Processes in Space

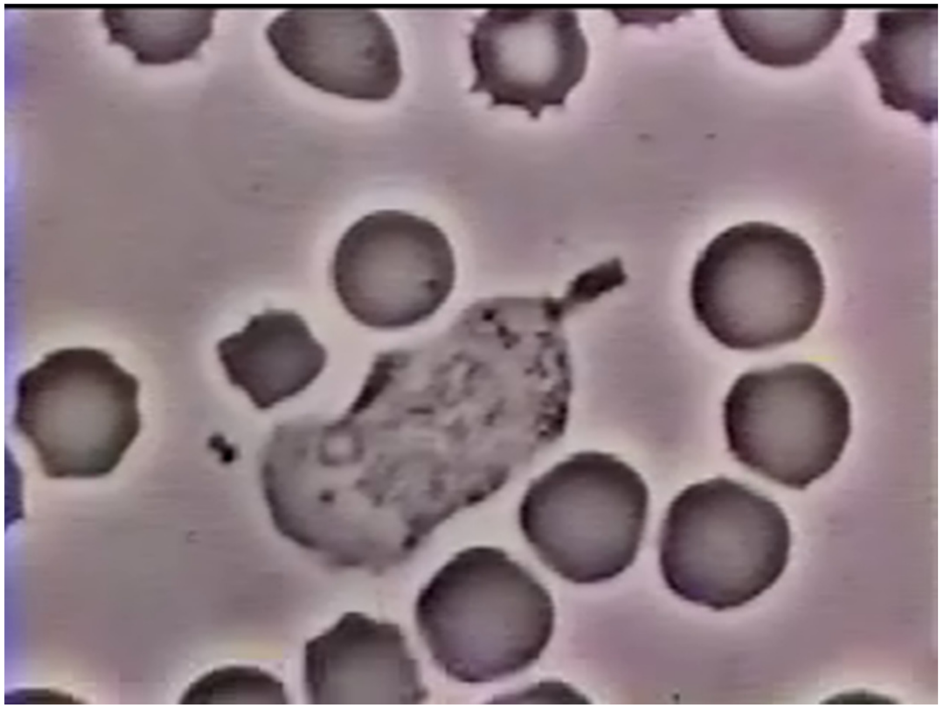
Luca Cardelli
Microsoft Research

with Philippa Gardner
Imperial College London

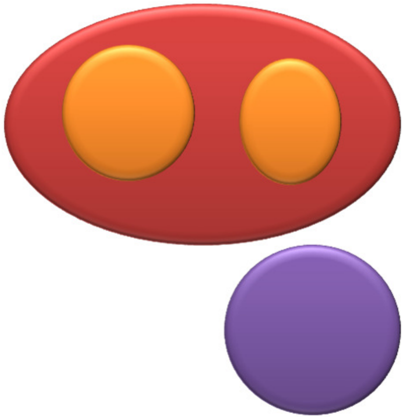
2010-07-02 CiE Ponta Delgada

http://LucaCardelli.name

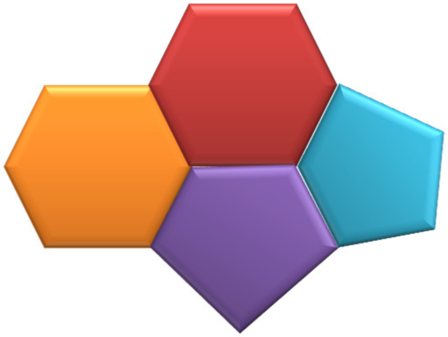
Introduction



From Topology to Geometry

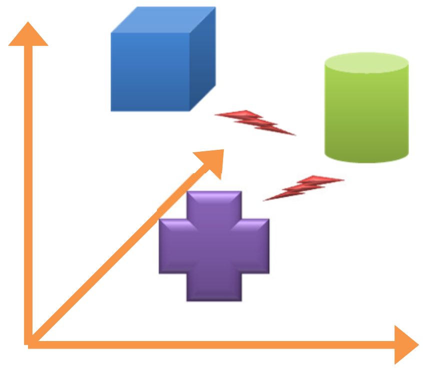


- Process Algebra and Membrane Computing
  - Successful in describing the hierarchical (*topological*) organization and transformation of complex systems.
- In many situations, however, *geometry* is necessary
  - Geometric constraints exist both at the subcellular level and at higher levels of cellular organization.
  - Developmental biology deals with dynamic spatial arrangements, and with forces and interactions.



- While many discrete geometric approaches exists
  - Cellular Automata, and graph models.
  - Geometric extensions of L-systems.
- Few cover both key aspects of development
  - Rich geometry.
  - Rich communication.

3π



- We have developed a geometric process algebra
  - Processes located in 3-dimensional space.
  - With a kinetic component (change of position over time).
  - With rich communication capabilities (as usual in π).
- Easy you say: just ‘add a position to each process’
  - Naive attempts result in awkward formal systems with too many features: coordinates, position, velocity, identity, force, collision, communication…
- Developmental biology is geometrically peculiar
  - The coordinate space is not fixed: it effectively expands, moves, and warps as the organism develops.
  - Approaches based fixed grid or coordinate systems are awkward.
- Algorithmic Botany has shown the way:
  - Affine geometry: the geometry of properties invariant under linear transformations and translations.

Shifting Reference Frames

- 3π is a π-calculus extended with a single new process construction able to shift the frame of reference. (N.B.: 'shifting' means 'composing'.)

- A *frame shift*, consists of dynamically applying a 3-dimensional affine transformation to a whole evolving process.
  - This is sufficient to express many dynamic geometric behaviors, thanks to the combined power of Affine Geometry and Process Algebra.
  - The π-calculus remains relatively simple, technically formulated in a familiar way, with a large but standard and modular geometric subsystem.

- From a Process Algebra point of view:
  - 3π adds powerful geometric data structures and transformations.
  - Position and communication collude to model *forces*.

- From an Affine Geometry point of view:
  - Key geometric invariants become theorems of 3π: *relativity*.
  - These invariants ensure that processes can be freely transformed, and that they cannot 'cheat' geometrically.

It's Still About Nesting

- While containment is the key notion in topological models:

  n[P] a process P in a container with interface n

  - Algebraic rules for manipulating nested n[-] brackets.

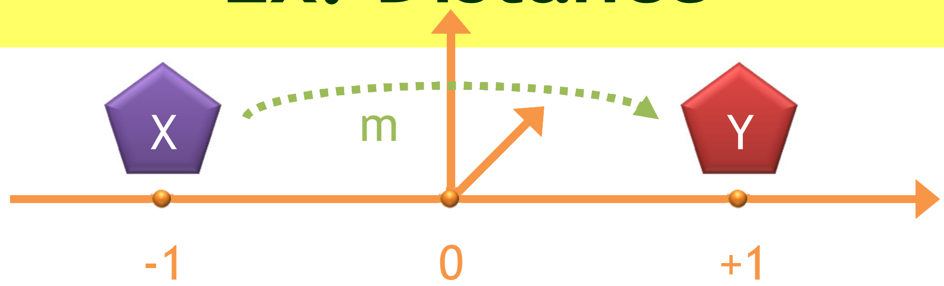
- Here frame shift is the key notion in geometric models:

  M[P] a process P transformed by an affine map M

  - Algebraic rules for manipulating nested M[-] brackets.

Examples

Ex: Distance

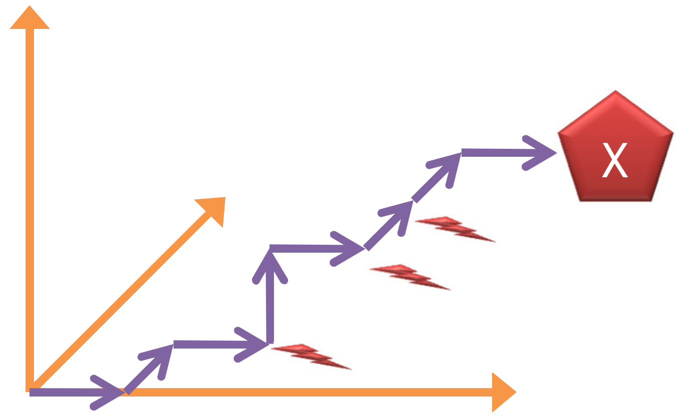


$$X = T(-↥_x)[P] \quad \text{where } P = !m(⊹)$$
$$Y = T(↥_x)[Q] \quad \text{where } Q = ?m(x).\ \|x ∸ ⊹\|=2.\ R$$

- **Two processes on the X axis:**
  - Process P is relocated at -1 on the x axis, by a translation $T(-↥_x)$ with unit x vector $↥_x$. Process Q is relocated at +1 by a translation $T(↥_x)$.
  - When P outputs *its origin* $⊹$ on m, the actual value sent is the point $\langle -1,0,0 \rangle$.
  - Process Q receives that value as x, and computes the size of the *vector* $x ∸ ⊹$. In the frame of Q that is the size of the vector $\langle -1,0,0 \rangle ∸ \langle 1,0,0 \rangle$, which is 2.
  - Therefore, the comparison $\|x ∸ ⊹\|=2$ succeeds, and process R is activated, having verified that the distance between P and Q is 2.

- Note: the *affine basis* $⊹, ↥_x, ↥_y, ↥_z$ is available to each process, but is interpreted relatively to the *current frame* of the process.

Ex: Motion

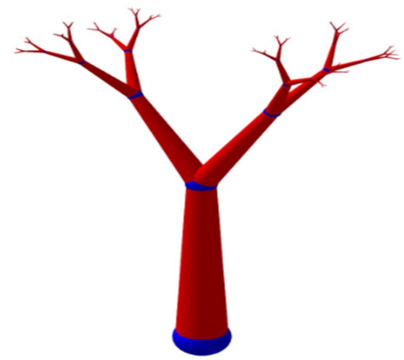


$$X = \tau.T(\uparrow_x)[X] + \tau.T(\uparrow_y)[X] + \tau.T(\uparrow_z)[X] + !c(\text{⊹}).X$$

- The recursive process X nonderministically applies unit translations to itself along the axes.
- It also nondeterminstically outputs its current origin on channel c, therefore producing a telemetry of its movements.

Ex: Spatial Arrangements

- Lung development in mouse is based on three splitting processes [9].
  - We show how to represent the third (orthogonal bifurcation, Orth).
  - Bifurcations alternate between orthogonal planes.



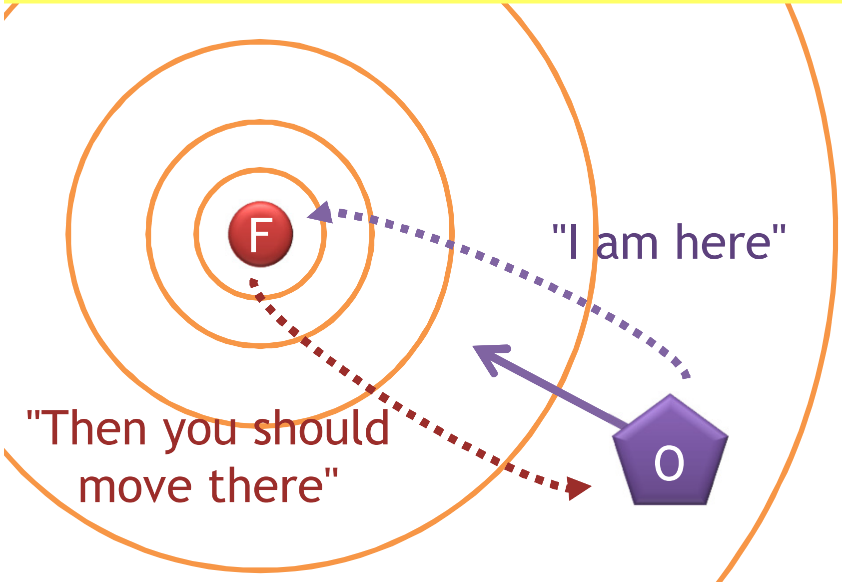
$$\mathrm{Orth} = !c(\text{⊹}).\ (\mathrm{M90}(\pi/6)[\mathrm{Orth}] \mid \mathrm{M90}(-\pi/6)[\mathrm{Orth}])$$
$$\mathrm{M90}(\theta) = \mathrm{R}(\mathrm{M}(\theta)[\uparrow_y], \pi/2) \circ \mathrm{M}(\theta)$$
$$\mathrm{M}(\theta) = \mathrm{Sc}(½) \circ \mathrm{R}(\uparrow_z, \theta) \circ \mathrm{T}(\uparrow_y)$$

  - M(θ): applies a translation $T(\uparrow_y)$ by $\uparrow_y$, a rotation $R(\uparrow_z,\theta)$ by θ around $\uparrow_z$, and a uniform scaling Sc(½) by ½.

  - M90(θ): first applies an M(θ) transformation in the XY plane, and then applies a further 90° rotation around the ‘current’ direction of growth, which is $M(\theta)[\uparrow_y]$, therefore rotating out of the XY plane for the next iteration.

  - Orth: Outputs the current origin ⊹ to the c channel at each iteration, providing a trace of the growing process that can be plotted. Opposite 30° rotations applied recursively to Orth itself generate the branching structure.

Ex: Forces



Force = (?f(x,p). !x(M{p}))*
Object = (νx) !f(x,⊹). ?x(Y). Y[Object]

f is the force field channel; M{p} is a map

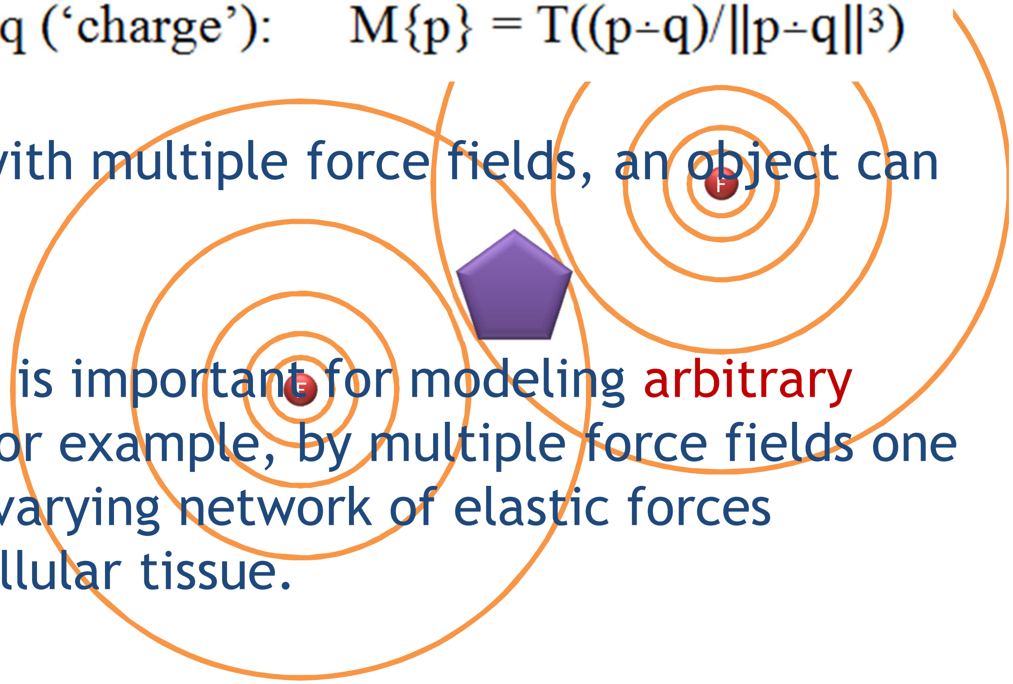
- A force field is a process F that receives the location of an ‘object’ process P (and, if appropriate, a representation of its mass or charge), and tells it how to move by a discrete step.

- The latter is done by replying to the object with a transformation that the object applies to itself.

Ex: Forces

- This force field transformation can depend on the distance between the object and the force field, and can easily represent inverse square and linear (spring) attractions and repulsions.

A uniform field ('wind'):	$M\{p\} = T(\hat{1}_x)$
A linear attractive field at q ('spring'):	$M\{p\} = T(\frac{1}{2}\cdot(q \dot{-} p))$
An inverse-square repulsive field at q ('charge'):	$M\{p\} = T((p \dot{-} q)/\|p \dot{-} q\|^3)$

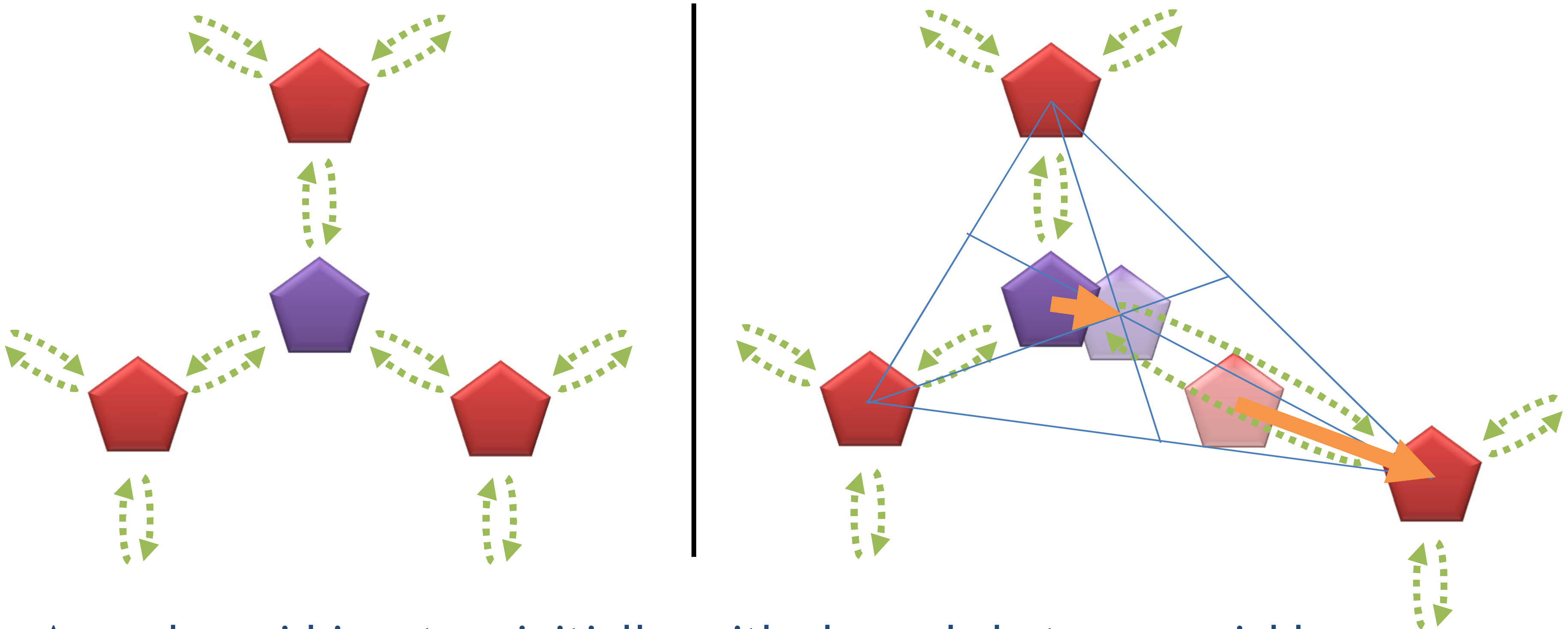


- By nondeterministic interaction with multiple force fields, an object can be influenced by several of them.

- The ability to express force fields is important for modeling arbitrary constraints in physical systems. For example, by multiple force fields one can set up an arbitrary and time-varying network of elastic forces between neighboring cells in a cellular tissue.

- Some forces have unlimited range: one should *not* restrict communication based on distance.

Ex: Tissues



- A regular grid is set up initially, with channels between neighbors.
- Each process repeatedly polls the position of its neighbors, and moves to their geometric mean position.
- If one process moves the others adjust, eventually recreating a regular grid.

Key Design Decisions

- Q1: How should the position of a process be represented?
  - By the affine basis relative to a global frame.
- Q2: How should a process move from one position to another?
  - By frame shift.
- Q3: How should processes at different positions interact?
  - By *standard* π-calculus communication supported by frame-shift structural congruence rules. Communication not limited by distance.
- Q4: What theorems can we prove that blend properties of geometry and of process algebra, to show we have reached a smooth integration?
  - Relativity (invariances of barbed congruence under affine transformations).

Processes

Syntax of Processes

- A pretty standard π-calculus syntax with data terms Δ (details later).

- Data can have one of five sorts σ = {scalar, point, vector, map, channel}

Δ ::=	$x_c \;\vdots\; \ldots \;\vdots\; M[\Delta]$	Data terms
π ::=	$?_\sigma x(x') \;\vdots\; !_\sigma x(\Delta) \;\vdots\; \Delta =_\sigma \Delta'$	Action terms
P ::=	$0 \;\vdots\; \pi.P \;\vdots\; P+P' \;\vdots\; P\vert P' \;\vdots\; (\nu x)P \;\vdots\; P^* \;\vdots\; M[P]$	Process terms

- The new addition is process frame shift M[P], where M is (a data term denoting) an affine map. And a similar data frame shift M[Δ].

- Process reduction is relative to a global frame; M[P] means running process P in a global frame that has been shifted by (composed with) M.

Reduction

- Process reduction is indexed by a global frame $\mathcal{A}$, which is an affine map:
- Uses a data evaluation relation from data terms Δ to data values ε, similarly indexed by a global frame:

$$P \;{}_{\mathcal{A}}\!\rightarrow Q$$

$$\Delta \;{}_{\mathcal{A}}\!\rightarrowtail \varepsilon$$

(Red Comm)	$\Delta \;{}_{\mathcal{A}}\!\rightarrowtail \varepsilon \;\Rightarrow\; !_{\sigma}x(\Delta).P + P' \mid ?_{\sigma}x(y).Q + Q' \;{}_{\mathcal{A}}\!\rightarrow\; P \mid Q\{y\backslash\varepsilon\}$	
(Red Cmp)	$\Delta \;{}_{\mathcal{A}}\!\curvearrowright\!\curvearrowleft \Delta' \;\Rightarrow\; \Delta{=}_{\sigma}\Delta'.P \;{}_{\mathcal{A}}\!\rightarrow P$	
(Red Par)	$P \;{}_{\mathcal{A}}\!\rightarrow Q \;\Rightarrow\; P \mid R \;{}_{\mathcal{A}}\!\rightarrow Q \mid R$	
(Red Res)	$P \;{}_{\mathcal{A}}\!\rightarrow Q \;\Rightarrow\; (\nu x)P \;{}_{\mathcal{A}}\!\rightarrow (\nu x)Q$	
(Red $\equiv$)	$P' \equiv P,\; P \;{}_{\mathcal{A}}\!\rightarrow Q,\; Q \equiv Q' \;\Rightarrow\; P' \;{}_{\mathcal{A}}\!\rightarrow Q'$	

eliminate impure terms

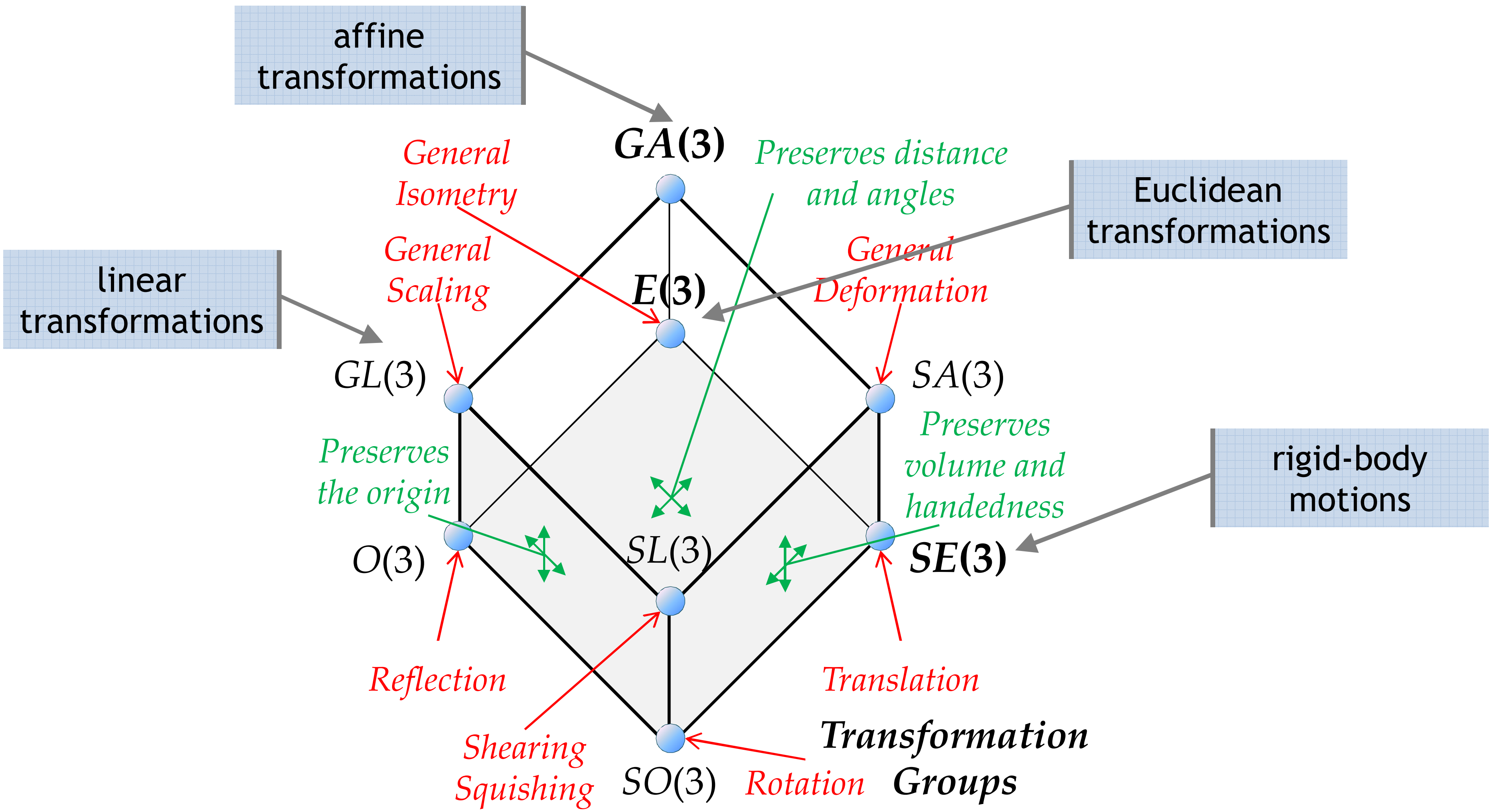
introduce impure terms

- Otherwise, a completely standard π-calculus reduction relation
  - Communication is by-value (using $\Delta \;{}_{\mathcal{A}}\!\rightarrowtail \varepsilon$ for evaluation).
  - *Data comparison* $\Delta{=}_{\sigma}\Delta'$ generalizes 'channel matching' to all data sorts.
  - The global frame $\mathcal{A}$ is just handed down to the evaluation relation.
  - There is no new rule here about frame shift: it is handled via (Red $\equiv$).

Structural Congruence

(≡ Refl)	$P \equiv P$	(≡ Sum Comm)	$P+Q \equiv Q+P$
(≡ Symm)	$P \equiv Q \Rightarrow Q \equiv P$	(≡ Sum Assoc)	$(P+Q)+R \equiv P+(Q+R)$
(≡ Tran)	$P \equiv Q,\ Q \equiv R \Rightarrow P \equiv R$	(≡ Sum Zero)	$P+0 \equiv P$
(≡ Act)	$P \equiv P' \Rightarrow \pi.P \equiv \pi.P'$	(≡ Par Comm)	$P \mid Q \equiv Q \mid P$
(≡ Sum)	$P \equiv P',\ Q \equiv Q' \Rightarrow P+Q \equiv P'+Q'$	(≡ Par Assoc)	$(P \mid Q) \mid R \equiv P \mid (Q \mid R)$
(≡ Par)	$P \equiv P',\ Q \equiv Q' \Rightarrow P \mid Q \equiv P' \mid Q'$	(≡ Par Zero)	$P \mid 0 \equiv P$
(≡ Res)	$P \equiv P' \Rightarrow (\nu x)P \equiv (\nu x)P'$	(≡ Res Zero)	$(\nu x)0 \equiv 0$
(≡ Repl)	$P \equiv P' \Rightarrow P^* \equiv P'^*$	(≡ Res Sum)	$(\nu x)(P+Q) \equiv P+(\nu x)Q \quad (x \notin fv_c(P))$
(≡ Map)	$P \equiv P' \Rightarrow M[P] \equiv M[P']$	(≡ Res Par)	$(\nu x)(P \mid Q) \equiv P \mid (\nu x)Q \quad (x \notin fv_c(P))$
(≡ Map Cmp)	$M[\Delta =_\sigma \Delta'.P] \equiv M[\Delta] =_\sigma M[\Delta'].M[P]$	(≡ Res Res)	$(\nu x)(\nu y)P \equiv (\nu y)(\nu x)P$
(≡ Map Out)	$M[!_\sigma x(\Delta).P] \equiv !_\sigma x(M[\Delta]).M[P]$	(≡ Repl Zero)	$0^* \equiv 0$
(≡ Map In)	$M[?_\sigma x(y).P] \equiv ?_\sigma x(y).M[P] \quad (y \notin fv_\sigma(M))$	(≡ Repl Par)	$(P \mid Q)^* \equiv P^* \mid Q^*$
(≡ Map Sum)	$M[P+Q] \equiv M[P]+M[Q]$	(≡ Repl Copy)	$P^* \equiv P \mid P^*$
(≡ Map Par)	$M[P \mid Q] \equiv M[P] \mid M[Q]$	(≡ Repl Repl)	$P^{**} \equiv P^*$
(≡ Map Res)	$M[(\nu x)P] \equiv (\nu x)M[P]$		
(≡ Map Comp)	$M[N[P]] \equiv (M \circ M[N])[P]$		

Data: Points, Vectors, Affine Maps



Observation

Relativity

- Relating process behavior and geometry: what can we prove?

- How does behavior change when changing frame?
  - Galilean relativity (about classical 4D-spacetime):

The laws of physics are the same in all inertial frames.

(inertial frame = rigid body motion at constant velocity)

The laws of physics are algebraic laws (equations) that remain valid (observationally invariant) under some class of transformations.

  - 3π relativity (about 3D space):

The laws of process algebra are the same in all rigid-body frames.

- What are the "laws of process algebra"?
  - All equations that are valid under observational equivalence!
  - An equation is a pair of *pure terms* P,Q, written $P = Q$.
  - A law in a frame $\mathcal{A}$ is a valid equation $P \ {}_{\mathcal{A}}\!\approx Q$ for all observations ($\approx$).
  - Where ${}_{\mathcal{A}}\!\approx$ is a fairly standard barbed congruence, indexed by a frame.

Global Frame Shift for Observation

- How do observations change when changing frame?

Theorem: Global Frame Shift for Barbed Congruence

$$C \models \mathrm{P},\mathrm{Q}, \quad \mathrm{P} \;{}_{\mathcal{A}}{\approx}\; \mathrm{Q} \;\Rightarrow\; C(\mathrm{P}) \;{}_{C\circ\mathcal{A}}{\approx}\; C(\mathrm{Q})$$

- If two processes P,Q are observationally equivalent in global frame $\mathcal{A}$ and transformation C is compatible with P,Q,
then they are observationally equivalent in global frame $C\circ\mathcal{A}$
- But if P/Q are impure, then C needs to be applied to the values inside.

Relativity Corollaries

- All process equations are invariant under rigid body transformations (rotations and translations, not reflections), implying that no process can observe the location of the origin, nor the orientation of the basis vectors in the global frame.

- Processes that do not perform absolute measurements (via $\bullet$ and $\times$) are invariant under all affine transformations, meaning that they are also unable to observe the size of the basis vectors and the angles between them.

- Processes that use $\bullet$ but not $\times$ are invariant under all the isometries, meaning that they cannot observe whether they have been reflected.

Conclusions

Summary

- Biological systems (particularly during development) use complex communication *and* complex geometry.
- Process Algebra can flexibly represent communication patterns, and topological arrangements, but needs to be extended to handle geometric arrangements, motions, and forces.
- The classical theories of process equivalence can be extended to state interesting facts about geometry.

Related Work

- Affine geometry is widely used in computer graphics; probably the most accessible reference for computer scientists is Gallier's book [5].
- It has been used in conjunction with L-Systems in very successful models of plant development [11]. However, L-systems are contextual term rewriting systems and, unlike 3π, do not have an intrinsic notion of interaction, which is important since biological development is regulated by sophisticated intra-cellular interactions.
- There is a solid body of work in functional computer graphics, but not so much in concurrent computer graphics.
- SpacePi [8] is an extension of π-calculus to model spatial dynamics in biological systems. Similar general aims to our work, but technically rather different. We do not restrict communication to a radius because that can be achieved by comparing data values, because some physical forces have infinite radius, and because geometric constraints on interaction are not necessarily of such a simple form (e.g., interaction restricted to adjacent cells of odd shapes).