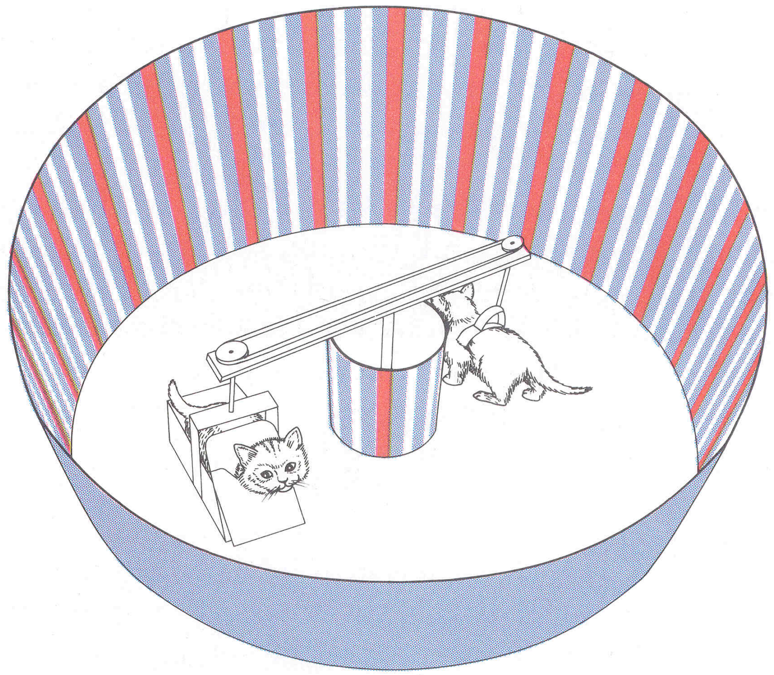
- Eight necessary operational requirements:
  - Environmental openness: muddy environments
  - High dimensional sensing, but without loss of essential information
  - Online
  - Real time speed, with a large memory
  - Incremental:
    for each fraction of second (e.g., 10-30Hz)
  - One-instance learning
  - Mixed learning modes
  - Muddy tasks
- Existing works (other than SAIL) aimed at some, but not all.
- SAIL deals with the 8 requirements altogether

3. Neuroscience and Developmental Psychology

Why Autonomous Mental Development?

- **Developmental mechanisms are easier to program:** ***lower level, more systematic, task-independent, clearly understandable***
- **Relieve humans from intractable programming tasks:** ***vision, speech, language, complex behaviors, consciousness***
- **User-friendly machines and robots:** ***humans issue high-level commands to machines***
- **Highly adaptive manufacturing systems (e.g., self-trainable, reconfigurable machining systems)**
- **Help to understand human intelligence**

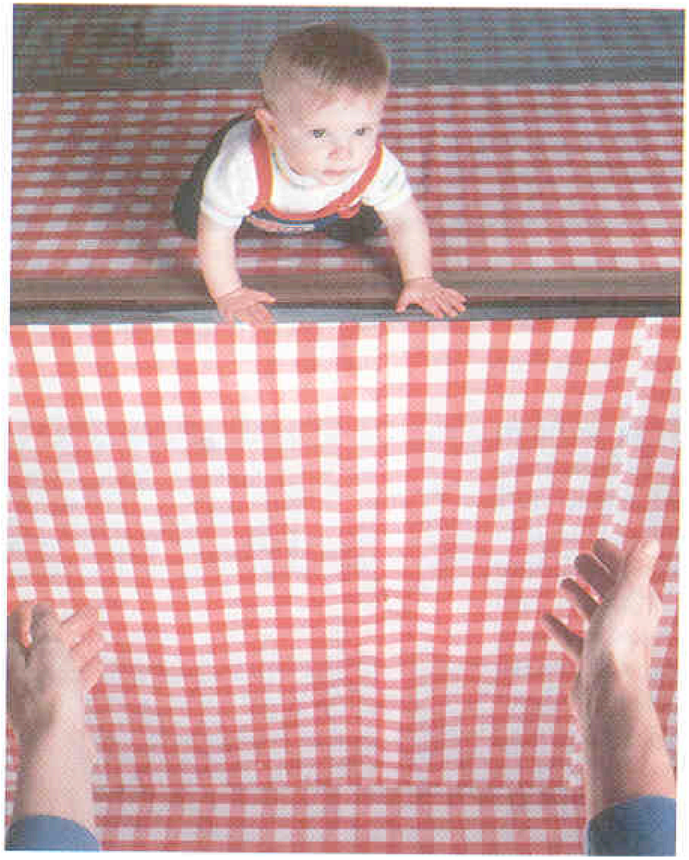
Why Active Body?
Kitten Carousel Experiment



- A classic study by Held & Hein 1963
- Kittens raised from birth in total darkness
- When old enough to walk, placed in "kitten carousel" for 42 days
- One kitten harnessed to pull the carousel
- Another just being carried in a box.
- The behavior of the kittens is strikingly different at 'visual cliff'.
- Thus, *autonomous* actions are very important to understanding.

Visual Cliff

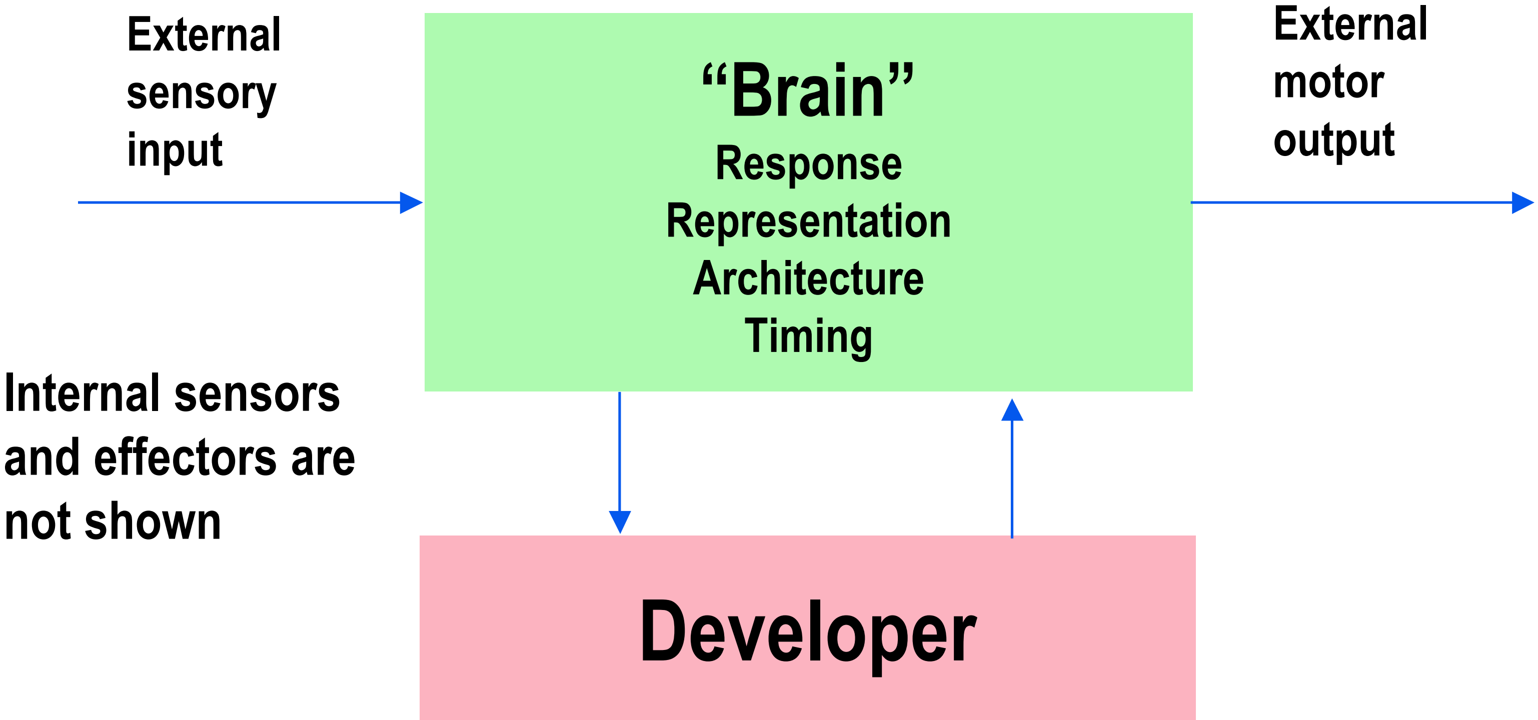
- Visual cliff:
  - A transparent platform
  - Visual sharp drop in elevation
- Human infants:
  - 6 – 8 months old, a week or two after they began to crawl
  - all would cross a visual cliff in initial trials
  - They became increasingly reluctant to cross in later trials, although nothing bad had happened during crossing.
- Carousel kittens:
  - Passive one does not fear
  - Active one does
- Implication:
  - Vision is very much developed from experience!



Stages of Cognitive Development

Age	Stage	Characteristics
Birth - 2	Sensorimotor	Coordinating sensory perception with motor behaviors; Not capable of symbolic representation
2 - 6	Pre-operational	Ego-centric; Captured by surface appearance;
6 - 12	Concrete operational	Operational thinking with concrete objects and actions; Doing so in the presence of concrete objects and events
12 -	Formal operational	Abstract, formal, deductive reasoning

Robot “Brain” and Its Developer



External sensory input

“Brain”

Response

Representation

Architecture

Timing

External motor output

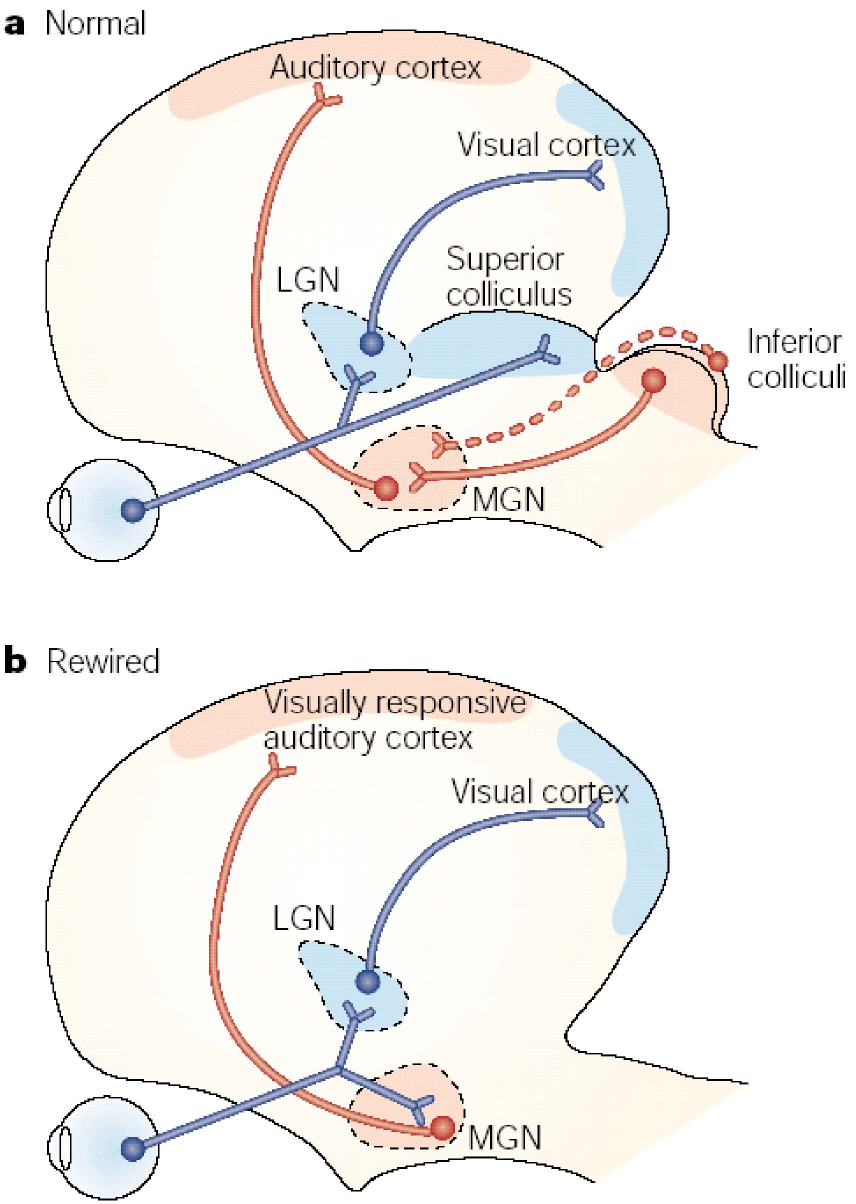
Internal sensors and effectors are not shown

Developer

Cross-Modality Cortex Plasticity
(Mriganka Sur and coworkers, *Nature* April 2000)

- Rewiring:
  - Ferret
  - Visual signal is rewired to auditory cortex early in life
- Results:
  - Orientation selectivity appeared in rewired auditory cortex, statistically identical
  - The ferrets have been successfully trained to perform vision tasks using auditory cortex
- Implication:
  - Similar developmental mechanisms shared by different sending modalities

Rewiring


a Normal
Auditory cortex
Visual cortex
LGN
Superior colliculus
Inferior colliculi
MGN
b Rewired
Visually responsive auditory cortex
Visual cortex
LGN
MGN

Receptive Fields Change with Experience
(Mike Merzenich and coworkers)

- Experiment:
  - Adult owl monkeys
  - Synchronized stimulus cross fingers
  - Repeated training for weeks
- Result:
  - Receptive fields cross fingers
  - Normal cases: receptive fields cover a single finger
- Implication:
  - Even adults are developing every day!

Developmental Psychology

- **Biological-maturation perspective**
  Maturation of central nervous system in explaining early behavior in infants
- **Environmental-learning perspective**
  Contribution of the environment
- **Constructivist perspective**
  Jean Piaget: active and constructive
- **Cultural-context perspective**
  The impact of custom and culture
- **Current lack of computational perspective**

Animal Learning Models

- Nonassociative learning:
  habituation. (e.g., bored to seeing the same toy)
  sensitization (e.g., after being startled by a snake, startle by a rope)
- Classical conditioning:
  Training: CS - US – UR (e.g., tone - food - salivation)
  Result: CS – CR (e.g., tone – salivation)
  C: conditional; U: unconditional; S: stimulus, R: response
- Instrumental conditioning (also reinforcement, shaping):
  Training: E - R – UR
  (e.g., red/green buttons – press red – shock;
  red/green buttons – press green – juice)
  Result: R or avoidance depending on UR (e.g., pressing green not red)
- Animal cognitive learning
  No apparent reinforcer, very complex behaviors, establishing value system
  (e.g., following owner's instructions, kids learning at schools)

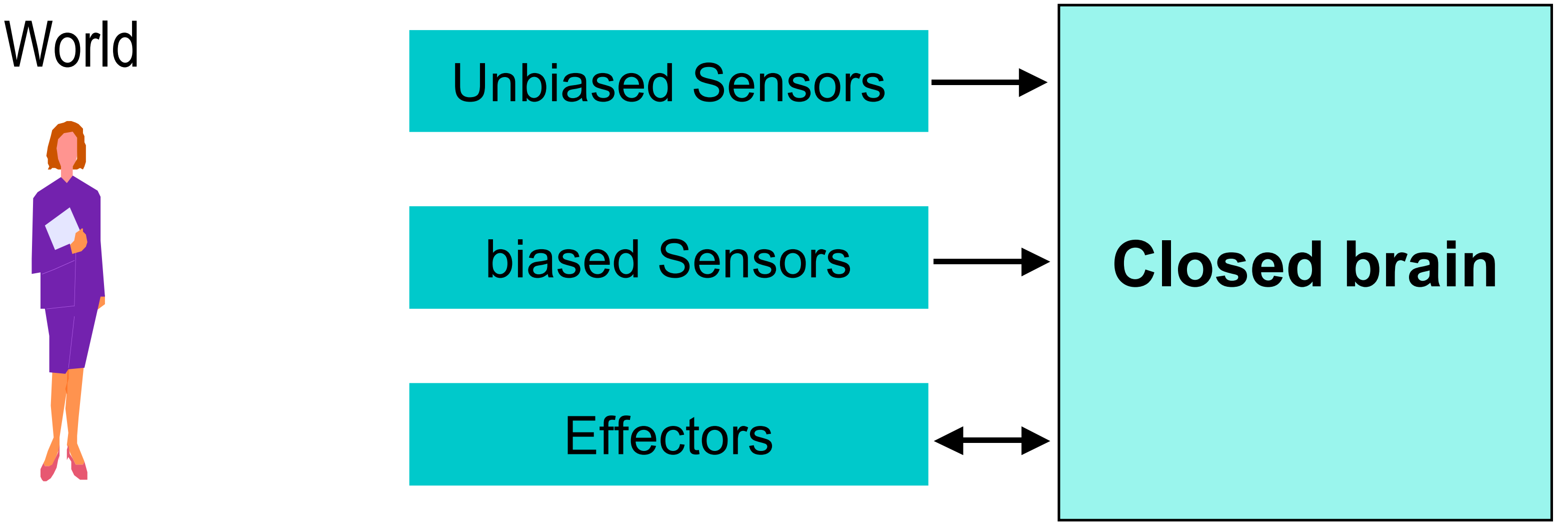
4. Learning Types for Machines

Existing Machine Learning Types

- Supervised learning
  Class labels (or actions) are given in training
- Unsupervised learning
  Class labels (or actions) are not given in training
- Reinforcement learning
  Class labels (or actions) are not given in training but reinforcement (score) is given

Mode of Development: AA-Learning

AA-learning: Automated animal-like learning



World

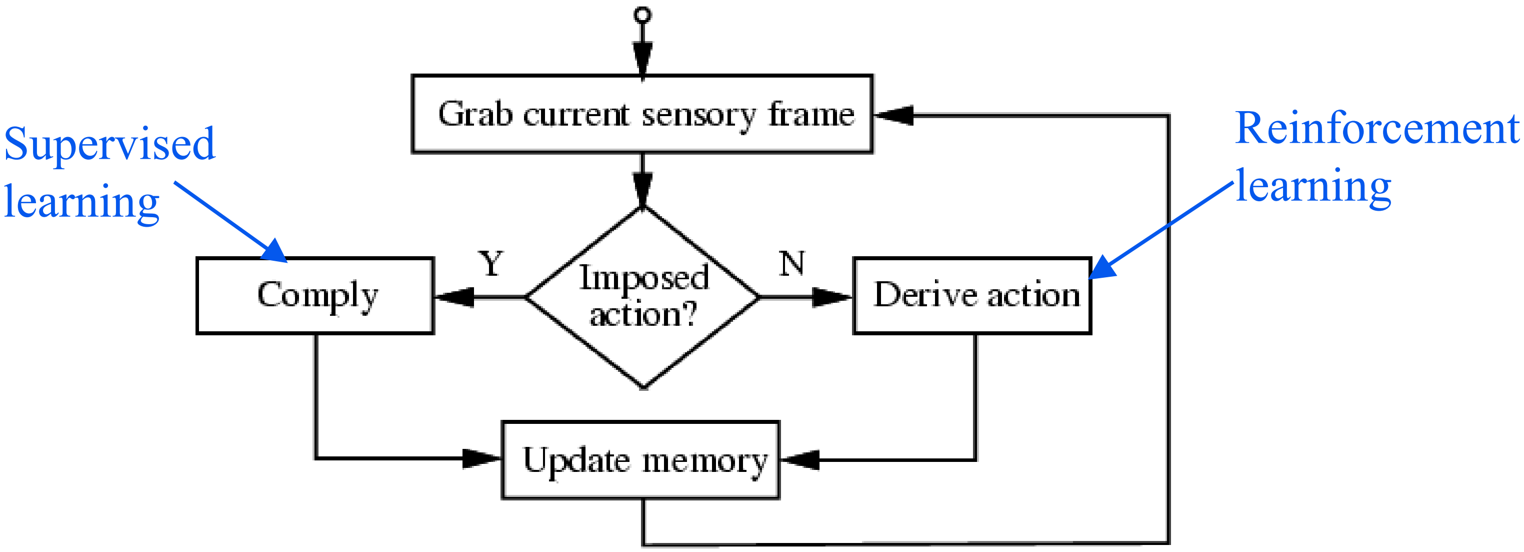
Unbiased Sensors

biased Sensors

Effectors

Closed brain

Flowchart of AMD Learning



- **Modified Q-learning: integration of supervised learning and reinforcement learning**

New Classification for Machine Learning

- Need for considering state imposability after the task is given
- 3-tuple (s, b, e):
  *s*tate imposable, *b*iased sensor, *e*ffector
- State: state imposable after the task is given
- Biased sensor: whether the biased sensor is used
- Effector: whether the effector is imposed

8 Types of Machine Learning

Learning type 0-7 is based on 3-tuple (s, b, e):

***S*tate imposable (s=1), *b*iased sensor used (b=1) *e*ffector-imposed (e=1)**

TypeStateBiasedEffector**0 (000)**state-autonomouscommunicativeeffect

Existing Developmental Robots

- **Darwin robot:**
  - Series:
    - Darwin IV: Classical condition
    - Darwin V: Reinforcement learning
  - Constrained environment (two types of cubes)
- **MSU:**
  - Series:
    - Cresceptron (91 – 95): grow architecture
    - SHOSLIF (93 – 00): scalable real-time regression
    - SAIL (95 – present): our first AMD robot
    - Dav (99 – present): the next generation of developmental humanoid
  - Unconstrained environments

5. Representation

Two Types of Concept

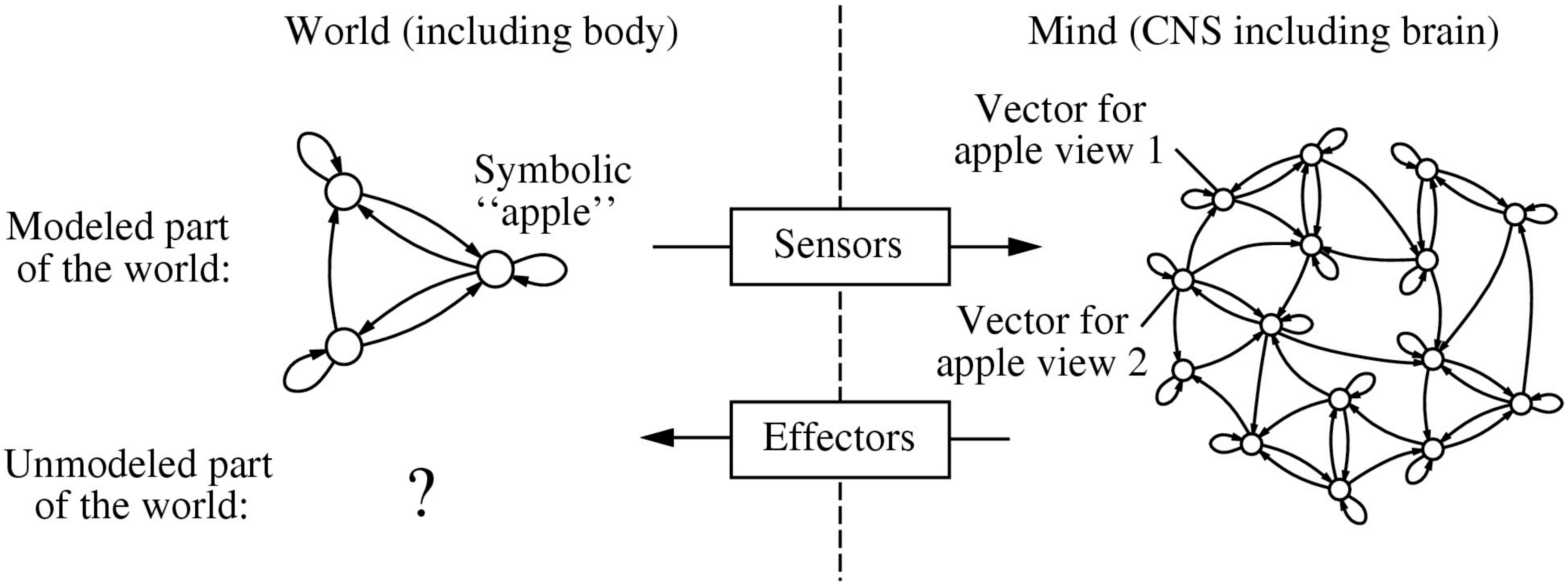
- **World concept:**
  a concept about objects in the external environment and their properties
  E.g., In front of the agent, there is an apple.
  Properties: **grounded in the world, well understood by the human society**
- **Mind concept:**
  a concept that is internal with respect to a nervous system
  E.g., In front of me (agent) there is a pear.
  Properties: **individualized, incomplete, not necessarily a correct representation of the real world**

World Centered vs Body Centered

- World centered:
  Every item corresponds to a world concept

- Body centered:
  Every item corresponds to a mind concept

World Concept and Mind Concept



World (including body)

Mind (CNS including brain)

Modeled part of the world:

Symbolic “apple”

Unmodeled part of the world:

?

Sensors

Effectors

Vector for apple view 1

Vector for apple view 2

Symbolic vs. Numeric Representation

- Symbolic:
  use symbols to represent objects.
  E.g., name, weight, house, neuron, signal
- Numeric:
  use numeric numbers to represent objects.
  E.g., value of a pixel, the firing rate of a neuron

World Centered Symbolic Representation

- World-centered symbolic representation:
  - World centered: one-to-one correspondence between a world object and an instance of representation type
  - In the form of $A = (v_1, v_2, \cdots, v_n)$
  - Example: Apple = (weight, color)
- Properties:
  - Each component (attribute) has a predefined meaning
  - Each attribute is represented by a unique variable
  - Each object (e.g., apple) is uniquely represented by an instance

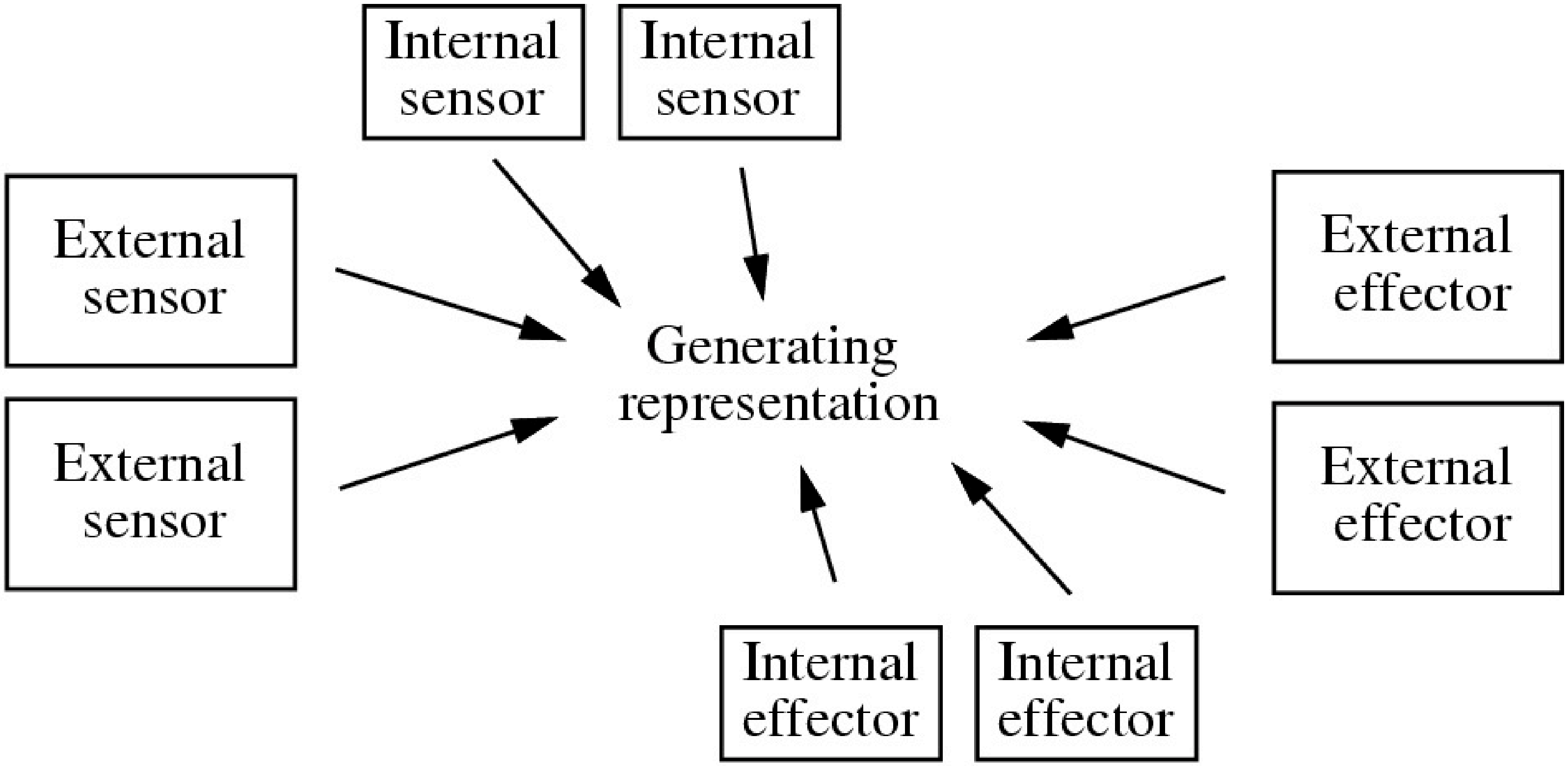
Mind Centered Numeric Representations

- **Body centered numeric representation:**
  - Body centered
  - In the form $A = (v_1, v_2, \cdots, v_n)$
  - But each component corresponds to
    - A sensory element
    - A motor control terminal
    - Or a function of a multiple of the above
  - Example: a brain image
- **Properties:**
  - Each component often does not correspond to any world concept

Type of Internal Representation for AMD

- Internal representation: non-terminal (not sensor and effector ends)
- World-centered symbolic representation is not suited for internal representation for AMD:
  It is *world centered, symbolic, not suited* for internal representation of a developing brain
- A body-centered numeric representation is suited for internal representation for AMD:
  It does not have to be a one-to-one correspondence to a world concept, could be body centered
- Implication to human brain?

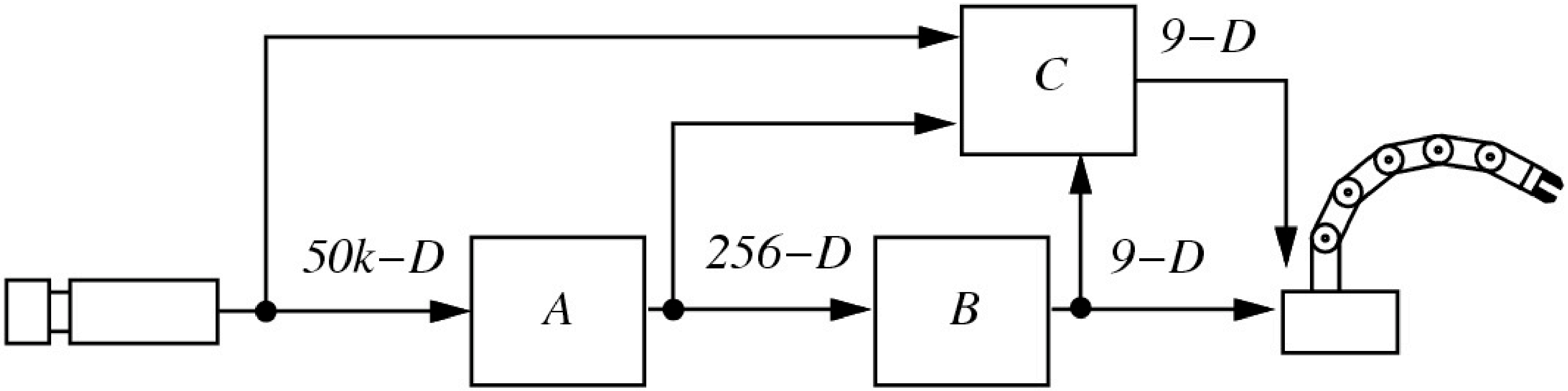
Sources for Grow Representation



Epigenetic Representation

1. Raw vector from sensors.
2. Raw control vector to effectors generated by task-nonspecific program.
3. Representation generated by a task-nonspecific program using the input of epigenetic representation.
4. Nothing other than those generated by recursive applications of the above three steps.

An Example

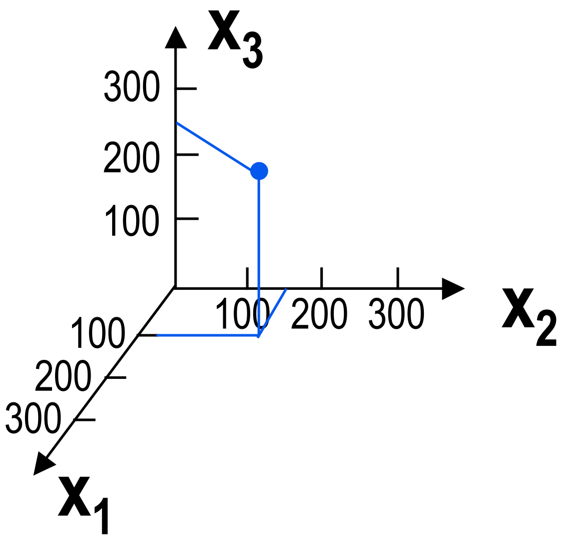


Raw Sensory Representation

- Vector representation
- Sensor independent:
  - Individual sensors: e.g. light sensors
  - Linear sensory array: e.g., linear camera
  - Surface sensory array: e.g., image
  - Volume sensory array: e.g., video

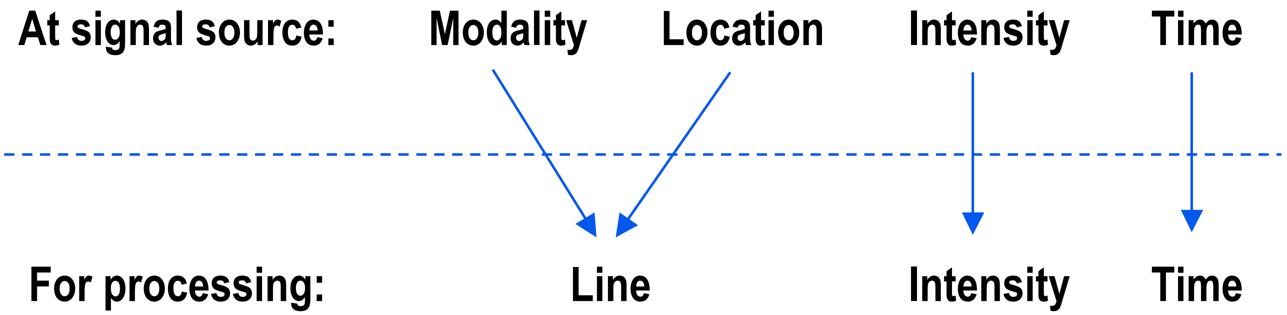
Image:	■	▨	□
Intensity:	100	150	250

Representation:



Attributes of Signal Source

- Avoid symbolic representation
- Four attributes at signal source:
  Modality, location, intensity and time
- Three attributes for processing:
  Line, intensity and time



At signal source: Modality Location Intensity Time

For processing: Line Intensity Time

Information Hierarchy

- **Response level**

  e.g. Inborn behaviors and learned cognitive skills.

- **Representation level**

  e.g., neural weights

- **Architecture level**

  e.g., a cortex area is prepared for eyes and how neurons are connected.

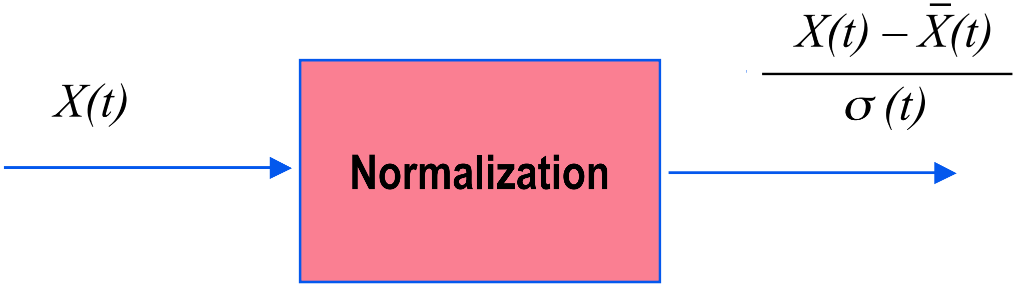
- **Timing level**

  e.g., the time schedule of neural growth

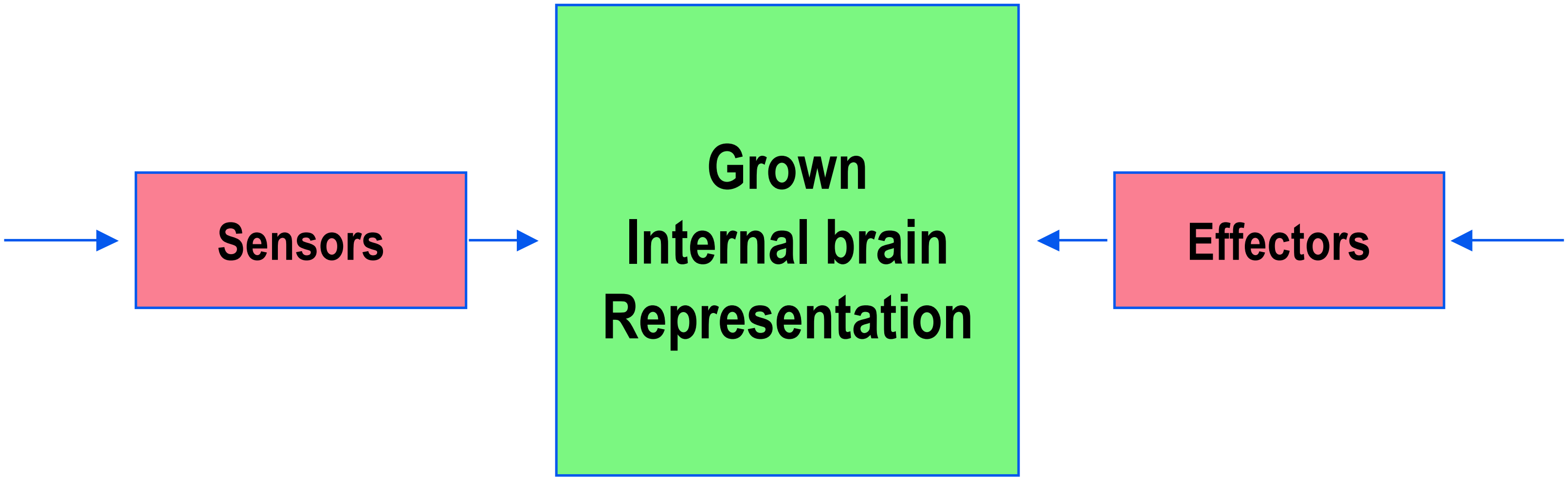
The lower the level, the more is wired in.
But all are experience-dependent.

Self-Adaptive Position and Scale

- Too many lines with different meanings
- Position: mean
- Scale: variance



How to Grow Internal Representation?



No consistent access of world concept into internal representation!

6. Mental Architecture

Past Major Work on Mental Architecture

- Perception architectures:
  - Neisser 1967: Two stages: pre-attentive then attentive.
  - Deldman & Ballard 1982: 100-step rule
  - Tsotsos’ 1990: complexity analysis of immediate vision
- Cognitive architectures (no perception):
  - Subsumption by R. Brooks 1986
  - Soar by Laird, Newell & Rosenbloom 1987
  - Outline by J. Albus 1991
  - ACT-R by J. Anderson 1993
- Developmental, incl. perception, cognition and behavior
  - Darwin V by Edelman et al. 1998 (inter-cortical adaptation)
  - SAIL-3 by Weng et al. 1998 (intra- and inter cortical adaptation)

AMD Architecture Considerations

- Intelligent controller or task decomposer? No.
  - No intelligent component
  - Each component is very mechanical and “dumb”
- Symbolic representation? No.
  - A symbol is unbreakable, abstract
  - Difficult for robot to generate new symbols
- Vector (distributed) multilevel representation? Yes.
  - Automatic context formation
  - Competing percepts
  - Competing behaviors

Soar

- Symbolic representation
- Model goal-oriented cognition and reasoning
- Task-specific knowledge built in representation
- Sequencing of decision learned from interactive training
- System example: robot-Soar by Laird at al.

Darwin V

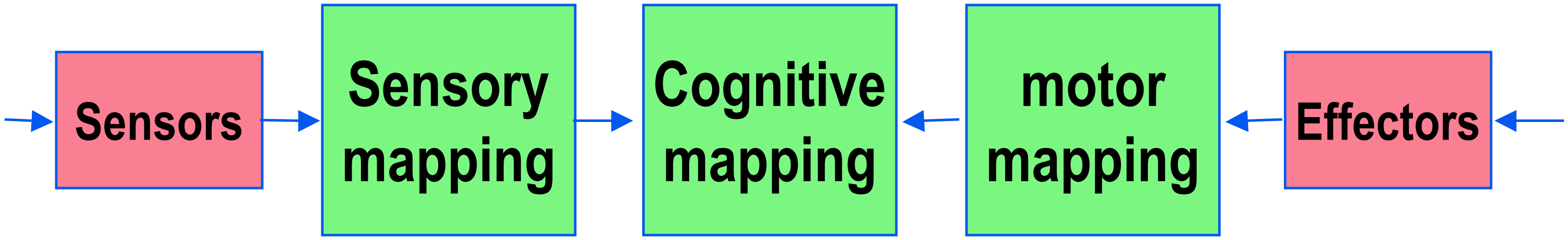
(Gerald Edelman, N. Almassy and Olaf Sprons and)

- Plasticity in feature integration and in behavior generation
- A set of programmed-in behaviors
- A value system
- “Taste” as wired-in appetitive and aversive stimuli
- Real-time, online, embodied development by a robot
- Developed capabilities:
  Feature invariance linked to behaviors
  Vision-based object selection behaviors

SAIL-3 Developmental Robot

- Automatically developed internal representation from sensory and effector space
- Sensory mapping: hierarchical feature spaces from input
- Cognitive mapping:
  - forming states as working memory from input of sensory mappings
  - Self-organizing cognitive maps as long-term memory
- Internal behaviors:
  attention selection, action release
- 4 learning modes, including effector-imposed, reinforcement and communicative learning
- Value system, vigilance, forgetting.

Three Mappings

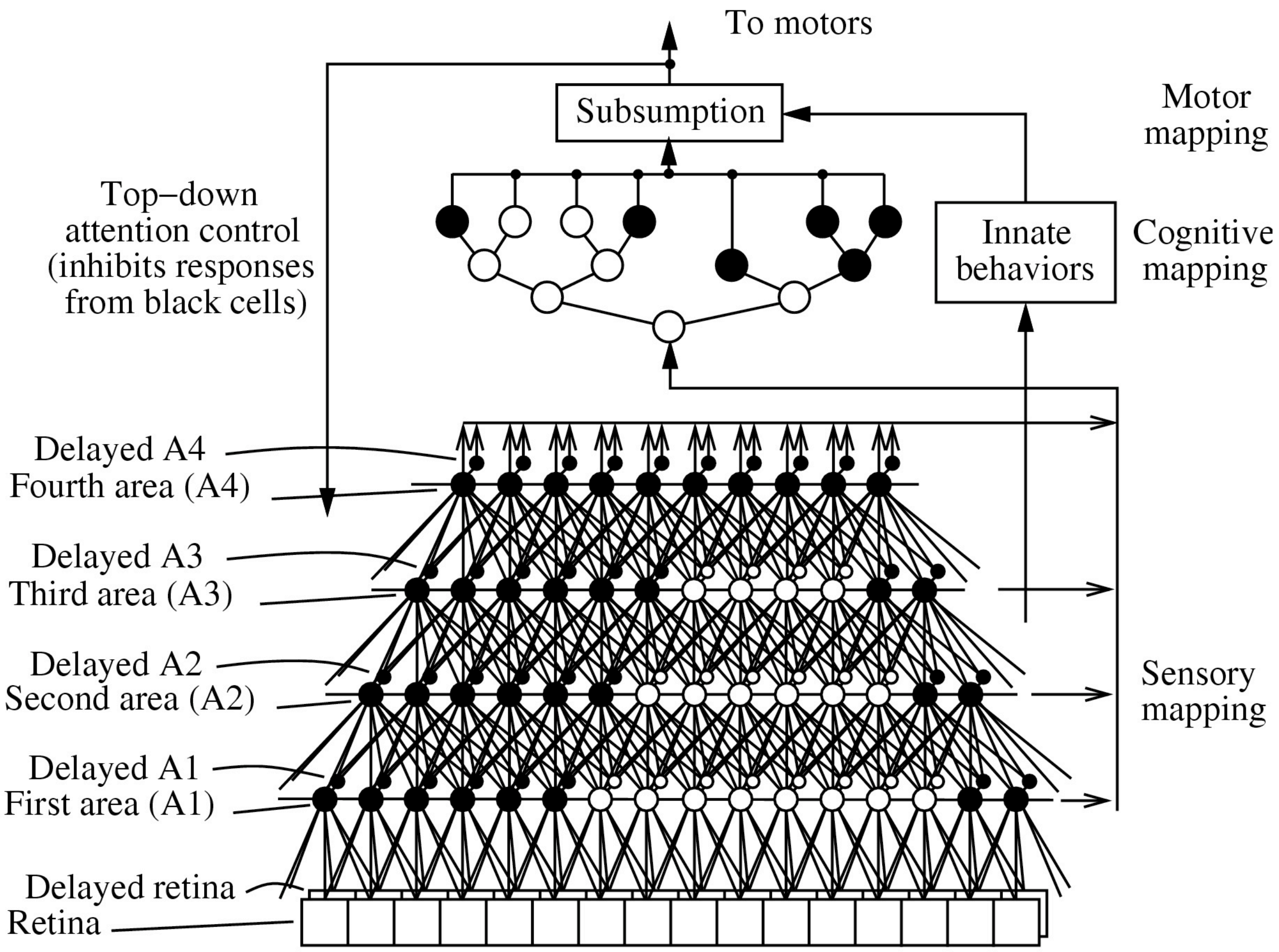


Only input space is available

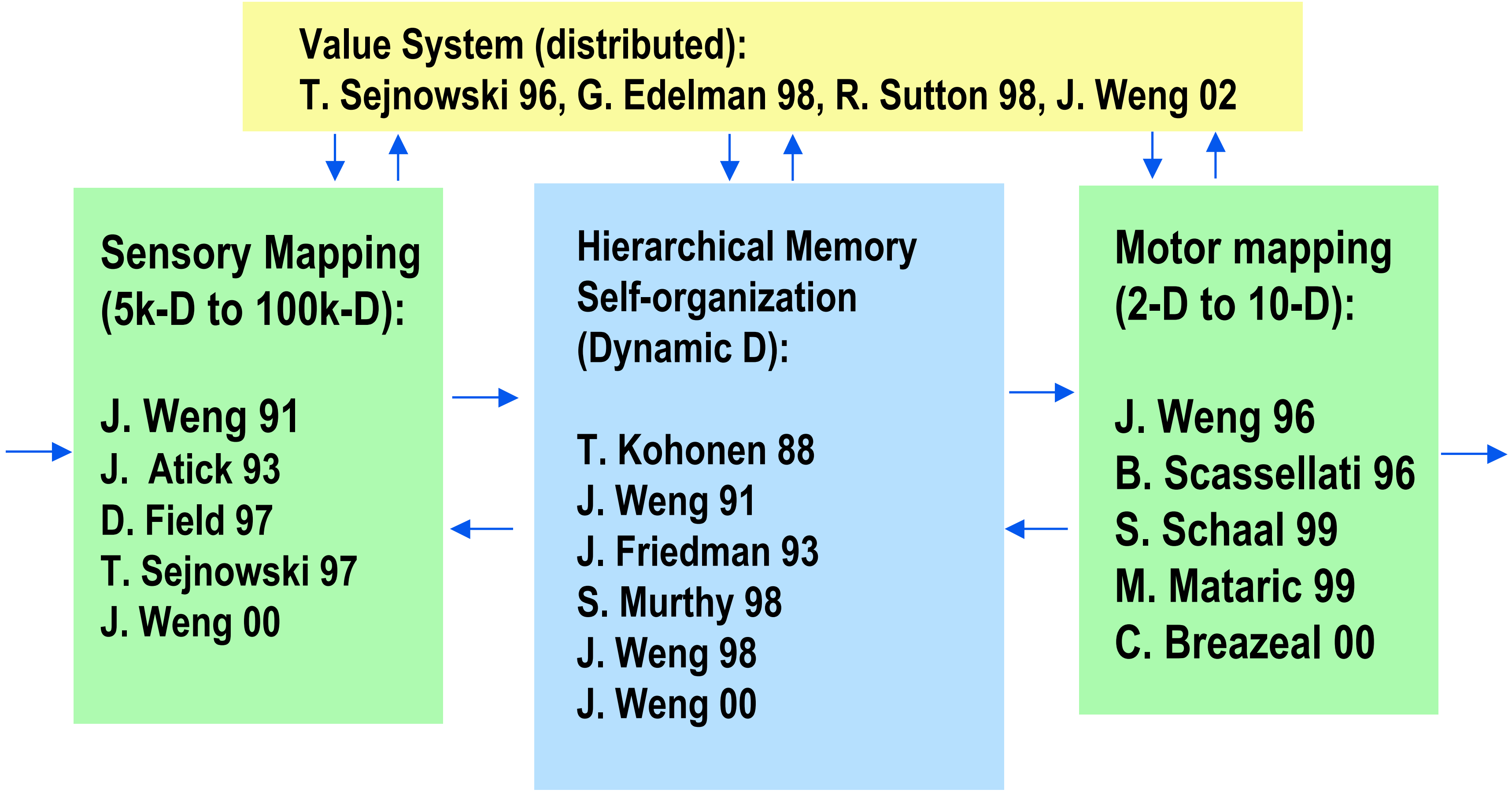
Input and output spaces are indirectly available

Only output space is available

Overview of Sensorimotor System



Some Related Work



Value System (distributed):
T. Sejnowski 96, G. Edelman 98, R. Sutton 98, J. Weng 02

Sensory Mapping
(5k-D to 100k-D):

J. Weng 91
J. Atick 93
D. Field 97
T. Sejnowski 97
J. Weng 00

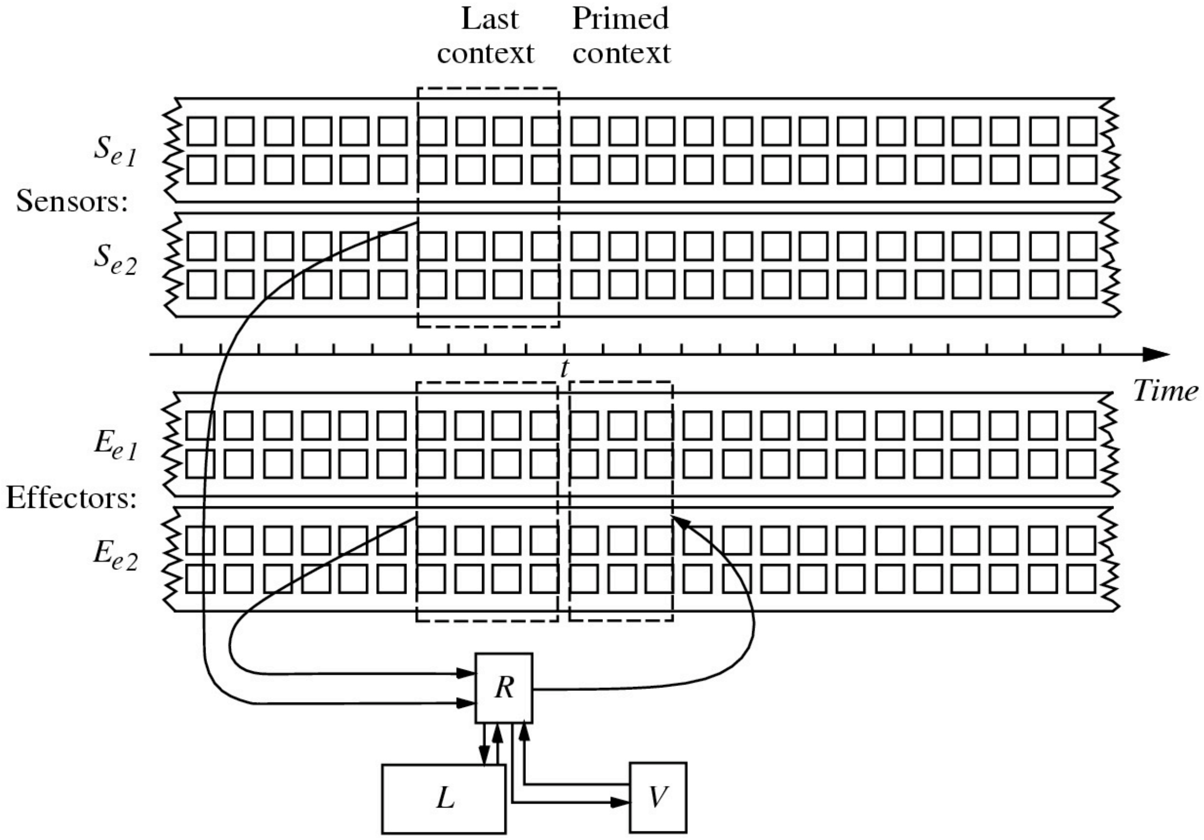
Hierarchical Memory
Self-organization
(Dynamic D):

T. Kohonen 88
J. Weng 91
J. Friedman 93
S. Murthy 98
J. Weng 98
J. Weng 00

Motor mapping
(2-D to 10-D):

J. Weng 96
B. Scassellati 96
S. Schaal 99
M. Mataric 99
C. Breazeal 00

Type 1: Observation-driven MDP



R: Regressor
L: Set of states
V: Value system

Formulation: POMDP and ODMDP

POMDP

$$H_t = \{s_{t-1}, s_{t-2}, ..., s_0, a_{t-1}, a_{t-2}, ..., a_0\}$$

$$l_t = \{s_{t-1}, s_{t-2}, ..., s_{t-k}, a_{t-1}, a_{t-2}, ..., a_{t-k}\}$$

$$P(s_t \mid H_t) = P(s_t \mid l_t)$$

ODMDP

$$H_t = \{x_t, x_{t-1}, ..., x_0, p_{t-1}, ..., p_0\}$$

$$l_t = \{x_t, x_{t-1}, ..., x_{t-k}, p_{t-1}, ..., p_{t-k}\}$$

$$P(p_t \mid H_t) = P(p_t \mid l_t)$$

Comparison of POMDP and ODMDP

- POMDP is world centered
  ODMDP is mind centered
- Each state of POMDP is hand specified
  Each state of ODMDP is automatically generated
- POMDP has two layers of probability
  $P(s_t \mid x_t, s_{t-1})$ and $P(x_t \mid s_t)$.
  ODMDP has one layer of probability
  $P(p_t \mid l_t)$.

From Sensory input to Behavior output

- **Cognitive mappings:**

$$\{p_1(t), ..., p_k(t)\} = R(l(t)).$$

$$R : \mathcal{L} \mapsto 2^{\mathcal{P}}$$

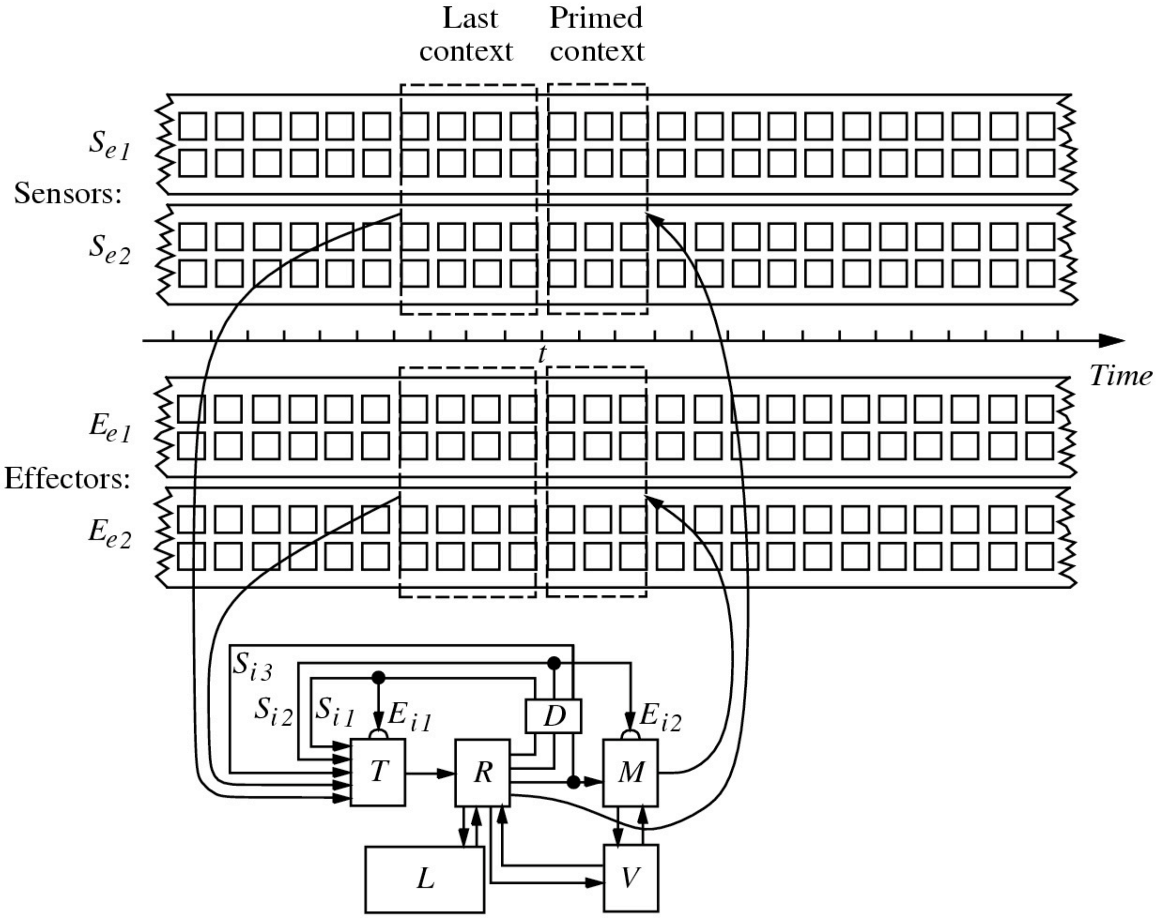
- **Value system:**

$$V(R(l(t))) = V(\{p_1(t), p_2(t), ..., p_k(t)\}) = p_i(t)$$

$$V : 2^{\mathcal{P}} \mapsto \mathcal{P}$$

Types 2 through 5

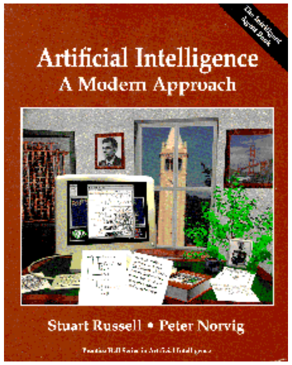
- Type 2: Add *T* (selective)
- Type 3: Add *M* (reheasable)
- Type 4: Add S_i (SASE)
- Type 5: Developmental



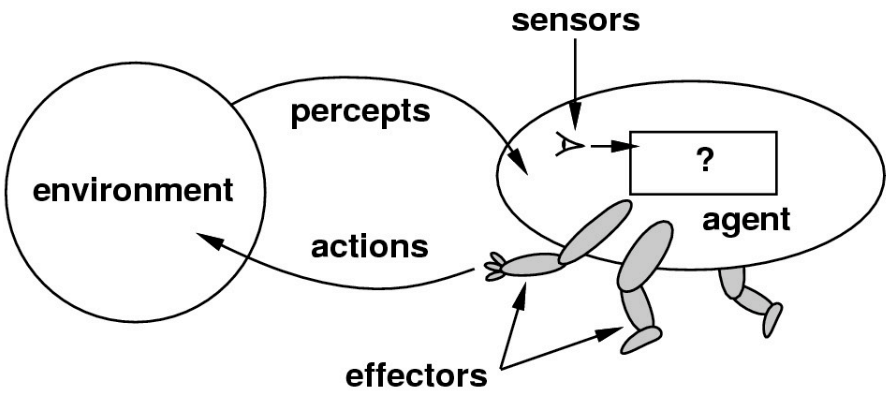
T: Attention selector
R: Regressor
L: Set of states
M: Motor mapping
V: Value system
D: Delay unit

A Flaw of the Model for Agents

- A well accepted model: Sense and respond to external world
  E.g., the excellent text by Russell & Norvig



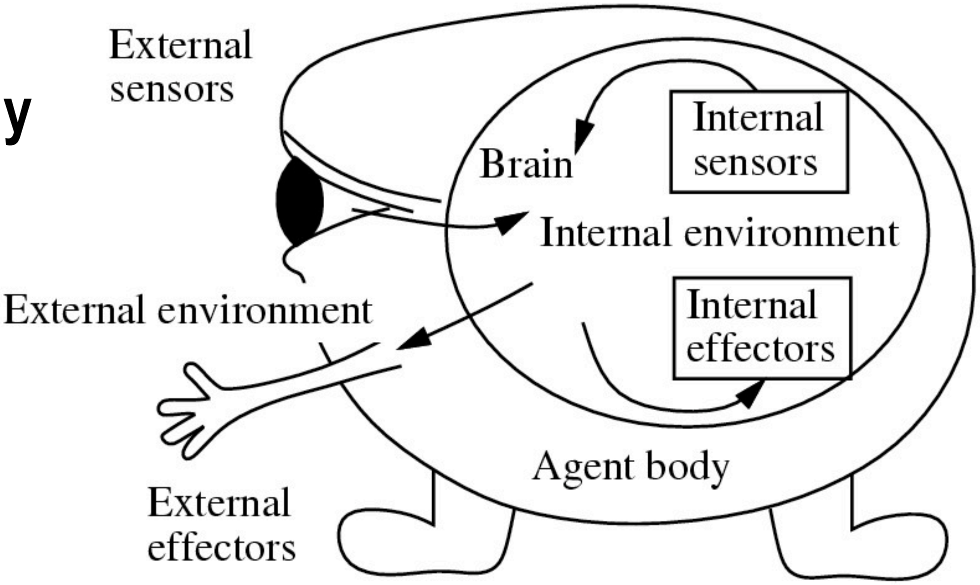
- The flaw:
  - Absence of self-generated representation
  - Lack of sensing and learning internal activities



The New Model: SASE Agent

(Weng ICDL'02)

- SASE:
  Self-Aware and Self-Effecting
- External world: the world around including the body
- Internal world: brain
- Sensing:
  - External:
  - internal: brain
    (e.g., primed sensation)
- Effecting:
  - external world
  - internal world
    (e.g., attention selection)



Necessary Conditions of Self-Awareness

Suppose an agent is aware of its mental activities (sensations and actions) about a task b in an environment E. Then:

1. It senses such mental activities using its (internal) sensors
2. It feeds the sensed signal into its perceptual entry point just like that for the external sensors

Why SASE?

- Self-generated internal representation (at least the later conscious part) should be a part of the internal world to be aware of: SA
- Autonomous operations on the internal representation is necessary: SE
- A half century of *mistake*:
  overlook the need for the machine to be aware of its own internal world and its operations (autonomous thinking process)
- Autonomous thinking using autonomously developed internal representation: an essence of *consciousness*?

How the Architecture Enables Generalization?

Several mechanisms for generalization:

- Value system: value-sensitive events
  - Pleasure seeking and pain-avoidance
  - Novelty seeking
  - Values of many contexts depending on experience
- Value-insensitive events (nearly-equal values)
  - Attention selection from new settings
  - Autonomous thinking:

$$s = \left((l_1, p_1), \begin{bmatrix} e_1 \\ i_1 \end{bmatrix}, (l_2, p_2), \ldots, (l_k, p_k), \begin{bmatrix} e_k \\ i_k \end{bmatrix} \right)$$

External and Internal Reasoning

- Three types of reasoning processes:
  - External, through e_j's
  - Internal, through i_j's
  - Mixed
- Attention model *T* selects which is attended
- Type-1 through Type-3 allow external reasoning, not internal ones
- Type-4 allows all three types of reasoning

Autonomous Planning

- Type-4 allows internal reasoning to realize autonomous planning
  - Plan (a):

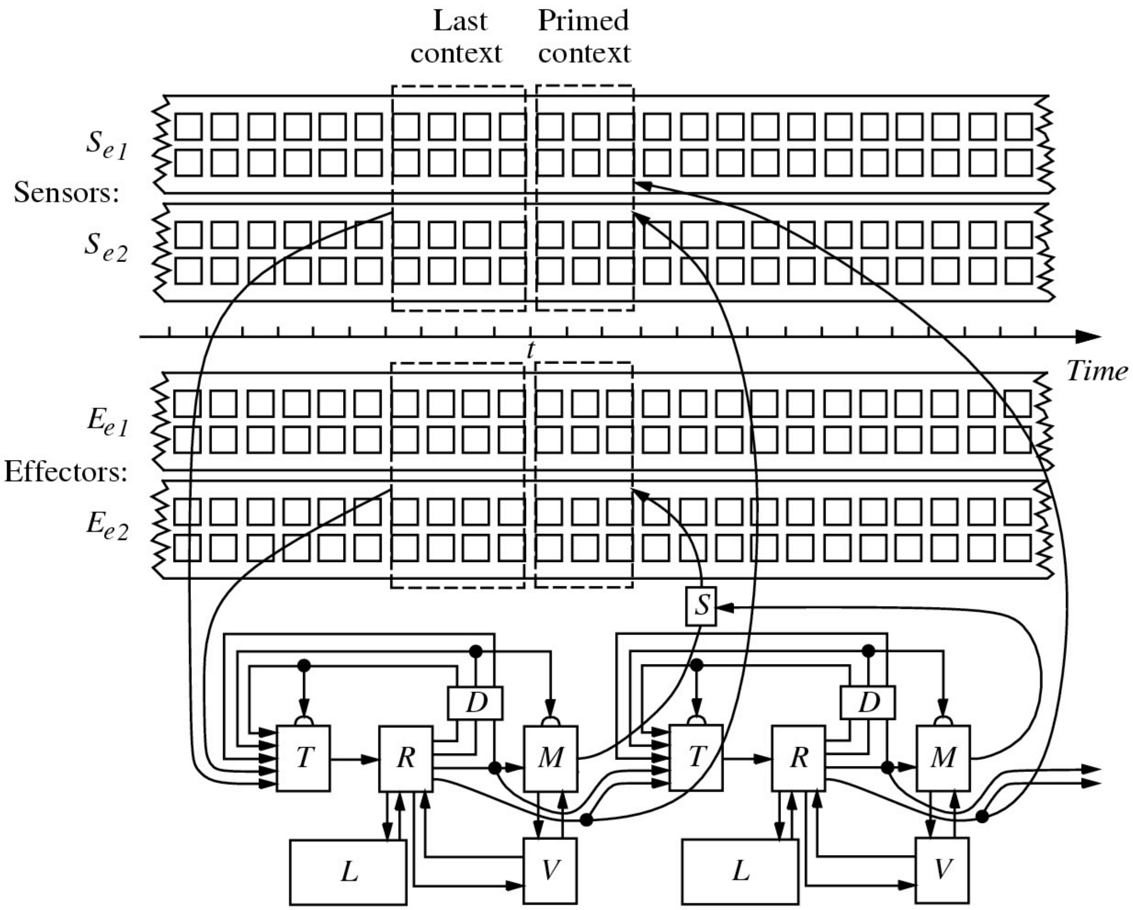
$$(l_1, p_{a,1}), (l_{a,2}, p_{a,2}), ..., (l_{a,i}, p_{a,i}))$$

  - Plan (b):

$$(l_1, p_{b,1}), (l_{b,2}, p_{b,2}), ..., (l_{b,j}, p_{b,j}))$$

  - Selection based on the value system

Type 6: Multi-level DOSASE MDP

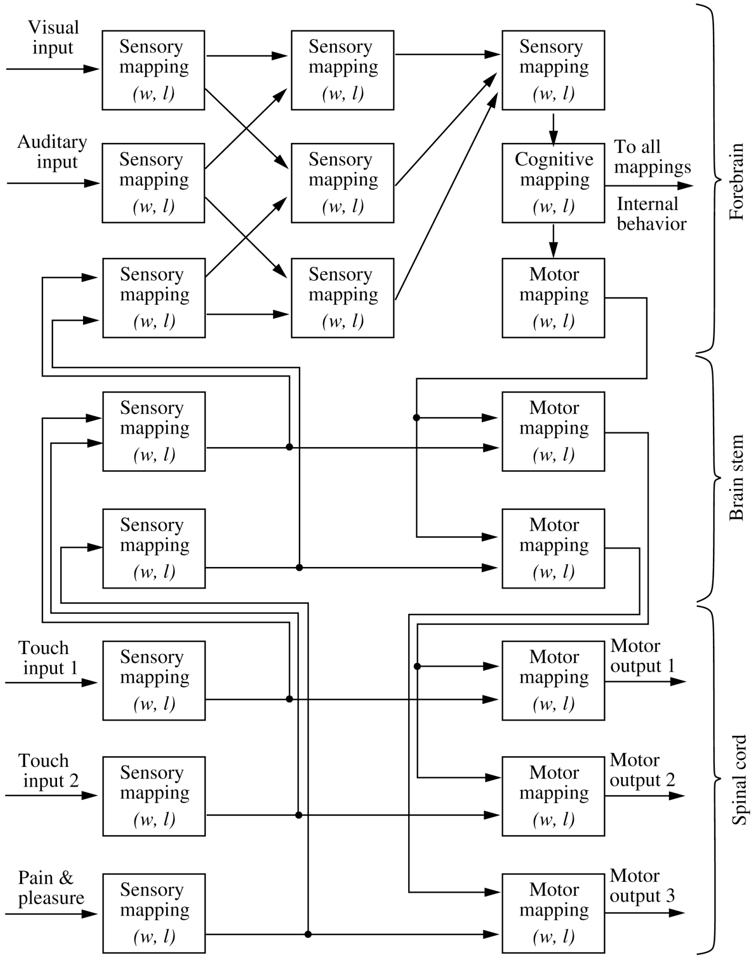


T: Attention selector
R: Regressor
L: Set of states
M: Motor mapping
V: Value system
D: Delay unit

Architecture Types

- Type 1: Observation-driven MDP
- Type 2: Observation-driven Selective MDP
- Type 3: Observation-driven Selective Rehearsed MDP
- Type 4: Observation-driven SASE MDP
- Type 5: Developmental Observation-driven SASE MDP
- Type 6: Multi-level DOSASE MDP

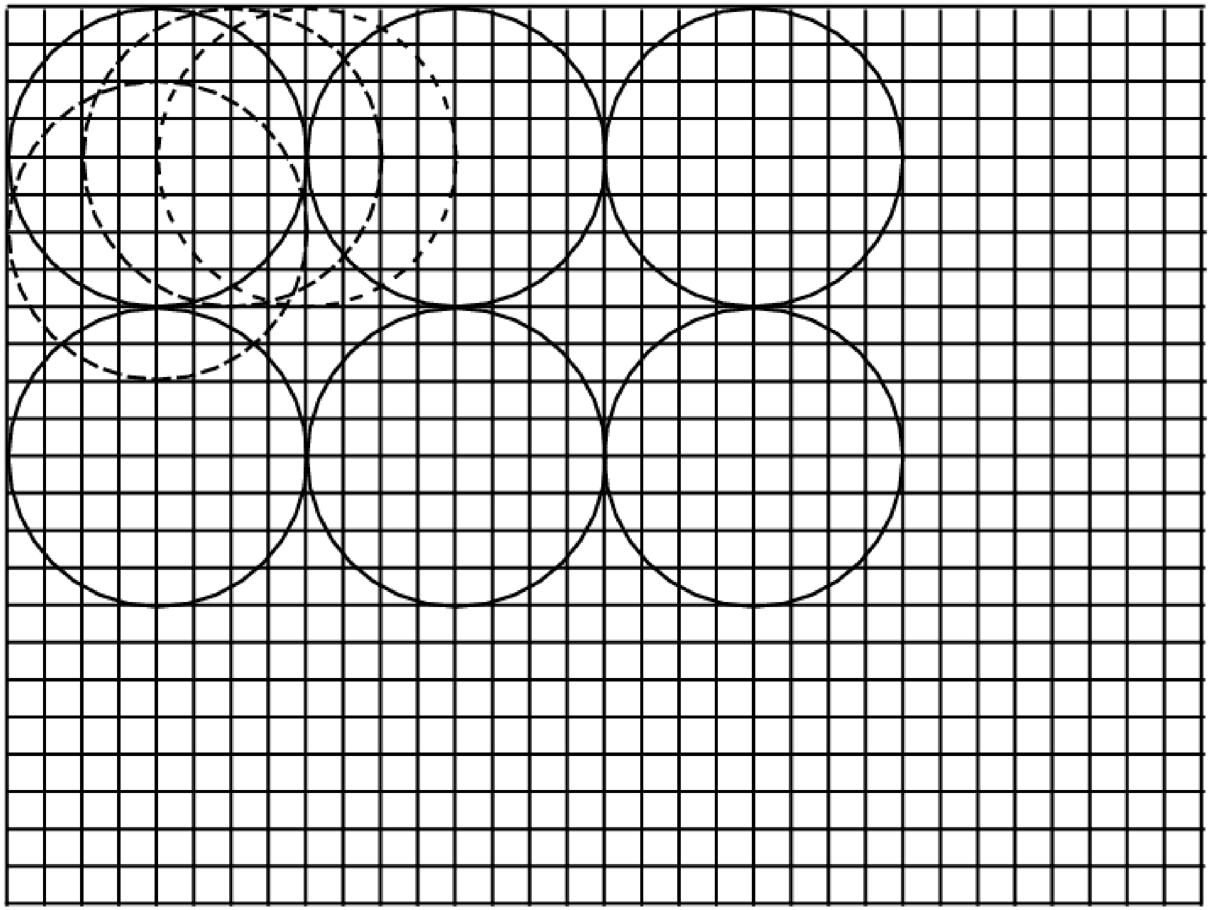
SAIL-3 Architecture



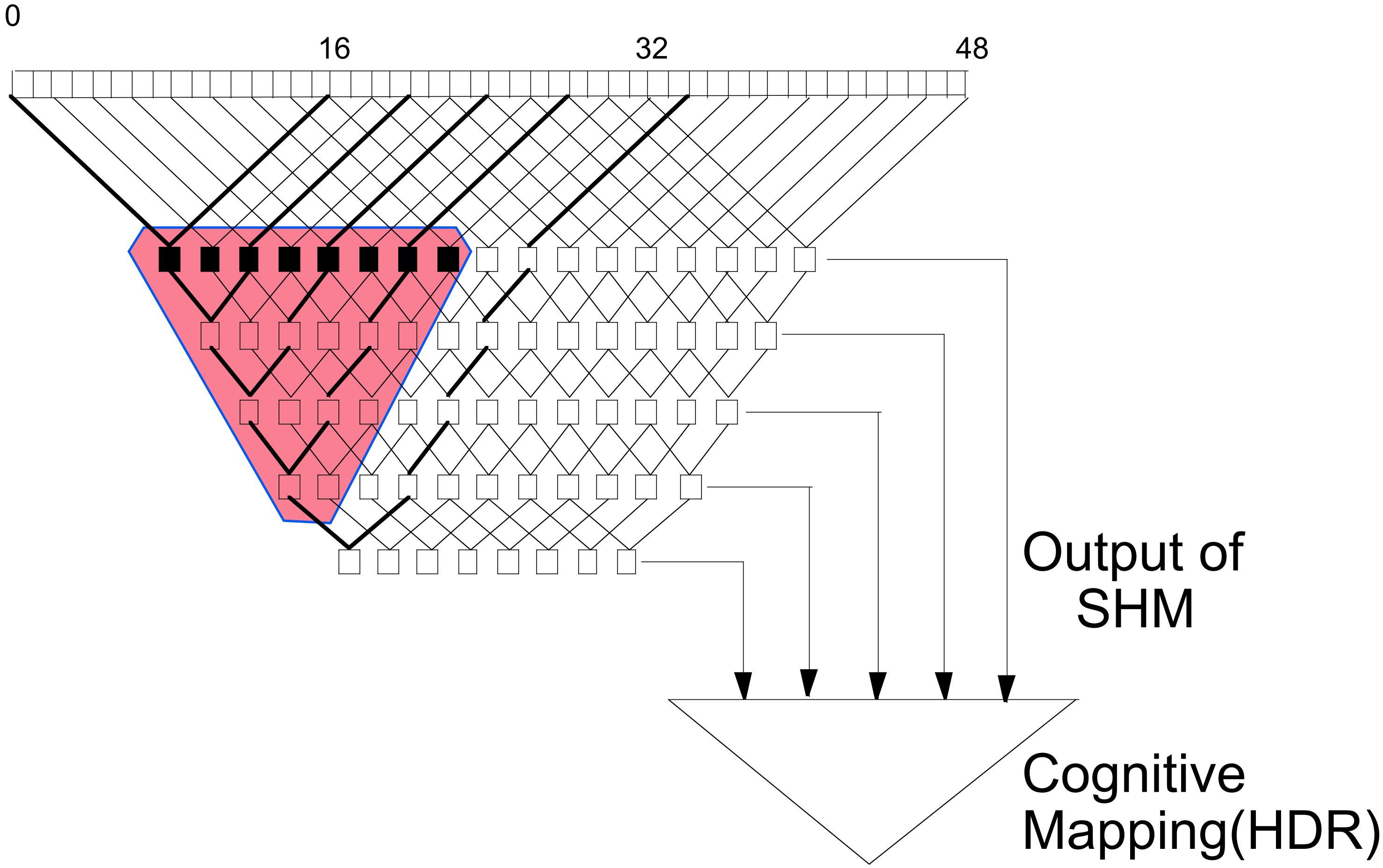
- Develop via experience
- Lower level:
  - More reflexive
  - Limited extent of sensory association
  - Fast behaviors
- Higher level:
  - More deliberative
  - Extensive sensory association
  - Slower response
- Mediation of levels

7. Sensory Mapping

Input Map and Receptive Fields

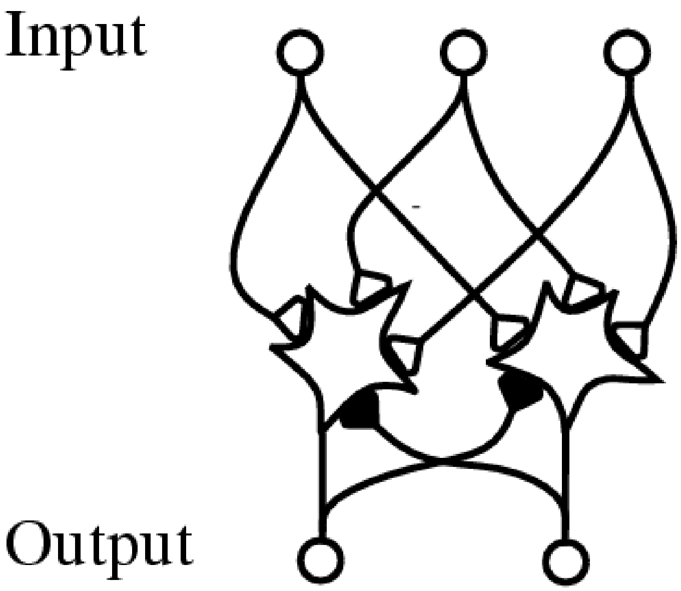


Staggered Hierarchical Mapping (SHM) (Zhang, Weng & Zhang ICDL 2004)

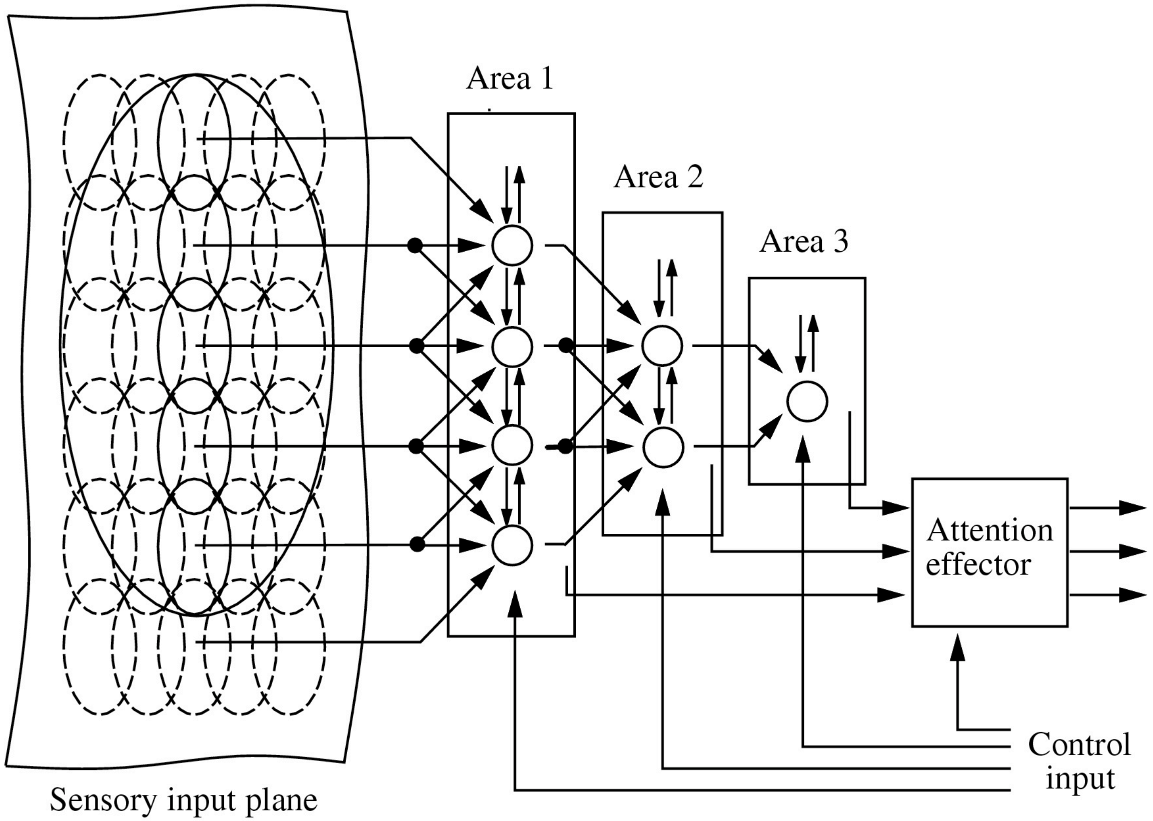


How Neurons Derive Features?

- Hebbian Rule: updating weights when output is strong
- Lateral Inhibition: suppressing neighbors when the neuron output is high
- Hebbian Rule + Lateral Inhibition develops feature detectors

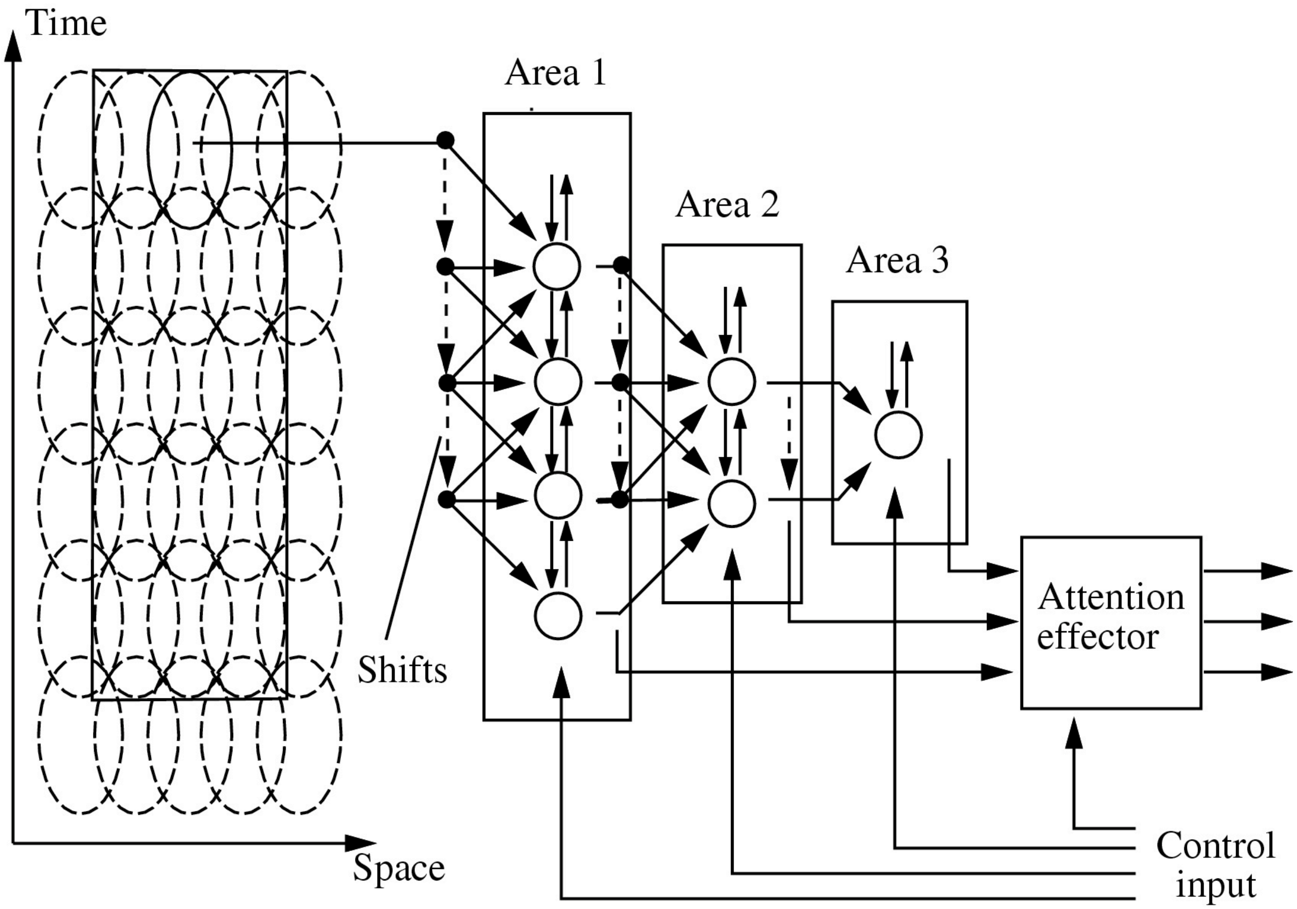


2-D Sensory Mapping



- **Goal:**
  - **reduce dimension**
  - **allow control**
- **Methods:**
  - **PCA, ICA, LCA**
  - **Attention selection**

SAIL: Spatiotemporal Sensory Mapping



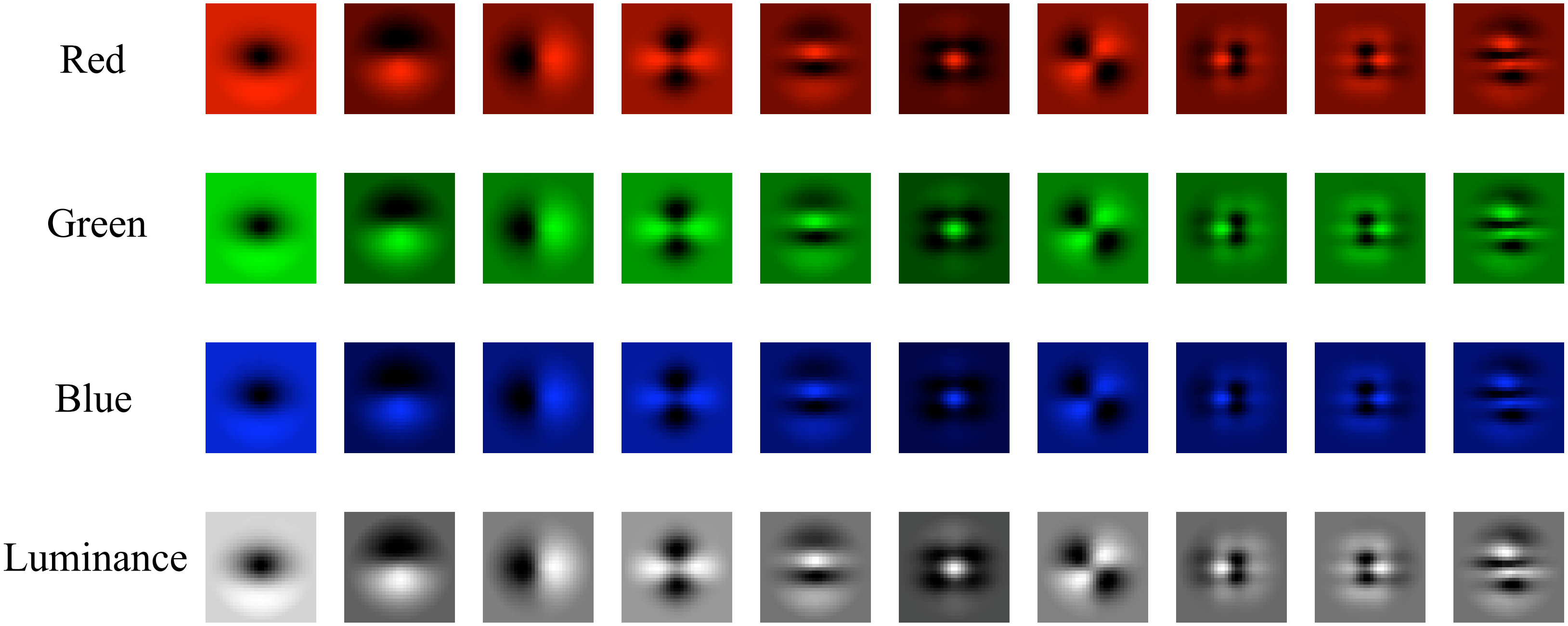
Example of Sensory Mapping Development

- Natural images digitized from video
- 5,000 image samples, each with 160x120 pixels
- Each receptive field cover 32x32 pixels
- Incremental PCA to generate and update wavelet filters

Samples of Training Images

Developed Visual Filters

- Similar to Gabor filters and Wavelets, but better and complete
- Automatically developed, not hand designed!
- Representation for higher perception becomes manageable



Incremental PCA (IPCA)

- Principal components:
  - Covariance matrix of *R*
  - eigenvectors associated with the largest eigenvalues of *R*
- *R* is too big and it is batch processing
- Incremental PCA:
  - Without using *R*
  - Update eigenvector (eigenvalue) one sample at a time

Candid Covariance-free Incremental PCA (CCIPCA) (PAMI Aug. 2003)

- Scatter vector:

$$u_i = x_i - \bar{x}$$

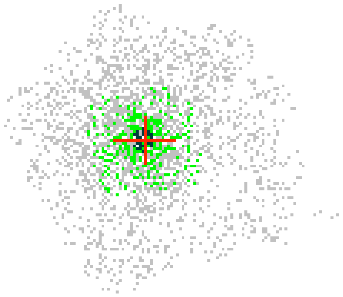
- Amnesic updating principal component vector

$$v_i^{(n)} = \frac{n-1-l}{n} v_i^{(n-1)} + \frac{1+l}{n\|v_i^{(n-1)}\|} (\hat{u}_i \cdot v_i^{(n-1)}) \hat{u}_i$$

- Compute output y_i as projection on the vector
- Residual vector for next principal component vector:

$$\hat{u}_{i+1} = \hat{u}_i - y_i v_i^{(n)} / \|v_i\|$$

IPCA: Most Efficient Estimate



- **A most efficient estimate is one that has the least variance from the real parameter, e.g. sample mean**

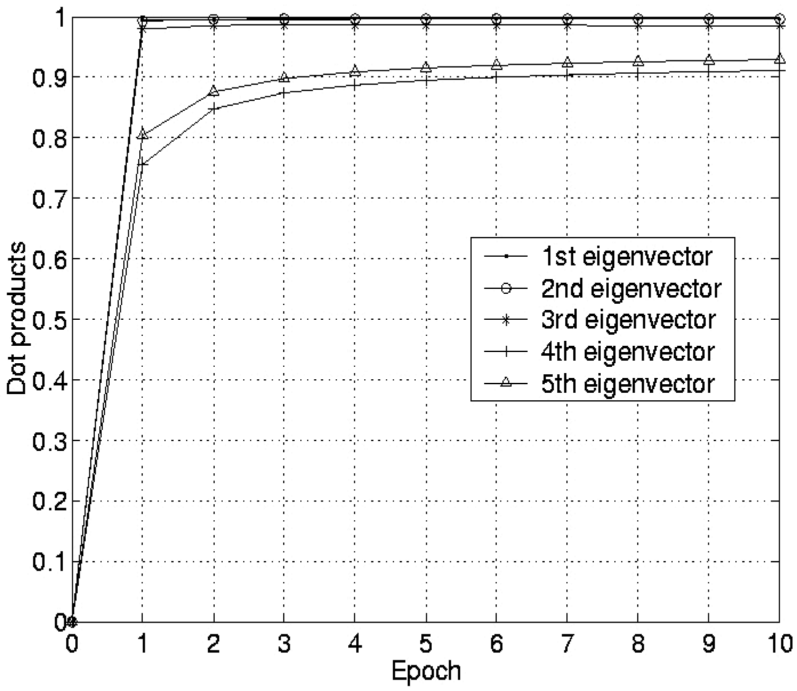
$$v(n) = \frac{n-1}{n} v(n-1) + \frac{1}{n} u(n) u^T(n) \frac{v(n-1)}{||v(n-1)||}$$

$$v(n) = \frac{n-1}{n} v(n-1) + \frac{1}{n} w(n)$$

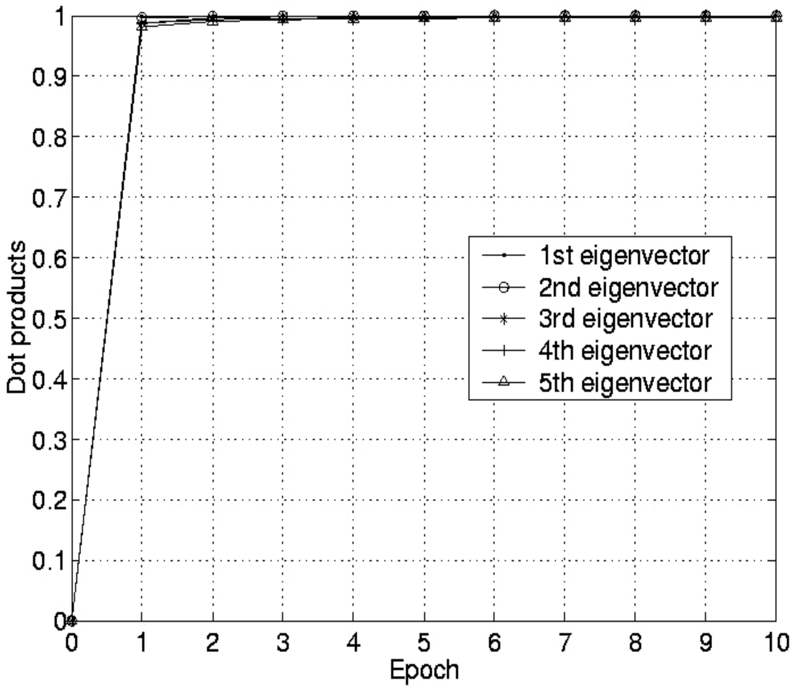
$$v(n) = \frac{1}{n} \sum_{i=1}^{n} w(i)$$

Convergence of IPCA Algorithms

Oja's SGA

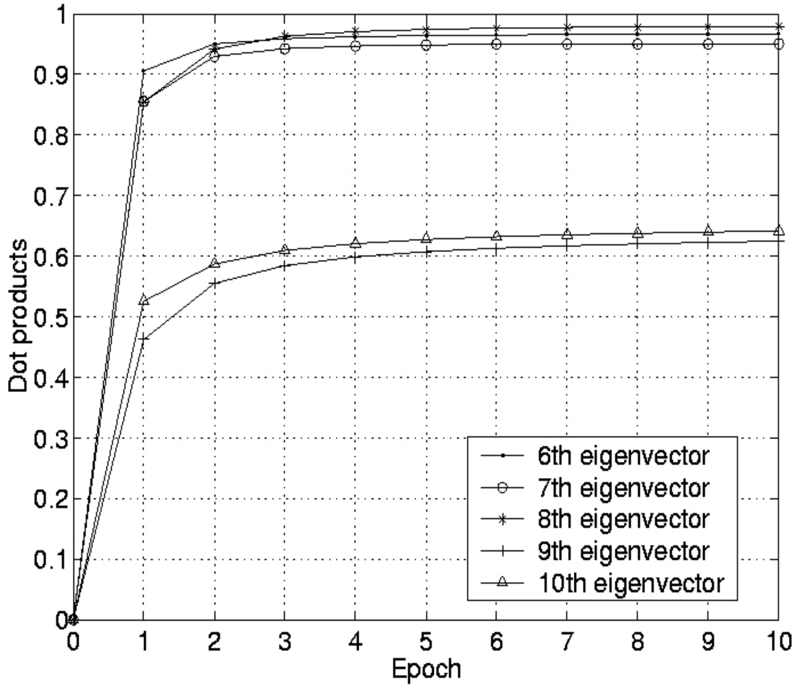


Our IPCA

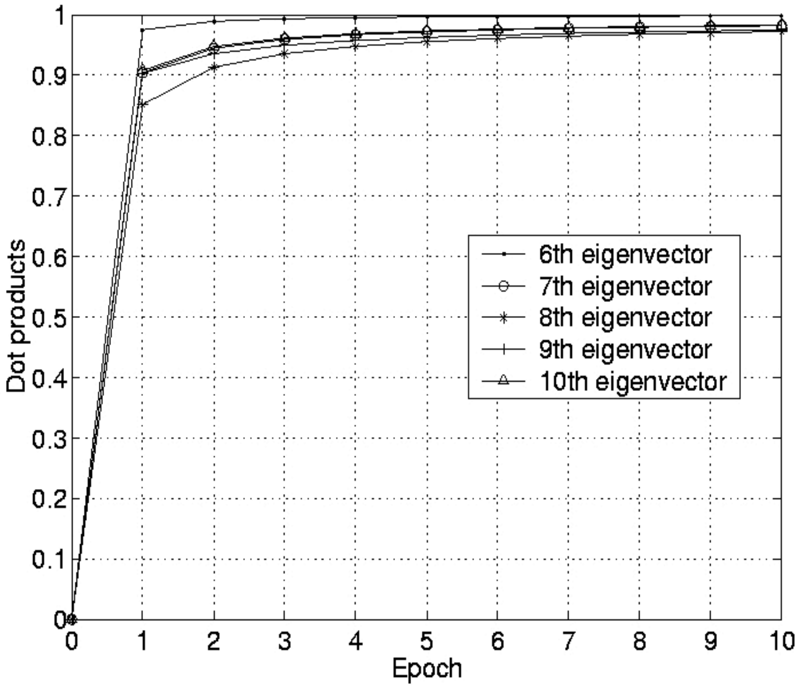


Other Eigenvectors

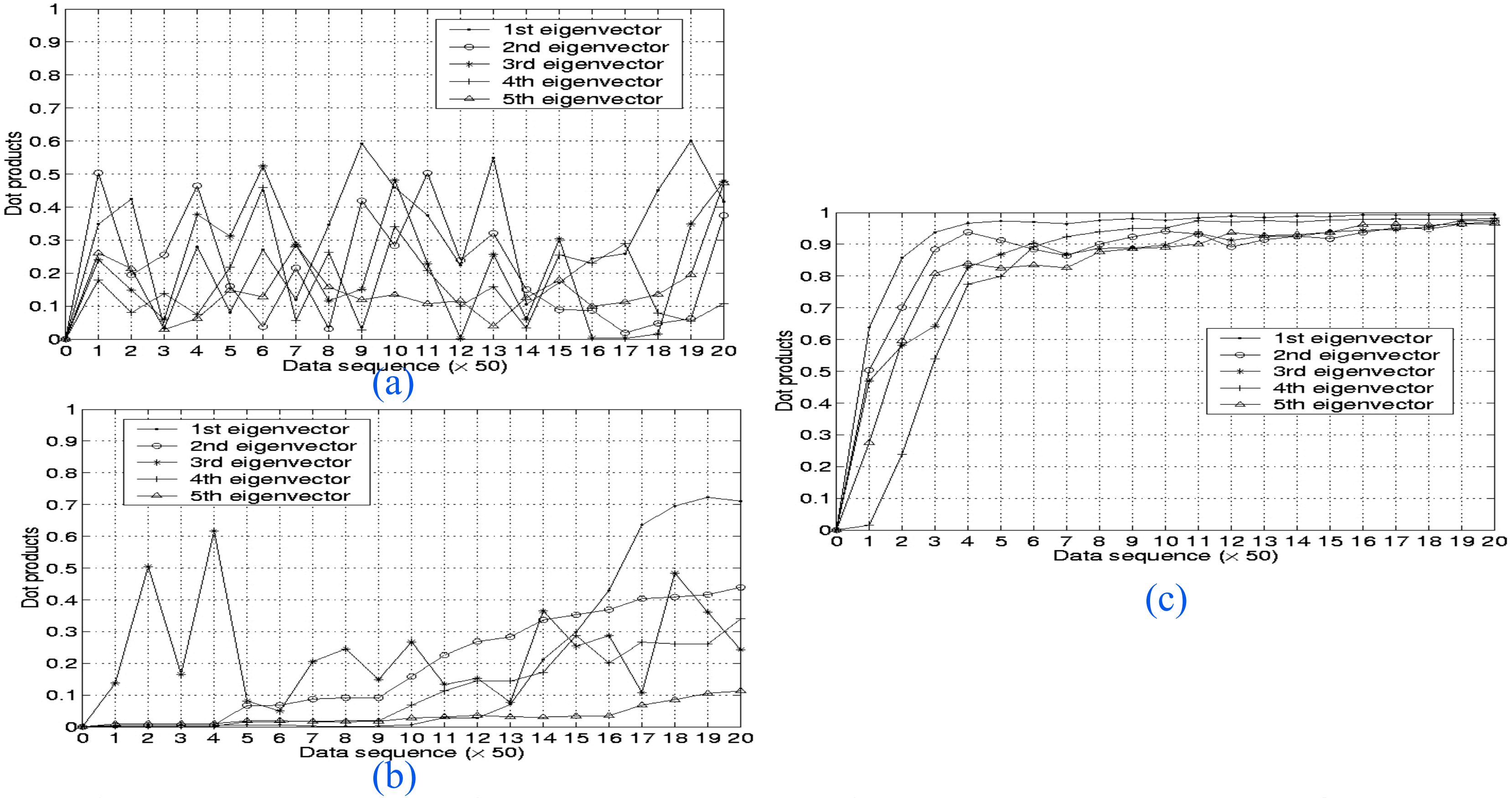
Oja's SGA



Our IPCA



IPCA Convergence Comparison



(a)

(b)

(c)

Convergence of the first 5 eigenvectors for 5632-dimension data (88-by-64 images): (a) Oja's SGA (b) Sanger's GHA (c) Proposed CCIPCA

IPCA and Most Efficient Estimate

IPCA: $$v(n) = \frac{n-1}{n} v(n-1) + \frac{1}{n} u(n) u^T(n) \frac{v(n-1)}{||v(n-1)||}$$

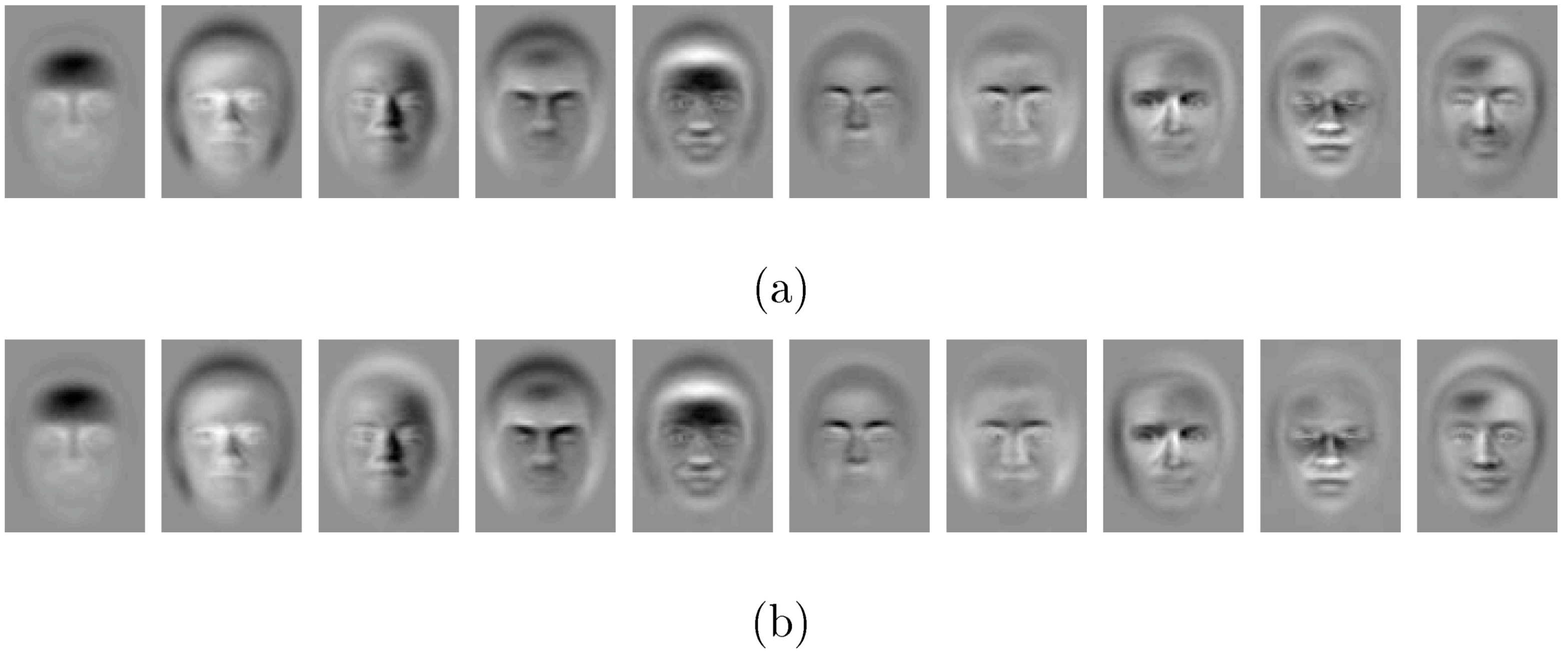
$$v(n) = \frac{n-1}{n} v(n-1) + \frac{1}{n} w(n)$$

$$v(n) = \frac{1}{n} \sum_{i=1}^{n} w(i)$$

The ***most efficient estimate:***

- **A most efficient estimate is one that has the least variance from the real parameter.**
- **The sample mean of data is a most efficient estimate of mean, if the distribution satisfies some regularity conditions**

IPCA: Eigenfaces



(a)

(b)

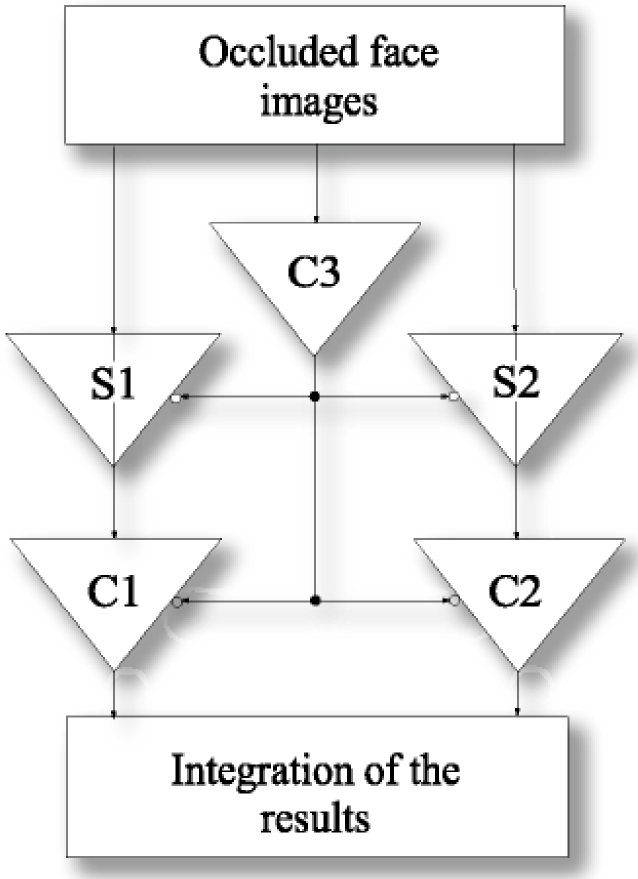
The first 10 eigenfaces obtained by (a) batch PCA, and (b) CCIPCA shown as images

CCI Lobe Component Analysis

- Lobe: concentration of probability density
- Whitening:
  Decorrelation of input components
  Normalize the power along each direction
  Lobe components are salient
- Lobe Component Analysis:
  corresponding to Independent Component Analysis (ICA) for super-Gaussians

Use of SHM: Occluded Face Recognition

- Training phase:
  Complete face (G) available
- Testing phase:
  Only occluded faces:
  Upper view (U)
  Lower view (L)
- Solution:
  - Training using active vision: Acquires U and L views during training
  - Testing detects U and L views
  - U and L integration



Summary of Occlusion Experiment

Method	Recognition Rate		
	U	L	U+L
Monolithic+NN	51.43%	75.83%	82.38%
SHM+HDR	92.86%	95.95%	98.57%

Method	Testing Time (ms)		
	U	L	U+L
Monolithic+NN	765.5	765.5	1131.5
SHM+HDR	702.4	702.4	1008.3

SAIL: Motor Mapping

- The reverse of sensory mapping
- Additional: signal reconstruction from projections
- Two action source at each level:
  - Innate behaviors, programmed in our learned offline
  - Learned behaviors from higher levels
- Mediating actions from high levels:

  Soft-subsumption

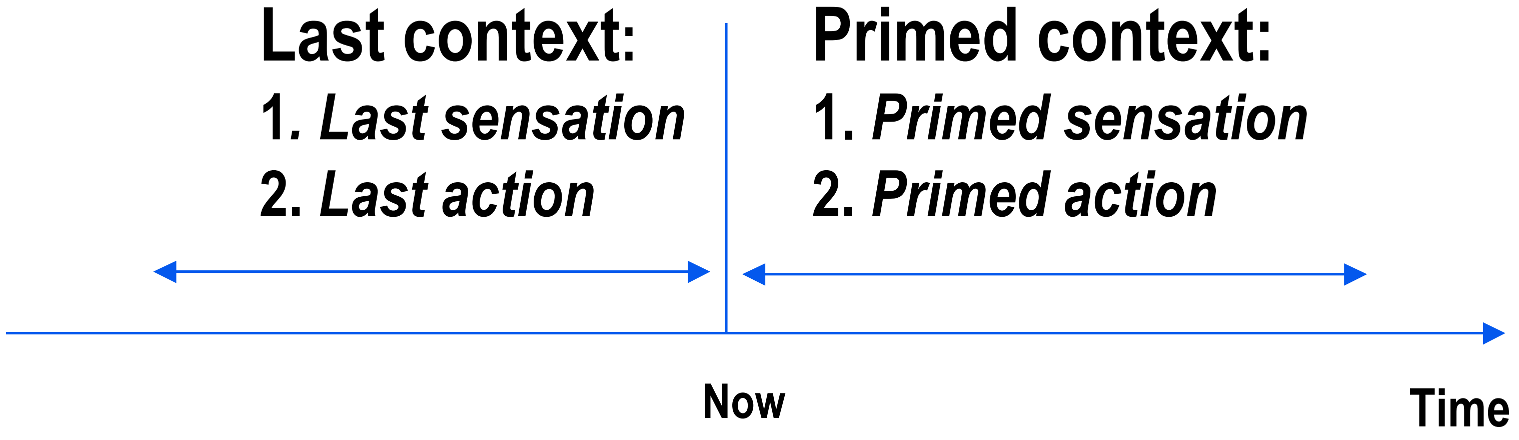
  From high level (v_h, c_h); from low level (v_l, c_l)

  winner: $\max\{c_h w_h, c_l w_l\}$, where $w_h > w_l > 0$

  The control signal of the winner is executed

8. Cognitive Mapping

The Last Context and the Primed Context



Last context:

1. *Last sensation*
2. *Last action*

Primed context:

1. *Primed sensation*
2. *Primed action*

Now

Time

Working Memory and Long Term Memory

- Long term memory $l(t)$:
  - Representation level
  - Architecture level
  - Timing level
- Working memory $w(t)$:
  - Context that the brain currently attend to
  - Depends on robot’s internal and external behaviors
  - E.g.,

$$w(t) = (x(t-1), x(t-2), x(t-3))$$

Cognitive Mapping: Regression

- **Last context:**

$$c(t) = (x(t), w(t))$$

- **Cognitive mapping:**
generate action and update long term memory

$$(l(t+1), a(t+1)) = f(x(t), w(t) \mid l(t))$$

Cognitive Mapping

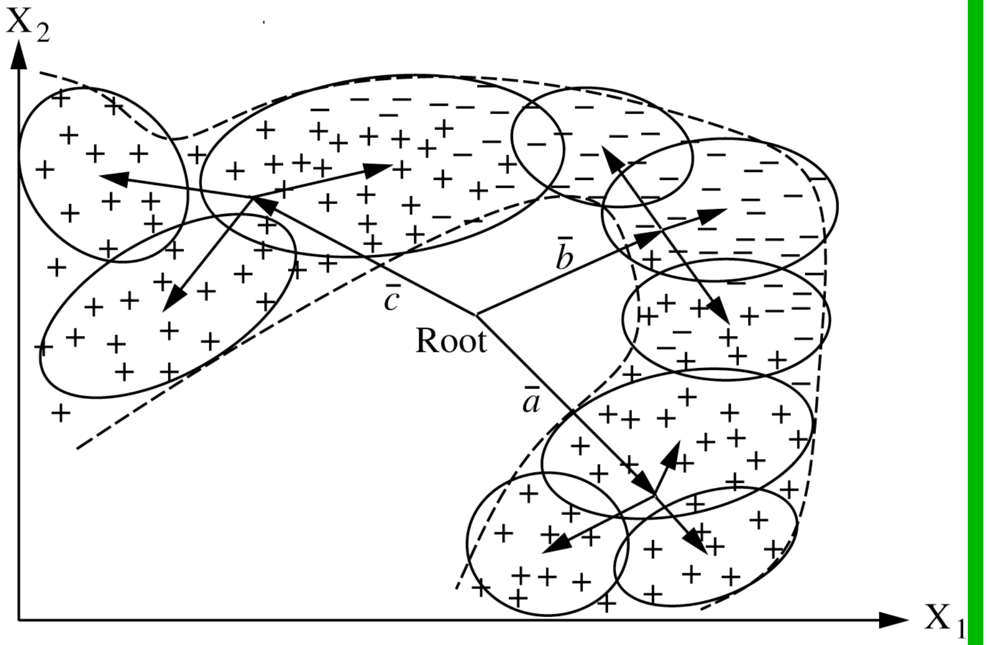
- **Goal: approximate a function** $y = f(x)$ where x is any vector in a d-dimensional space (e.g., $d = 10{,}000$)
- **Allow supervised learning:** training samples $(x_i, y_i), i=1, 2, \ldots$
- **Allow reinforcement learning**

Seven Regression Requirements

1. High dimensional (5k-D and more)
2. One-instance learning
3. Adapt to increasing complexity
4. Deal with local minima problem
5. Incremental
6. Long term memory without catastrophic memory loss, but forget old details
7. Very low time complexity with large memory

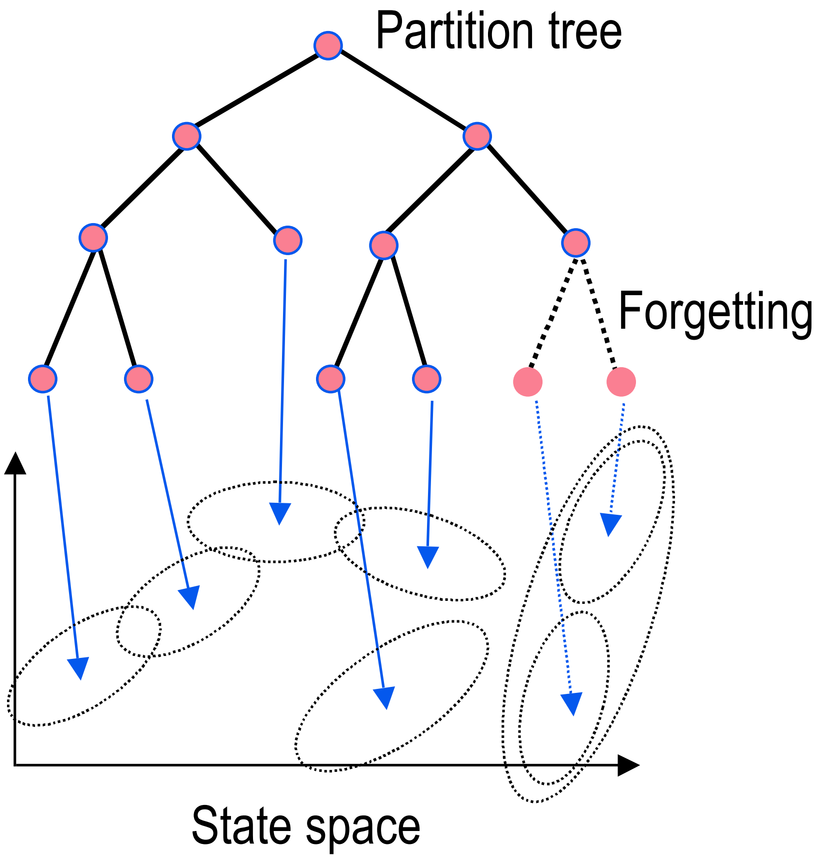
Coarse to Fine

- Input: *x*
- Starting from root
- Coarse-to-fine search using a tree
- Each leaf node has sample pairs *(x_i, y_i)*
- Output: y_i from the best matches x_i

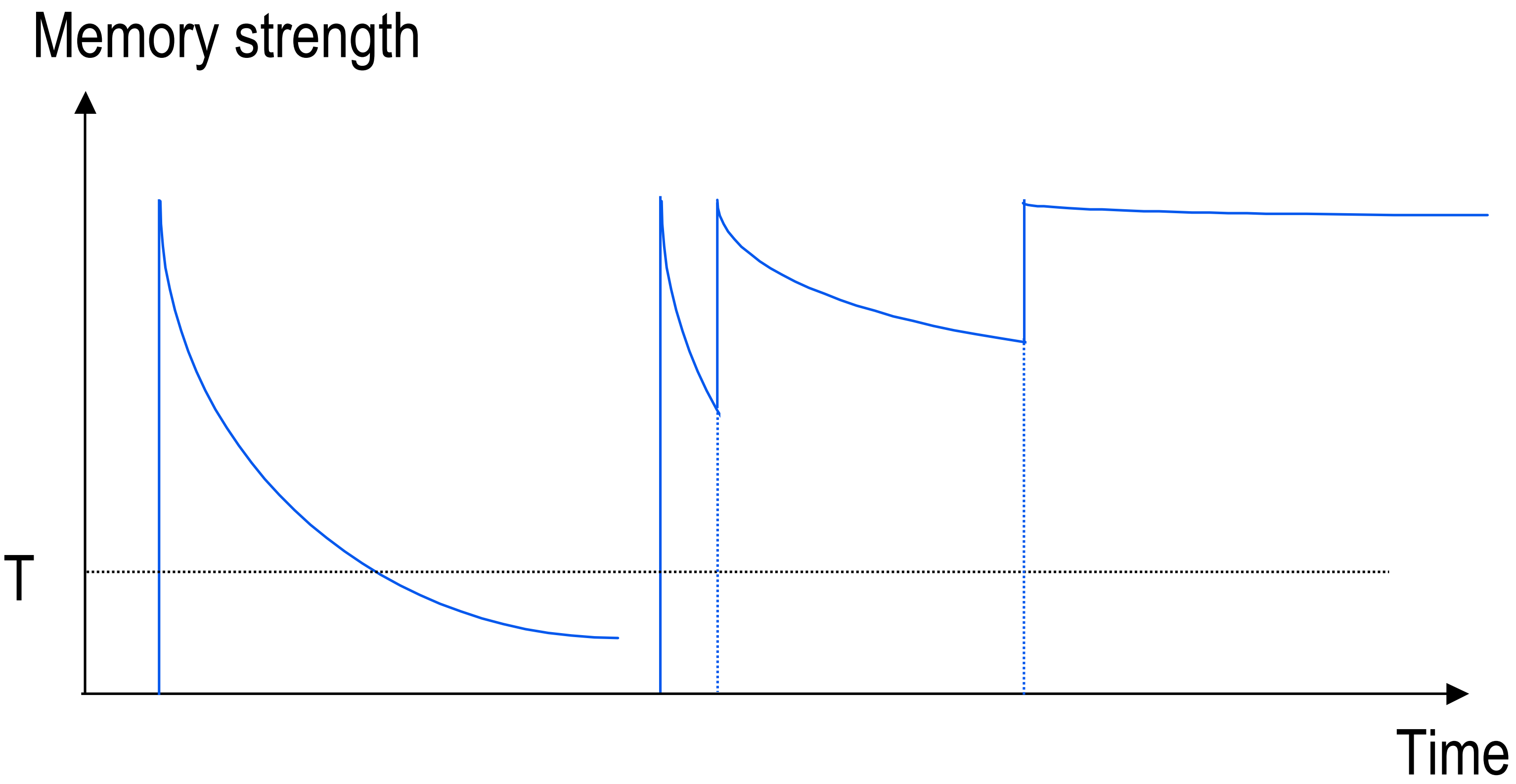


Hierarchical Discriminant Regression (PAMI Nov. 2000)

- The tree is constructed incrementally from (x_i, y_i), $i = 1, 2, \dots$ where y_i may be missing
- Given unknown x, the tree finds the best match x_i fast
- Each node has a memory trace register
- A tree node is forgotten (deleted) if this memory trace is low

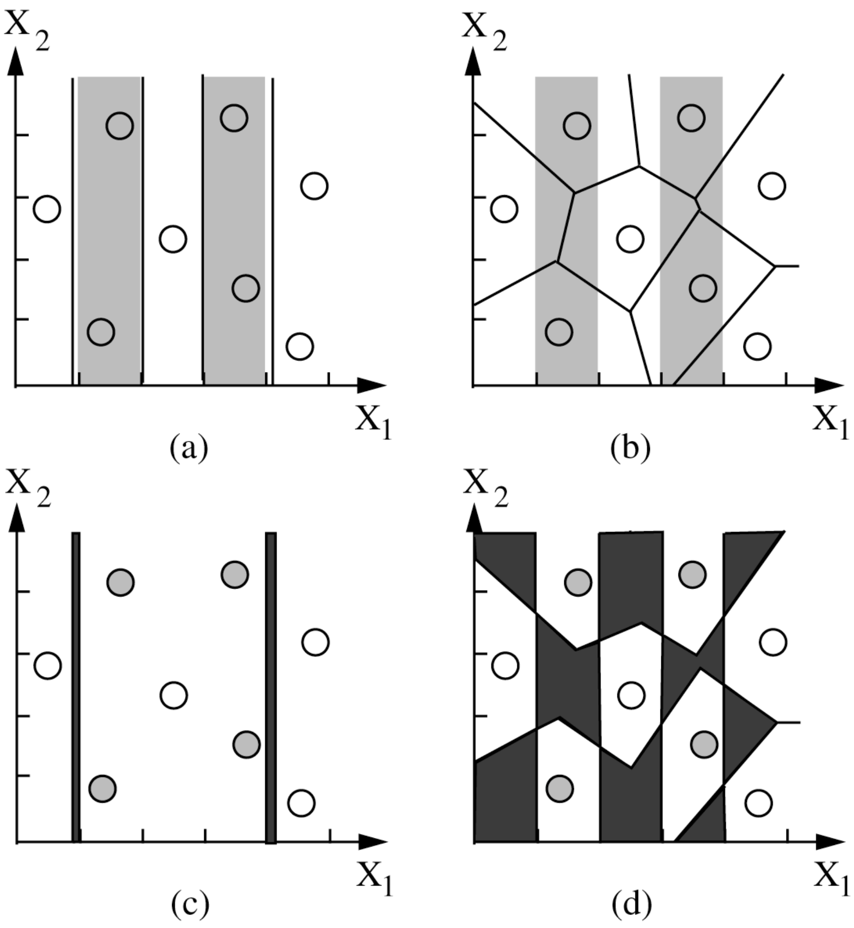


Forgetting Using Memory Trace Decay

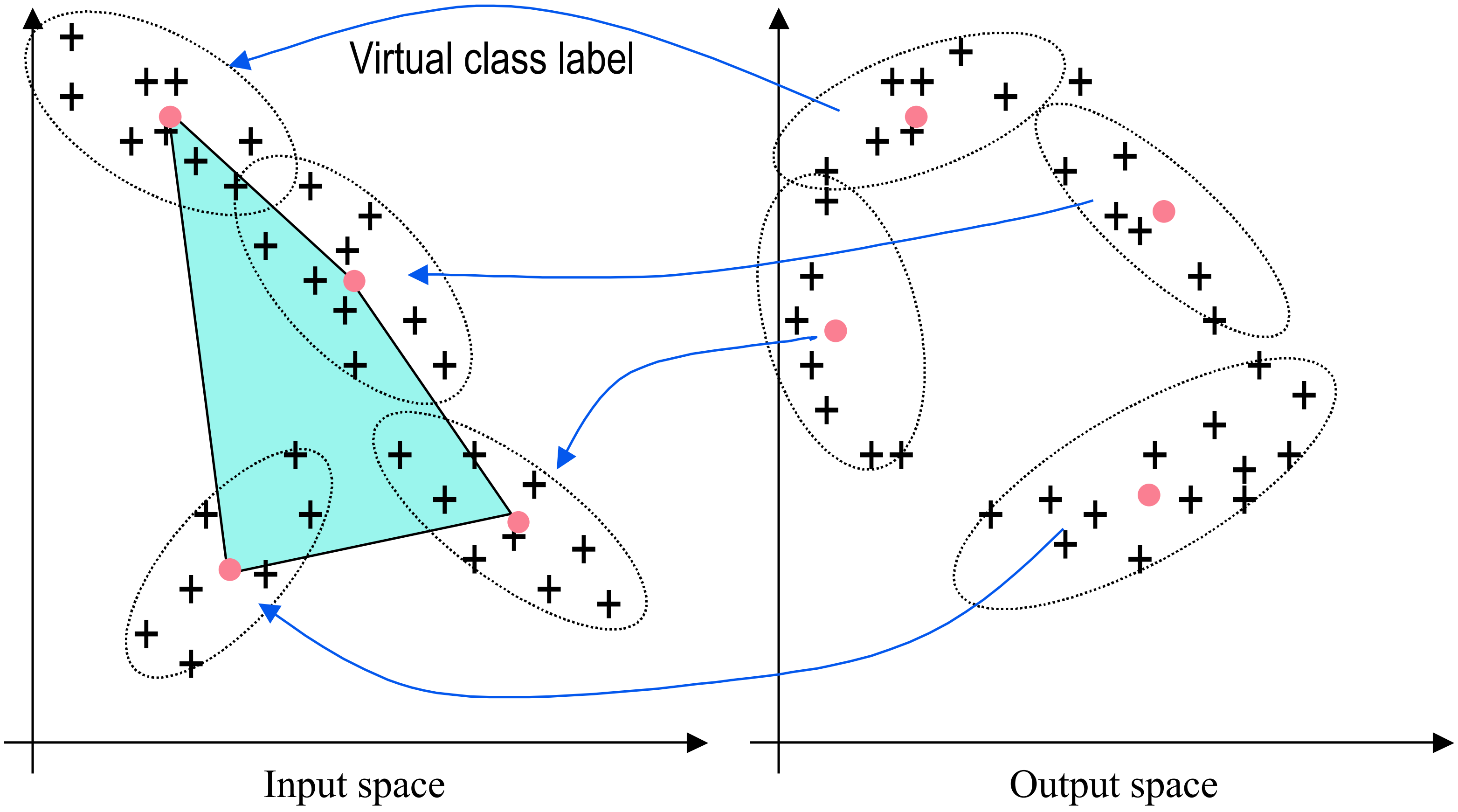


Why Features?

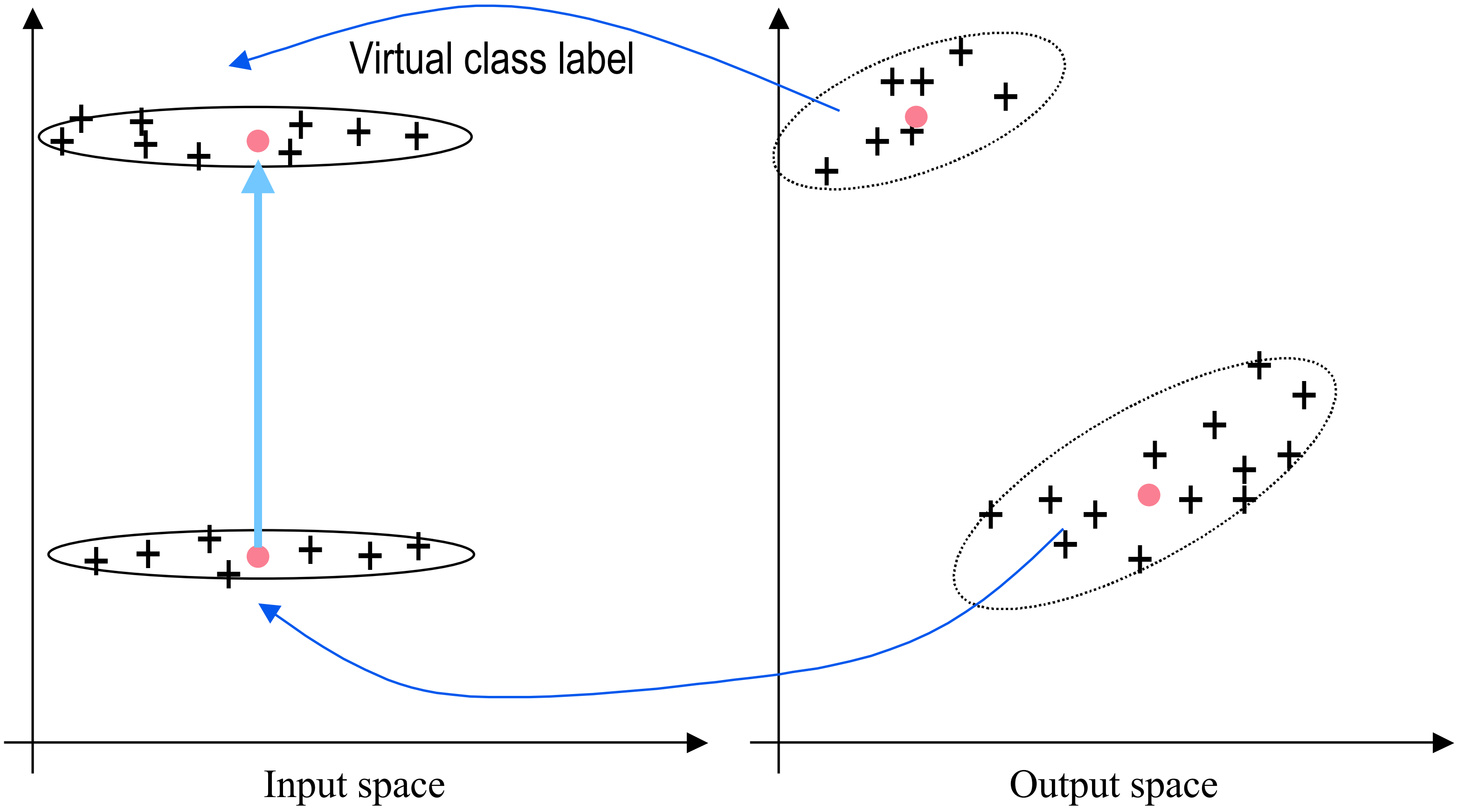
- Many components in the raw input are irrelevant to output
- Impractical to use nearest-neighbor rule: *Cannot exhaust all the possible combinations!*



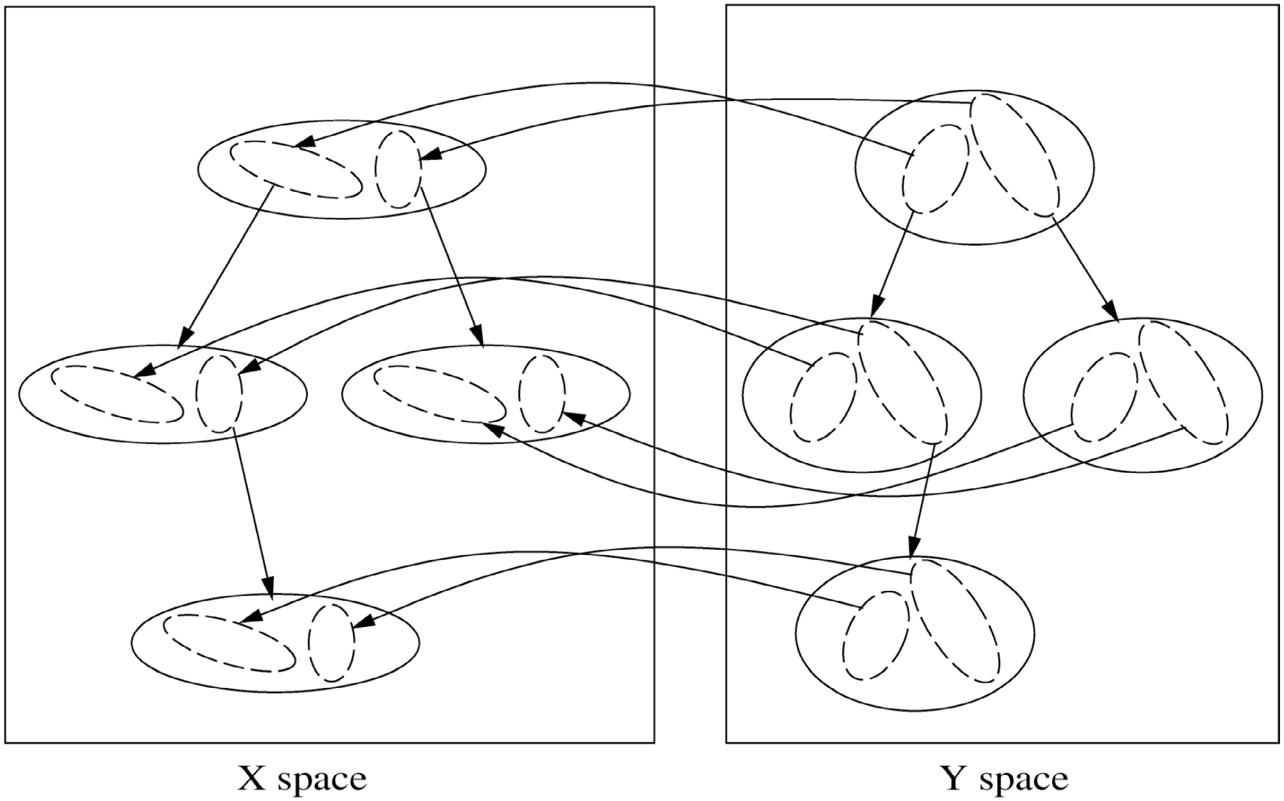
Each Node Automatically Derives its Most Discriminating Feature Subspace



Disregarding Irrelevant Input Components



HDR: Hierarchical Structure



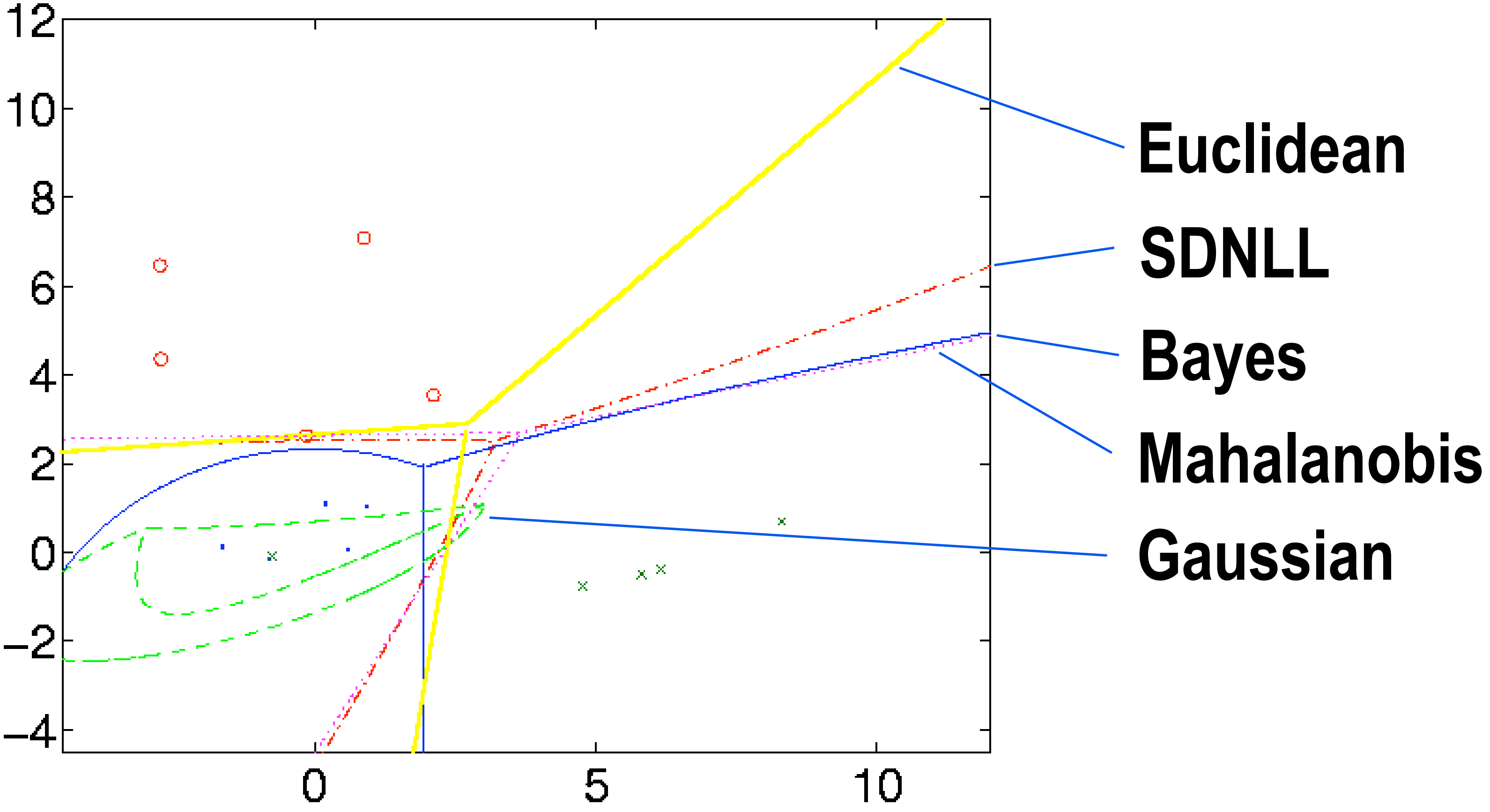
Handling Unbalanced Samples: SDNLL

- SDNLL: size-dependent negative log likelihood
- Smooth transition among three types of likelihood: Euclidean, Mahalanobis, and Gaussian
- Transition points are automatically determined by the statistics of estimates

$$L(x, c_i) = 1/2\ (x - c_i)^T\ W_i^{-1}(x - c_i) + 1/2\ ln(\ |W_i|\ )$$

$$W_i = w_e\, r^2 I + w_m\, S_w + w_g\, G_i$$

Fitting Class Boundaries: Few Samples



Euclidean

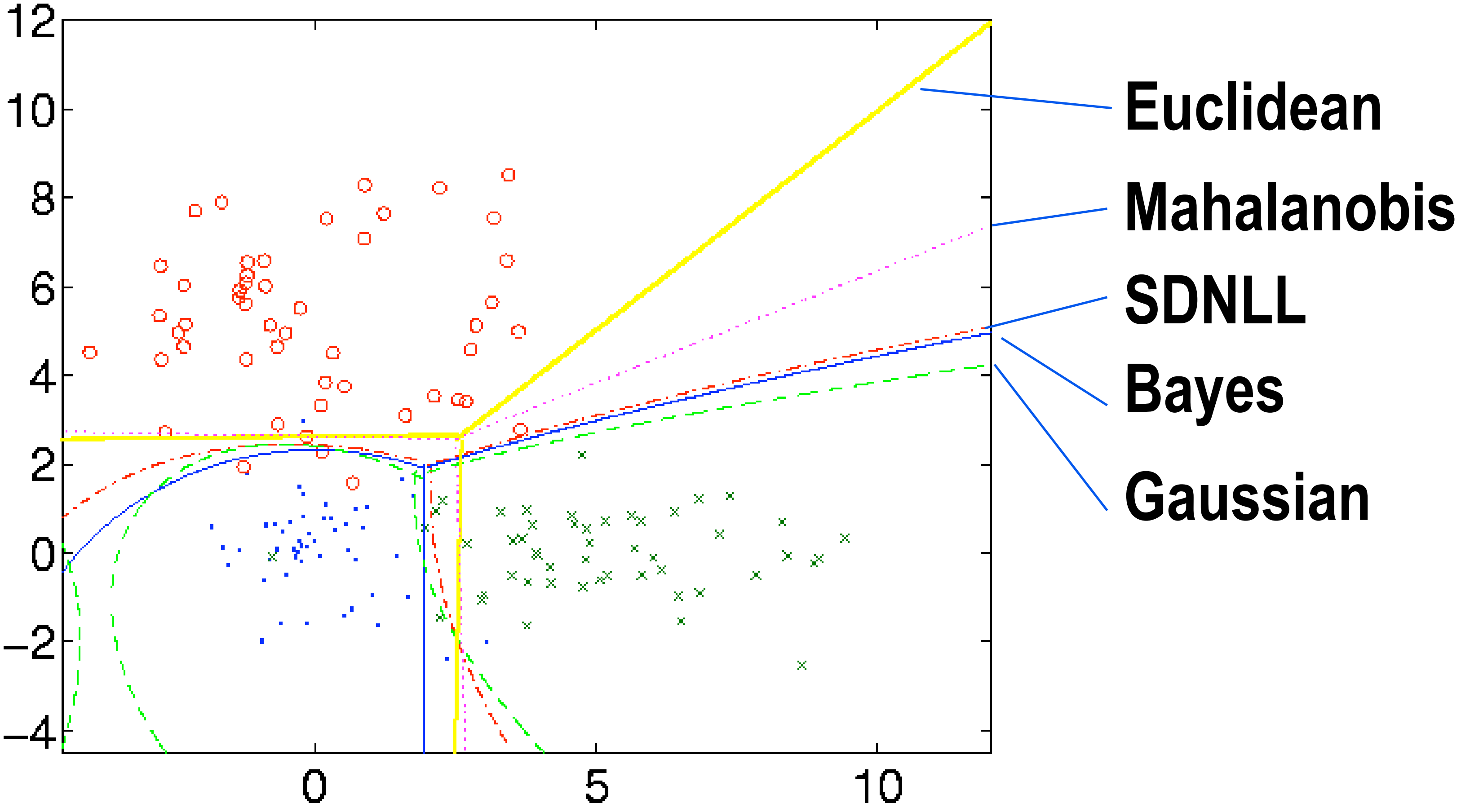
SDNLL

Bayes

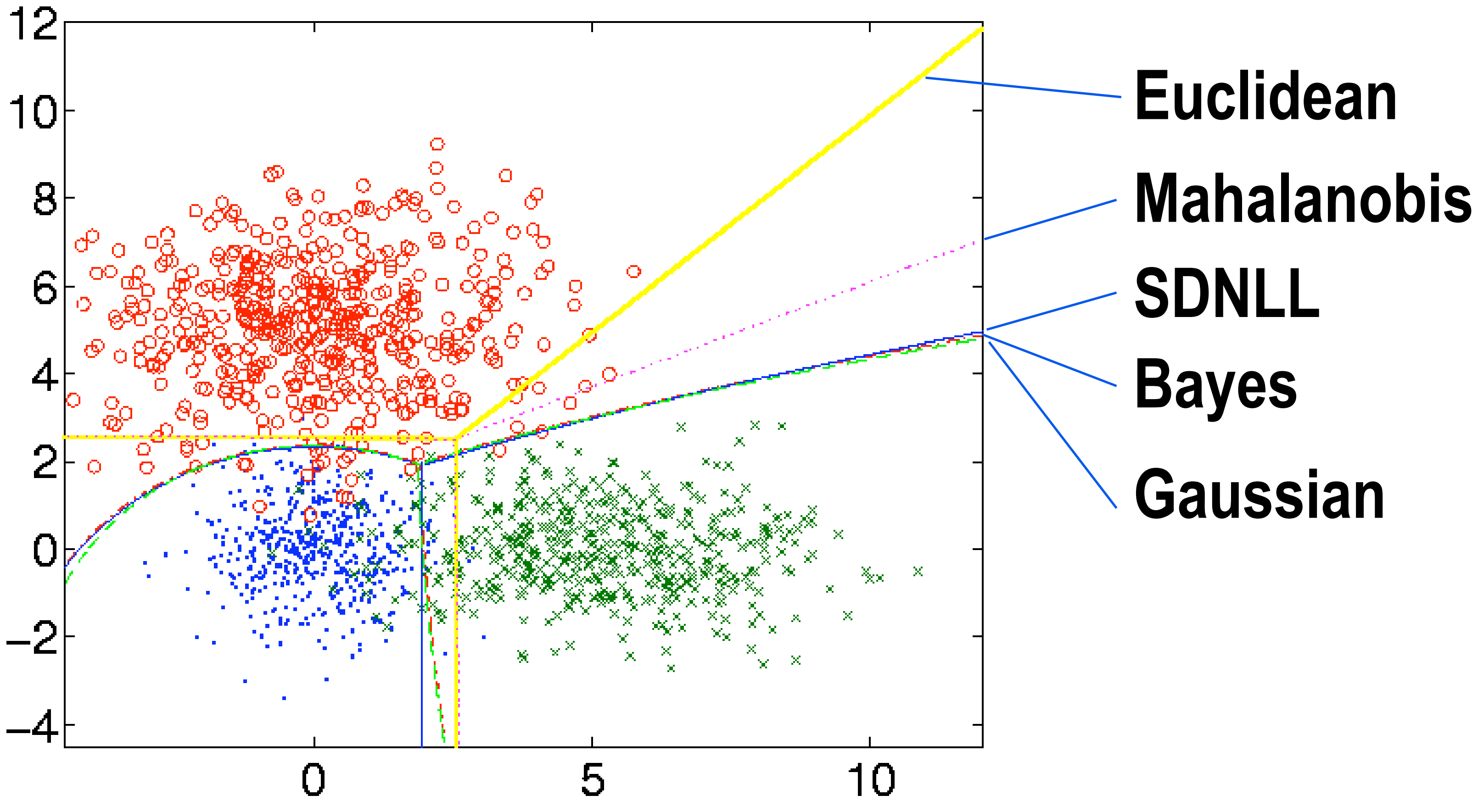
Mahalanobis

Gaussian

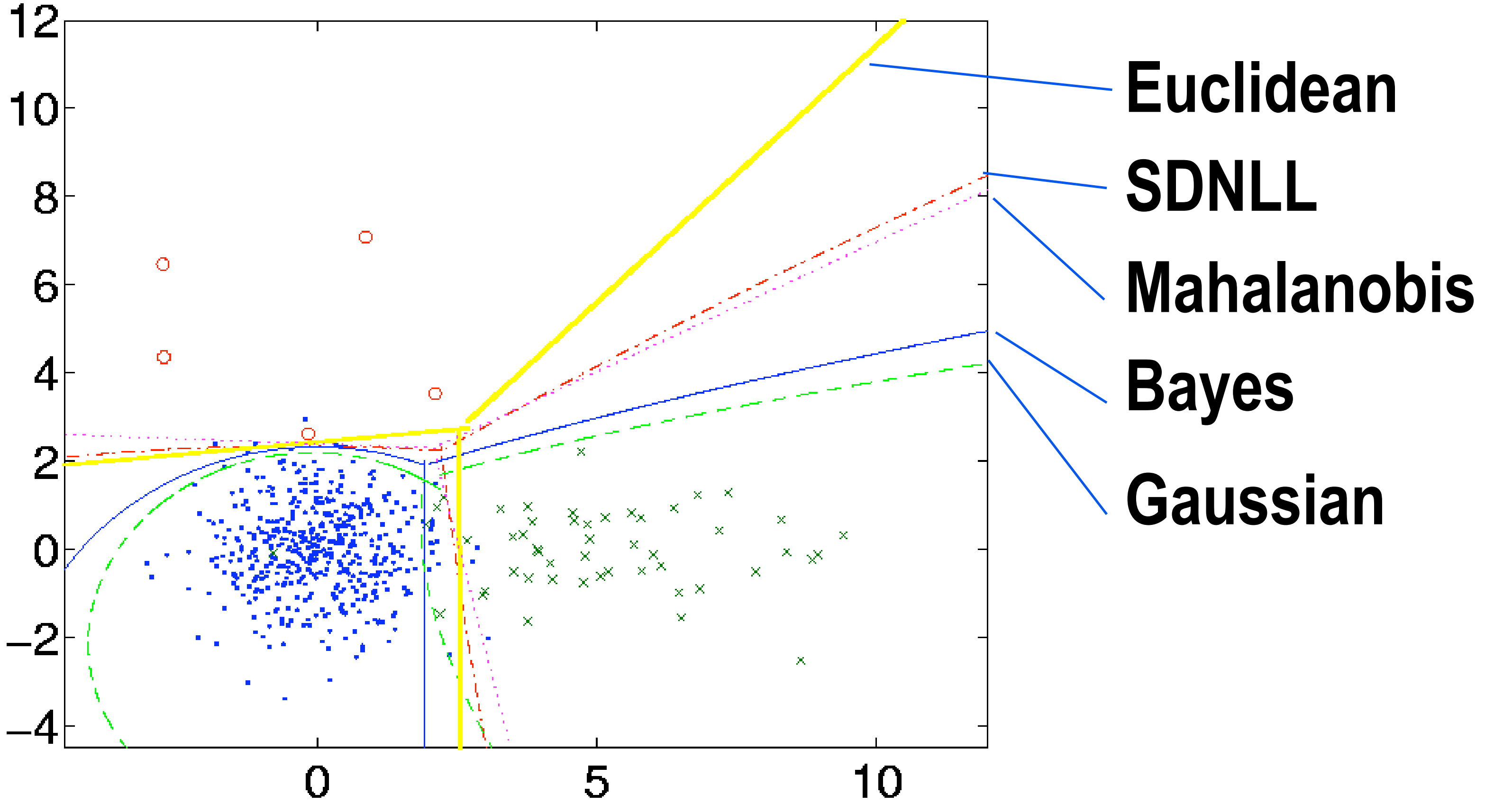
Fitting Class Boundaries: Median Sample Sizes



Fitting Class Boundaries: Large Sample Sizes



Fitting Class Boundaries: Unbalanced Sample Sizes



Industrial Applications of HDR

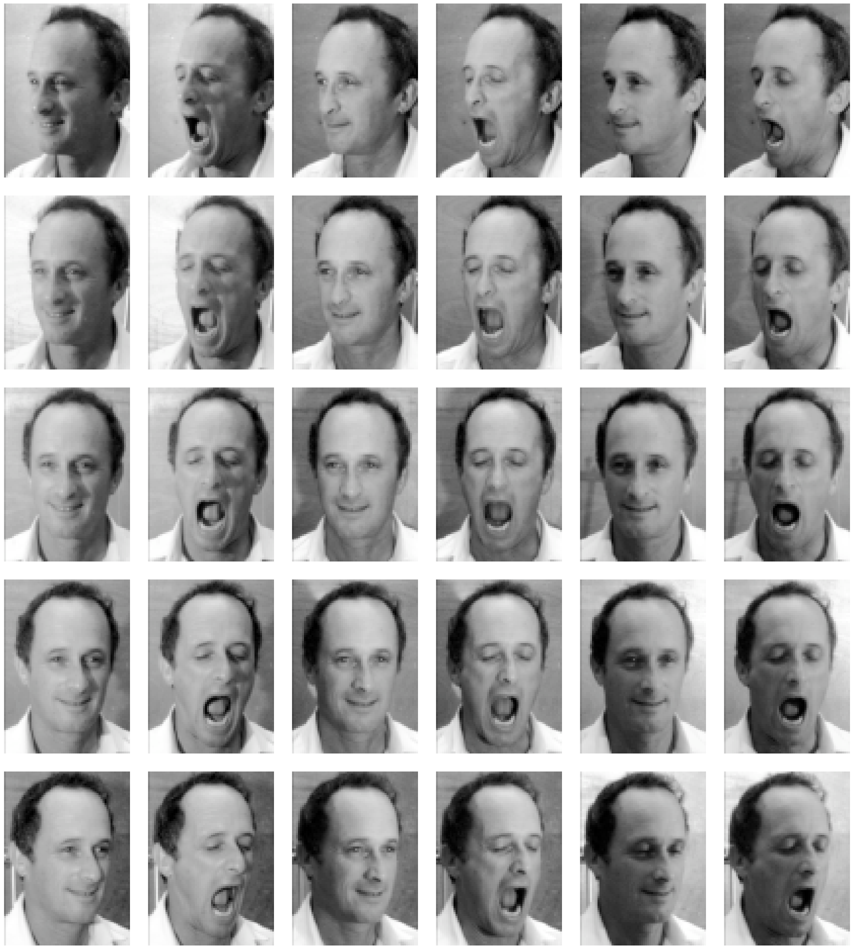
- Innovation:
  - Automatic derivation of features, instead of human designing features
  - Fast real time speed, easier for system development
- Applications:
  - Recognition: recognize shapes or patterns
  - Defect detection: position and types
  - Detection for missing component
  - Pose estimation: given a known pattern, determine its position, orientation, etc
  - Sensor and effector calibration: mapping from sensory space to effector space (learning based calibration)

An Overview of Existing Trees

- Well-known classification and regression trees: not suited for high-dimension input
  - CART and C5.0: univariate tree
  - OC1: multivariate tree
- SAIL: hierarchical discriminant regression (HDR) for high-dimensional input
  - Multivariate tree
  - Automatic subspace derivation: doubly clustered
  - Unify classification and regression problems

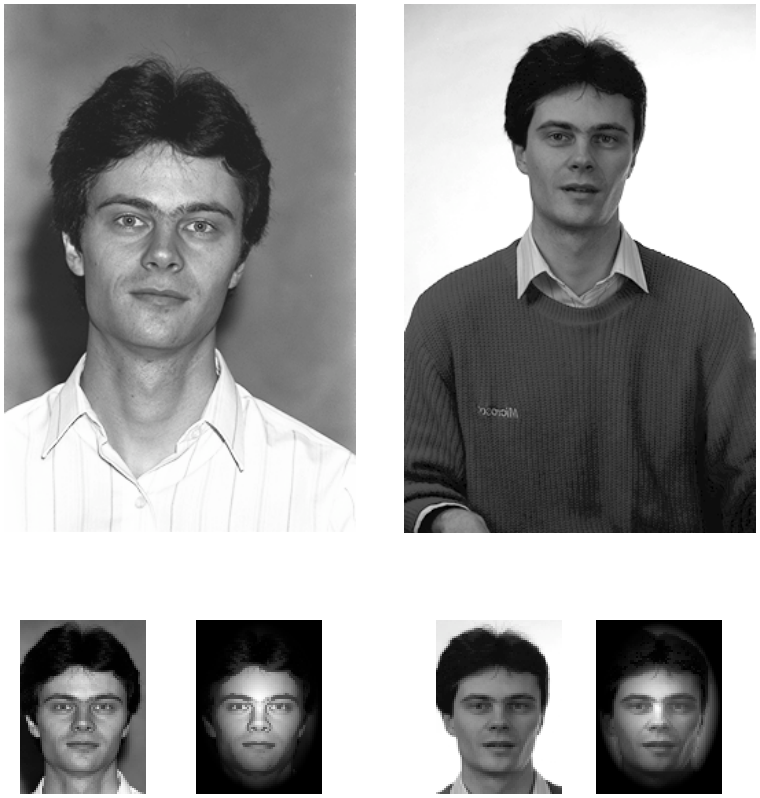
Classification: Weizmann Set

- Total 28 subjects
  - Each subject under:
    5 orientations
    3 lighting conditions
    2 expressions
  - 30 frontal views each
- Leaving-one-out test and cross validation



Classification: FERET Set

- Total 457 subjects
  - 1 subject: 6 images
  - 34 subjects: 4 each
  - 423 subjects: 2 each
- Images are normalized to the same size and intensity range
- Gaussian mask applied to suppress periphery
- Leaving-one-out test



HDR: Performance Comparison

Weizmann face database

Method	Error rate	Search time (s)
PCA	12.80%	0.115
PCA tree	14.58%	0.034
LDA	2.68%	0.105
NN	12.80%	0.164
SVM+PCA	12.5%	0.090
HDR	1.19%	0.078

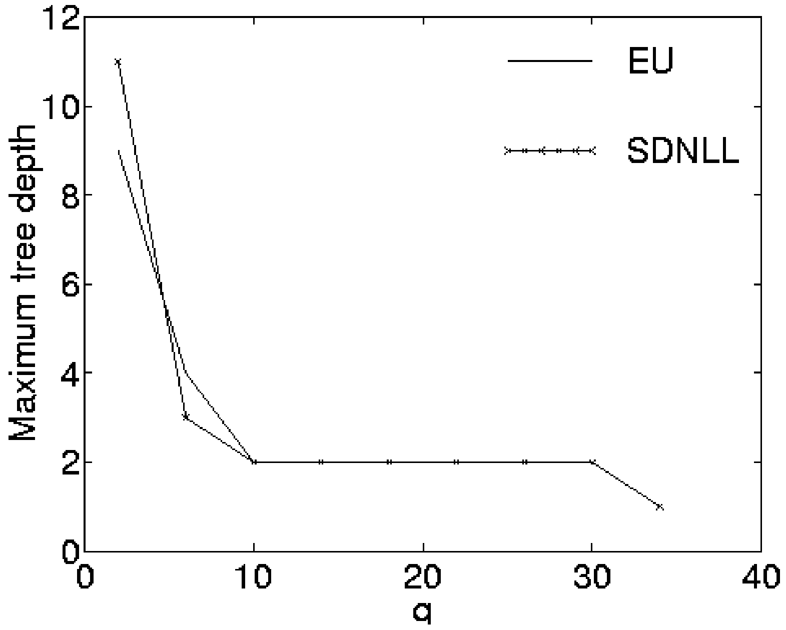
FERET face database

Method	CART

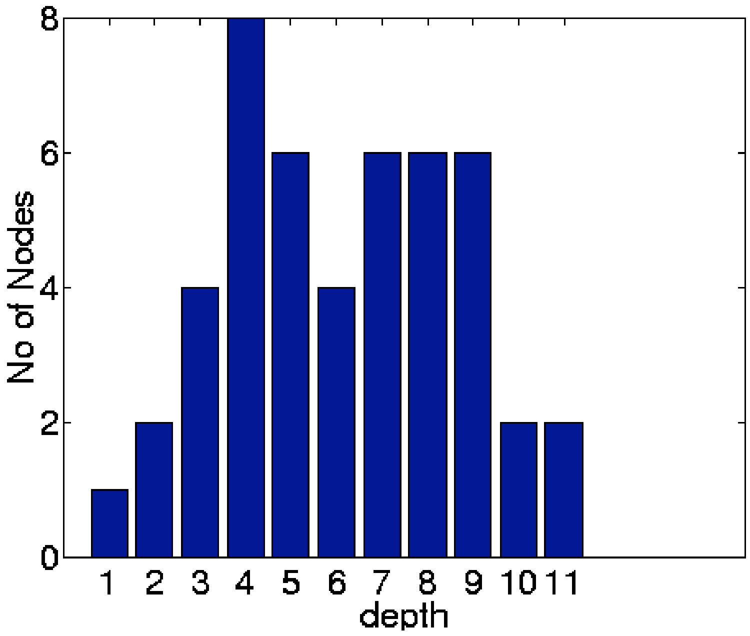
Characteristics of HDR tree

Measured from the trees automatically constructed from FERET face data set

Depth vs degree of tree (q)

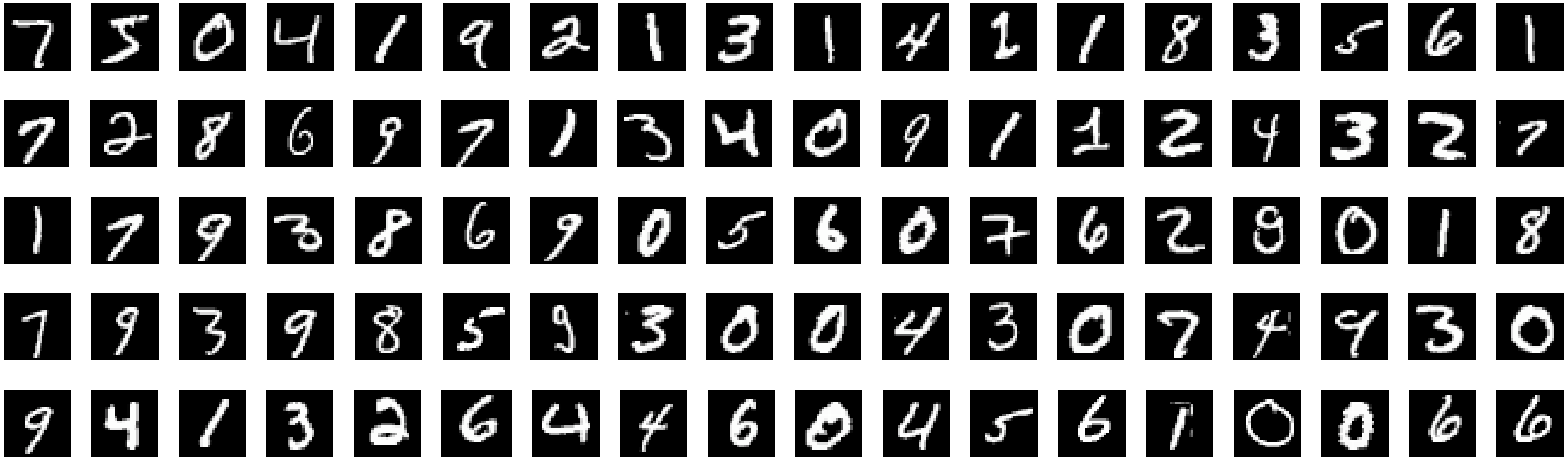


Distribution of nodes over levels (q=2)

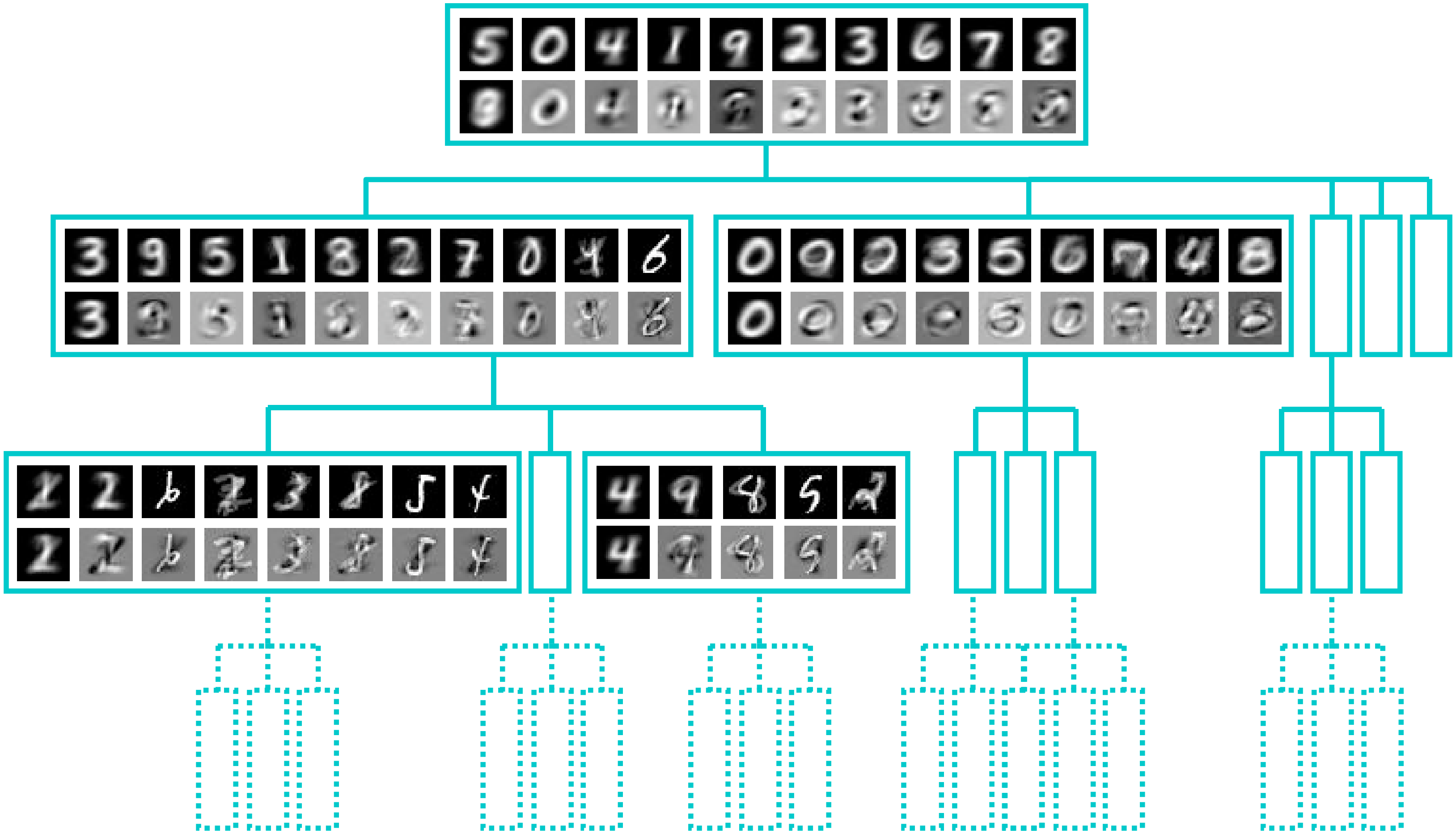


Hand-Written OCR Images

- MNIST hand-written digits
- 60,000 training samples, 10,000 test samples
- IHDR: 3.24% error rate



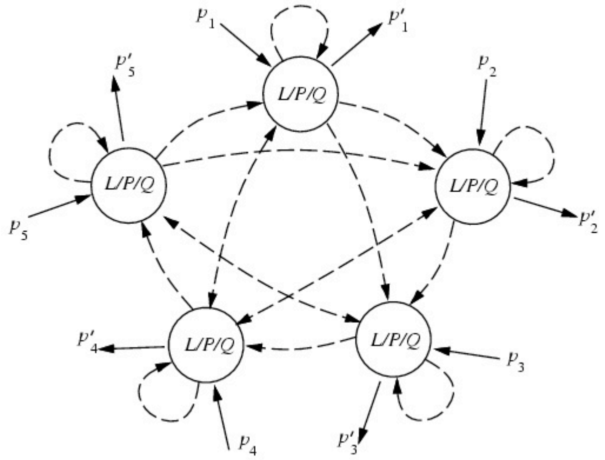
Features in IHDR Tree for OCR

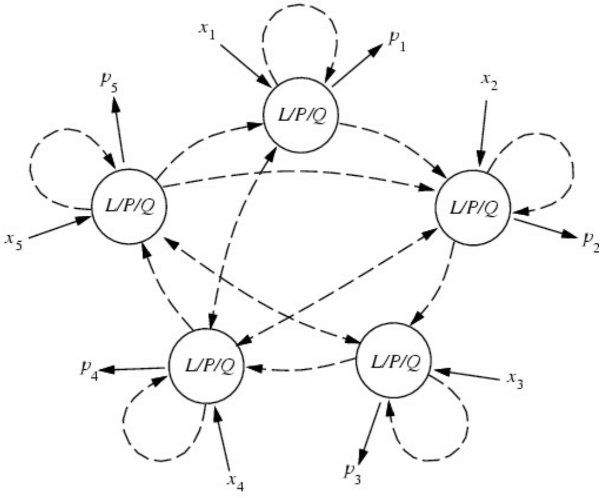


9. Abstraction Levels

SAIL: Open View

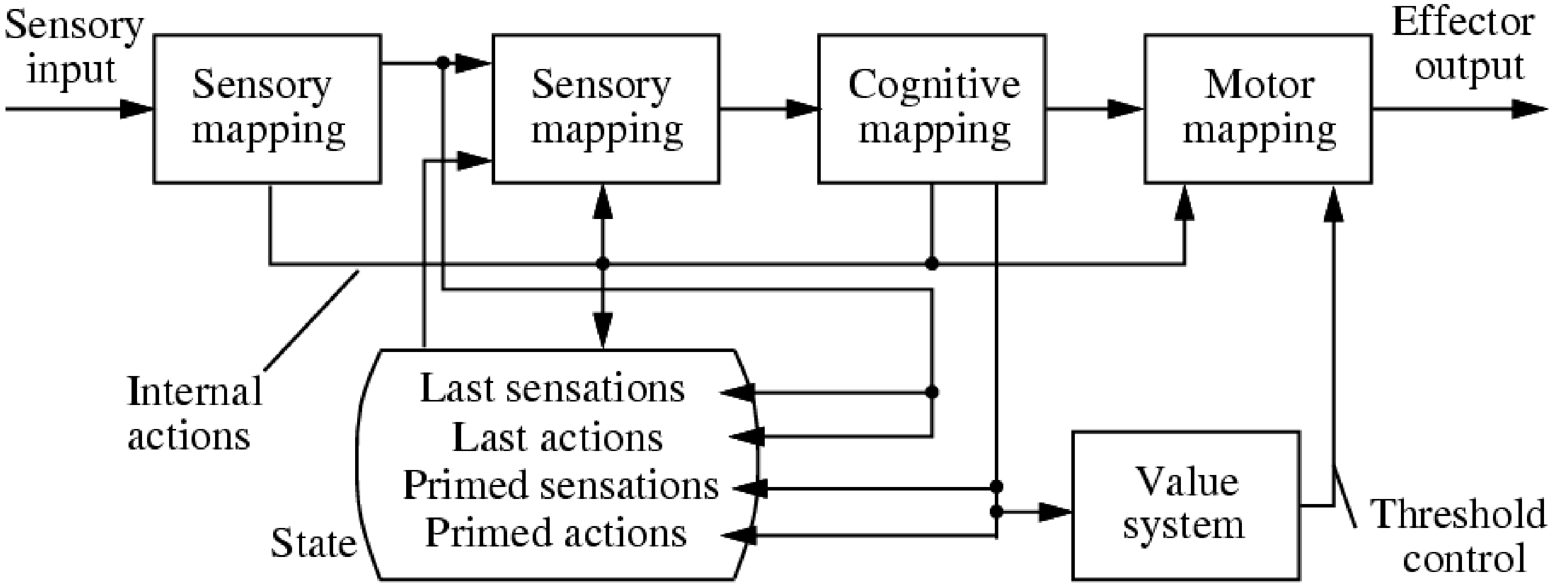
- Each state is a vector in a high dimensional space
- At each state:
  - take a vector input
  - update memory
  - output a vector output
- The finite state machine is autonomously generated from experience!





SAIL: Recursive View

- Active: what is in the state depends on internal behavior



Context state

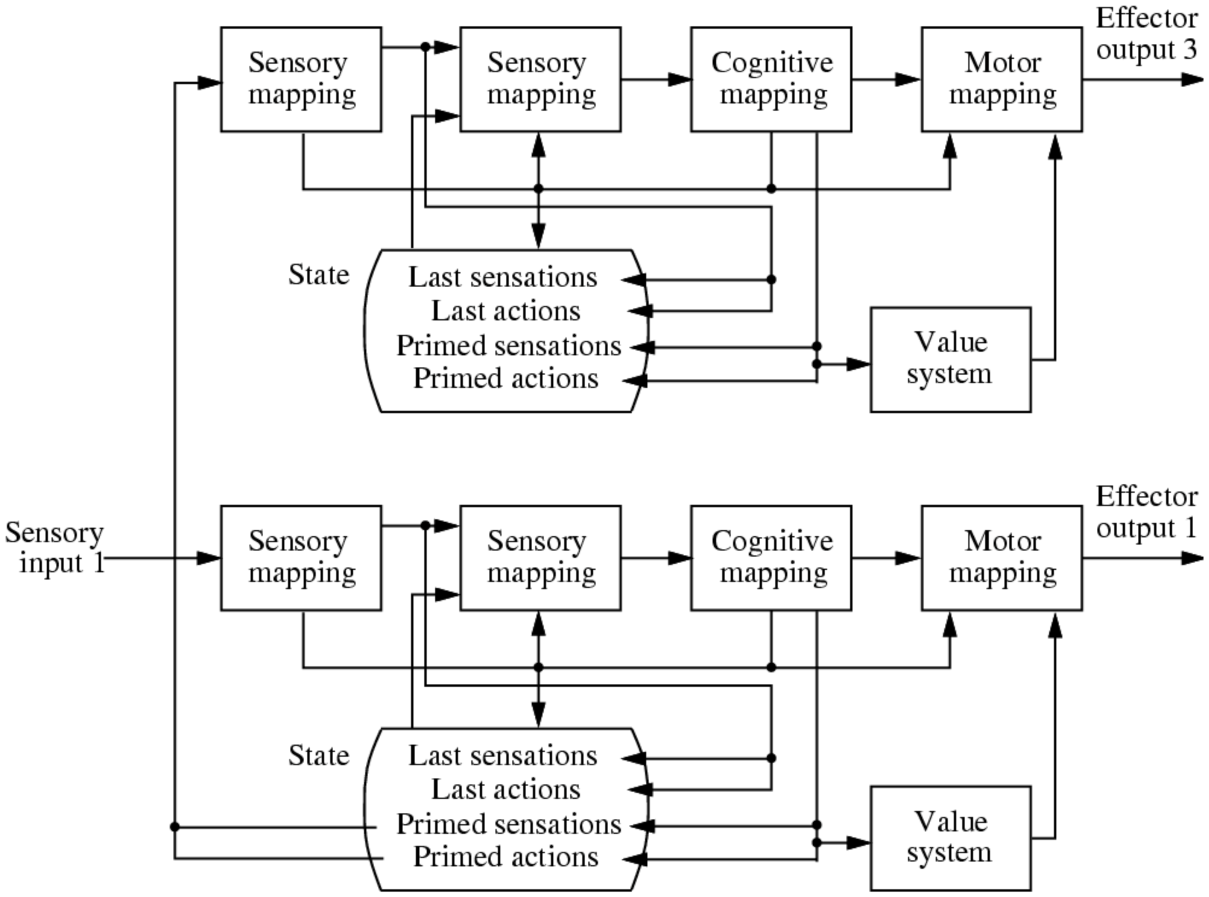
- Content: last experience
  - Sensory input, including “pain” and “pleasure”
  - Recalled action
- Internal action on state
  - Push in: moving on through time
  - Hold: pondering
  - Clear: take the new setting
- How internal actions are learned: shaping

Abstraction Levels

- Higher levels cover larger spatial and temporal scale
- State in the lower level as input to higher level
- Attention as internal action that turn to levels
- Action from attended level is selected as current pending system action
- "Go ahead" internal behavior releases pending action to motor mapping

SAIL: From Low to Higher Levels

- Primed sensations and actions new clustere space
- Composite effects of multi-level internal behaviors



Developing Behaviors

- Early period:
  - Some “innate” behaviors, programmed or learned off-line
  - Occasionally imposed actions
  - Reinforcement learning
  - Behavior shaping: changing reinforcement schedule according agent performance
- Later period:
  - Learn language, from simple to complex
  - Learn the value system: criteria for success
  - Mostly learn through communications in language
  - Less use of low-level reinforcers.

10. System Integration

A Unified View of Multimodality

All sensori-state-motor:

- Vision: image stream
- Speech: cepstrum stream
- Language:
  - sign language: image stream and motor actions
  - spoken language: speech stream in and out
  - written language: image stream
- Reasoning, thinking and decision making

Projects

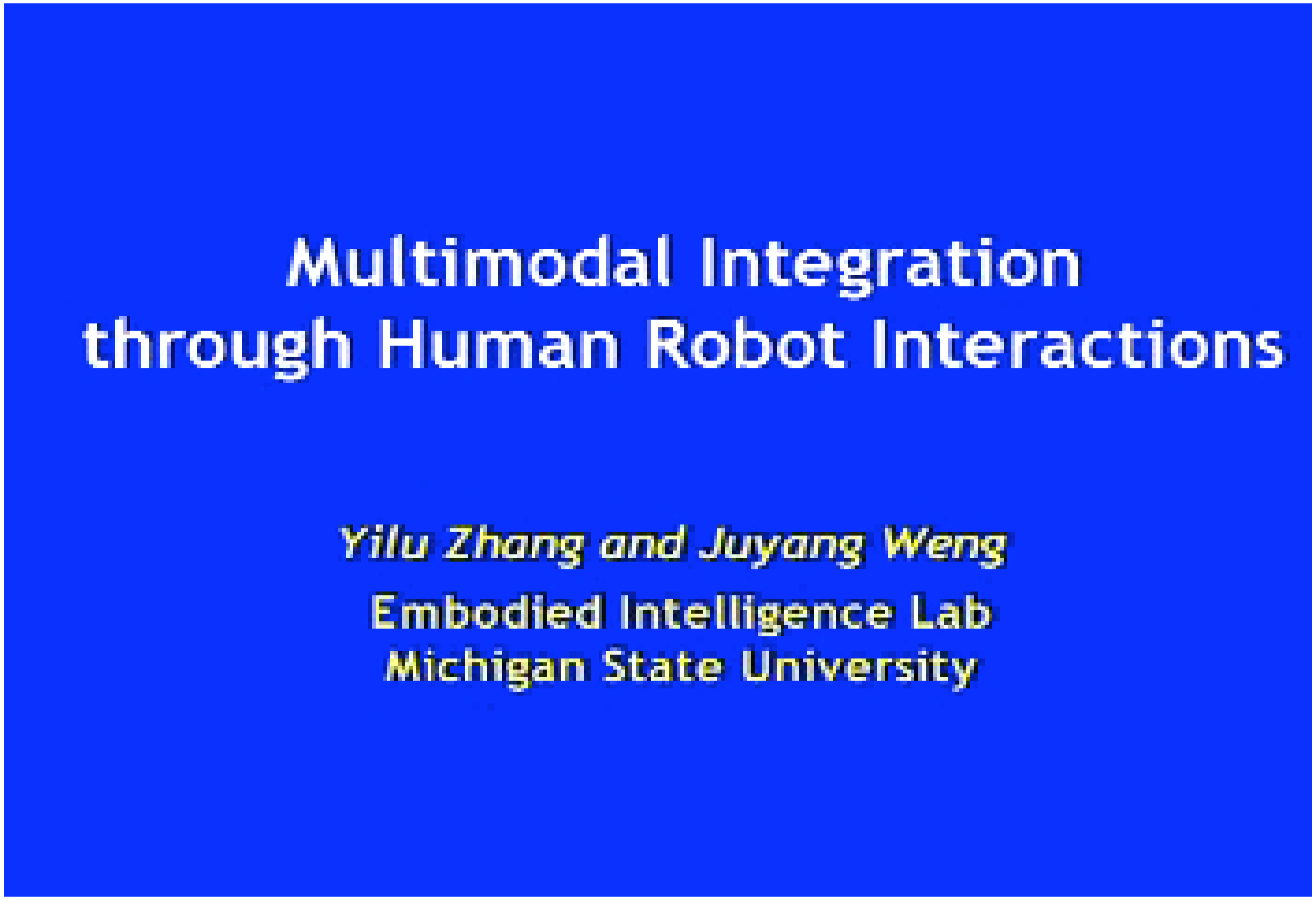
- SHOSLIF (1993 - 2000)
  - Classification and regression tree for high dimensional inputs (D > 5000)
  - Use of PCA and LDA for automatic derivation of features for space partition
- SAIL (1996 - present)
  - Developmental algorithm: SAIL (Self-organizing Autonomous Incremental Learner)
  - SAIL robot, custom made
  - Multimodal integration: vision, speech, language, navigation, object manipulation, and attention.

Participants

- SHOSLIF
  - Laura Blackwood
  - Shaoyun Chen
  - Yuntao Cui
  - Kaman Guentchev
  - Alavi Hamid
  - Sally Howden
  - Gongjun Li
  - Guo L. Liu
  - Hu Mao
  - Jason Sperber
  - Dan L. Swets
  - Jamal Wills
- SAIL
  - Leif E. Alton
  - Micky Badgero
  - Amy L. Bardenhagen
  - Greg J. Bloy
  - Kevin A. Brown
  - Yi Chen
  - Barbara Clark
  - Matthew P. Ebrom
  - Colin H. Evans
  - Wey-Shiuan Hwang
  - Xiao Huang
  - Raja Kanjikuta
  - Ameet Joshi
  - Syed Z. Kazim
  - Jeffrey A. Kohler
  - Yong-Beom Lee
  - Jason Massey
  - Chris Osborn
  - Becky Smith
  - Jason Sperber
  - Changjiang Yang
  - Nan Zhang
  - Yilu Zhang
- Dav
  - Micky Badgero
  - David A. Bordoley
  - Kevin Brown
  - David Cherba
  - Jianda Han
  - Keng Y.Tham
  - Jingliang Wang
  - Shuqing Zeng

SAIL: Reinforcement Leaning and Communicative Learning

Demo: Multimodal Integration



Multimodal Integration
through Human Robot Interactions

Yilu Zhang and Juyang Weng

Embodied Intelligence Lab
Michigan State University

SAIL: Action Chaining Video

SAIL Developmental Robot
- Action Chaining

Yilu Zhang and Juyang Weng

Embodied Intelligence Lab
Michigan State University
East Lansing, MI

February, 2002

SAIL Novelty Test

SAIL Robot

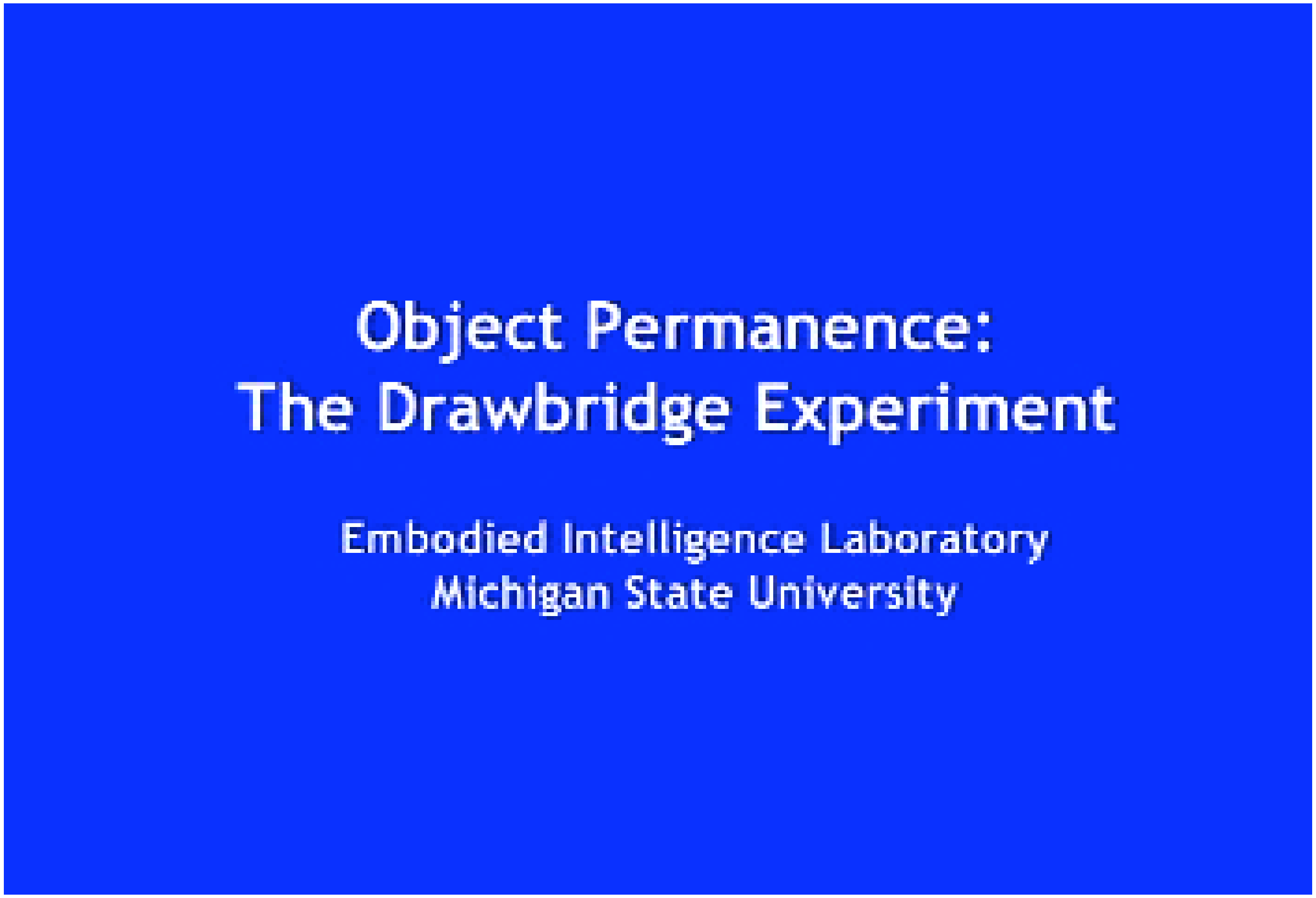
Novelty-Based Visual Attention

Xiao Huang and Juyang Weng

Embodied Intelligence Laboratory

Michigan State University

SAIL’s “Draw-Bridge” Test



Object Permanence:
The Drawbridge Experiment

Embodied Intelligence Laboratory
Michigan State University

Dav: Range-based Collision Avoidance with Attention Selection

Some SAIL References

- J. Weng, “The Living Machine Initiative ,” Technical Report MSU-CSE-96-60, 1996
- International Conf. Humanoid Robots, 1999, Japan
- International Conf. Humanoid Robots, 2000, MIT
- W. Hwang and J. Weng, HDR, IEEE PAMI, Nov. 2000
- Weng et al. “Mental Development by Robots and Animals,” *Science*, Jan. 26, 2001
- Weng, “Developmental Robotics: Theory and Experiments,” International Journal of Humanoid Robotics, vol. 1, no. 2, 2004

Future

- A new industry:
  - New type of software industry
  - Service robots and smart toys entering homes
  - Robots widely used in defense and public environments
- Systematic break throughs in artificial intelligence along all fronts:
  - Vision
  - Speech
  - Natural language
  - Robotics
  - Creative intelligence