A Basis Set of Operators for Space-Time Computations

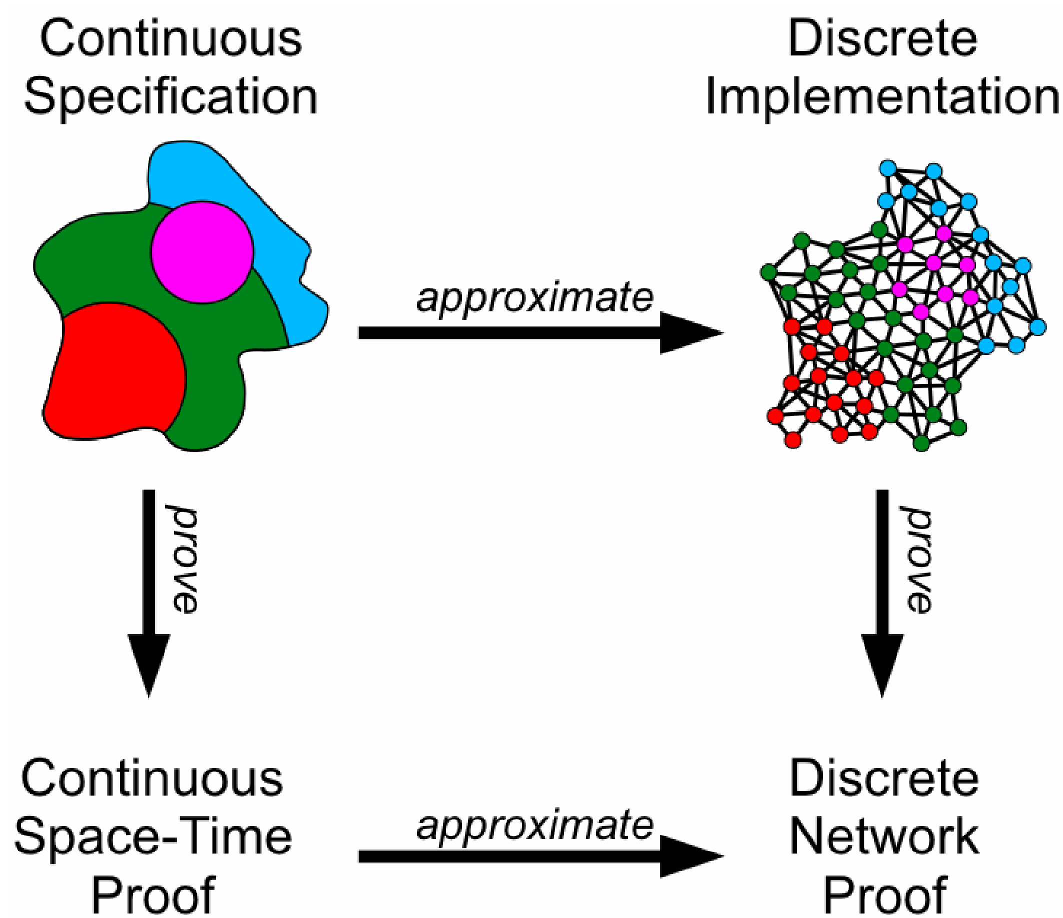
Jacob Beal

3rd Spatial Computing Workshop
SASO 2010

BBN TECHNOLOGIES

Problem: Analysis for Spatial Computing

- Model-to-model comparison: MGS, Proto, TOTA, LDP, etc... equivalent? complete?
- Platform-to-platform comparison: can we prove algorithms in the continuous model instead?

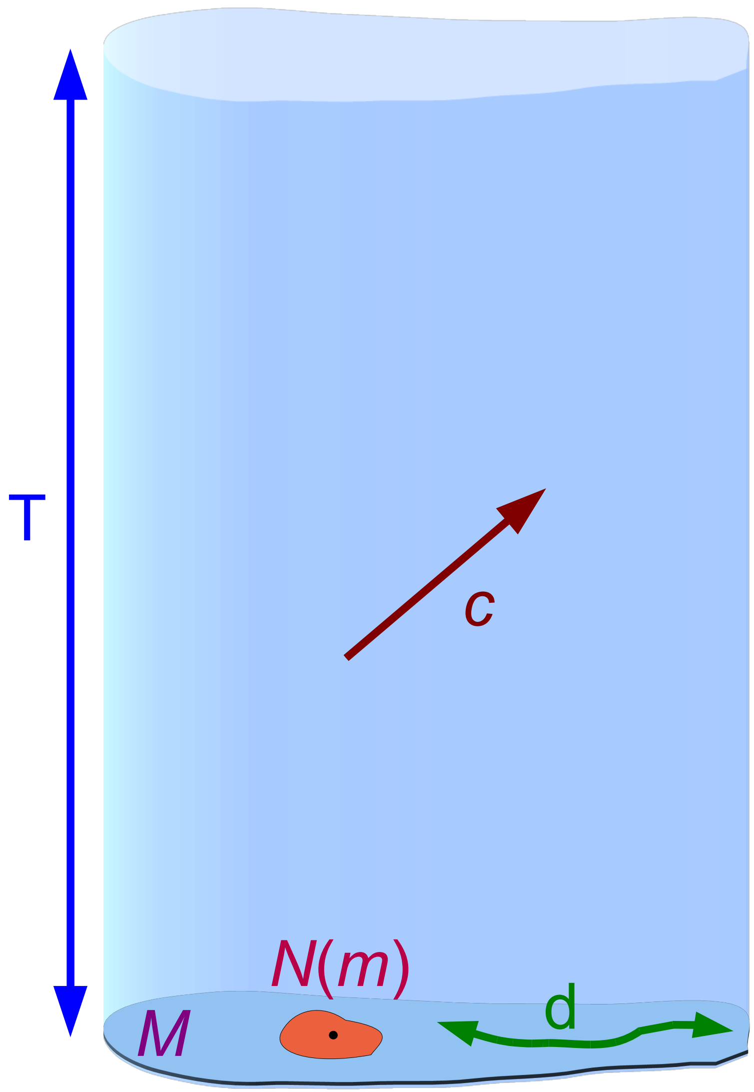


Continuous model = super-Turing?

Talk Outline

- **General definition of space-time computation**
- Basis set of operators
- Is Proto universal?

Amorphous Medium



Var.	Definition	Type
M	Spatial region	compact Reimannian manifold
T	Time interval	$T \subseteq (-\infty, \infty)$
d	Distance fn on M	$d : M \times M \rightarrow \mathbb{R}$
c	Max speed of information	meters per second
N(m)	Neighborhoods on M	$N : M \rightarrow P(M)$

Computation as Function

- Computed state:
  - Instant: $S_t : M \to V$
  - Initial: $S_0 : M \to V$
  - Interval: $S_T : M \times T \to V$
- Sensing:
  - $E : M \times T \to V$
- *Computation:*
  - $C : M \times T \times E \times S_0 \to S_T$

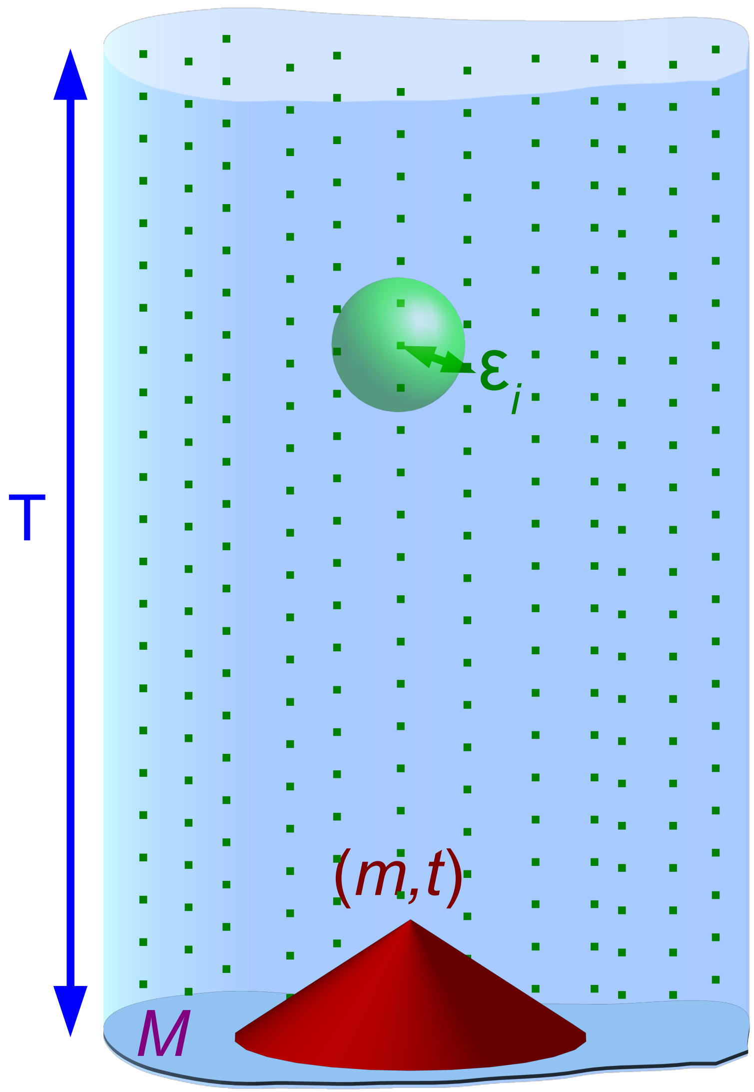
Var.	Definition	Type
M	Spatial region	compact Reimannian manifold
T	Time interval	$T \subseteq (-\infty,\infty)$
d	Distance fn on M	$d : M \times M \to \mathbb{R}$
c	Max speed of information	meters per second
$N(m)$	Neighborhoods on M	$N : M \to P(M)$
V	Function values	$\bigcup_{k\geq 0} \mathbb{R}^k$
S_t	State at time t	$S_t : M \to V$
S_0	Initial state	$S_0 : M \to V$
S_T	State on interval T	$S_T : M \times T \to V$
E	Environmental state	$E : M \times T \to V$
C	Computation	$C: M \times T \times E \times S_0 \to S_T$

Space-Time Universality

- Definition: a computation *C′* **implements** computation *C* if there is a restriction of S_T that is equal to S_t almost everywhere, and if for any non-equal point *p*, there is a sequence of points p_i converging on *p* such that $\lim_{i\to\infty} S'_T(p_i) = \lim_{i\to\infty} S_T(p_i)$.
- A basis set of operators *B* is **space-time universal** if, for any computation *C* that can be specified by some besis set of operators (we need not know what or how), it is possible to implement an equivalent comptuation *C′* using operators in *B*.

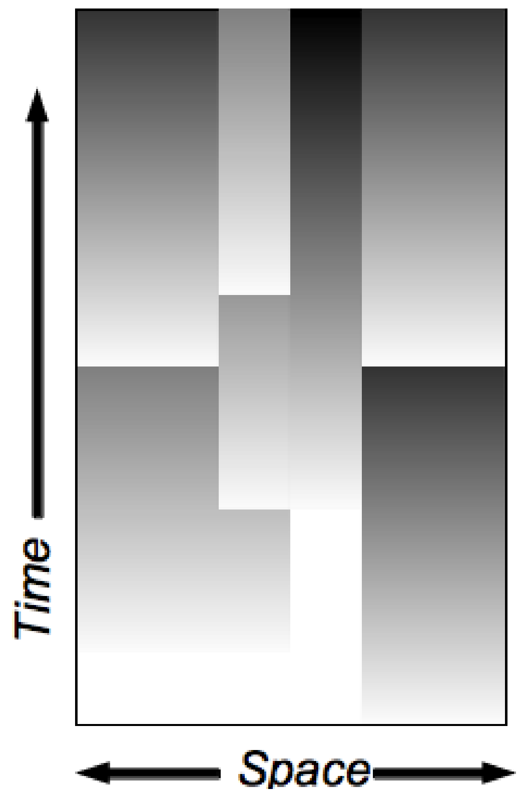
Note: definition of universality not dependent on a model.

Causal & Finitely-Apprxomable

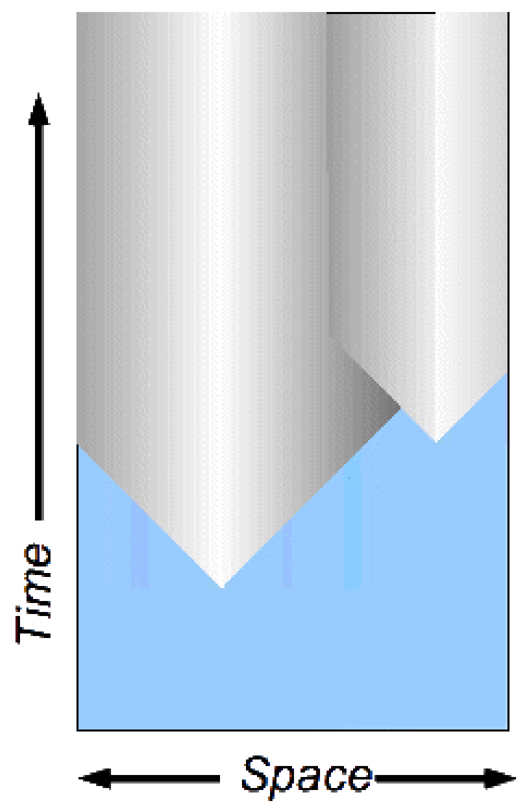


- A computation *C* is **causal** if at every point *(m,t)*, the value depends only on the past light cone.
- A computation is **finitely-approximable** if all countable sequences of ε_i-approximations C_i with decreasing ε_i converge to an implementation of *C*

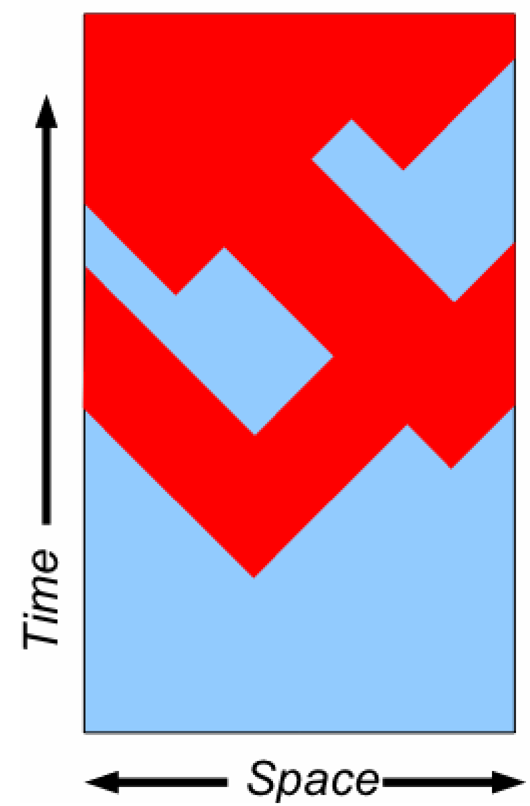
Examples of Finitely-Approximable Causal Computation



Elapsed time since environmental cue



Distance to nearest environmental cue



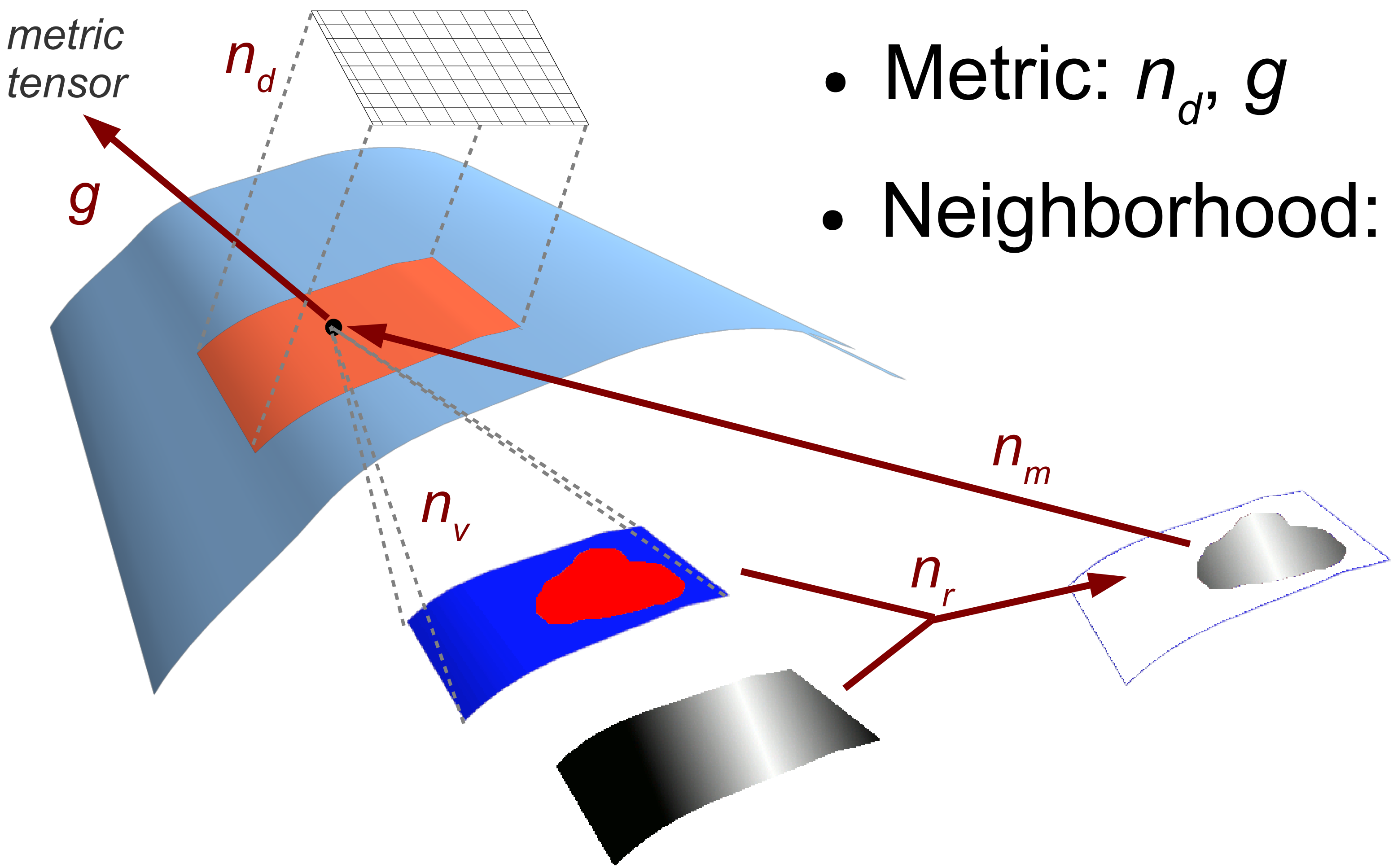
Is an environmental cue currently present?

Talk Outline

- General definition of space-time computation
- **Basis set of operators**
- Is Proto universal?

Basis Set of Operators

- Pointwise Turing-universal: P
  - Metric: n_d, g
  - Neighborhood: n_v, n_r, n_m

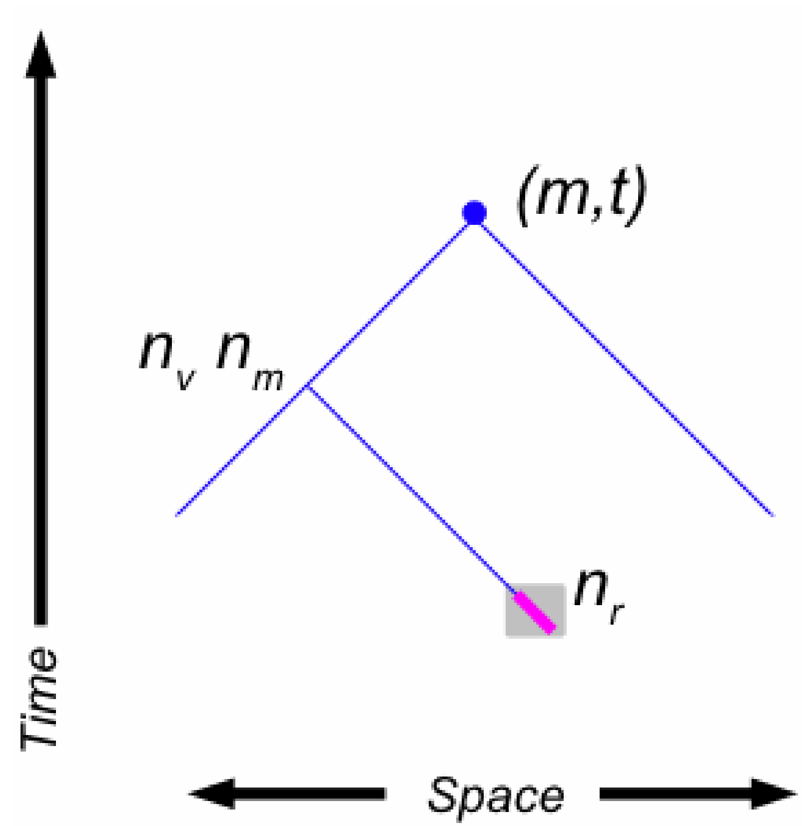


Universality of Basis

Theorem: any finitely-approximable causal computation C can be implemented using the basis set of operators $\{g, n_d, n_v, n_r, n_m\} \cup P$.

Intuition:

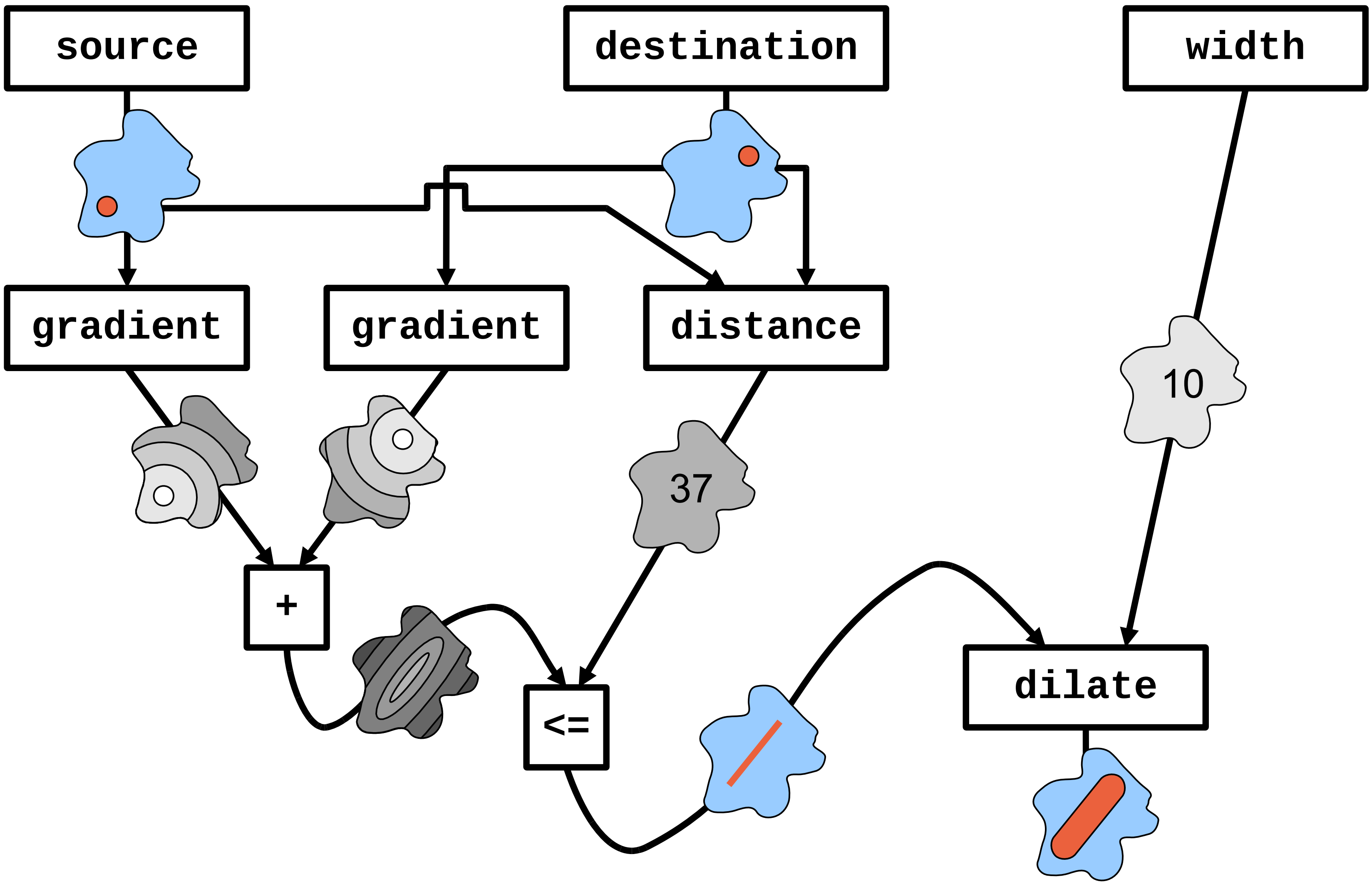
- Use n_* to sample past state, environment, g
- Use P to compute approximate value
- Increasing sampling resolution converges



Talk Outline

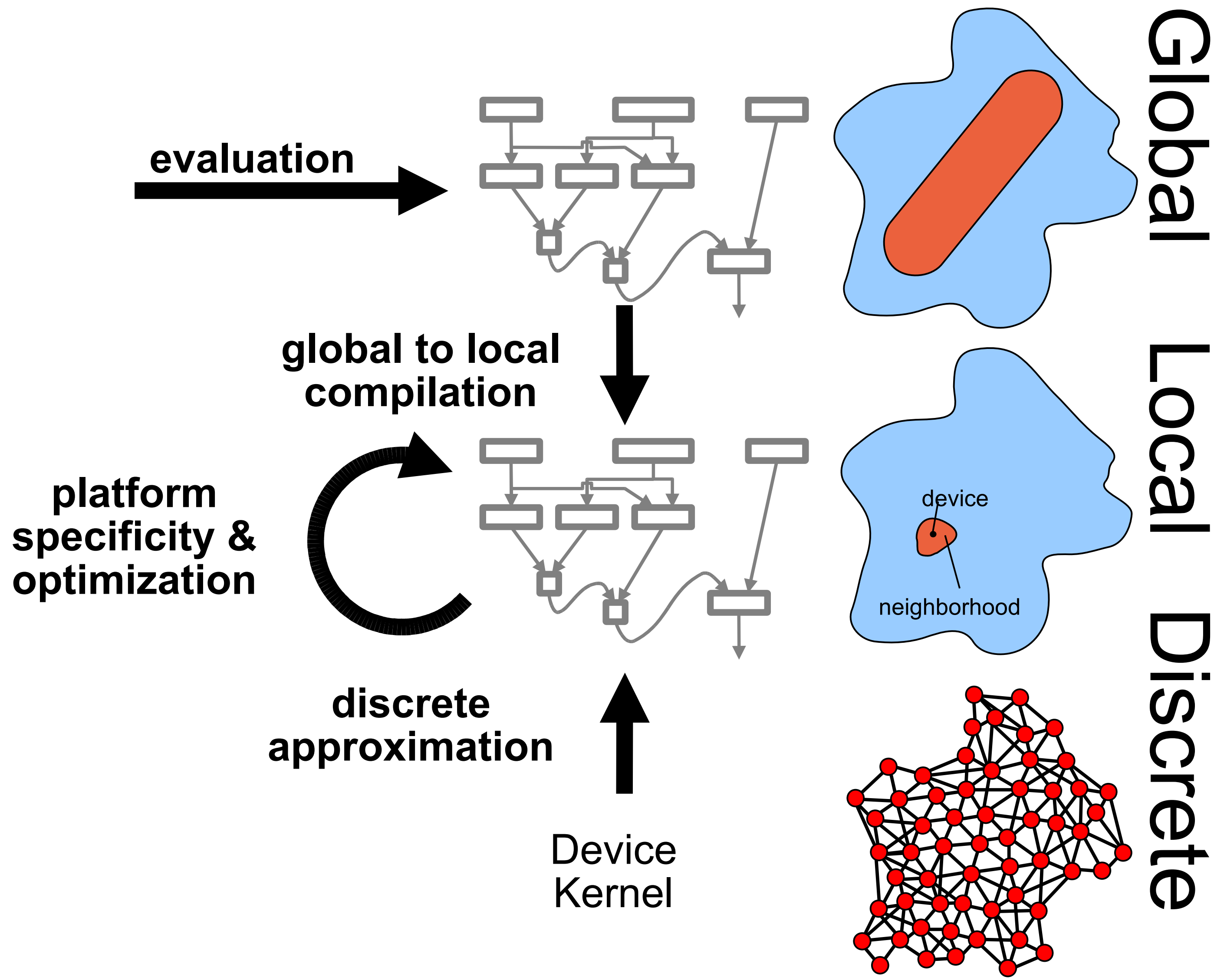
- General definition of space-time computation
- Basis set of operators
- **Is Proto universal?**

Proto: Computing with fields



Proto: Continuous/Discrete Relation

```
(def gradient (src) ...)
(def distance (src dst) ...)
(def dilate (src n)
  (<= (gradient src) n))
(def channel (src dst width)
  (let* ((d (distance src dst))
         (trail (<= (+ (gradient src)
                       (gradient dst))
                    d)))
    (dilate trail width)))
```



Application to Proto

Most operators are directly implemented:

- *P* implemented by Proto's point-wise operators
- n_d = **`nbr-vec`**
- n_v = **`nbr`**
- n_r = **`if`** applied to field types
- n_m = **`min-hood`**

Missing: *g* … but Proto has other metric ops, e.g. **`density`**, **`nbr-lag`** … partial gap cover?

Open Problems

- What are appropriate computational cost models, and what finitely-approximable operators minimize cost?
- How can we do “Nyquist rate” approximation analysis?
- Can we establish function approximability bounds?
- What families of continuous proofs can be automatically translated to discrete proofs?
- Extension of theory to dynamic manifolds?
- How can Proto be extended to cover *g*?
- How powerful are other spatial computing models?

Contributions

- Direct-proof motivation for super-Turing models
- Operator-free definitions for space-time computation
- Basis operators for finitely-approximable causal computations: $\{g, n_d, n_v, n_r, n_m\} \cup P$
- Gap analysis for universality of Proto