

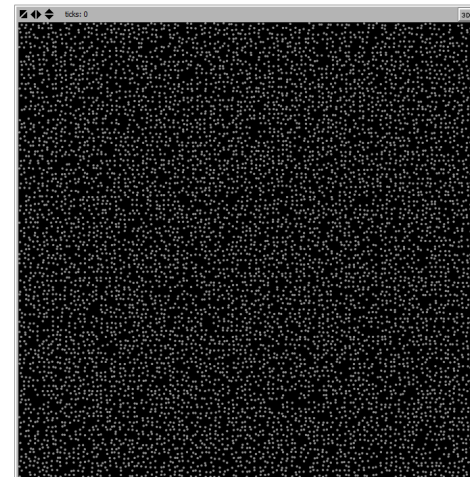
Exercise 5

Institut für Technische Informatik

08.01.2019

Amorphous Computing

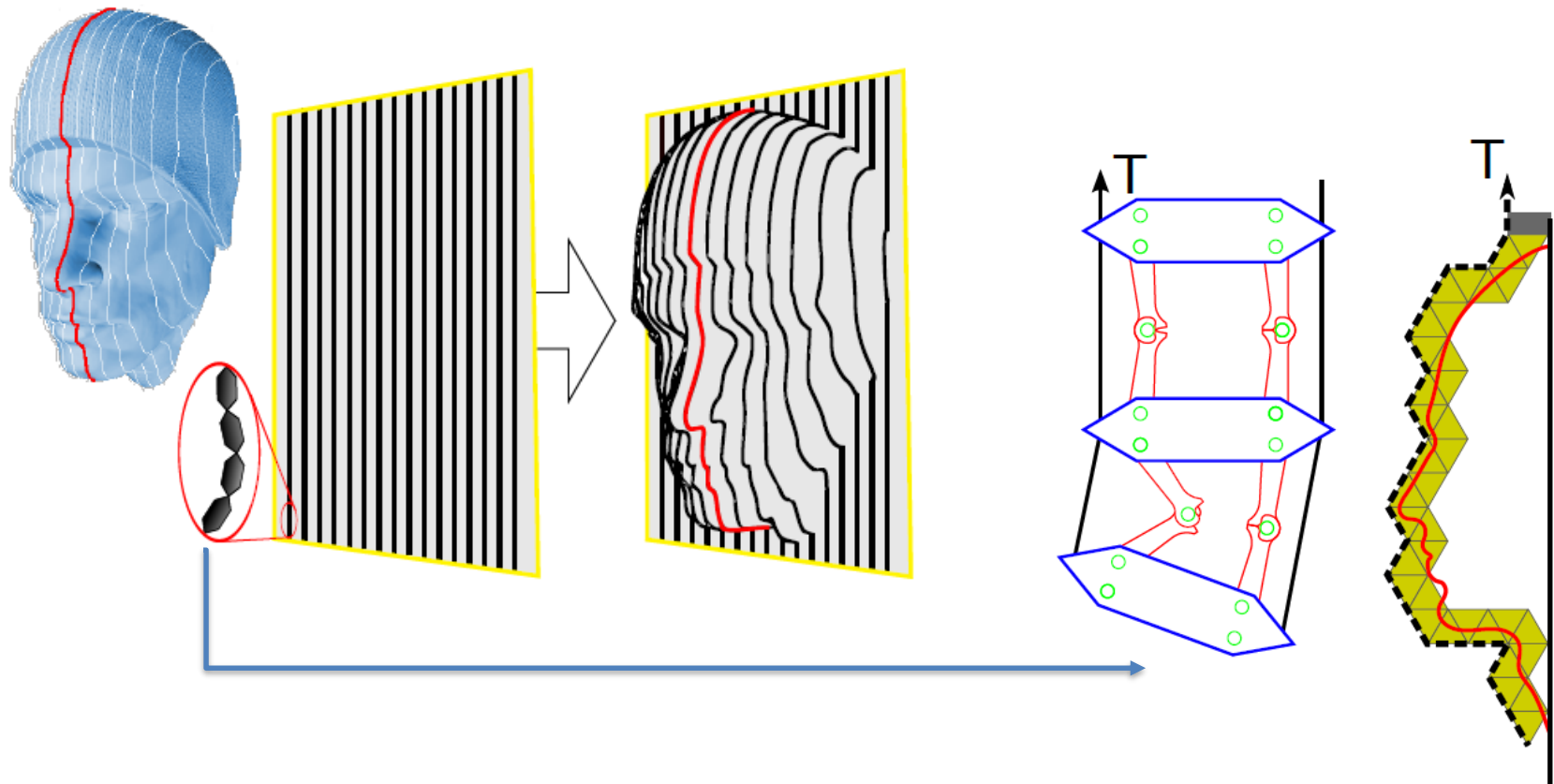
- Programming paradigm
 - Inspired by biology, cellular engineering
 - Objective is to create the algorithmic and technological foundations for exploiting **programmable materials**
- These materials incorporate a vast number of irregularly placed, locally interacting, asynchronous, programmable computing **particles**
 - Densely distributed
 - Can communicate with nearby neighbors
 - No priori knowledge of position
 - Reacting to the environment
 - May produce actions



Amorphous Computing

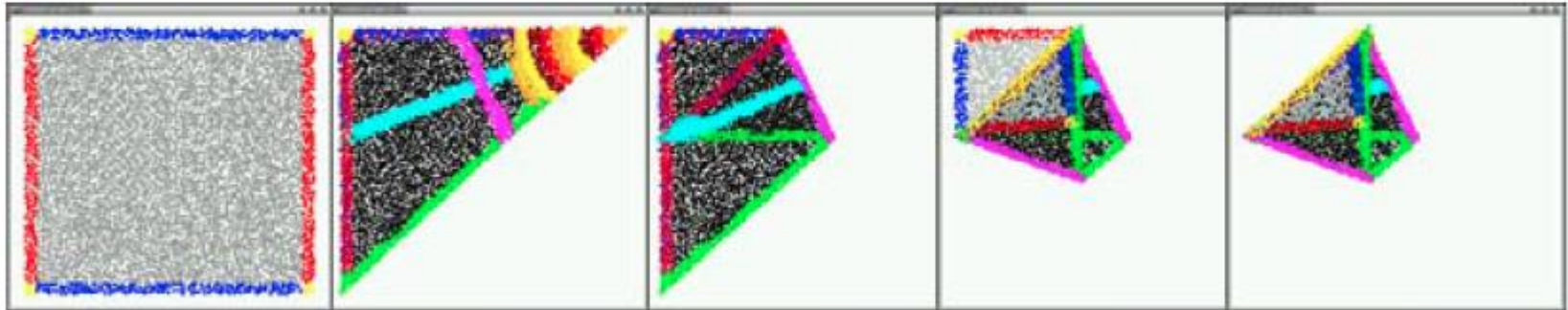
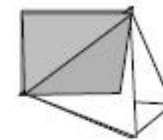
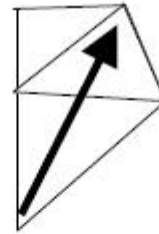
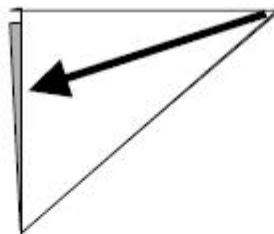
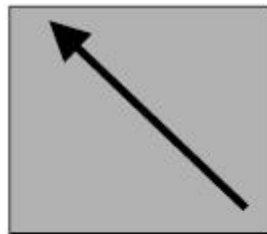
■ Example: Robotic particle chain

(<http://dl.acm.org/citation.cfm?id=2695932>)



Amorphous Computing

- Example: Origami programmable cell sheet
 - Global language to compile local-interaction rules from a high-level description of an origami-like folded structure



Amorphous Computing

■ Example: Wave propagation

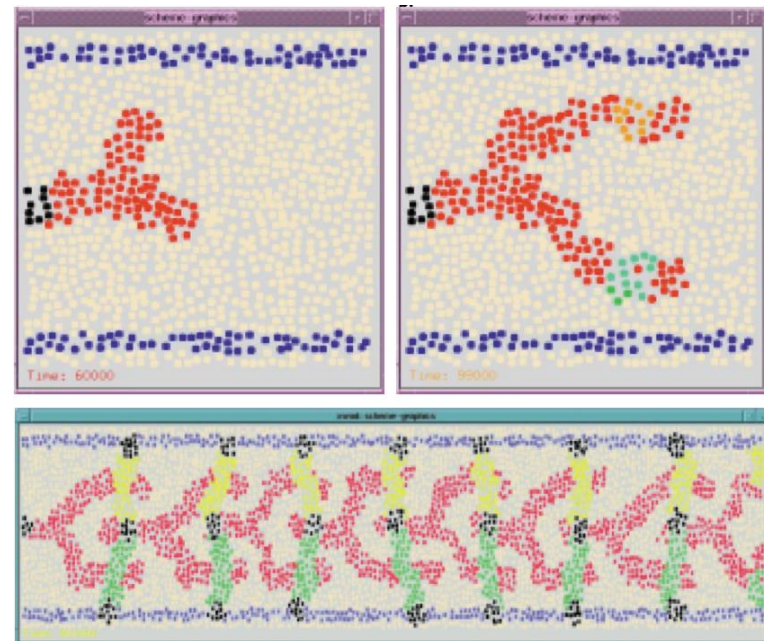
(<http://cacm.acm.org/magazines/2000/5/7669-amorphous-computing/fulltext>)

- Initial “anchor” particle broadcasts a message to its neighbors
- Neighbors propagate message to their neighbors
- Diffusion wave that spreads throughout the system
- Each message contains a hop count that each particle stores and increment before rebroadcasting, ignoring any subsequent higher values to prevent the wave from propagating backwards
- Hop-count provides estimates of distance from the anchor
 - One can produce 2D coordinate systems by propagating waves from two anchors

Amorphous Computing

- From wave propagation to pattern formation
(<http://cacm.acm.org/magazines/2000/5/7669-amorphous-computing/fulltext>)
 - To enable programmers to specify complex patterns, such as the interconnect topology of an electronic circuit, Coore et al. have developed the **growing point language** (GPL)

Example: graph of a
chain of CMOS inverters

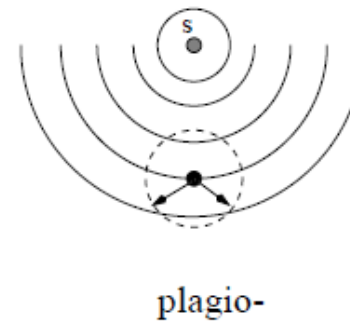
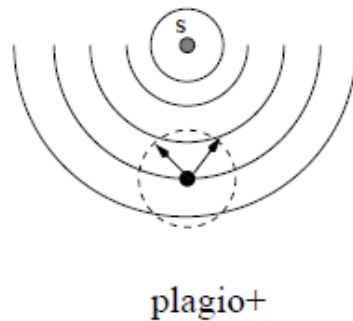
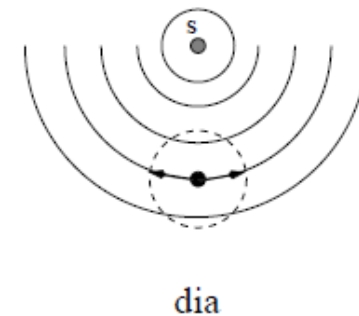
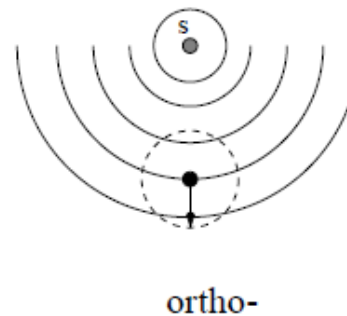
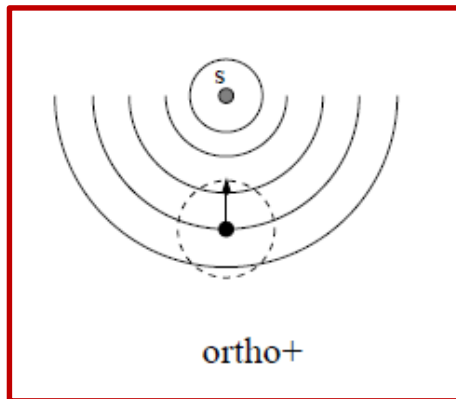


Exercise 5: Growing Point Language

- GPL: high-level language to specify complex patterns
- Three main abstractions
 - Pheromones (distinguishable by name)
 - Materials (distinguishable by color)
 - Growing Point
- Growing points:
 - Can secrete long-range pheromone concentrations to establish a gradient field
 - Can move from processor to processor based on a tropism
 - As growing points travel, they label the visited processors with material

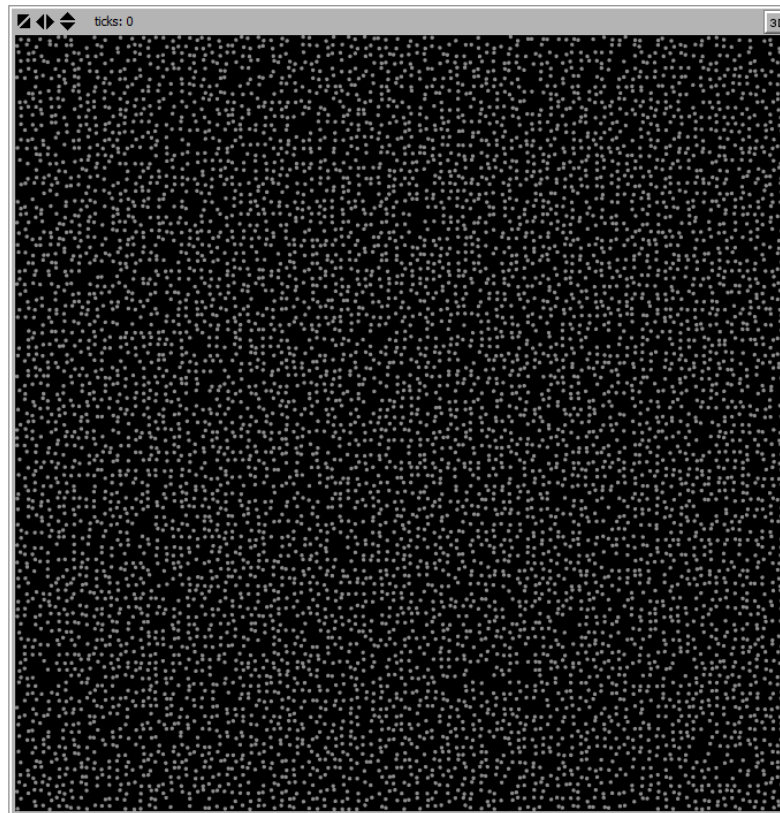
Exercise 5: Growing Point Language

- Tropism (Tropismus, Reizhinwendung)



Exercise 5: Growing Point Language

- Processors are spread with high-density in the field (skeleton provided: download it from the TeachCenter)



Exercise 5: Growing Point Language

- User is asked for the number of growing points
 - For each GP, the user is asked to place it by clicking the mouse inside the world and for the pheromone radius
- User interaction with NetLogo
 - NetLogo embeds commands to interact with the user
 - **user-message** "This is a info box"
 - **user-input** "What is your name?" (asks user for a value)
 - If the value is a number, boolean or list you can parse it with the command **read-from-string**
 - Example: crt **read-from-string** user-input "How many turtles?"



Exercise 5: Growing Point Language

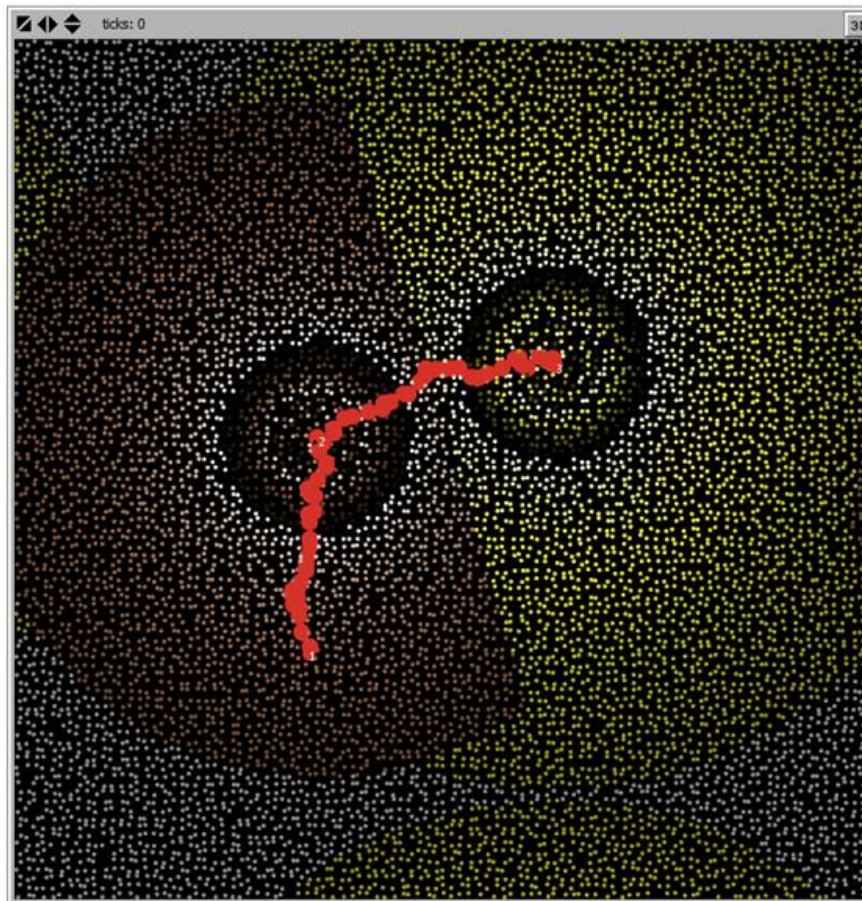
- User interaction with NetLogo
 - `mouse-xcor` and `mouse-ycor` report the x and y coordinates of the mouse at a given instant
 - `mouse-down?` Reports true if the mouse is pressed inside the world. If not pressed or outside the world it reports false
 - Check NetLogo documentation for a full list

- When pressing the “go” button, the first GP should “travel” according to the *ortho+* tropism to the second; afterwards from the second to the third, and so on

- When traveling, the GP should “drop” material on its way, i.e., all visited processors should be colored red

Exercise 5: Growing Point Language

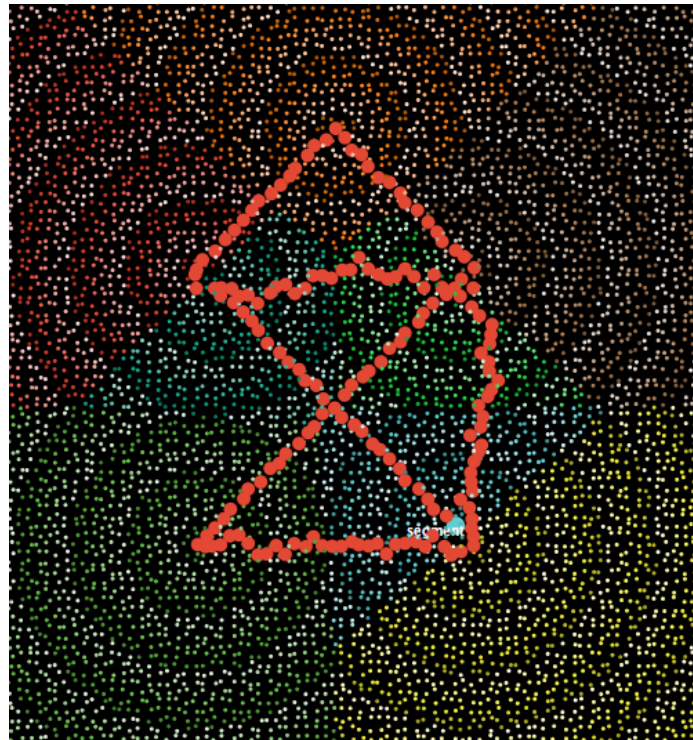
- Build the NetLogo model accordingly
- Example:



Exercise 5: Growing Point Language

- Paint “Das Haus vom Nikolaus” for testing purposes

http://de.wikipedia.org/wiki/Haus_vom_Nikolaus



- Deadline is **Sunday, 20.01.2019**, at 23:59 CET

Questions?

