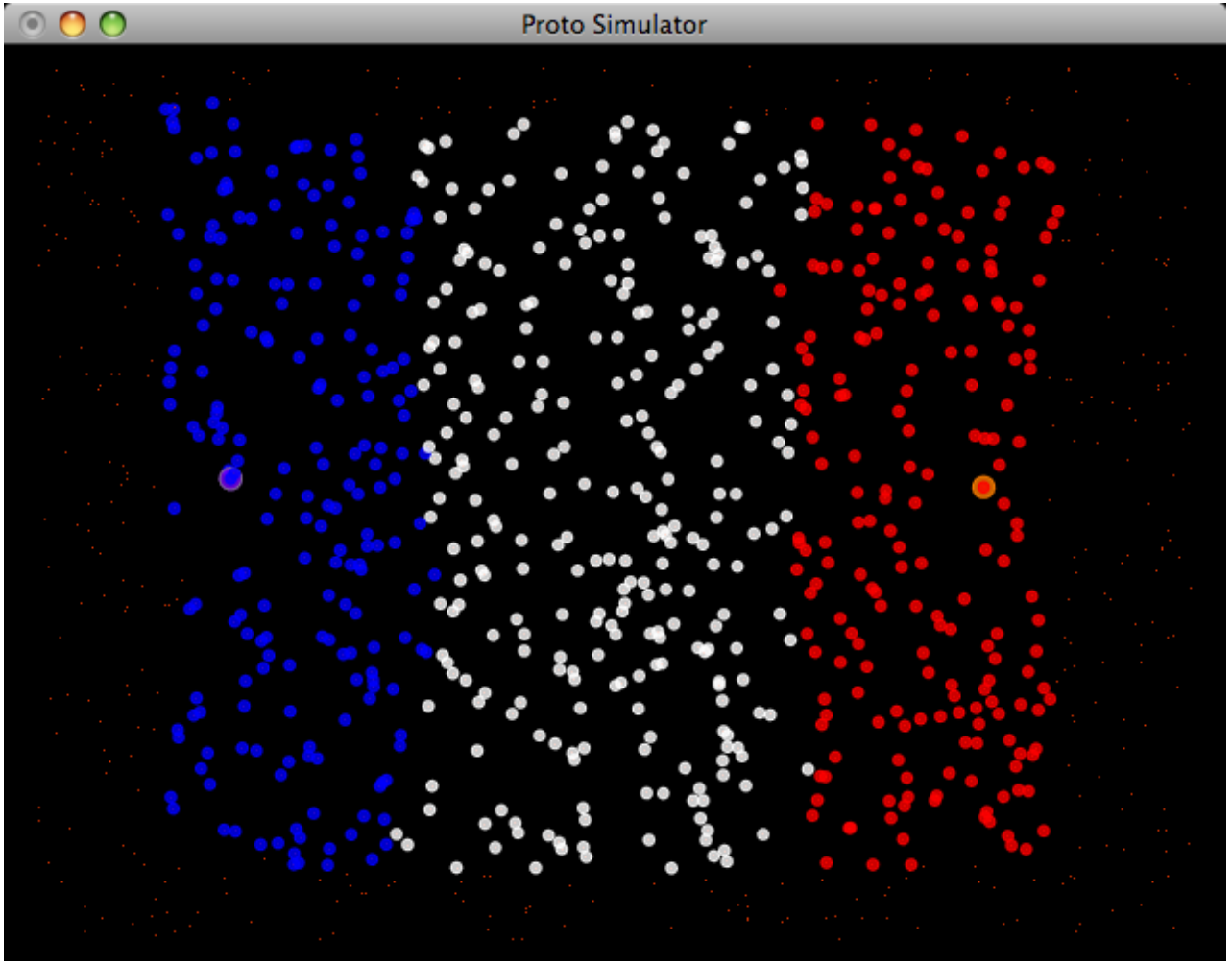
Spatial Computers & Fields

Jacob Beal

Lecture 1 of 5 on Spatial Computing
ISC-PIF Summer School, 2009

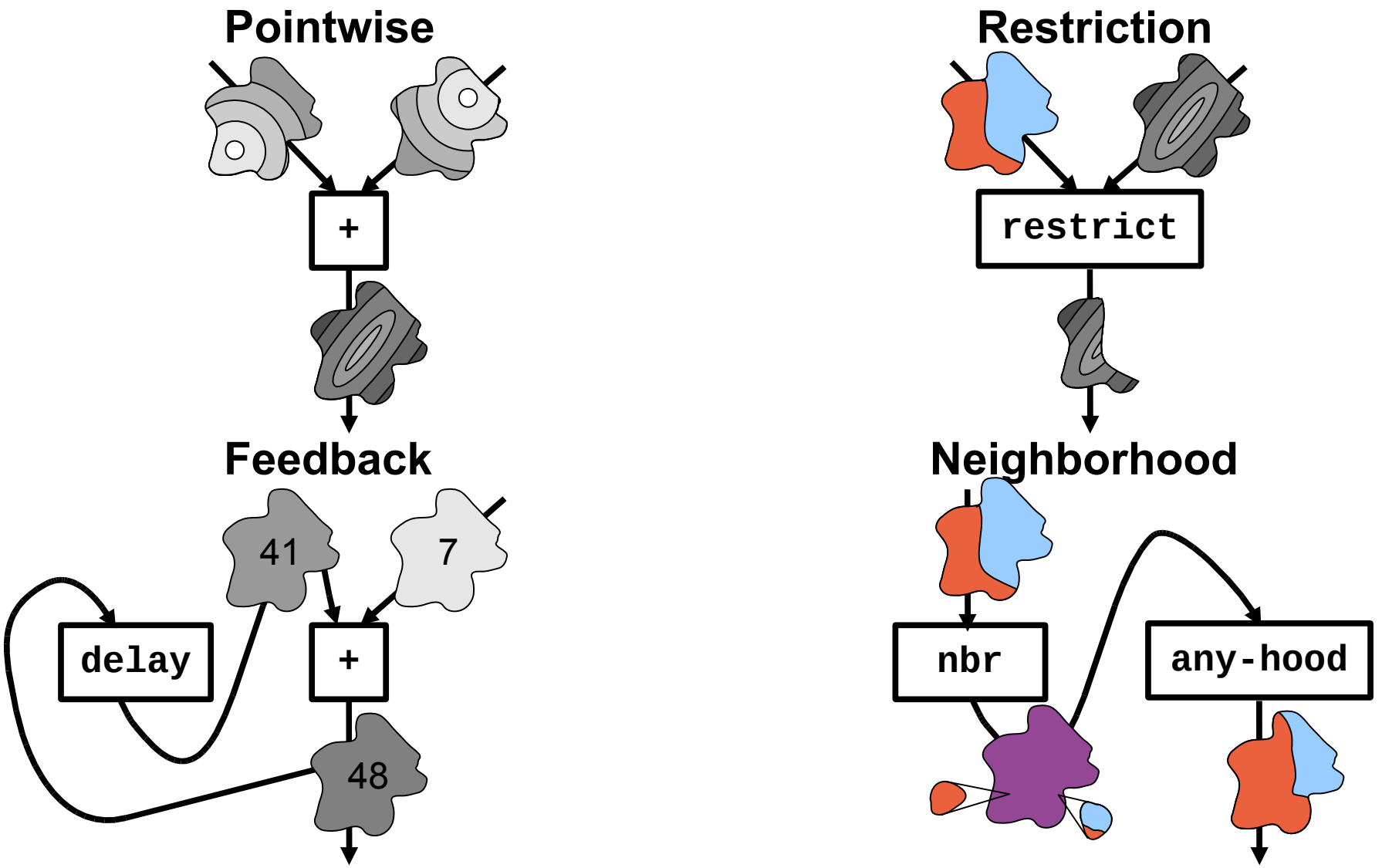
BBN TECHNOLOGIES

Today: geometric programming

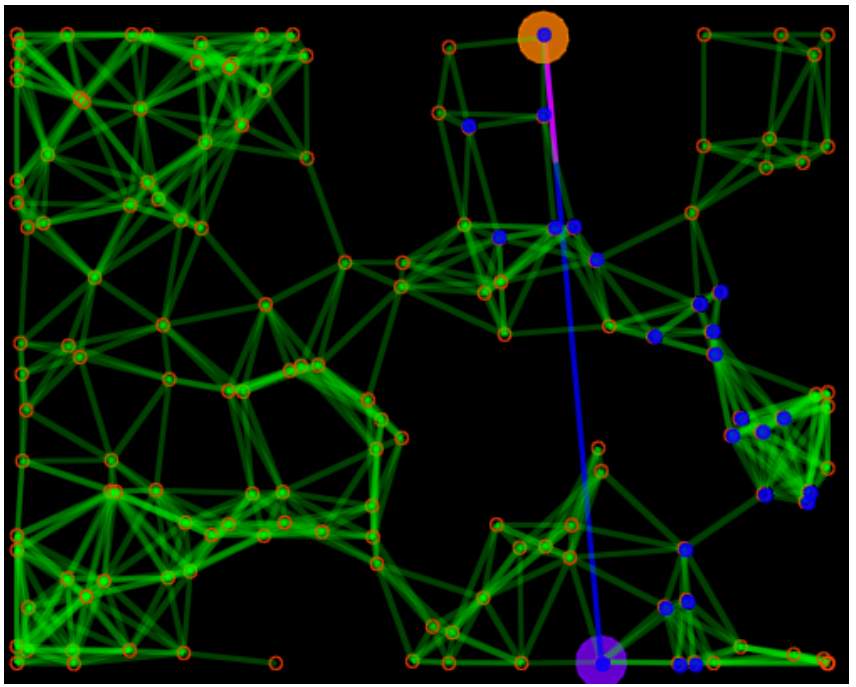


La Belle France!

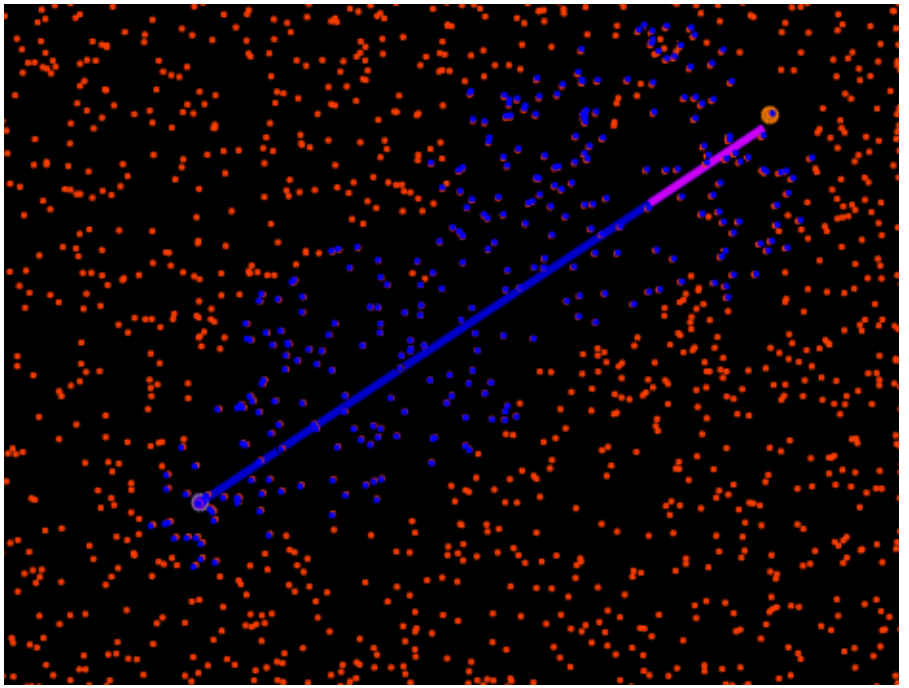
Lecture 2: Continuous Space-Time Programs



Lecture 3: Discrete Approximation & Self-Healing

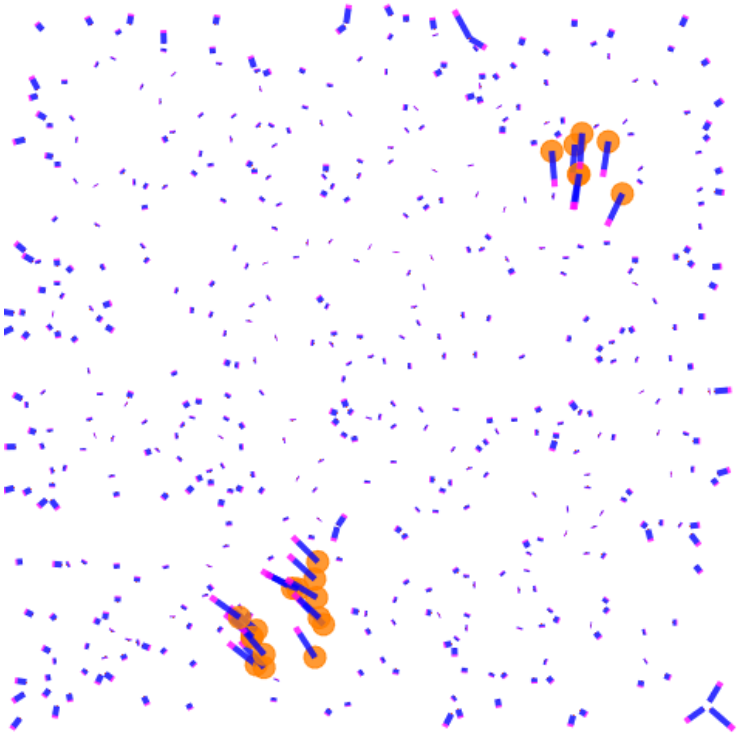


150 devices



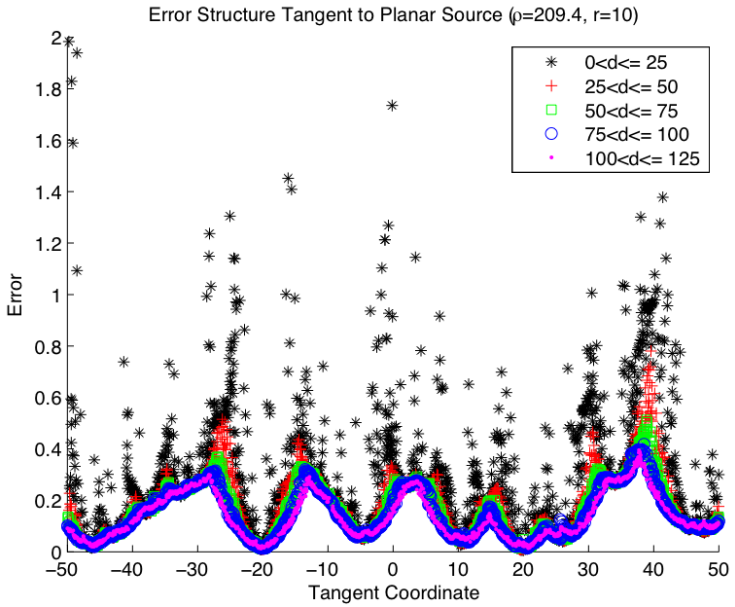
2000 devices

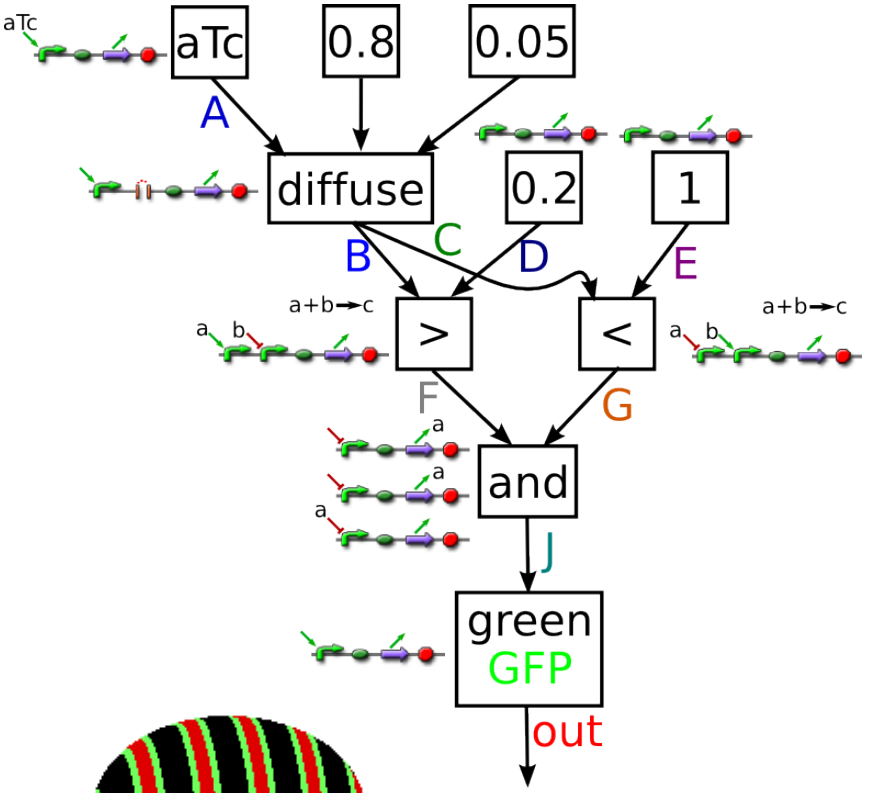
Lecture 4: Moving Devices

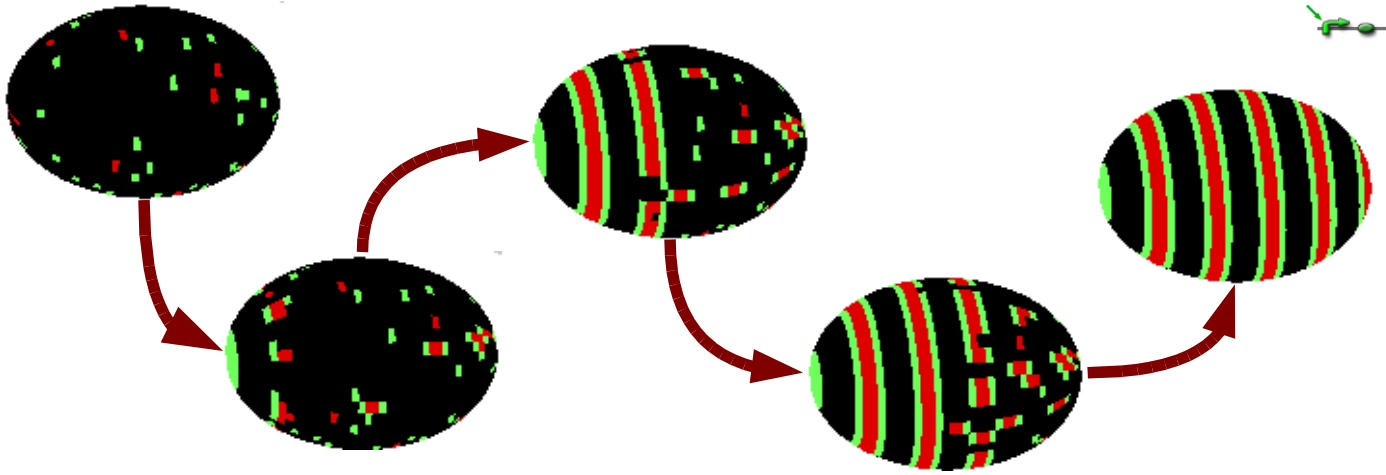


Robot motion = vector fields

Lecture 5: Current Frontiers







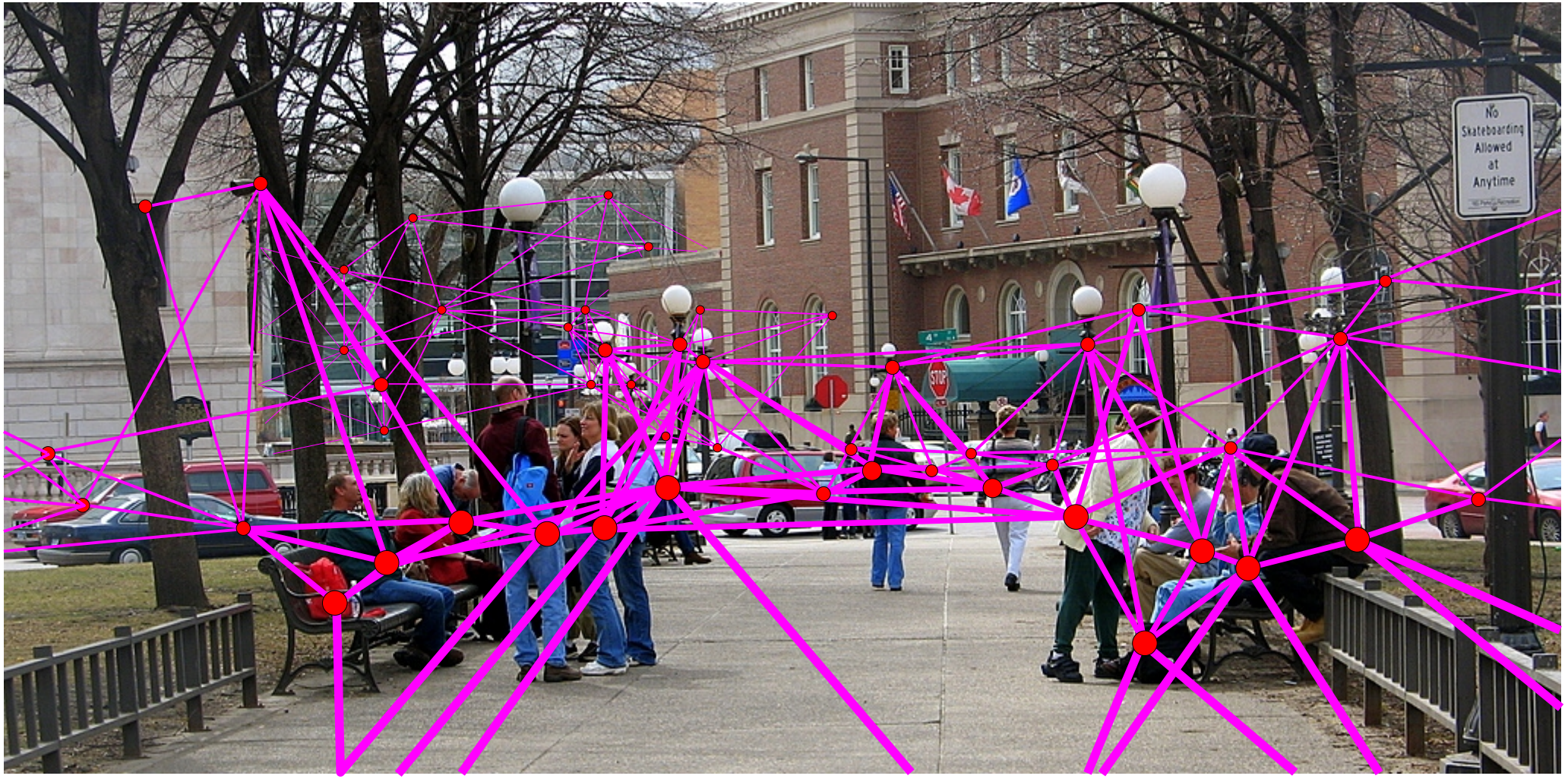
Agenda

- What is spatial computing?
- Geometric Programming

Networked devices are **filling** our environment...

Networked devices are **filling** our environment...

Networked devices are **filling** our environment...



How do we program aggregates robustly?

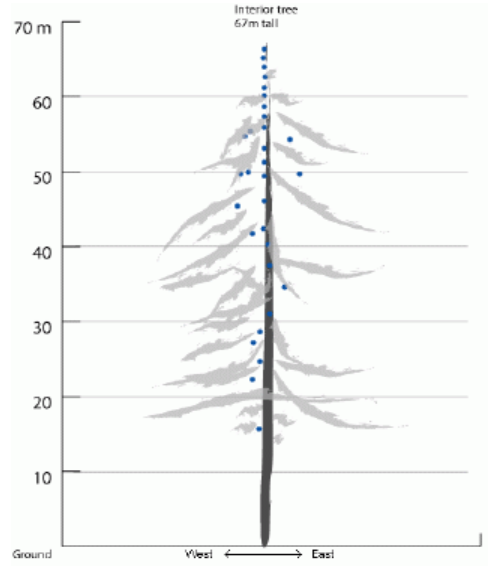
Wireless-enabled Embedded Systems

- >3.3B cell phones vs. 600M Internet-connected PC's in 2007
  - >600M cell phones with Internet capability, rising rapidly
- New cars come equipped with navigation systems and will soon have wireless interfaces (WiFi/DSRC, cellular, WiMax)
- Sensor deployment just starting, but some estimates ~5-10B units by 2015
- Military/emergency response wireless robots, unmanned vehicles, unmanned aircraft

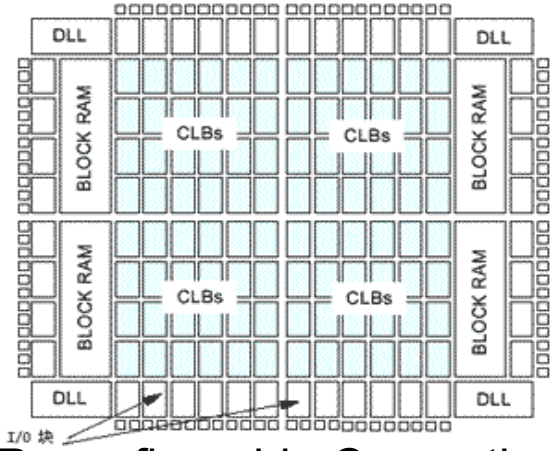
Spatial Computers

Robot Swarms

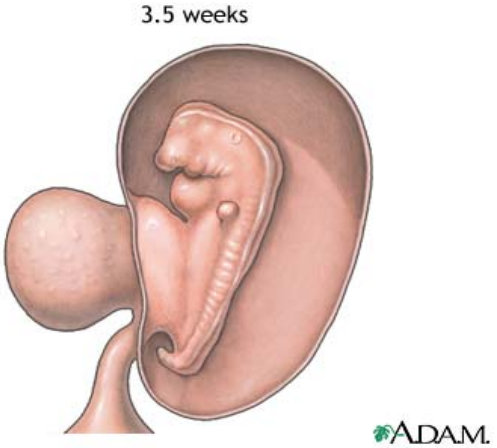
Biological Computing



Sensor Networks



Reconfigurable Computing



Cells during Morphogenesis

Modular Robotics

More formally...

- A spatial computer is a collection of computational devices distributed through a physical space in which:
  - the difficulty of moving information between any two devices is strongly dependent on the distance between them, and
  - the “functional goals” of the system are generally defined in terms of the system's spatial structure

More formally...

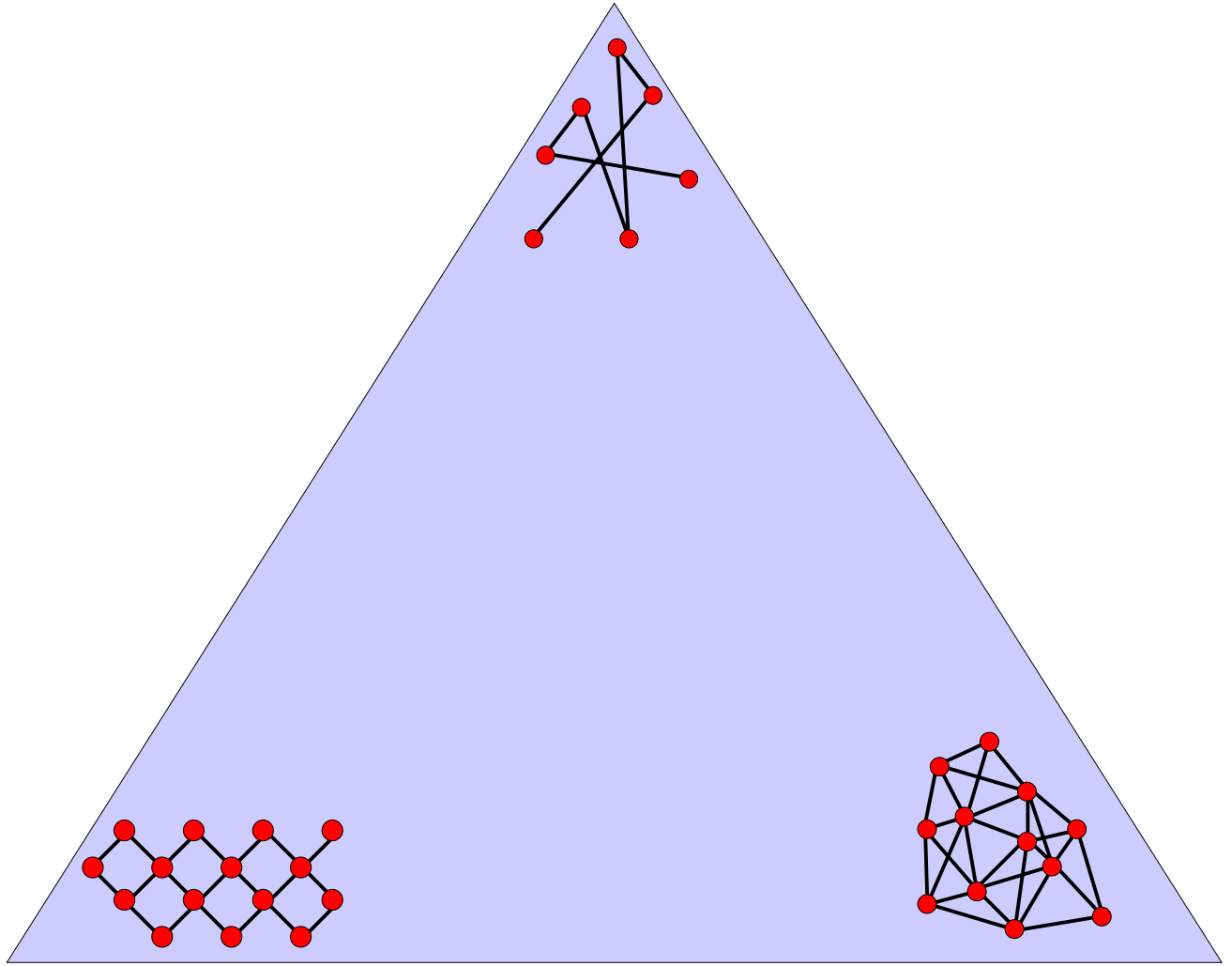
- A spatial computer is a collection of computational devices distributed through a physical space in which:
  - the difficulty of moving information between any two devices is strongly dependent on the distance between them, and
  - the “functional goals” of the system are generally defined in terms of the system's spatial structure

Notice the ambiguities in the definition

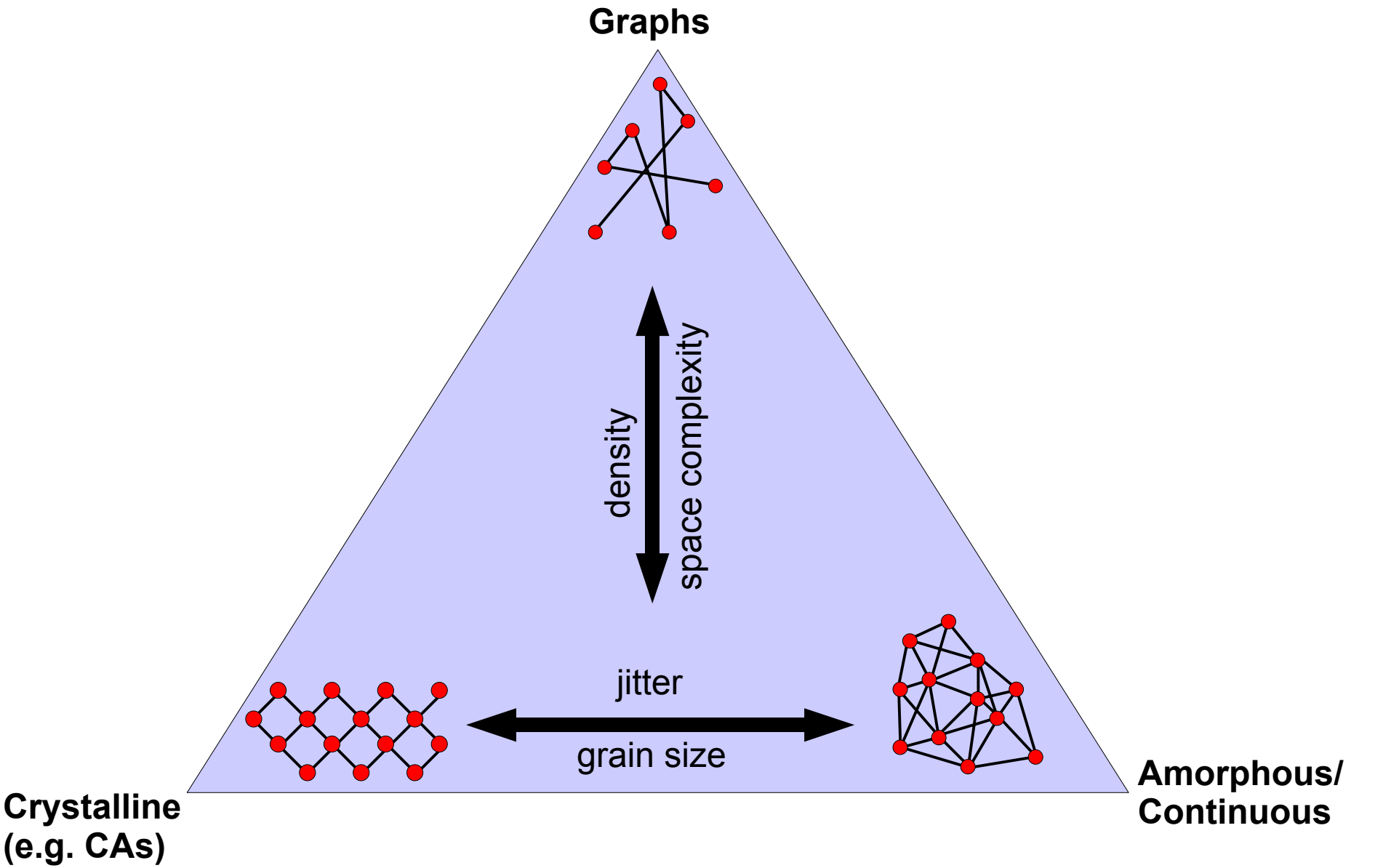
Tentative Mathematical Definition

- A spatial computer is any set of n devices s.t.
  - Graph $\{V,E\}$ with edge weights $w(v_1,v_2)$
  - Manifold M, with distance function d
    - M is compact, Riemannian *(may be stronger than needed)*
  - Position function p: V → M
  - $w(v_1,v_2) = O(1/d(p(v_1),p(v_2)))$

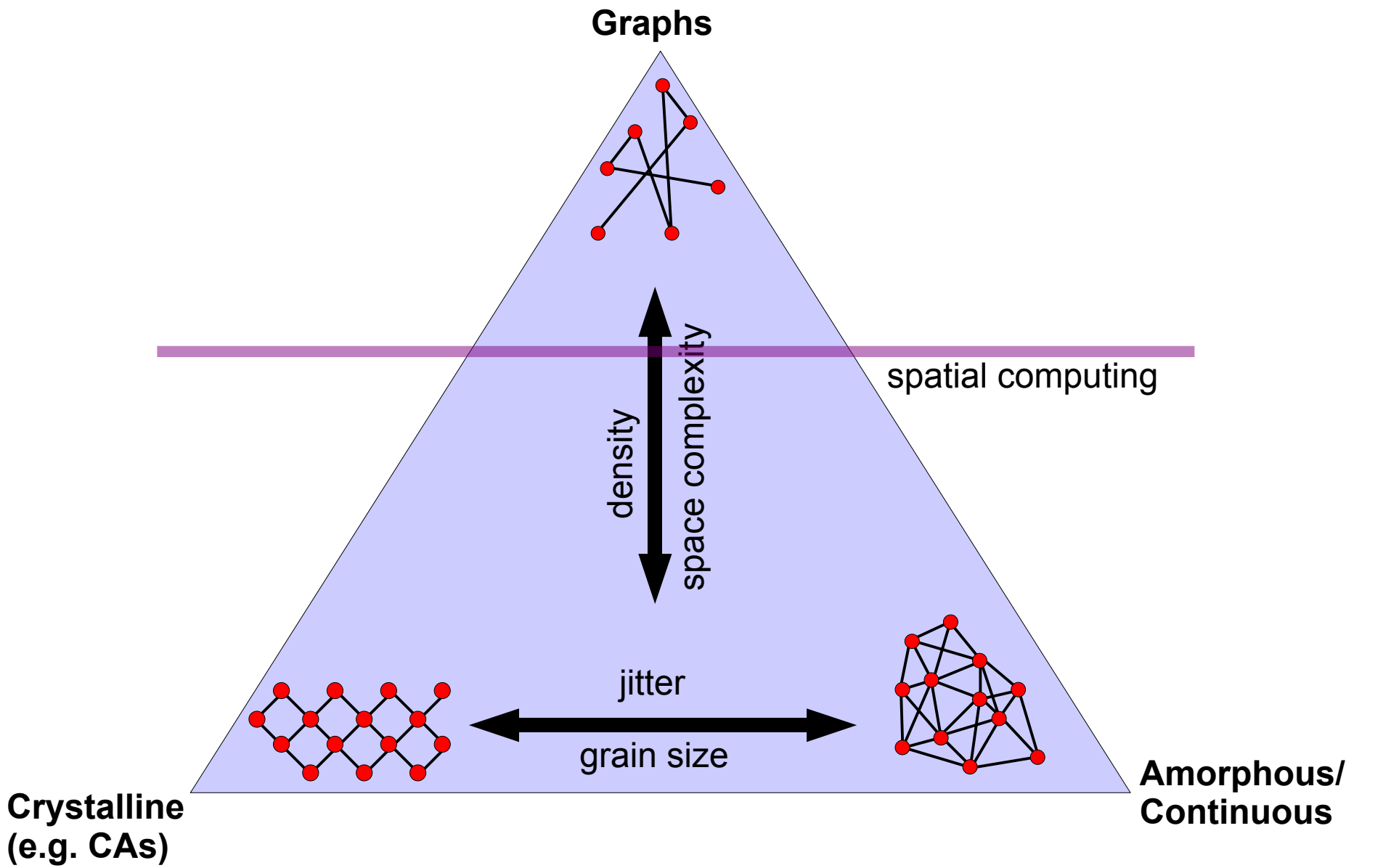
Examples: unit disc network, chemical diffusion



(w. Dan Yamins)


Graphs
density
space complexity
jitter
grain size
Crystalline
(e.g. CAs)
Amorphous/
Continuous

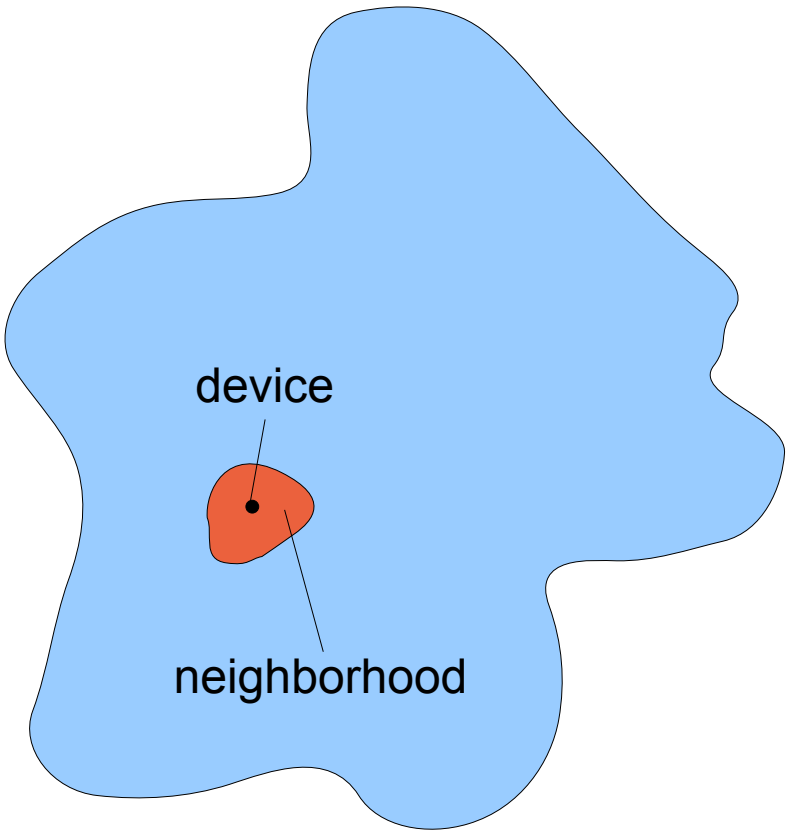
(w. Dan Yamins)

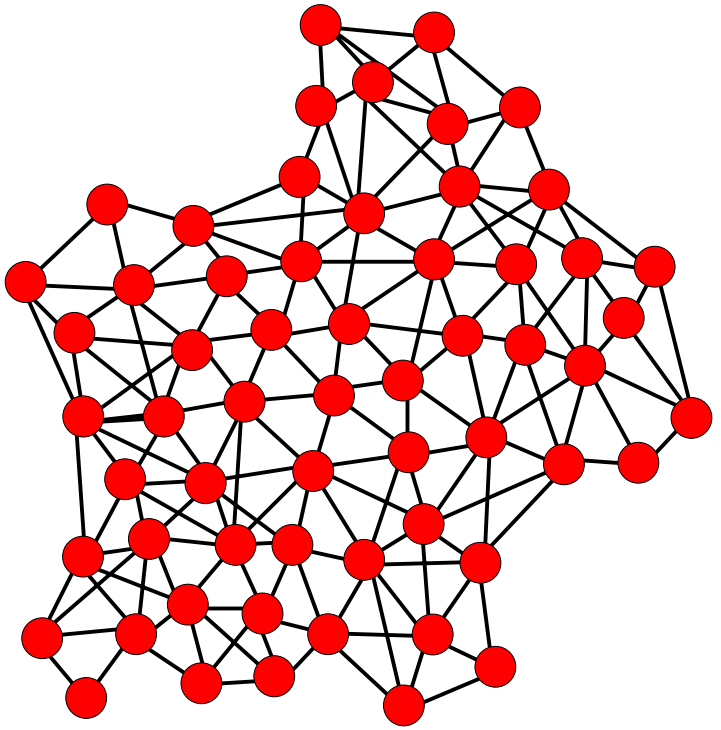

Graphs
spatial computing
density
space complexity
jitter
grain size
Crystalline
(e.g. CAs)
Amorphous/
Continuous

(w. Dan Yamins)

What do you think is or is not a spatial computer?

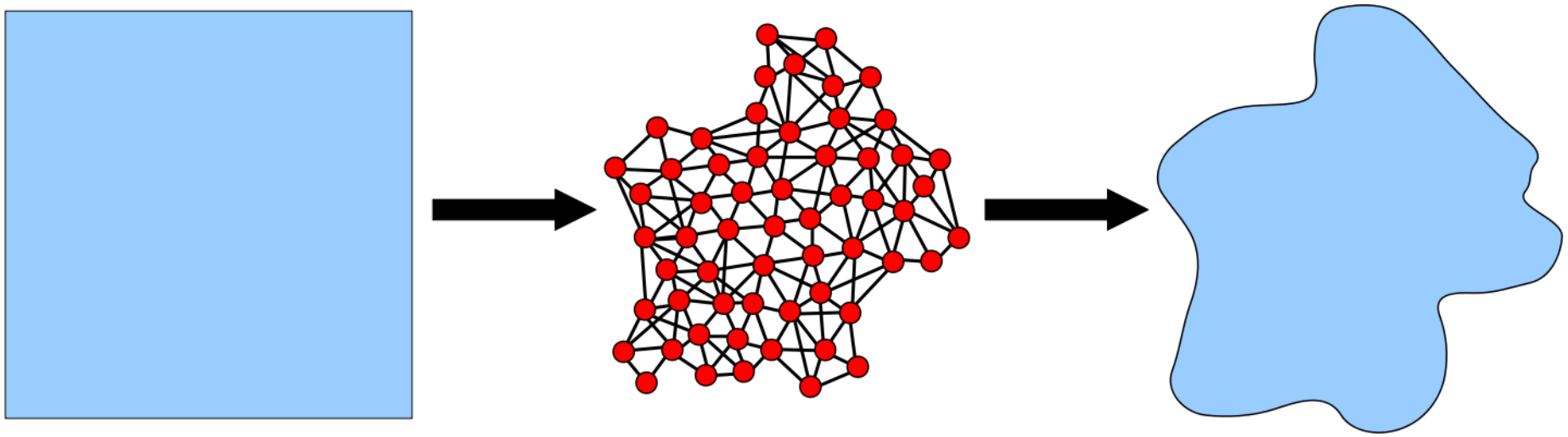
Space/Network Duality





Space/Network Duality

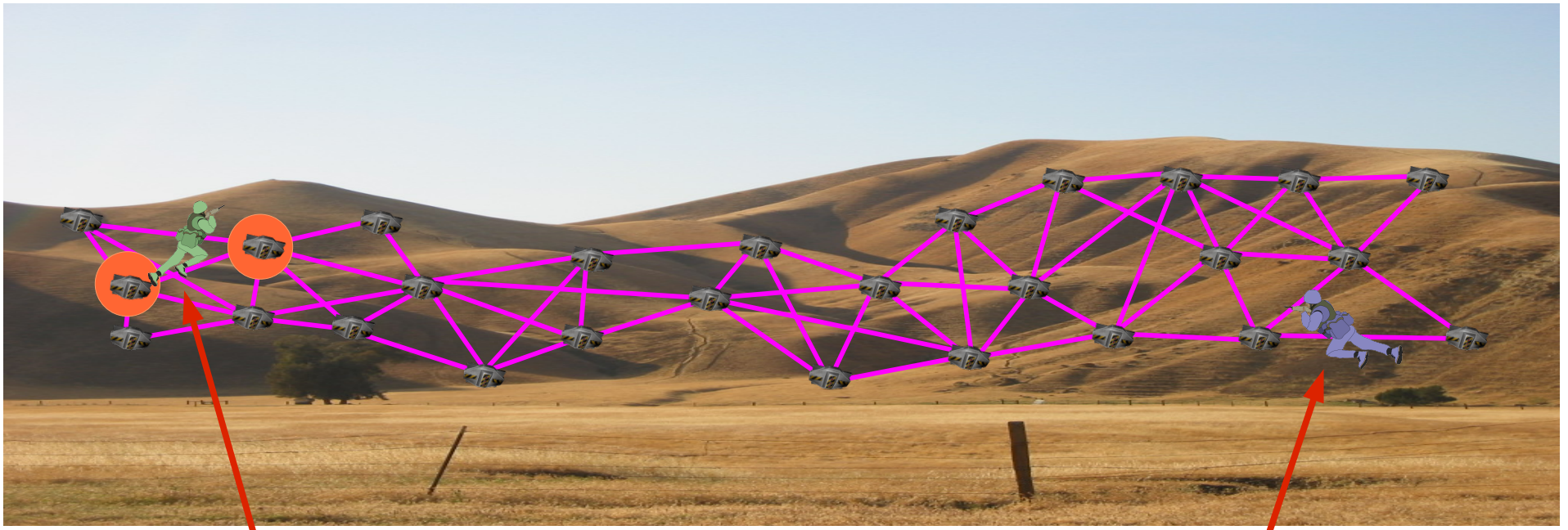
How well does the network cover space?



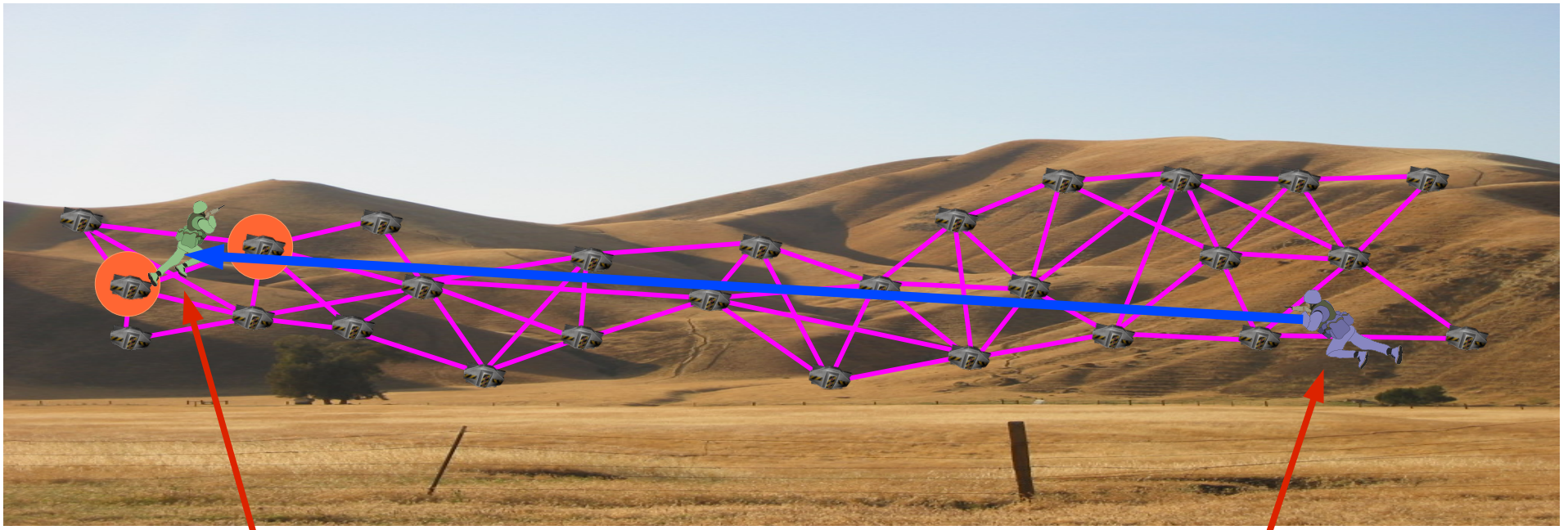
What space is covered well by the network?

Example: Target Tracking

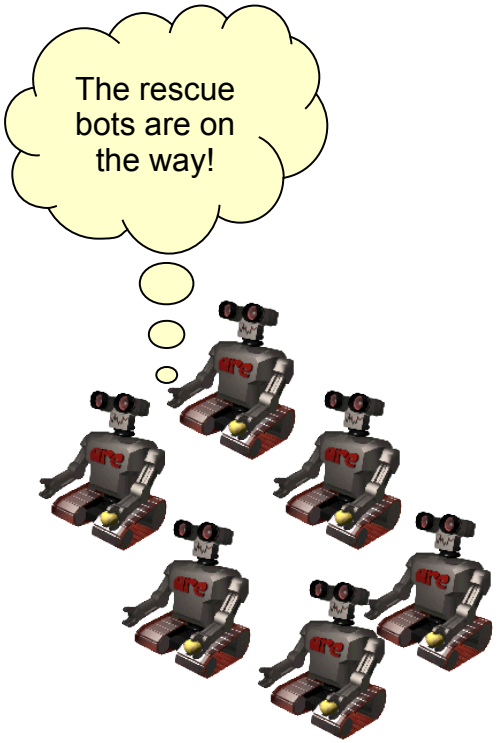
Example: Target Tracking



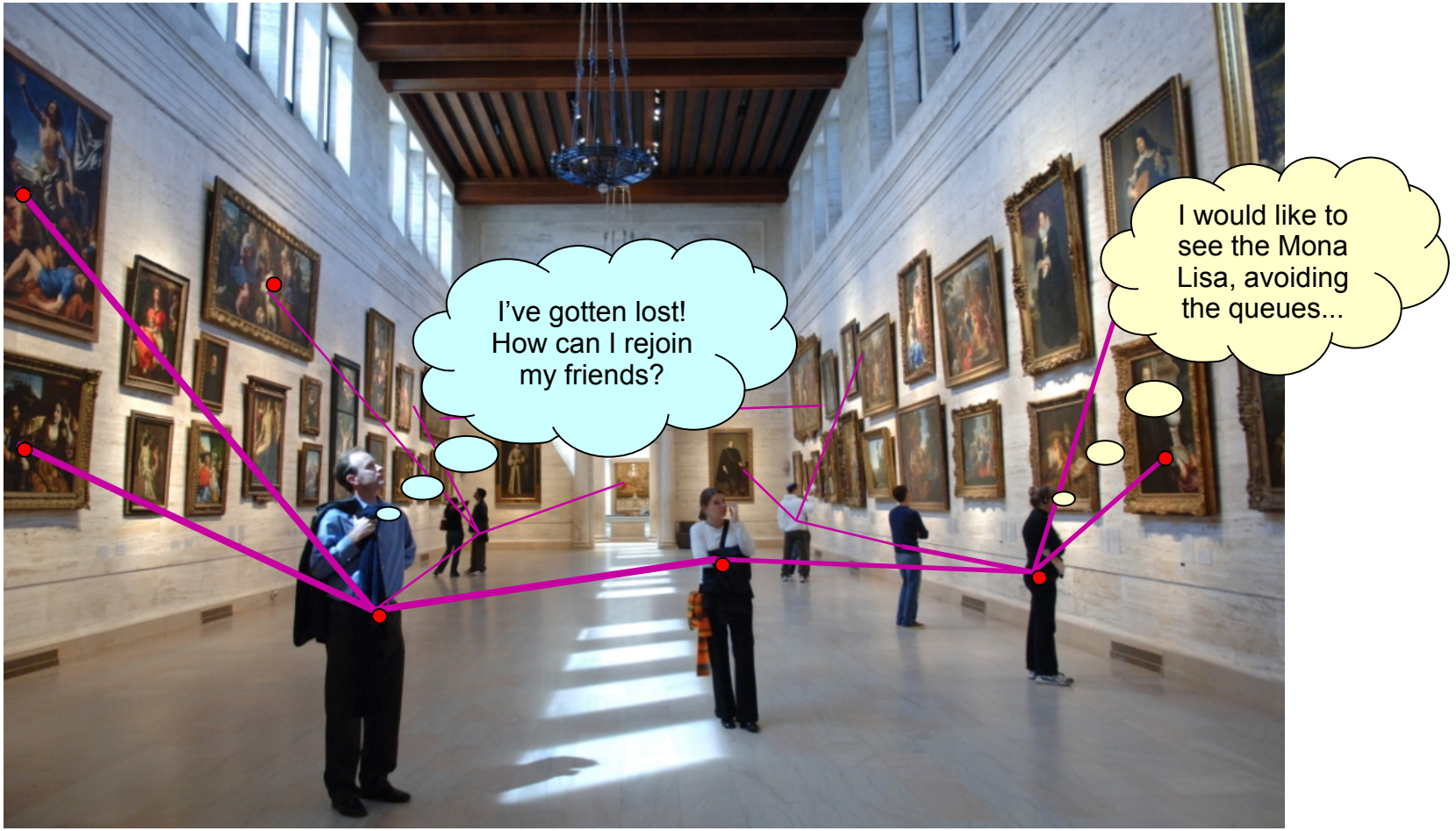
Example: Target Tracking



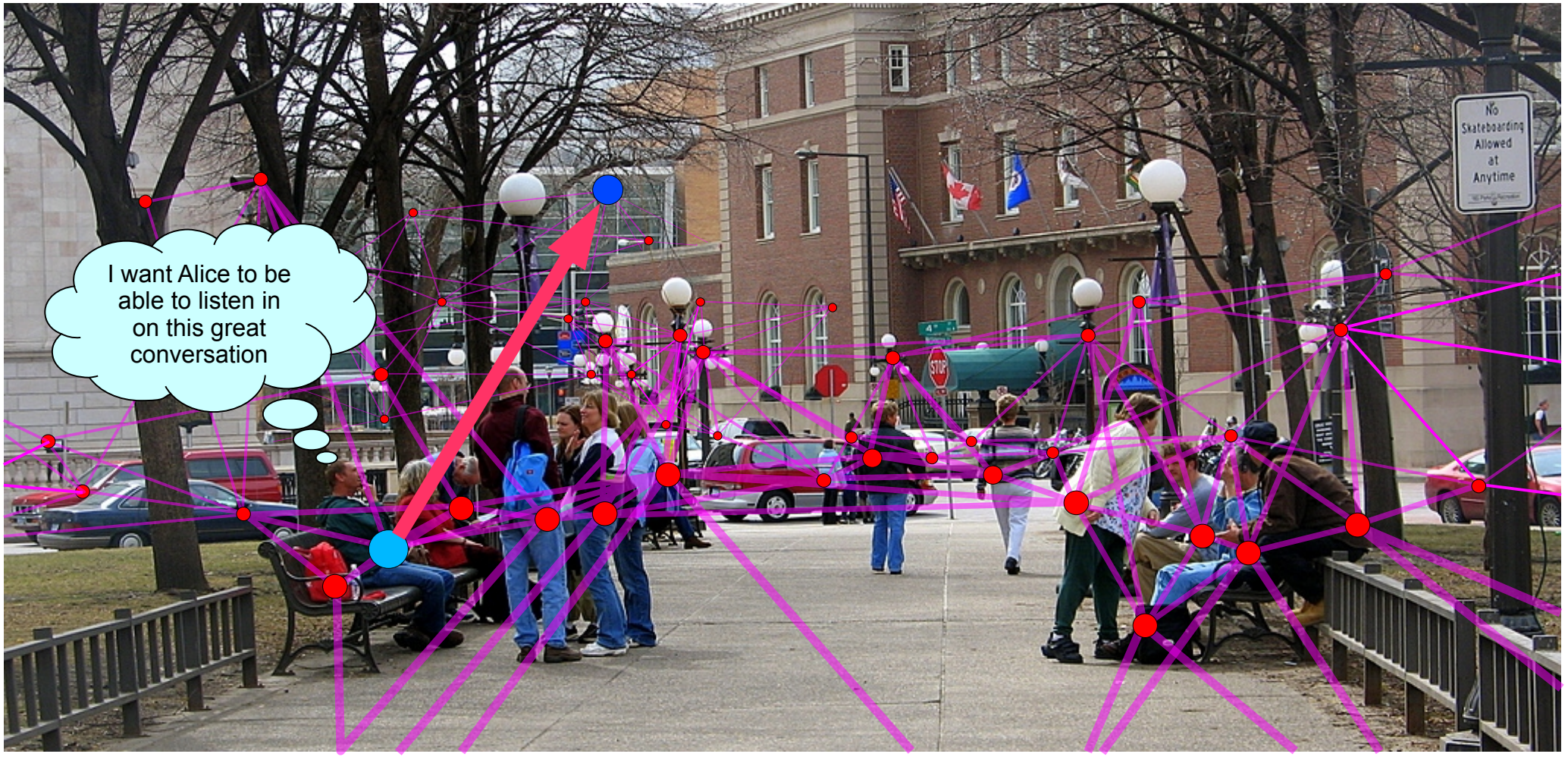
Example: Search & Rescue



Example: Museum Guide



Example: Mobile Streaming



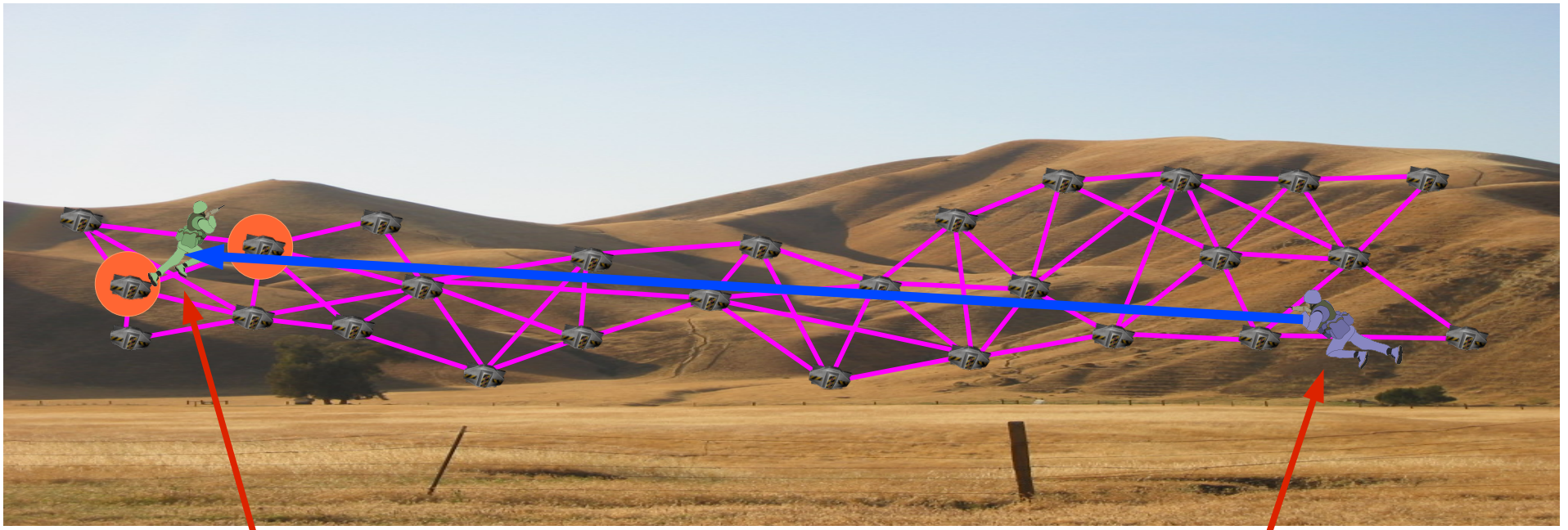
How can we program these?

- Desiderata for approaches:
  - Simple, easy to understand code
  - Robust to errors, adapt to changing environment
  - Scalable to potentially vast numbers of devices
  - Take advantage of spatial nature of problems

Agenda

- What is spatial computing?
- **Geometric Programming**

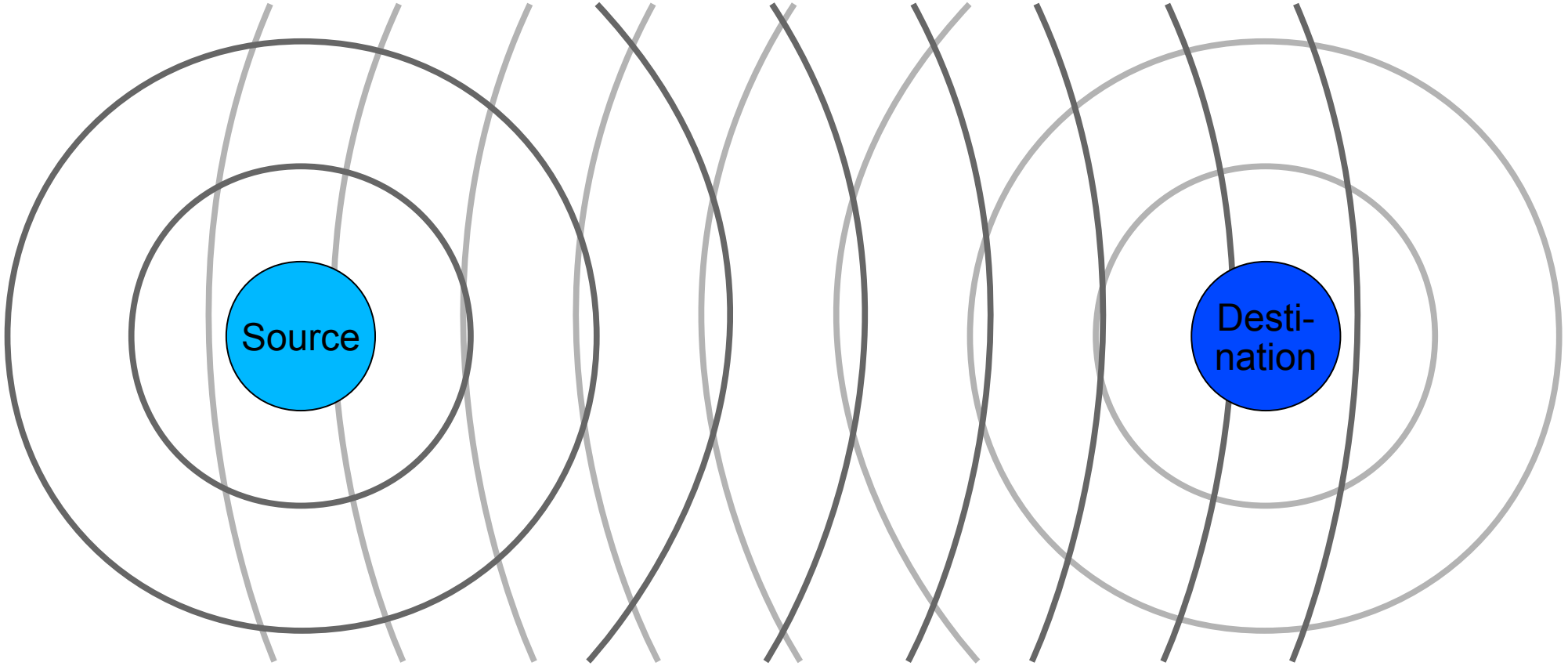
Example: Target Tracking



Geometric Program: Channel

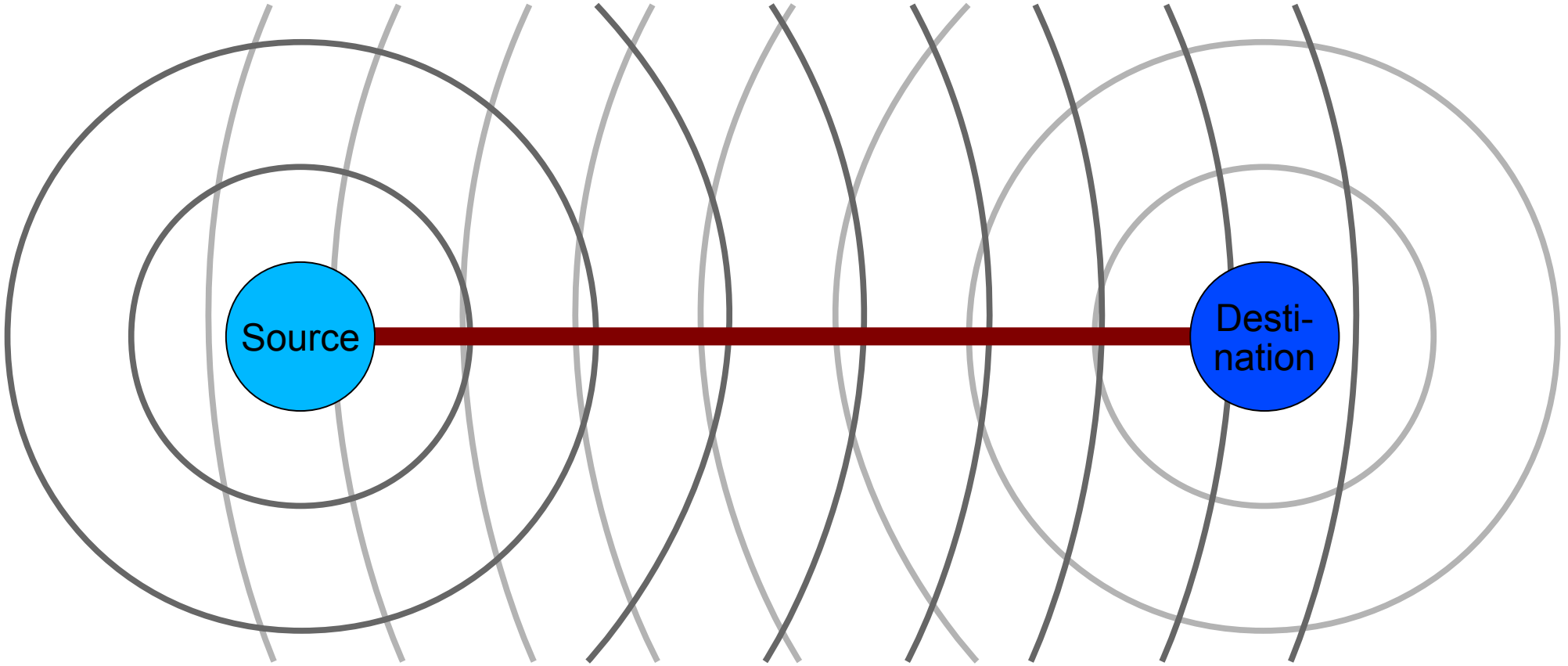
(cf. Butera)

Geometric Program: Channel



(cf. Butera)

Geometric Program: Channel

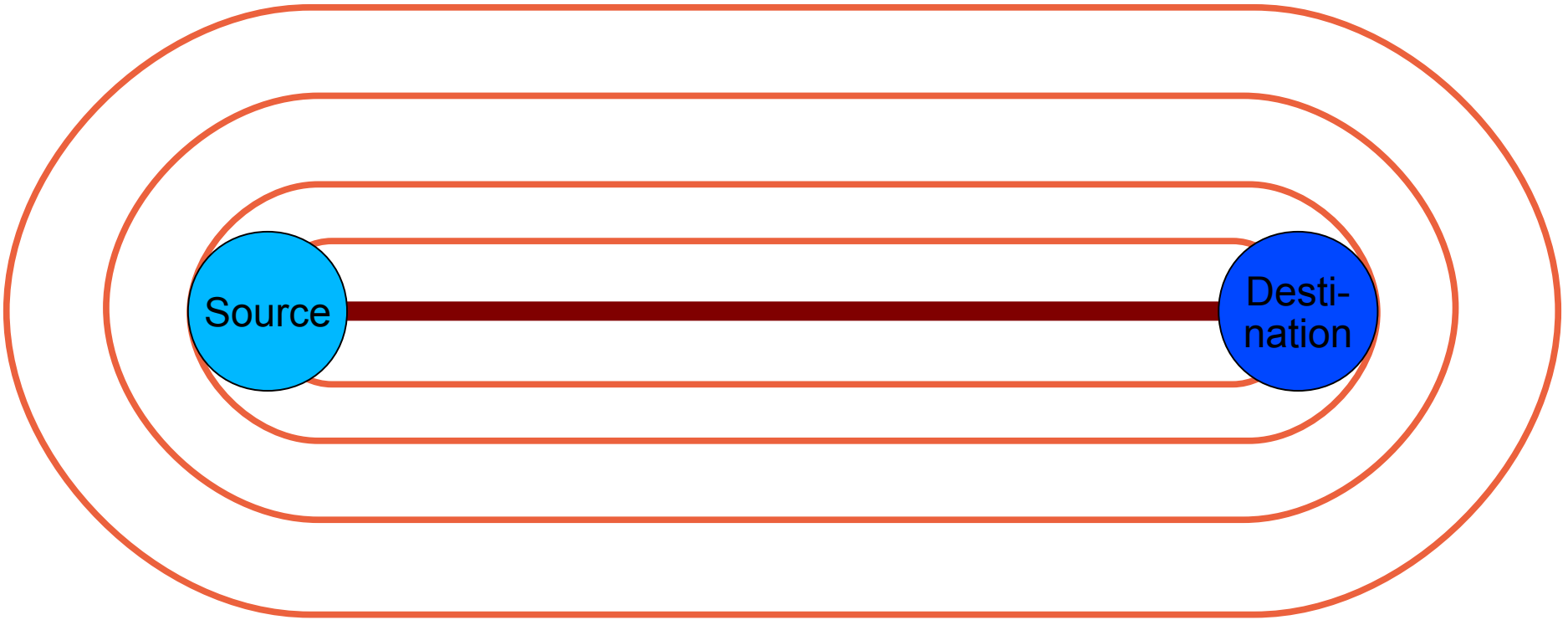


(cf. Butera)

Geometric Program: Channel

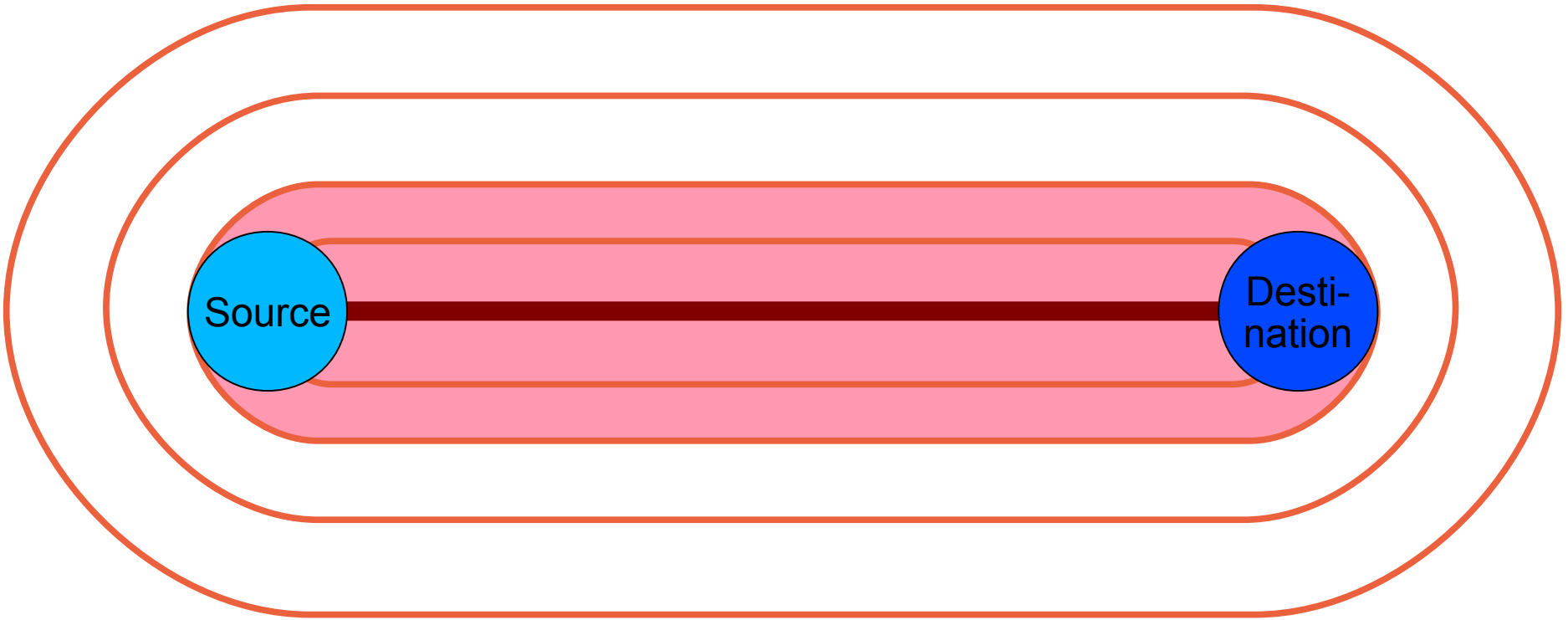
(cf. Butera)

Geometric Program: Channel



(cf. Butera)

Geometric Program: Channel

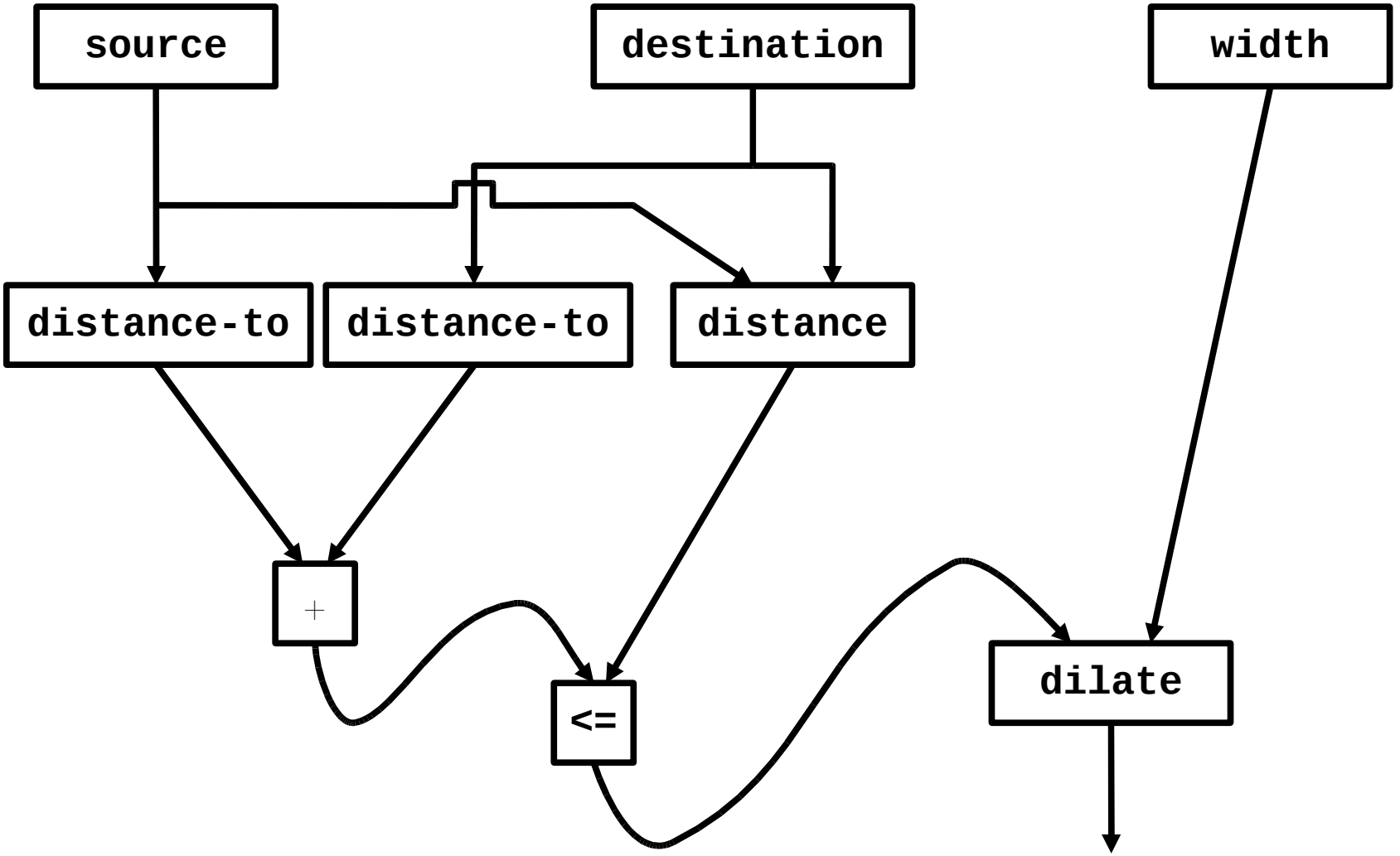


(cf. Butera)

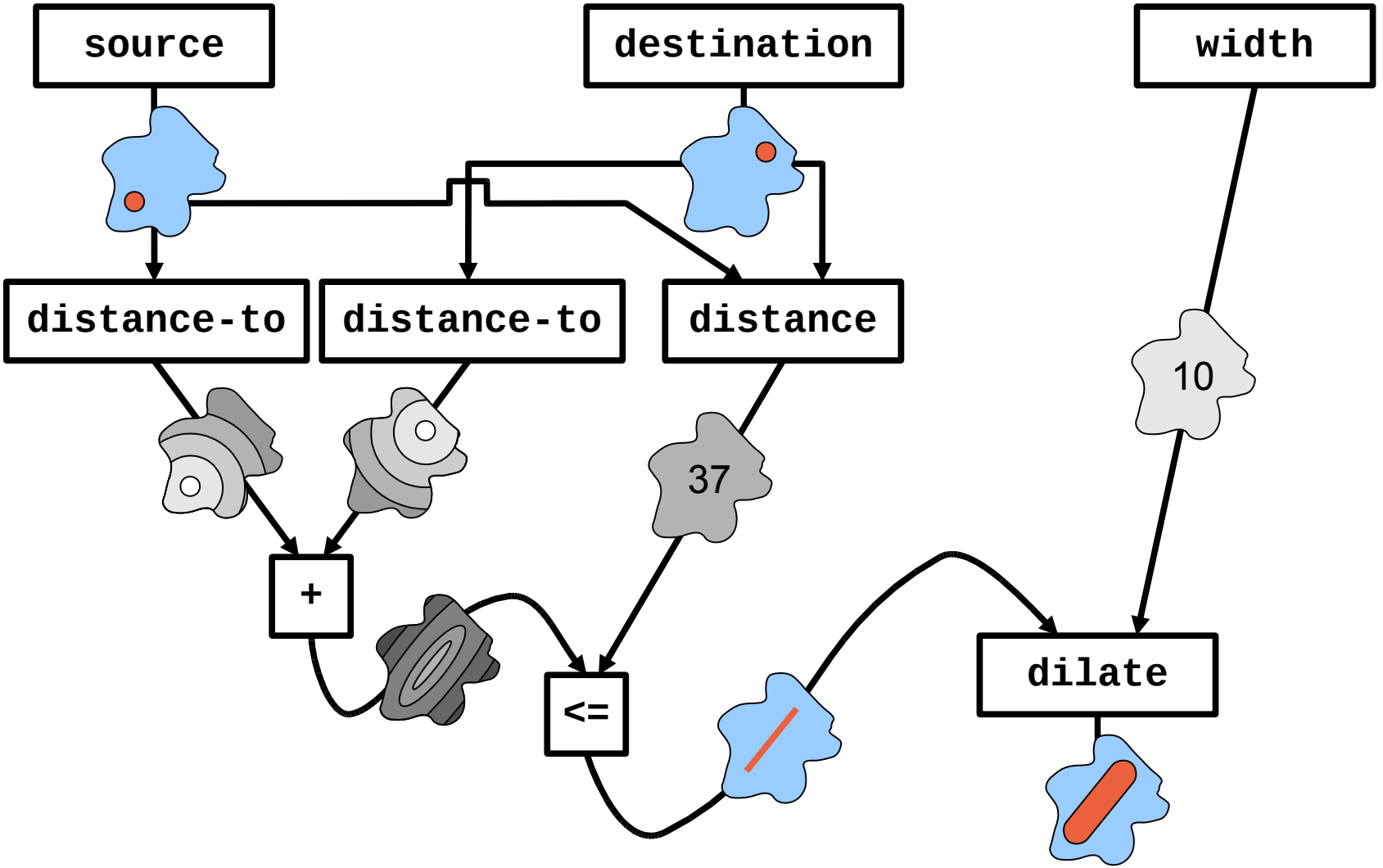
Geometric Program: Channel

(cf. Butera)

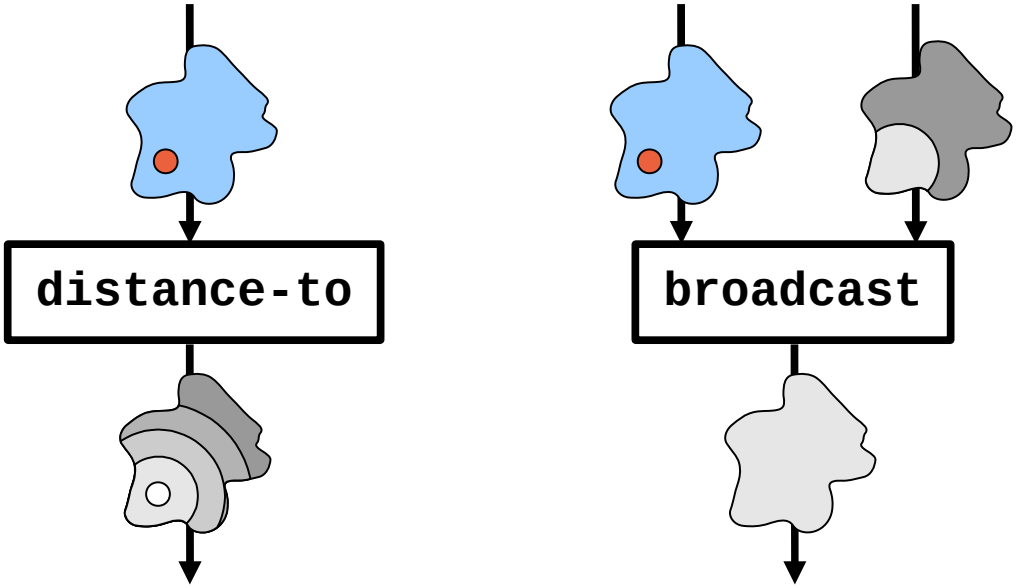
Computing with fields



Computing with fields



Simple Geometric Programming

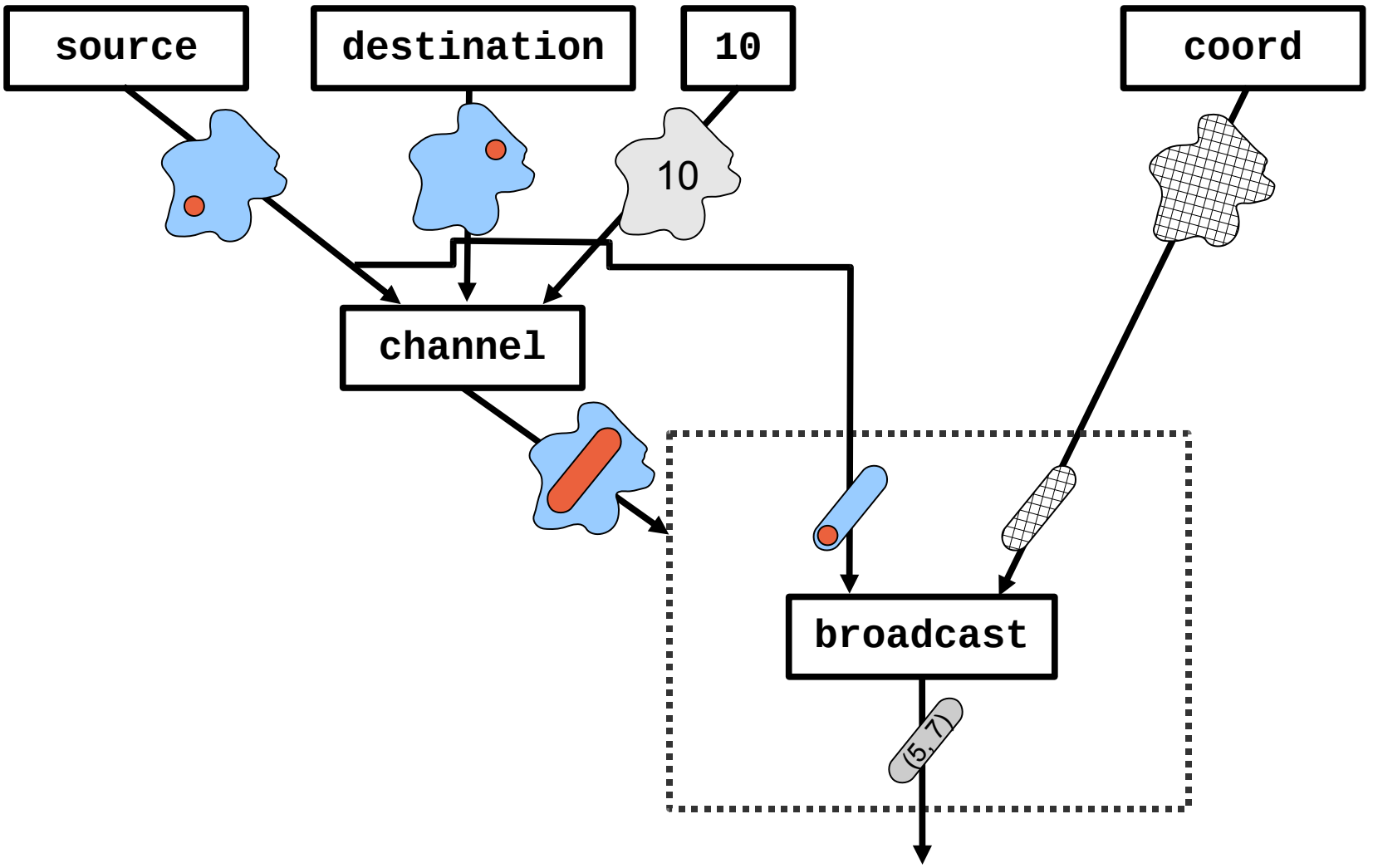


LISP Pointwise: e.g. **`2`**, **`+`**, **`sin`**, **`if`**, **`let`**

Sensors: **`(sense n)`**, **`(coord)`**, **`(hood-radius)`**

Actuators: **`(red v)`**, **`(green v)`**, **`(blue v)`**

Branching = Restriction



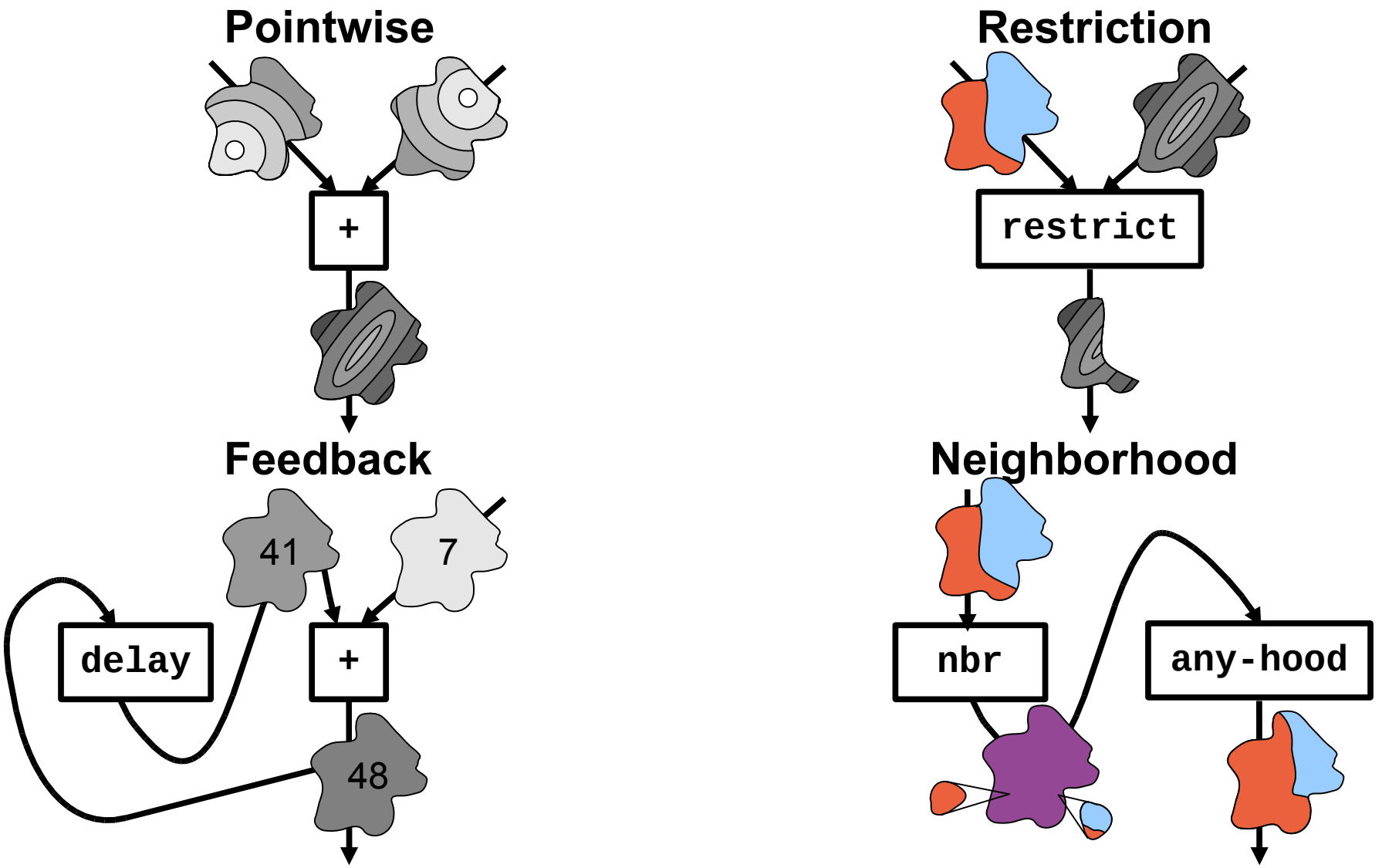
Simple Geometric Programming

Let's go try writing some programs...

Summary

- Spatial computers are space-filling sets of locally-communicating devices
  - Sensor networks, robot swarms, pervasive computing, embryos, biofilms, …
  - Network and space are dual
- Geometric programs can be expressed as operations on fields

Lecture 2: Continuous Space-Time Programs



Proto is available

http://stpg.csail.mit.edu/proto.html

(or google “MIT Proto”)

- Includes libraries, compiler, kernel, simulator, platforms
- Licensed under GPL (w. libc-type exception)

Further Questions

- Must a spatial computer's manifold be compact and Riemannian?
- How else can we smoothly change between crystalline, amorphous, and graph networks?
- What other geometric primitives are needed?
- What limits are there on patterns we can create using geometric primitives?