Nonstandard engineering principles of brain circuits

R.Granger et al.
Brain Engineering Laboratory
Dartmouth

© R.Granger 2009

Collaborators

Andrew Felch
Andrew Parker
Ashok Chandrashekar
James Hughes
Yune Lee
Amy Palmer
Casey Murray
Sergey Fogelson
Matthew Elkherj
John Eikens
Margaret McCue
Cory Kendrick
Kevin Miller
Stephanie Gagnon
Maria Barsky
Geethmala Srida
Nimit Dhulekar
Karn Seth
Lauren Ladolcetta
Brandyn Webb

BRAIN ENGINEERING LABORATORY at Dartmouth

www.BrainEngineering.org
Richard.Granger@Dartmouth.edu

Yale Cohen
Jay Kralik
Bill Kelley
Peter Tse
Laura Ray
Raj Raizada
Jim Haxby

C.Breazeal, MIT
N.Dutt, Irvine
D.Hammerstrom, Oregon
J.Moorkanikara, Irvine
A.Nicolau, Irvine
C.Priebe, Johns Hopkins
A.Veidenbaum, Irvine

© R.Granger 2009

Nonstandard engineering principles of brain circuits

R.Granger et al.
Brain Engineering Laboratory
Dartmouth

Aim:	Specification of brain circuit computation
Why:	Brains do tasks on which engineering methods still fail
Characteristics	- Intrinsically parallel - Scaling (~10^5 x) - Learning-based - End-to-end integration
Status:	- Algorithms, Architecture, Analyses - Linear scaling - Parallel silicon implementations - Fielded initial sensor & robot implementations
Principles:	**WHY** do brain circuits succeed? **WHAT** internal constructs do they build? **HOW** can we imitate them?

© R.Granger 2009

Nonstandard engineering principles of brain circuits

R.Granger et al.
Brain Engineering Laboratory
Dartmouth

N cores ≠ N x speedup

"Moore's Gap": software / algorithm speed vs hardware

Amdahl's law: $$\frac{1}{S + \frac{(1-S)}{N}}$$

Key question: **Architecture design** for parallelism

WHY do brain circuits succeed?
WHAT internal constructs do they build?
HOW can we imitate them?

© R.Granger 2009

OUTLINE

Circuits

Algorithms

$$W = \frac{\alpha}{\eta L}\,\frac{\log(1 - P_{RE}^{1/\eta L})}{\log(1 - \frac{1}{\sigma})}$$

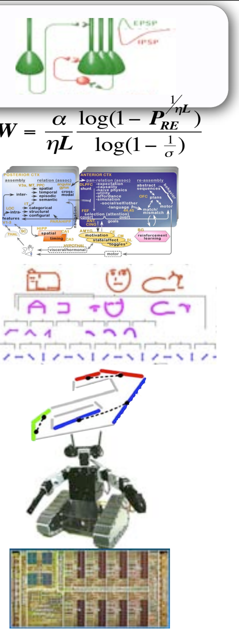
Architecture

Learning

Abilities

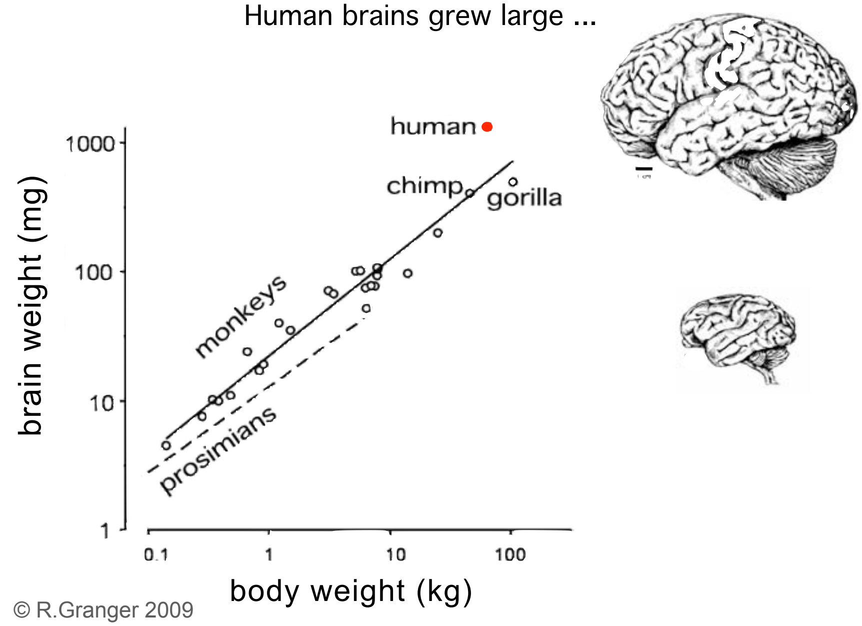
Robots

Silicon



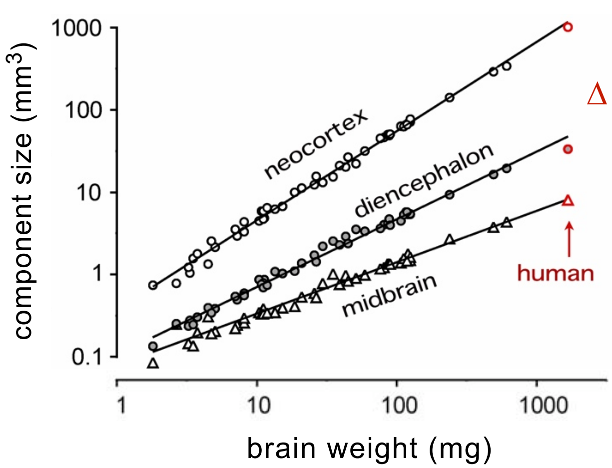
© R.Granger 2009

Human brains grew large ...



© R.Granger 2009

Components scale allometrically



Δ Size

⇓

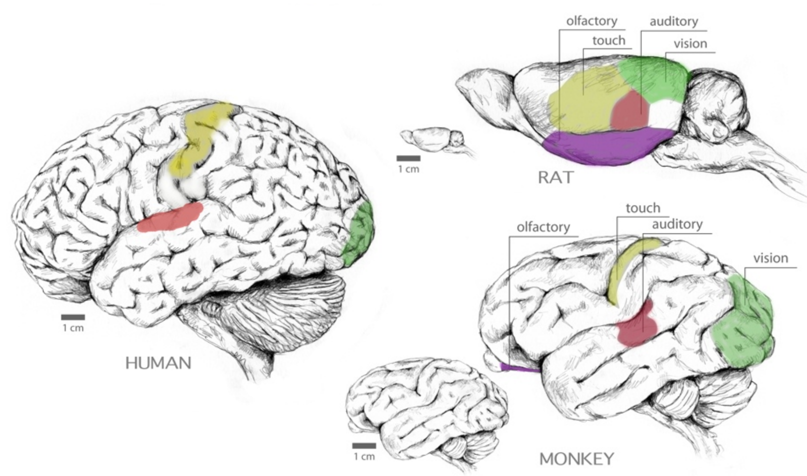
Δ Abilities

(?)

© R.Granger 2009

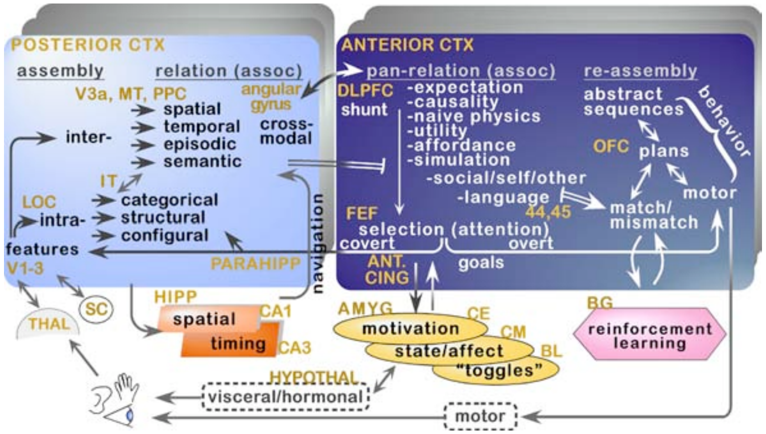
(Lynch & Granger 2008)

Brain size ↑, specializations ↓ : Shared mechanisms ↑



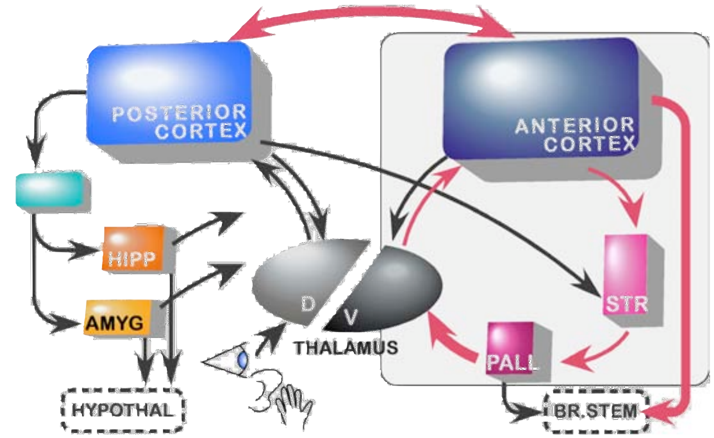
© R.Granger 2009

Architectural organization



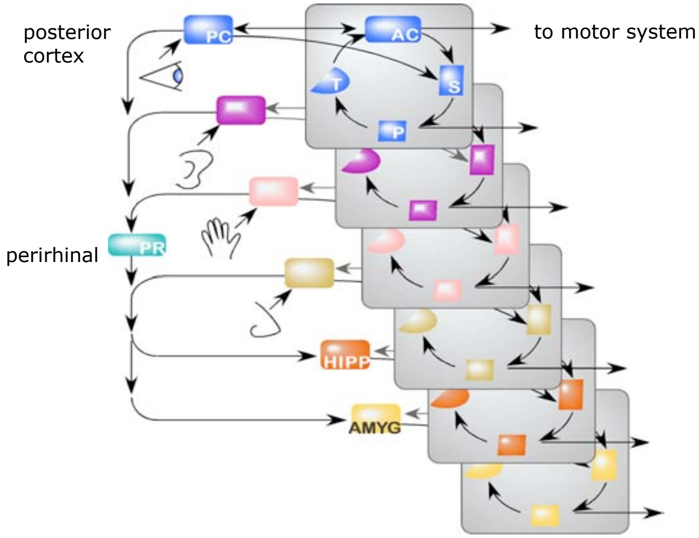
© R.Granger 2009

Architectural organization



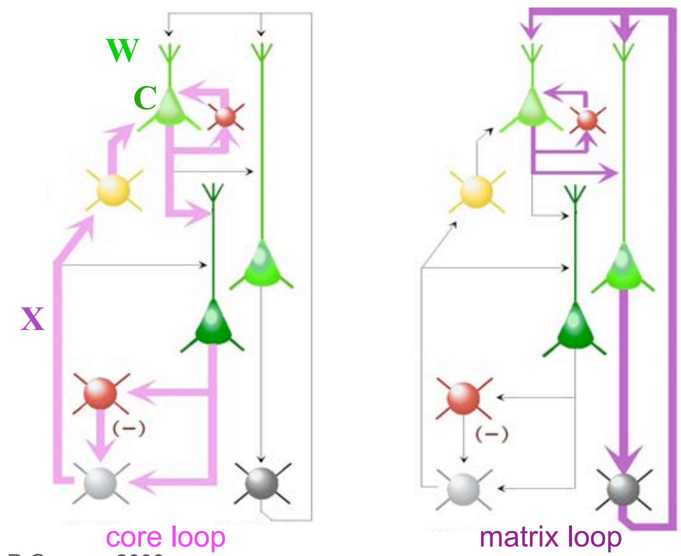
© R.Granger 2009

Architectural organization



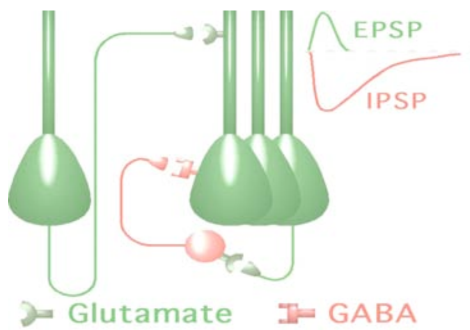
© R.Granger 2009

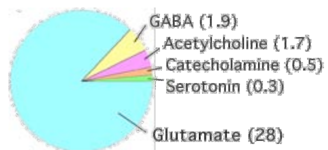
Basic circuits: Thalamocortical loops



© R.Granger 2009

Circuit components





- **probabilistic**
- **slow**
- **sparse**
- **low-precision**

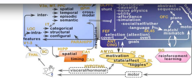
© R.Granger 2009

OUTLINE

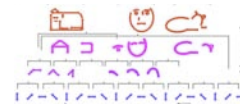
Circuits

Algorithms $W = \frac{\alpha}{\eta L}\frac{\log(1 - P_{RE}^{1/\eta L})}{\log(1 - \frac{1}{\sigma})}$

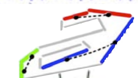
Architecture



Learning



Abilities

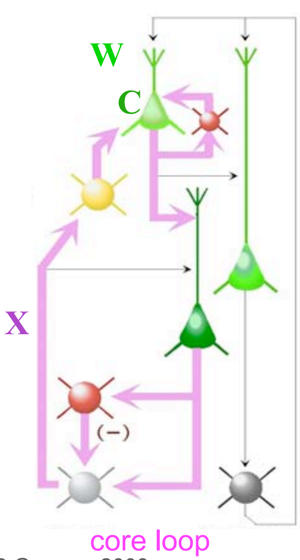


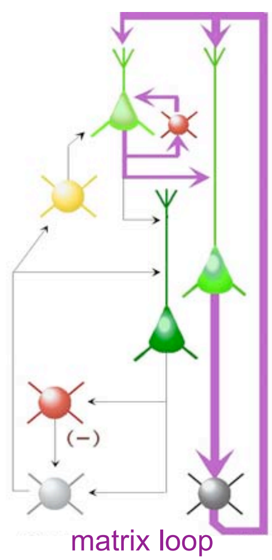
Robots

Silicon

© R.Granger 2009

Basic circuits: Thalamocortical loops

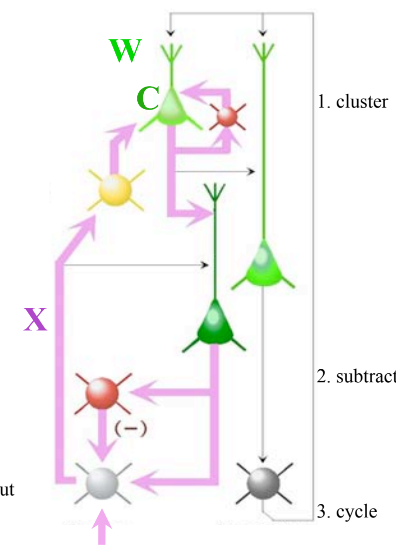




© R.Granger 2009

Thalamocortical core loop

Static inputs (fixation)

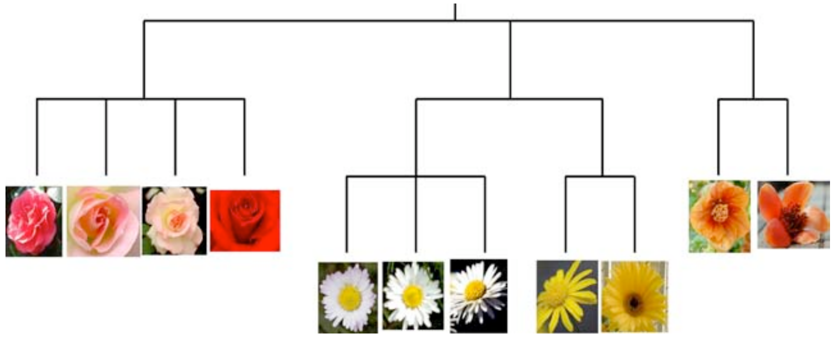


© R.Granger 2009

(Granger et al., 1996; Rodriguez et al. 2004; Granger et al. 2005)

Thalamocortical core loop

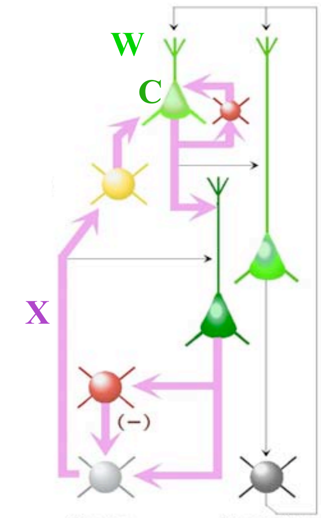
successive approximation / hierarchical clustering



High-precision computations from low-precision components

© R.Granger 2009

Core loop: Iterative re-use to compute successive sub-categories



for input X
  for C $\in$ win(X,W)
    $W_j \Leftarrow W_j + k(X - C)$
  end_for
$X \Leftarrow X - \text{mean}(\text{win}(X,W))$
end_for

where
X = input activity pattern (vector);
W = layer I synaptic weight matrix;
C = responding superficial layer cells (col. vector);
k = learning rate parameter;
win(X,W) = cell most responsive to X, [e.g., $\forall \mathbf{j}$, $\max(X \cdot W_j)$]

© R.Granger 2009

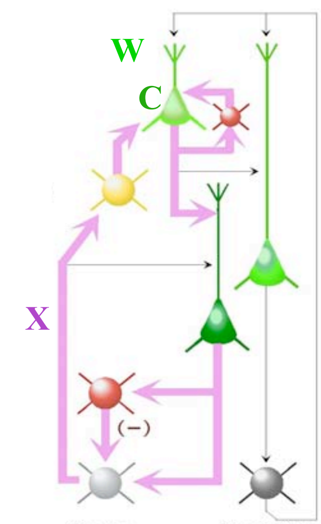
(Rodriguez et al., 2004)

Engineering principle

Re-use of components over time yields
high-precision computation
from low-precision components

© R.Granger 2009

Core loop: Iterative re-use computes successive sub-categories



for input X
  for C $\in$ win(X,W)
    $W_j \Leftarrow W_j + k(X - C)$
  end_for
$X \Leftarrow X - \text{mean}(\text{win}(X,W))$
end_for

where
X = input activity pattern (vector);
W = layer I synaptic weight matrix;
C = responding superficial layer cells (col. vector);
k = learning rate parameter;
win(X,W) = cell most responsive to X, [e.g., $\forall \mathbf{j}$, $\max(X \cdot W_j)$]

© R.Granger 2009

(Rodriguez et al., 2004)

Core loop: Efficient space & time complexity

Space costs

For complete separability of n cues,
Bottom of hierarchy must contain at least **n** units
tree consists of $\log B_n$ hierarchical layers,
where B = avg branching factor at each level

=> complete hierarchy contains $\sim n\left[\frac{B}{B-1}\right]$ units,

=> Space to learn n cues of dimensionality N = $O(nN)$

Time costs

Three costs:

i) Summation	ii) compute winners	iii) weight mod

In serial, after processing all levels

i) = $O(nN)$	ii) = $O(n)$	iii) = $O(N \log n)$

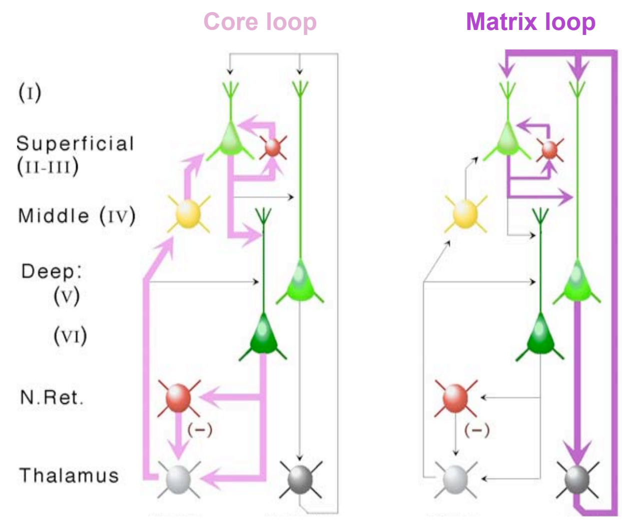
In parallel

i) = $O(\log N)$	ii) = $O(\log n)$	iii) = constant

© R.Granger 2009

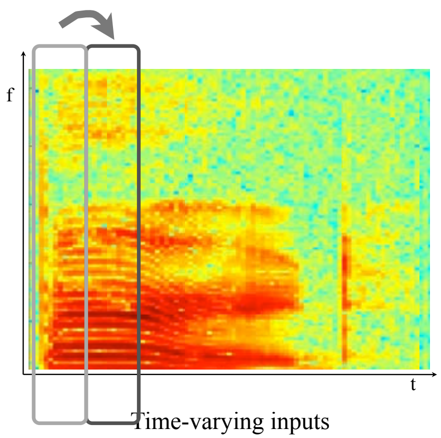
(Rodriguez et al., 2004)

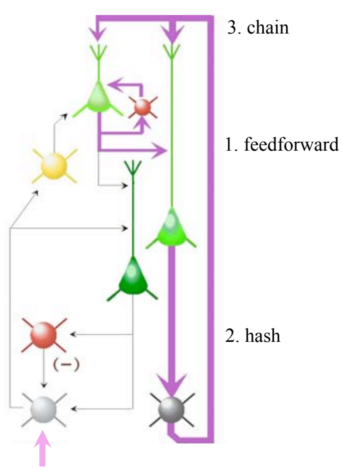
Basic circuits: Thalamocortical loops



© R.Granger 2009

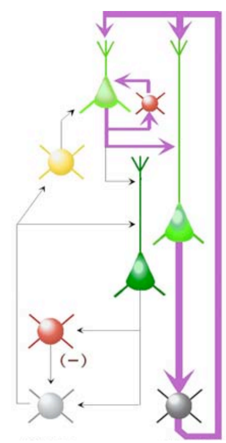
Matrix loop: Learning sequences





© R.Granger 2009

Matrix loop: Learning sequences



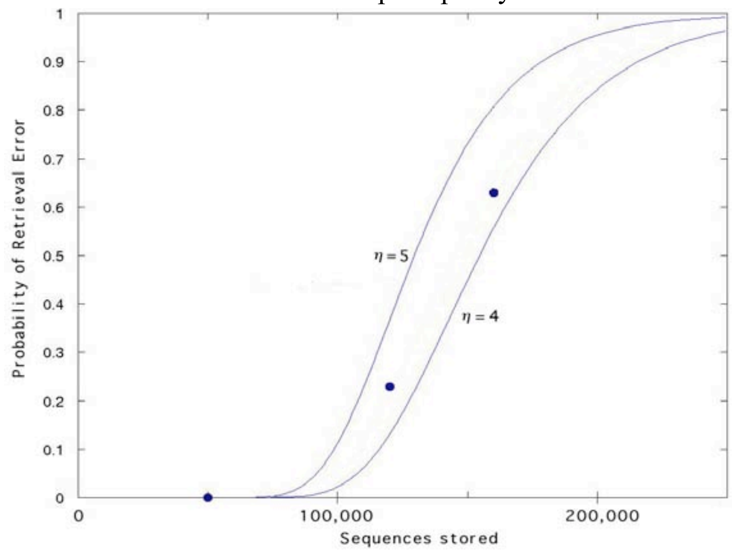

```
for input sequence X(L)
    for C ε TopogSuperficialResponse(X(L))
        for V(s) ε C ∩ NNtResponse(X(L-1))
            Potentiate( V(s) )
            NNt(L) ⇐ NontopogDeepResponse(V)
        end_for
    end_for
end_for
```

(where
L = length of the input sequence;
C = columns activated at step X(L);
V(s) = the synaptic vector of a layer V cell,
NNt(L) = nonspecific nucleus response to layer V feedback.)

© R.Granger 2009

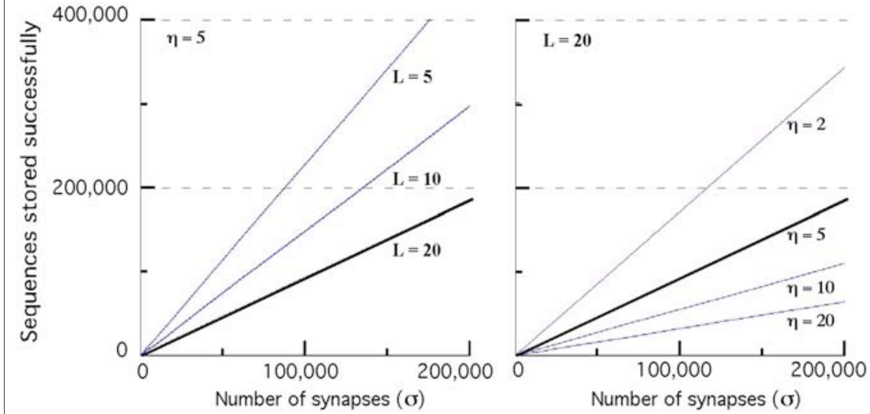
(Granger et al., 1996; Rodriguez et al. 2004; Granger et al. 2005)

Matrix loop: Capacity



© R.Granger 2009 $P_{RE} = P_E^{\eta L} = [1-(1-\frac{1}{\sigma})^{WL\eta/\alpha}]^{\eta L}$ (Rodriguez et al., 2004)

Matrix loop: Capacity scales linearly



Strings (***W***) successfully stored within a fixed error rate (0.001 for this graph) for a network of specified size (σ synapses) as a function of string length (***L***) and number of synapses potentiated per trial (η).

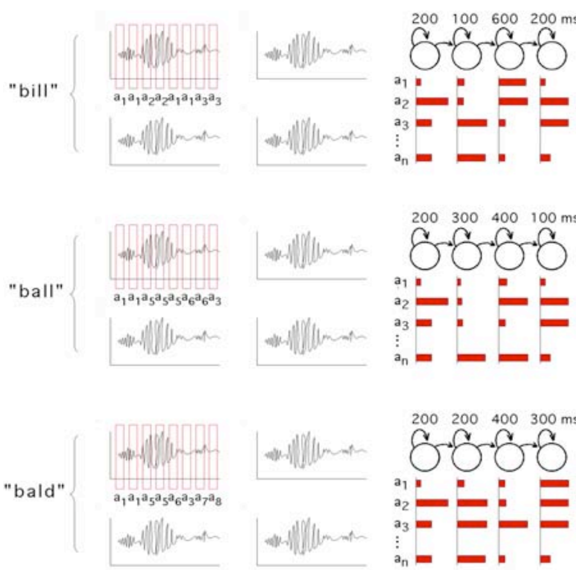
$$W = \frac{\alpha}{\eta L}\frac{\log(1-P_{RE}^{1/\eta L})}{\log(1-\frac{1}{\sigma})}$$

$$= O(\sigma)$$

© R.Granger 2009

(Rodriguez et al., 2004)

Matrix loop: Comparison w/ HMMs



HMM time / space costs:

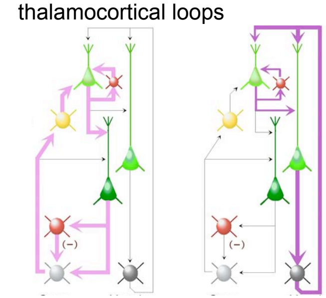
training:
- time: $O(CN(M^2 + MD^2))$
- space: $O(NM^2 + MD^2)$

operation:
- time: $O(NM^2 + NMD^2))$
- space: $O(M^2 + MD^2)$

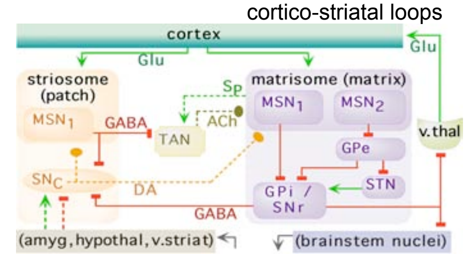
(Rodriguez et al., 2004)

Distinct telencephalic circuits

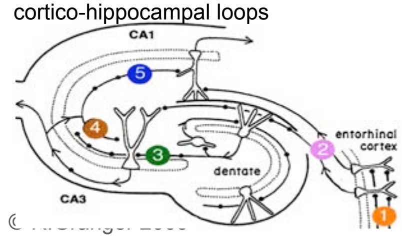
thalamocortical loops



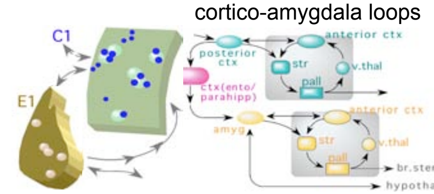
striatal complex / BG
cortico-striatal loops



hippocampal fields
cortico-hippocampal loops

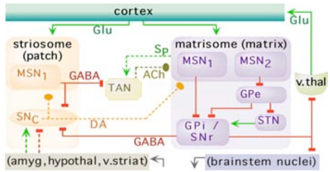


amygdala fields
cortico-amygdala loops



Summary: Striatum / Hippocampus / Amygdala

Basal ganglia / striatal complex

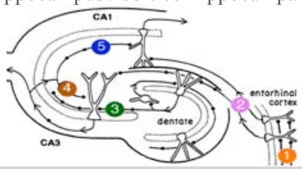


Reinforcement learning
(critics; selection)

Five circuit designs
Eight receptor types
Six neuron types
Two plasticity rules

Schultz et al., 2000
Granger et al., 2004
Granger et al., 2005

Hippocampus / cortico-hippocampal loops

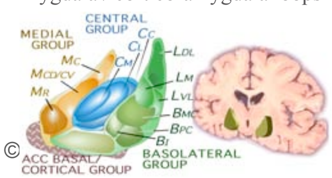


Compression / dilation
(redundancy)

Six circuit designs
Four receptor types
Four neuron types
Three plasticity rules

Gluck et al., 1993
Granger et al., 1994
Myers et al., 1995
Kilborn et al., 1997

Amygdala / cortico-amygdala loops



Filters / region selection
(toggles)

Four circuit designs
Five receptor types
Five neuron types
Two plasticity rules

Parker et al., in prep

©

Derived brain circuit mechanisms: "Instruction set"

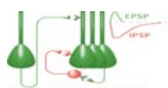
Summary of algorithms derived from circuits

thalamocortical circuits 1: clustering, hierarchies
thalamocortical circuits 2: sequences, chaining, hash codes
striatal complex / basal ganglia: reinforcement learning
hippocampal fields: time dilation / compression
amygdala nuclei: filters / toggles

© R.Granger 2009

OUTLINE

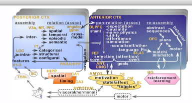
Circuits



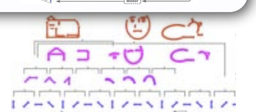
Algorithms

$$W = \frac{\alpha}{\eta L}\,\frac{\log(1 - P_{RE}^{1/\eta L})}{\log(1 - \frac{1}{\sigma})}$$

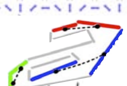
Architecture



Learning

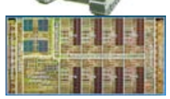


Abilities



Robots

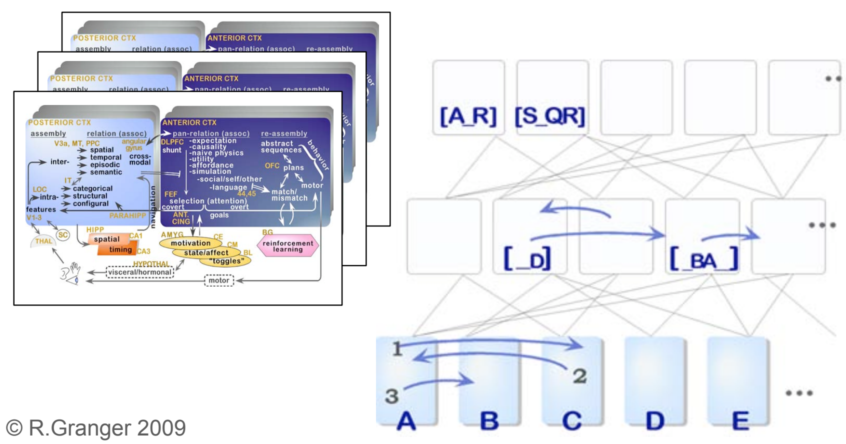
Silicon



© R.Granger 2009

Integrated operation

- Sequences (of categories) of sequences (of categories) of sequences . . .
- Grammars



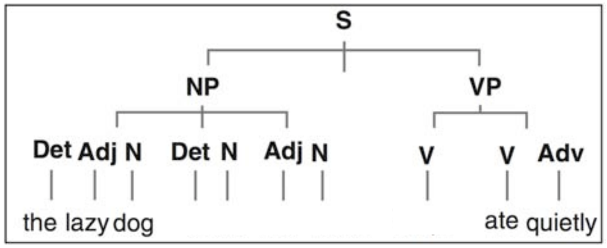
© R.Granger 2009

Grammars = nested sequences of categories

S → NP VP

NP → Det Adj N | Det N | Adj N

VP → V | V Adv



Learned grammatical relations among components

(

Integrated operation

Summary of algorithms derived from circuits

- thalamocortical circuits 1: clustering, hierarchies
- thalamocortical circuits 2: sequences, chaining, hash codes
- striatal complex / basal ganglia: reinforcement learning
- hippocampal fields: time dilation / compression
- amygdala nuclei: filters / toggles

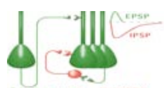
Shared data structures

nested sequences of categories (grammars)

© R.Granger 2009

OUTLINE

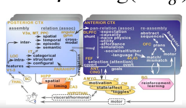
Circuits



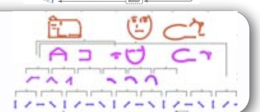
Algorithms

$$W = \frac{\alpha}{\eta L}\,\frac{\log(1 - P_{RE}^{1/\eta L})}{\log(1 - \frac{1}{\sigma})}$$

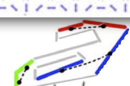
Architecture



Learning

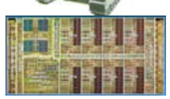


Abilities



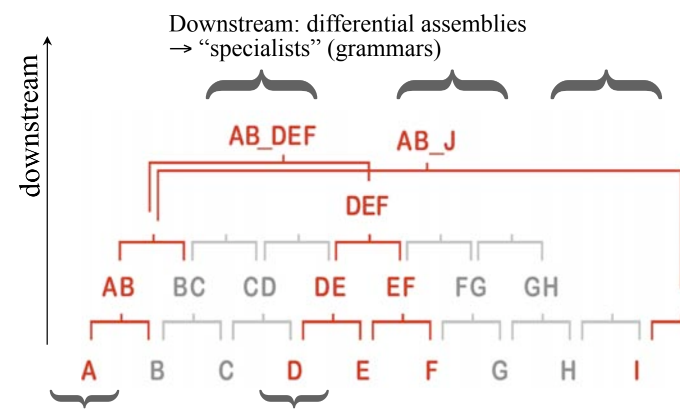
Robots

Silicon



© R.Granger 2009

Data structures: learned perceptual grammars



Learned grammatical relations among components

Adding new entries ⇒ qualitatively new behaviors

© R.Granger 2009

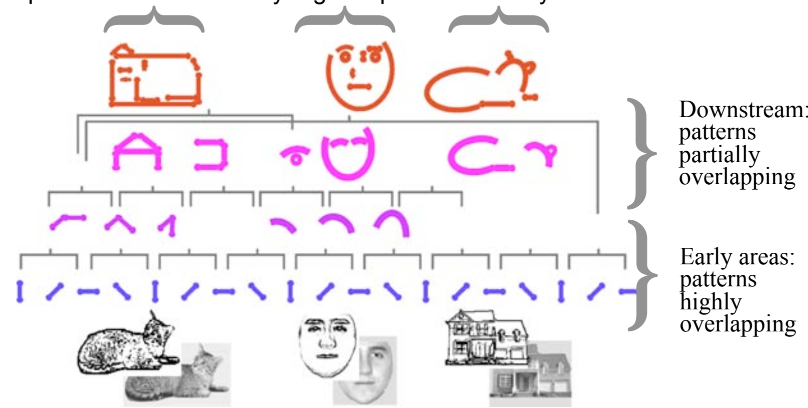
Engineering principle

In grammars,
quantitative change ⇒ qualitative change

© R.Granger 2009

Memory characteristics

- distributed representations
- different regions → different feature assemblies
- representations scaled by region-specific similarity

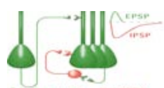


Learned grammatical relations among components

© R.Granger 2009

OUTLINE

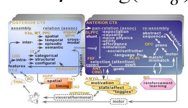
Circuits



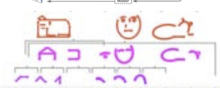
Algorithms

$$W = \frac{\alpha}{\eta L}\,\frac{\log(1 - P_{RE}^{1/\eta L})}{\log(1 - \frac{1}{\sigma})}$$

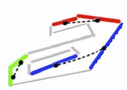
Architecture



Learning

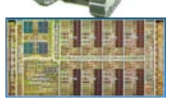


Abilities



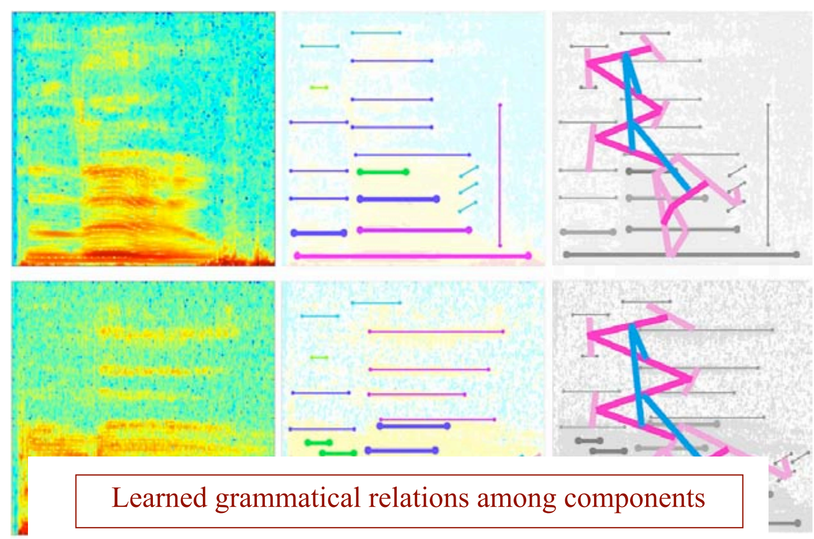
Robots

Silicon



© R.Granger 2009

Auditory processing: Learning auditory "landmarks"



Learned grammatical relations among components

© R.Granger 2009

Engineering principle

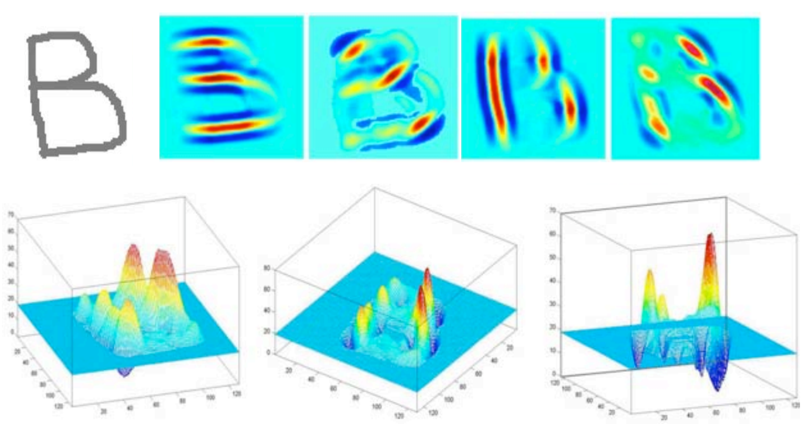
Grammar recognition enables “partial matching” of otherwise-intractable complex signals

© R.Granger 2009

Visual recognition 1: Difficult images

© R.Granger 2009

Visual TCSL algorithm: specialized front-end processing (multiple frequencies,scales)

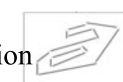


$$g_{\lambda,\theta,\varphi}(x,y) = e^{-(x'^2+\gamma^2 y'^2)/2\sigma^2} \cos(2\pi \frac{x'}{\lambda} + \varphi)$$

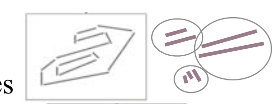
© R.Granger 2009

Method summary

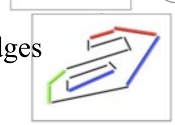
1) Front end: Partial edge extraction



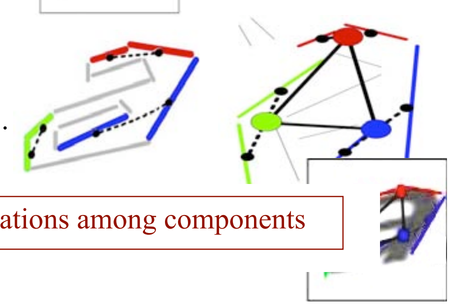
2) Categorization of similar edges



Sequential relations between edges



3) Hierarchical iteration: sequences of categories of …



Learned grammatical relations among components

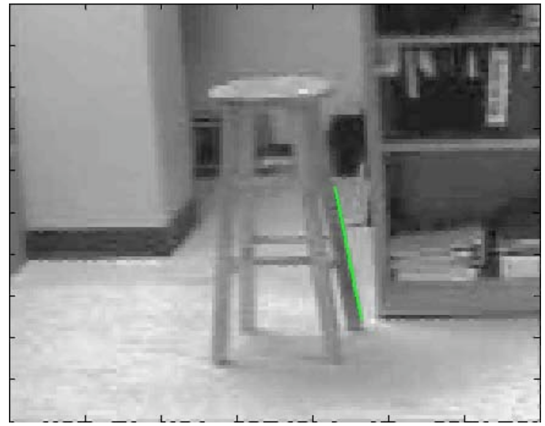
© R.Granger 2009

Engineering principle

Brain circuit methods are applicable across different domains

© R.Granger 2009

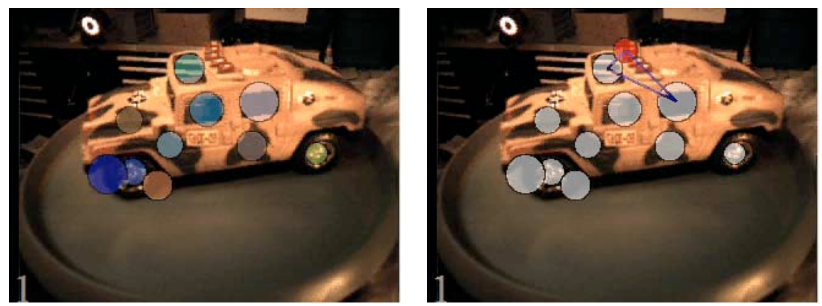
Visual learning and active recognition



© R.Granger 2009

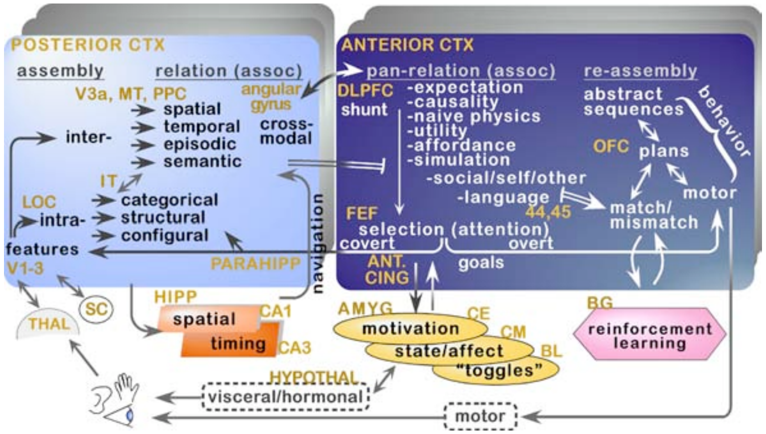
© R.Granger 2009

Visual learning and active recognition



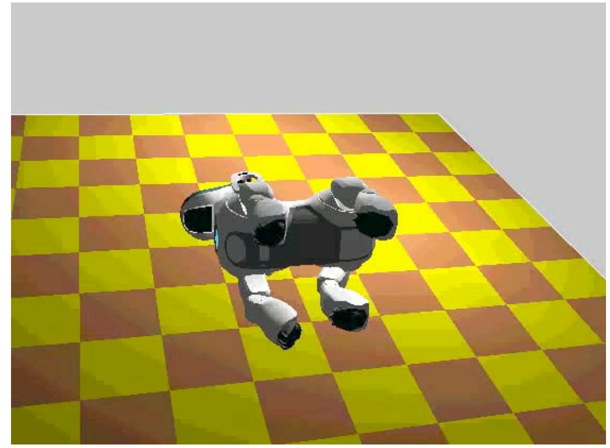
© R.Granger 2009

Integrated architecture: Perception and motor learning



© R.Granger 2009

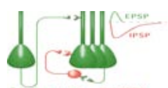
Integrated architecture: Perception and motor learning



© R.Granger 2009

OUTLINE

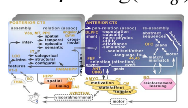
Circuits



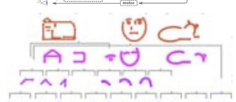
Algorithms

$$W = \frac{\alpha}{\eta L}\,\frac{\log(1 - P_{RE}^{1/\eta L})}{\log(1 - \frac{1}{\sigma})}$$

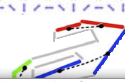
Architecture



Learning



Abilities

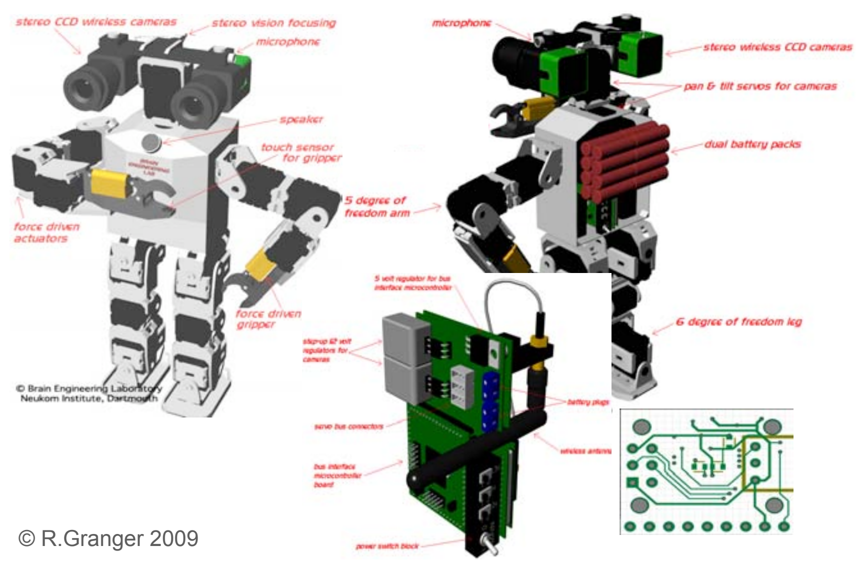


Robots

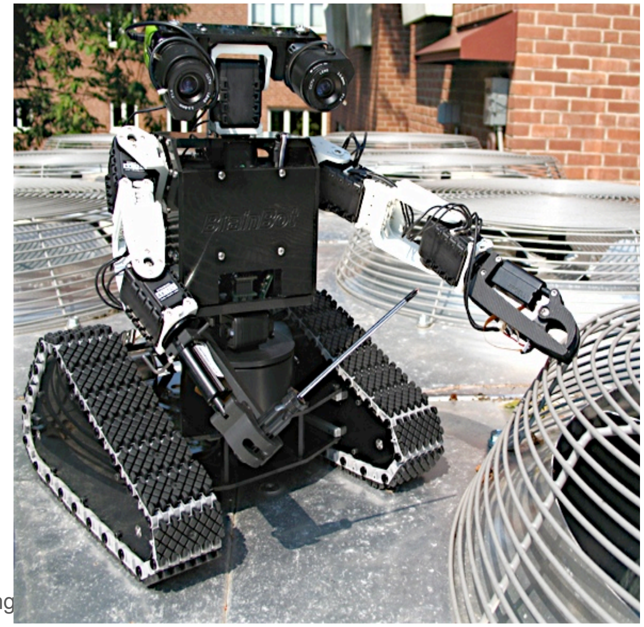
Silicon

© R.Granger 2009

Design of reconfigurable, trainable sensor-rich robot platforms

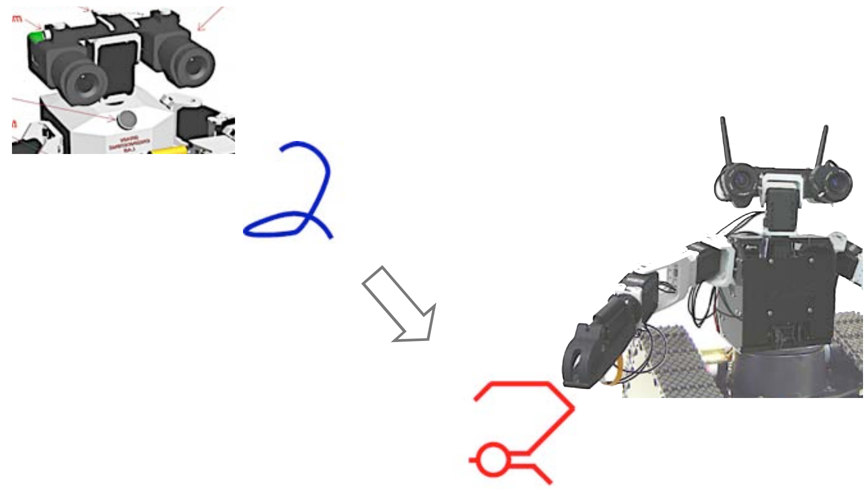


© R.Granger 2009



© R.Grang

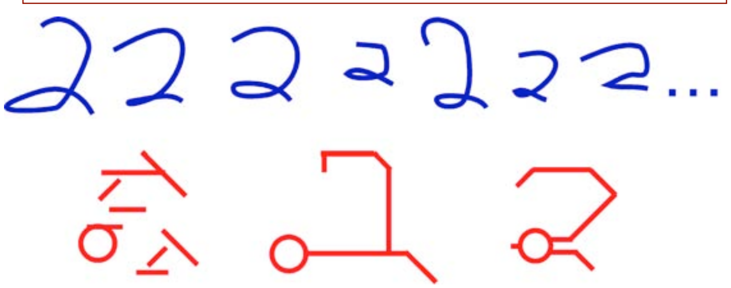
Interleaved perception and action: Learning by doing



© R.Granger 2009

Interleaved perception and action: Learning by doing

⇒ Production becomes part of the training input.



Train on multiple instances

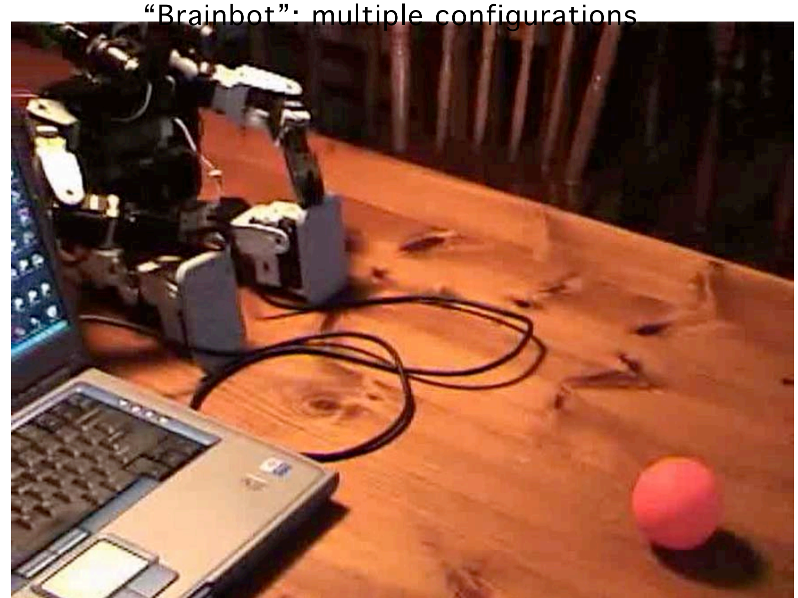
Produce recognition-representation (4 angles, 8 lengths, 1 circle)
Use recognition-representation for initial production

Recognize what was produced (first negative exemplar)
Match/mismatch (successive negative exemplars)
Striatal goal/punishment reinforcement learning

© R.Granger 2009

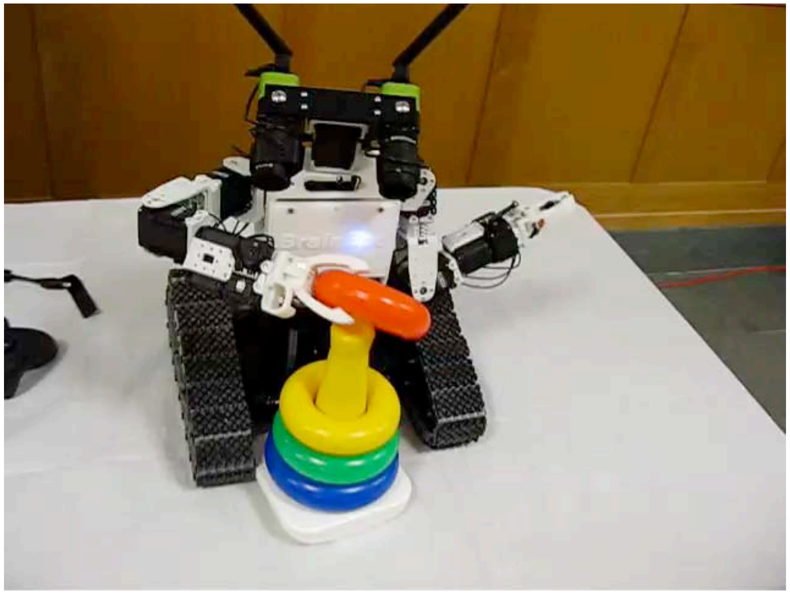
Hearn & Granger (2008)

“Brainbot”: multiple configurations



© R.Granger 2009

(Felch et al.)

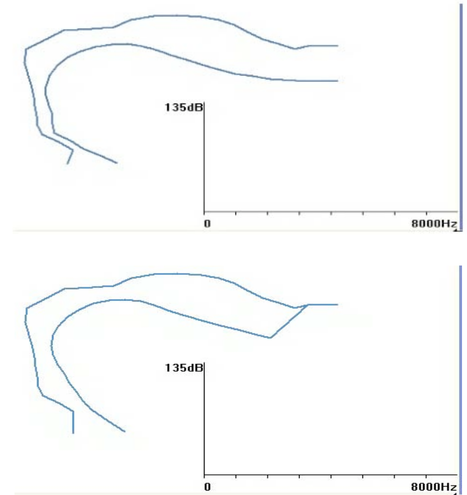


© R.Granger 2009

(Felch et al.)

Perceptual-motor learning by imitation

Learns sounds; attempts to repeat



© R.Granger 2009

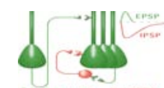
Engineering principle

Learning can require
interleaved perception and action

© R.Granger 2009

OUTLINE

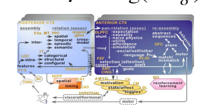
Circuits



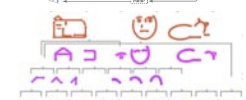
Algorithms

$$W = \frac{\alpha}{\eta L}\,\frac{\log(1-P_{RE}^{1/\eta L})}{\log(1-\frac{1}{\sigma})}$$

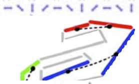
Architecture



Learning



Abilities

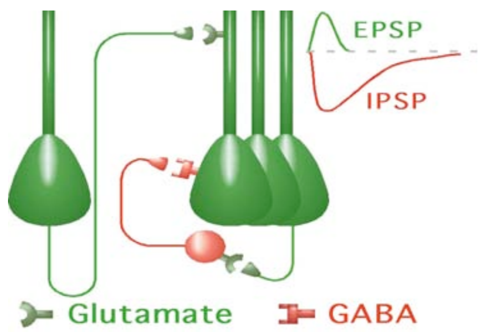


Robots

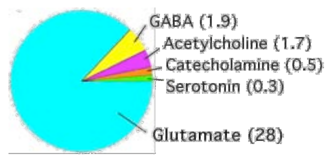
Silicon

© R.Grange

Circuit components



- probabilistic
- slow
- sparse
- low-precision



© R.Granger 2009

Characteristics of implementation

Key implementation characteristics

Simple learning rule

Low-precision connects

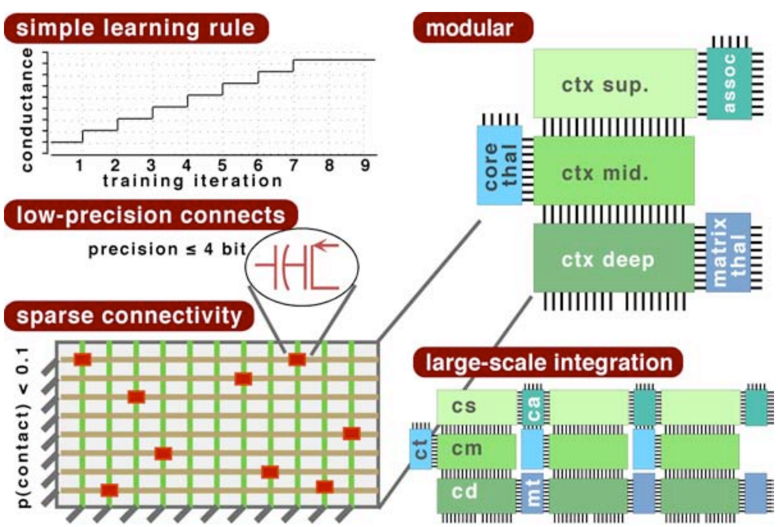
Sparse connectivity

Modular components

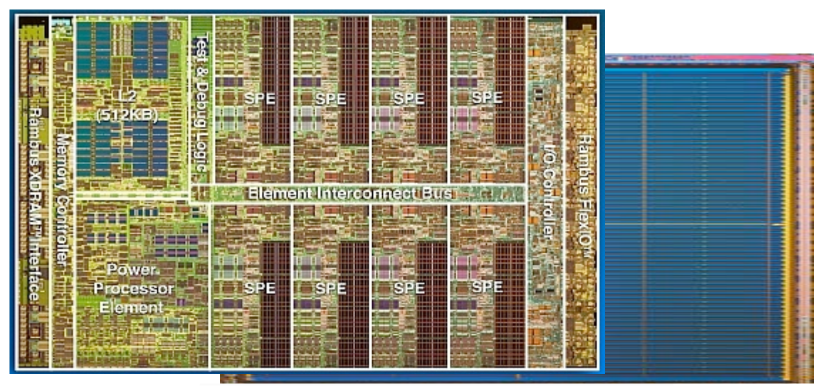
Large-scale integration

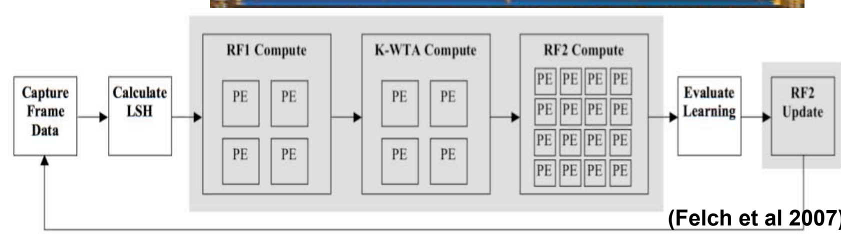
© R.Granger 2009

Characteristics of implementation



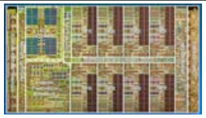
© R.Granger 2009





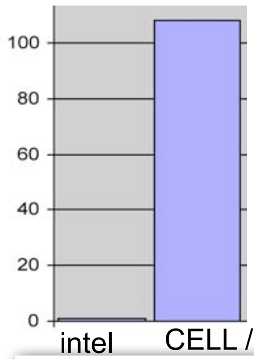
(Felch et al 2007)

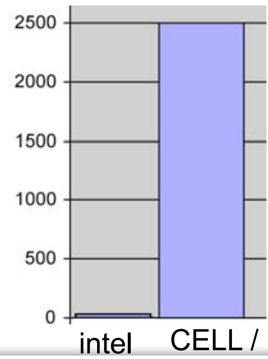
Performance improvement (over Intel Core 2)



speedup: 100 x

improvement per watt: 2500 x





IBM Fall '07 international CELL competition

1st place / 80,000 entries

(Felch et al 2007)

© R.Granger 2009

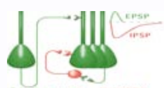
Engineering principle

Intrinsically parallel algorithms lend themselves to scalable direct hardware implementation

© R.Granger 2009

OUTLINE

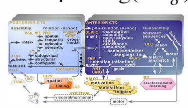
Circuits



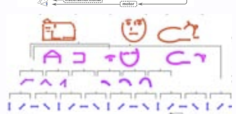
Algorithms

$$W = \frac{\alpha}{\eta L}\,\frac{\log\left(1 - P_{RE}^{1/\eta L}\right)}{\log\left(1 - \frac{1}{\sigma}\right)}$$

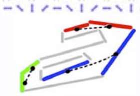
Architecture



Learning



Abilities



Implementations

Robots

© R.Granger 2009

Nonstandard engineering principles of brain circuits

R.Granger et al.
Brain Engineering Laboratory
Dartmouth

N cores ≠ N x speedup

"Moore's Gap": software / algorithm speed vs hardware

Amdahl's law: $$\frac{1}{S + \frac{(1-S)}{N}}$$

Key question: **Architecture design** for parallelism

WHY do brain circuits succeed?
WHAT internal constructs do they build?
HOW can we imitate them?

© R.Granger 2009

Engineering principles of brain circuitry

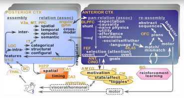
- High-precision computation from low-precision components
- Quantitative changes $\Longrightarrow$ qualitative changes (grammars)
- Adhere to constraints from neuroimaging
- Intractable tasks may succumb to nonstandard engineering
- Shared mechanisms apply across multiple domains
- Learning may require interleaved perception and action
- Perception and language are both grammar-based
- High-order behaviors composed from brain “instruction set”
- Direct parallel hardware implementation

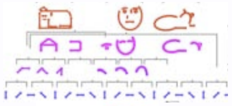
www.BrainEngineering.org

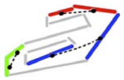
© R.Grang

www.BrainEngineering.org
(further information; publications)

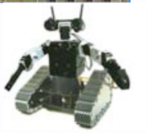
OUTLINE

Architecture 

Learning 

Abilities 

Implementations 

Robots 

© R.Granger 2009