Spatial Computers & Fields

Jacob Beal
Lecture 1 of 5 on Spatial Computing
ISC-PIF Summer School, 2009
Today: geometric programming

La Belle France!
Lecture 2: Continuous Space-Time Programs

Pointwise

Feedback

Restriction

Neighborhood

delay

41

48

7

restrict

nbr

any-hood
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 \( p: V \rightarrow M \)
  - \( w(v_1,v_2) = O(1/d(p(v_1),p(v_2))) \)

Examples: unit disc network, chemical diffusion
Graphs

Crystalline (e.g. CAs)

Amorphous/Continuous

(w. Dan Yamins)

density

space complexity

jitter

grain size
Graphs

Crystalline (e.g. CAs)

Amorphous/Continuous

spatial computing

density

space complexity

jitter

grain size

(w. Dan Yamins)
What do you think is or is not a spatial computer?
Space/Network Duality

device

neighborhood
Space/Network Duality

How well does the network cover space?

What space is covered well by the network?
Example: Target Tracking
Example: Target Tracking

Intruder

Guard
Example: Target Tracking

Intruder

Guard
Example: Search & Rescue

I've found a victim!

The rescue bots are on the way!
Example: Museum Guide

I've gotten lost! How can I rejoin my friends?

I would like to see the Mona Lisa, avoiding the queues...
Example: Mobile Streaming

I want Alice to be able to listen in on this great conversation.
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

Intruder

Guard
Geometric Program: Channel

(cf. Butera)
Geometric Program: Channel

(cf. Butera)
Geometric Program: Channel

(cf. Butera)
Geometric Program: Channel

Source — [ Connection ] — Destination

(cf. Butera)
Geometric Program: Channel

(cf. Butera)
Geometric Program: Channel

(cf. Butera)
Geometric Program: Channel

(cf. Butera)
Computing with fields

source

destination

distance-to

distance-to

distance

width

<=

+}

dilate
Computing with fields

source

distance-to

+ 37

destination
distance-to
distance

dlitate

10

width
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

source

destination

10

channel

10

coord

broadcast
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

Pointwise

Feedback

delay

Restriction

Neighborhood

Restrict

nbr

any-hood
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?