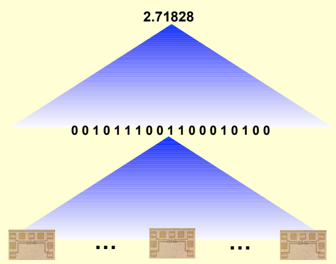
Embodiment and Non-Turing Computation

Bruce MacLennan
Dept. of Electrical Eng. & Computer Science
University of Tennessee, Knoxville

Post-Moore's Law Computation

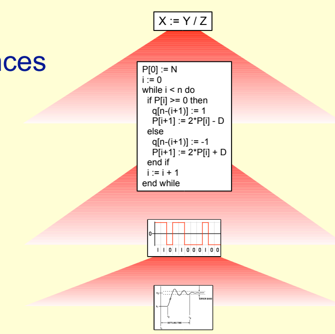
- The end of Moore's Law is in sight!
- Physical limits to:
  - density of binary logic devices
  - speed of operation
- Requires a new approach to computation
- Significant challenges
- Will broaden & deepen concept of computation in natural & artificial systems

Differences in Spatial Scale



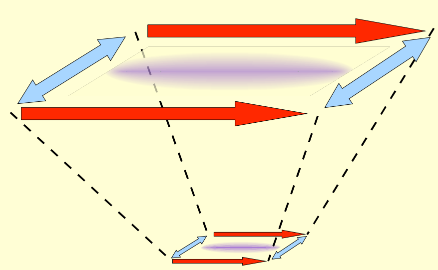
(Images from Wikipedia)

Differences in Time Scale



(Images from Wikipedia)

Convergence of Scales



Implications of Convergence

- Computation on scale of physical processes
- Fewer levels between computation & realization
- Less time for implementation of operations
- Computation will be more like underlying physical processes
- Post-Moore's Law computing ⇒ greater assimilation of computation to physics

Computation is Physical

"Computation is physical; it is necessarily embodied in a device whose behaviour is guided by the laws of physics and cannot be completely captured by a closed mathematical model. This fact of embodiment is becoming ever more apparent as we push the bounds of those physical laws."

— Susan Stepney (2004)

Cartesian Duality in CS

- Programs as idealized mathematical objects
- Software treated independently of hardware
- Focus on *formal* rather than *material*
- Post-Moore's Law computing:
  - less idealized
  - more dependent on physical realization
- More difficult
- But also presents opportunities…

Embodied Cognition

- Rooted in pragmatism of James & Dewey
- Dewey's *Principle of Continuity*:
  - no break from most abstract cognitive activities
  - down thru sensory/motor engagement with physical world
  - to foundation in biological & physical processes
- Cognition: emergent pattern of purposeful interactions between organism & environment

Embodiment, AI & Robotics

- Dreyfus & al.:
  - importance & benefits of embodiment in cognition
  - there are many things we know merely by virtue of having a body
  - embodiment essential to cognition,
  - not incidental to cognition (& info. processing)
- Brooks & al.: increasing understanding of value & exploitation of embodiment in AI & robotics

Embodied Computation

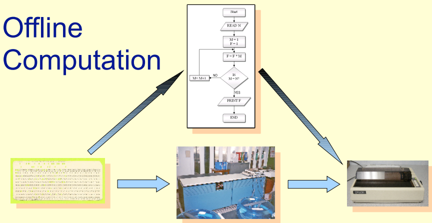
Embodiment & Computation

- Embodiment = "the interplay of information and physical processes"
  — Pfeifer, Lungarella & Iida (2007)
- *Embodied computation* =
  information processing in which
  physical realization & physical environment
  play unavoidable & essential role

Three Modes of Computation

- Offline computation
- Embedded computation
- Embodied computation

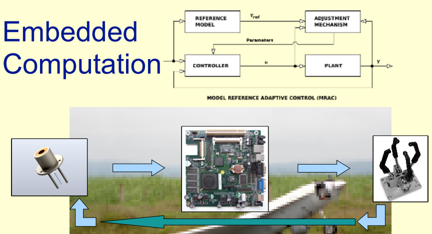
Offline Computation



- Physical input conv. to computational medium
- Abstract computation
- Physical representation of results
- Computation as evaluation of function

(Images from Wikipedia)

Embedded Computation



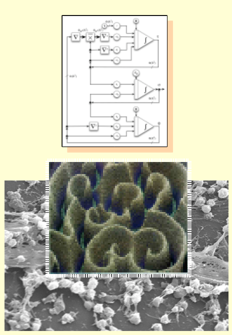
- In ongoing interaction with environment
- Non-terminating
- Real-time feedback through environment

(Images from Wikipedia)

Embedded Computation

- Sensors & actuators still convert to/from computational medium
- Computation is effectively abstract
- Physical considerations confined to:
  - embedding device
  - environment
  - transducers
  - basic physical characteristics of processor

Embodied Computation



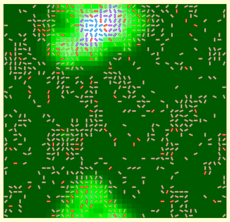
- Embodied (vs embedded) computation:
  - little or no abstract computation
  - computation as physical process in continuing interaction with other physical processes

(Images from Wikipedia)

Strengths of Embodied Computation

- Information often implicit in:
  - its physical realization
  - its physical environment
- Many computations performed “for free” by physical substrate
- Representation & info. processing emerge as regularities in dynamics of physical system

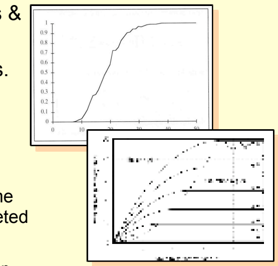
Example: Diffusion



- Occurs naturally in many fluids
- Can be used for many computational tasks
  - broadcasting info.
  - massively parallel search
- Expensive with conventional computation
- Free in many physical systems

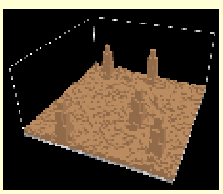
Example: Saturation

- Sigmoids in ANNs & universal approx.
- Many physical sys. have sigmoidal behavior
  - Growth process saturates
  - Resources become saturated or depleted
- EC uses free sigmoidal behavior



(Images from Bar-Yam & Wikipedia)

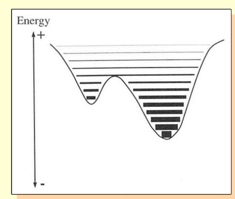
Example: Negative Feedback



- Pos. feedback for growth & extension
- Neg. feedback for:
  - stabilization
  - delimitation
  - separation
  - creation of structure
- Free from
  - evaporation
  - dispersion
  - degradation

Example: Randomness

- Many algs. use randomness
  - escape from local optima
  - symmetry breaking
  - deadlock avoidance
  - exploration
- For free from:
  - noise
  - uncertainty
  - imprecision



(Image from Anderson)

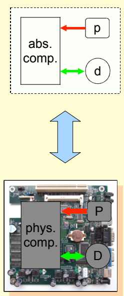
"Respect the Medium"

- Conventional computer technology "tortures the medium" to implement computation
- Embodied computation "respects the medium"
- Goal of embodied computation:

  Exploit the physics, don't circumvent it

Computation for Physical Purposes

EC for Action



- EC uses physics for information processing
- Inf. system governs matter & energy in physical computer
- EC uses info. proc. to govern physical proc.

EC for Action

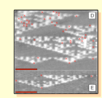
- Natural EC:
  - governs physical processes in organism's body
  - physical interactions with other organisms & environment
- Often, result of EC is not *information*, but *action*, including:
  - self-action
  - self-transformation
  - self-construction

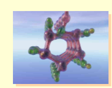
*Dis*embodied Computation

- *If* purpose is information processing
- *Then* represent information with small quantities of matter or energy
- Objective: state change involves small change of matter or energy
- Limit: *disembodied* computation & communication
- Pure *form* without need for *matter*

EC Controlling Matter & Energy

- May want to move *more* rather than *less*
- Physical effects may be direct results of computation
- No clear distinction between processors & actuators
- Examples:
  - Algorithmic assembly by DNA computation (Winfree)
  - Nanostructure synth. & control by molecular combinator reduction (MacLennan)

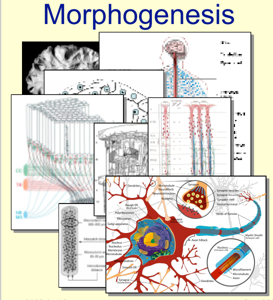

(figure from Rothemund)



Active Materials

- EC may be applied to active materials
- E.g., artificial tissue that can
  - recognize environmental conditions
  - open or close channels controlling transport
  - react mechanically (e.g., contraction)
  - self-organize

Artificial Morphogenesis



- Morphogenesis ⇒ EC can coordinate:
  - proliferation
  - movement
  - disassembly
- to produce complex, hierarchical systems
- Future nanotech.: use EC for multiphase self-org. of complex, functional, active hierarchical systems

(Images from Wikipedia)

Natural Computation

- Challenge of EC: little experience with it
- Nature provides many examples of effective EC
- Nature shows how computation can
  - exploit physics
  - without opposing it
- Shows how information processing systems can interact fruitfully with physical embodiment of selves & other systems

Design of Emergent Computation Systems

(1) Understand

(2) Abstract

(3) Realize

(1) Understand

- Understand how information processing occurs & interacts with physical reality in natural systems
- Look to studies of specific systems relevant to intended application
- Also look to more general information about embodied computation in nature

(2) Abstract

- Abstract process from physical specifics
  - may amount to a mathematical model
  - but is not disembodied
- Physical processes not ignored, but included in essential form
- E.g., diffusion:
  replace *specific* quantity by *generic* quantity
- Some processes will be more generally useful than others

(3) Realize

- Realize abstract computation in appropriate medium by selecting:
  - substances
  - forms of energy
  - quantities
  - processes
  - etc.
- More difficult than traditional computing
- But necessary in post-Moore's Law era

General Design Principles

- Natural EC suggests computational primitives that are:
  - generally useful
  - realizable in a variety of media
- EC for morphogenesis:
  - *discrete primitives*: individual elements
  - *continuous primitives*: spatial masses of them
  - *coordinated algorithms*: temporal organization

"Discrete Primitives"

- Physical processes involving single elements, responding passively or actively
- Examples:
  - mobility (translation, rotation)
  - adhesion & release
  - shape change
  - differentiation or state change
  - collision & interaction
  - proliferation & apoptosis

"Continuous Primitives"

- Physical processes pertaining to spatially distributed masses of elementary units
- Examples:
  - elasticity
  - diffusion
  - degradation
  - fluid flow
  - gradient ascent

Coordinated Algorithms

- Biological morphogenesis ⇒ EC organizes complex, multistage processes operating in parallel at microscopic and macroscopic levels
- Coordinated algorithms in wasp nest construction (Bonabeau, Dorigo & Theraulaz)
- Sequential, parallel, or overlapping
- What are the principles of coordinated algorithm design for EC?

But is it Computing?

Is EC a Species of Computing?

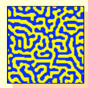
- The Turing Machine provides a precise definition of computation
- Embodied computation may seem imprecise
- & difficult to discriminate from other physical processes
- Expanding concept of computation beyond TM requires an expanded definition

What is Computation?

- What distinguishes computing (physically realized information processing) from other physical processes?
- Computation is a *mechanistic process,* the purpose or function of which is the *abstract manipulation* (processing) of *abstract objects*
- Purpose is *formal* rather than *material*
- Does not exclude embodied computation, which relies more on physical processes

Material Effects Inherent to EC

- Goal of EC may be specific material effects
- But can be understood abstractly
- Example: activator-inhibitor system
  - produces characteristic "Turing patterns"
  - can be characterized mathematically
- May be degrees of computational/non-comp.
- May be degrees of essential embodiment vs. independence of specific phys. realization



Nature Combines Functions

- Artificial systems often have clear purposes
- Nature often combines multiple functions into one system
- Example: ant foraging brings food to nest
- But also does computational tasks:
  - adaptive path finding
  - path minimization
  - exploration

Shanley Principle

- Well-engineered artificial systems obey *Shanley Principle*: multiple functions should be combined into single parts
- Orthogonal design is important in prototyping
- But should be followed by integration of function (Knuth)
- Pushing limits of tech. & deeper embedding
  - will have to combine functions
  - inf. processing systems will be less purely computational & more essentially embodied

Related Work: Hamann and Wörn (2007)

- An EC system has at least two levels
  - adaptive SO & collective behavior at higher levels results from
  - spatially local interactions of "microscopic control devices"
- aspects of embodiment:
  - lack of separation between processor and memory
  - essential dependence of computation on physical world
- Seems to conflate embodiment with other issues

Related Work: Susan Stepney (in press)

- Material computation and in materio computers
- Systems in which physical substrate "naturally" computes
- Focus on non-living substrates
- Primarily concerned with use of physical materials to implement computations
- Less concerned with use of computational processes to organize & control matter and energy
- Cautions against ill-advised application of notions from Turing computation

Non-Turing Computation

Frames of Relevance

- CT computation is a *model* of computation
- All models have an associated *frame of relevance*
  - determined by model's simplifying assumptions
  - by aspects & degrees to which model is similar to modeled system
- Determine questions model is suited to answer
- Using outside FoR may reflect model & simplifying assumptions more than modeled system

FoR of CT Model

- CT computation developed to address issues in effective calculability & formalist mathematics
- In FoR makes sense to consider something computable if it can be computed in
  - finite number of steps
  - of finite but indeterminate duration
  - using finite (but unbounded) amount of memory
- Makes sense to treat computation as function evaluation
- And define computability in terms of sets of functions

Unsuitability to EC

- CT model is not well-suited to address relevant issues in EC (or in natural computation)
- Its simplifications & approximations are bad ones for EC
- E.g., CT model ignores real-time rates of operations, but they are highly relevant in EC
- Also, CT notions of equivalence & universality do not address the efficiency with which one system simulates another

Frame of Relevance for EC

- Premature to define model of embodied computation
- We do not yet understand which issues are relevant or not
- Premature formalization can impede progress
- Some relevant issues:
  - robustness
  - generality
  - flexibility
  - adaptability
  - morphology & steric constraints
  - physical size
  - consumption of matter & energy
  - reversible reactions
  - real-time response

Conclusions

- Embodied computation will be important in post-Moore's Law computing
- But need new models of computation that:
  - are orthogonal to CT model
  - but address relevant issues of EC
- There will be a fruitful interaction between investigations of embodiment in computation and philosophy

More Information?

- A written version of this presentation, "Aspects of Embodied Computation," can be found at: www.cs.utk.edu/~mclennan/papers/AEC-TR.pdf
- Or by looking under "Recent reprints etc." at my website: www.cs.utk.edu/~mclennan [sic]