Programming Spatial Computers

Jacob Beal
Project MAC/Space-Time Programming Group
September, 2006
Space-filling Computers

- Sensor networks
- Biological Computing
- Robot Swarms
- Distributed Control Systems
- FPGAs
- Programmable Matter
Amorphous Medium Approach
Amorphous Medium Approach
Amorphous Medium Approach

Program the space... approximate with a network
Global v. Local v. Discrete

Compiler

Kernel

Program

Global

Local

Discrete
Target Tracking
Target Tracking

(defun local-average (v) (/ (reduce-nbrs v integral) (reduce-nbrs integral 1)))
(defun gradient (src)
  (let* ((n infinity
           (+ 1 (mux src 0 (reduce-nbrs min (+ n nbr-range)))))
         (- n 1))
(defun grad-value (src v)
  (let ((d (gradient src)))
    (let* ((x 0 (mux src v (2nd (reduce-nbrs min (tup d x)))))
           (x))
(defun distance (p1 p2) (grad-value p1 (gradient p2)))
(defun channel (src dst width)
  (let* ((d (distance src dst))
          (trail (<= (+ (gradient src) (gradient dst)) d))
          (dilate width trail)))
(defun track (target dst coord)
  (let ((point
          (if (channel target dst 10)
            (grad-value target
             (mux target
               (tup (local-average (1st coord))
               (local-average (2nd coord)))
               (tup 0 0)))
             (tup 0 0))))
    (mux dst (vsub point coord) (tup 0 0))))
Threat Avoidance

(def exp-gradient (src d)
  (letf ed ((n src (max (* d (reduce-nbrs max n)) src)))
    n))
(def sq (x) (* x x))
(def dist (p1 p2)
  (sqrt (+ (sq (- (1st p1) (1st p2)))
           (sq (- (2nd p1) (2nd p2))))))

(def l-int (p1 v1 p2 v2)
  (pow (/ (- 2 (+ v1 v2)) 2) (+ 1 (dist p1 p2))))

(def max-survival (dst v p)
  (letf ed
    ((ps 0 (mux dst
            1
            (reduce-nbrs max (* (l-int p v (local p) (local v)) ps)))
        ps)))
  (def greedy-ascent (v coord)
    (- (2nd (reduce-nbrs max (tup v coord))) coord))
  (def avoid-threats (dst coords)
    (greedy-ascent
      (max-survival
        dst
        (exp-gradient (sense :threat) 0.8) coords) coords))
Ongoing Projects

• Programming bacteria
• Robotic swarms
• Simulating biological systems
• Power management
• Hardware layout
• Theoretical analysis
• Compiler improvement
etc.