EXPERIMENTS OF MORPHOGENESIS IN SWARMS OF SIMPLE ROBOTS

MARCO MAMEI, MATTEO VASIRANI, FRANCO ZAMBONELLI
Dipartimento di Scienze e Metodi dell'Ingegneria,
Università di Modena e Reggio Emilia

In this paper we focus on the problem of having a multitude of simple mobile computational "particles" (i.e., sorts of small mobile robots) self-organize their global shape so as to obtain a variety of spatial configurations. The problem has a variety of applications in mobile robotics, modular robots, sensor networks, and computational self-assembly. The approach we investigate in this paper attempts at minimizing the local capability of particles and at verifying how and to which extent a variety of global shapes can be obtained by exploiting simple self-organizing algorithms and emergent behaviors. Several experiments are reported showing the effectiveness of the approach.

INTRODUCTION

Over the next decades, MEMS sensors will be everywhere, and sensing virtually everything. Scavenging power from sunlight, vibration, thermal gradients, and background RF, these autonomous sensors motes will be immortal, completely self contained, single chip computers with sensing, actuation, communication, and power supply built in. Entirely solid state, and with no natural decay processes, they may well survive the human race (Pister 2003). In this perspective, we envision the possibility of exploiting these technologies to build sorts of multi-cellular computational organisms, made up of millions of interacting autonomous computational particles, capable of assembling and dynamically re-assembling themselves into a variety of complex shapes (as the T1000 robot in the Terminator 2 movie) for interacting with – and acting in – an environment.

While electronics and communication technologies are advancing, computer scientists and software engineers will be asked to answer two key questions to contribute to the vision:

- How can one achieve reliable patterns of activities in systems made up of a large number of simple computational components interacting in amorphous and dynamic networks?
- How does one translate pre-specified global goals into the local interactions of vast numbers of parts?

In general, the critical task is to identify appropriate (self)organization principles and programming methodologies for controlling the overall behavior of such complex systems. In particular, our goal is to study how and to which extent a group of simple mobile autonomous particles can be programmed to coordinate their respective movements and create variety of global shapes. Apart from futuristic nano-technology scenarios such as computational self-assembly (Nagpal 2002) and the T-1000 vision, such a problem has more practical short-term applications, e.g. coordinate the movements of navigator-equipped cars (Mamei and Zambonelli 2004b), coordinate the movements of a rescue team provided with PDA (Mamei and Zambonelli 2004a), enforcing self-deployment of sensor networks (Estrin 2002) and robots in a landscape (Bay 1995; Fredslund 2002).

Biological organisms, achieving coherent, reliable and complex behavior from the local cooperation of large numbers of identically "programmed" cells, are of course the most natural source of inspiration for all these kinds of problems. A variety of phenomena hint at powerful underlying mechanisms that can adapt to variation, while maintaining constraints that may be geometric, topological or functional (Lawrence 1992; Wolpert 1998). In particular, the diffusion of chemicals among cells and the possibility for cells to be driven in their behavior by the locally sensed gradients of diffused proteins ("morphogen gradients") (Day 2000), due to its simplicity, seems suitable for applications to the problem of pattern formation in simple computational particles.

A large number of papers deal with pattern formation in mobile robots (Sugihara 1990; Bay 1994; Spears 1999; Fredslund 2002), and some exploit approaches somewhat similar to the one of morphogen gradients (Shen 2002; Nagpal 2002). The key contribution here is to show how a variety of patterns (from regular to non-regular ones, also involving differentiation in particles) can be achieved even in the absence of those capability (e.g., global perception, distance and direction sensing) that are required by most other approaches. Also, we show that these results can be achieved in a scalable way and without significant losses in performances.

The following of this paper is organized as follows. Section 2 introduces the concept of morphogen gradients, as the basic mechanism underlying our proposal. Section 3 details our model for particles and our methodological approach, and compares it with related work in the area. Section 4 presents some

experiments in pattern formation. Section 5 discusses the performances of the approach. Section 6 concludes and sketched open research directions.

MORPHOGEN GRADIENTS

Morphogenesis is one of the major outstanding problems in the biological sciences. It concerns the fundamental question of how biological form and structure is generated starting from an immense number of biological cells. Translating this problem into our future nano-technology scenario means answering the following question: Given a bucket of autonomous mobile particles, how can we build the T1000 Terminator? *(Un)fortunately*, we are still far from a solution. Nevertheless, some of the mechanism driving morphogenesis in biology are currently being discovered.

One example of a mechanism common throughout development is the use of morphogen gradients to determine positional information and polarity. In the Drosophilae embryo, cells at one end of the embryo emit a morphogen (protein) that diffuses along the length of the embryo. The concentration of this morphogen is used by other undifferentiated cells to determine whether they lie in the head, thorax or abdominal regions (Lawrence 1992). Different morphogens are used for determining the dorsal-ventral axis, wing and limb development, and even leg bristle polarity. Gradients of morphogens are believed to play an important role in providing position and polarity information in many different organisms, and even in regeneration (Wolpert 1998).

Reproducing morphogen gradients in dense network of short-range wirelessly interacting particles is dramatically simple. A "source" particle can create a gradient by broadcasting a simple message to its local neighborhood (i.e., to all nodes within the wireless connection range). The message can be a simple tuple containing the morphogen name and the morphogen value, initially at zero. Neighbor particles re-broadcast the message in their turn after having modified its value, typically incrementing it, and so on, until the morphogen has propagated through the entire population. Each particle stores and forwards only the minimum value it has heard for a particular morphogen name, thus the morphogen gradient typically represents the shortest path from the source.

The above very simple mechanism can be used in powerful ways to influence the local activities of particles.

1. Leader Election**:** Gradients can be used to elect leader particles in the group. Randomly elected leaders could propagate 'leader' gradients through the network inhibiting others to become leaders on their turn (Nagpal 2002).
2. Selective Propagation**:** Particles can be programmed to selectively choose which morphogen gradients to propagate, also based on the perceived value of morphogen gradients. Thus, particles can act as barriers/inhibitors to

specific morphogen gradients, or as obstacles around which the morphogen must travel.

3. Region Selection*:* If a leader particle propagates a gradient, by having other particles inhibit its propagation when it reach a maximum value, one can create (approximately circular) regions of controlled size, or have cells recognize their being in a specific circular region (see Figure 1a). We outline that if the gradient value is incremented by one at each step, it provides an estimate of distance from the source: a perceived value of n steps implies a distance nr from the source, where r is the wireless communication range of particles. The quality of this estimate depends on the density of particles (Nagpal 2003).
4. Coordinate System*:* Gradients can be used to self-organize coordinate systems. Particles, in fact, can recursively evaluate their coordinates by triangulating the distances – expressed by means of morphogen gradients – from elected beacons (Nagpal 2003).
5. Patterning of Activities*:* When a groups of particles (or even all the particles) is the source for the same morphogen, and when particles can also inhibit the diffusion of some morphogen gradients, complex patterning in particles activities (see Figure 1b) can be created in a sort of agent-based reaction-diffusion model (Bonabeau 1997).
6. Communication*:* Gradient can be used to broadcast messages to other particles and, by having these messages follow other gradients previously laid down, effective routing mechanism can be enforced (Poor 2001).
7. Adaptive Morphogens*:* We can allow a source particle to continuously inject in the system a morphogen message, and have the morphogen value stored by any particle lose significance if not constantly reinforced. The result is that morphogen gradients (and all related activities of particles) adapt as particles move, appear, or disappear (Mamei and Zambonelli 2004a).
8. Driving Motion*:* if a particle can perceive the local slope of any gradient, it can also move following gradients uphill, downhill or along equipotential lines (Mamei and Zambonelli 2004a; Mamei and Zambonelli2004b; Coore 2001; Nagpal 2002).

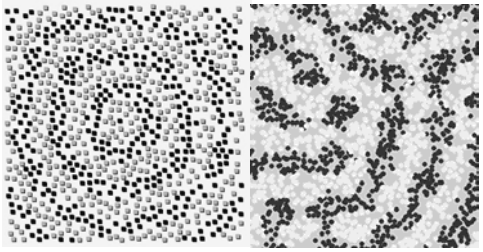

Figure 1. Morphogen gradients: (a) propagating from a source and defining spatial circular spatial regions; (b) propagating from several particles and being inhibited in their propagation, in a sort of reaction-diffusion model.

OUR AND RELATED APPROACHES

In this section, we first describe our model for particles and how they can exploit morphogen gradients towards pattern formation. Then we present related approaches and possible objections to our model

Our approach

To be "compliant" with foreseeable future nano-technology scenarios, we focus on particles with minimal capabilities. Specifically:

1. Particles are autonomous (i.e. have a separate thread of execution and control) and are equally programmed (i.e. they run the same code). Differentiation in their activities – if needed – must be established run-time on the basis of perceived morphogen gradients).
2. In any case, each particle is provided with a random number generator enabling an additional simple form of symmetry breaking and particle identification (with high probability – e.g. cast ten random numbers and let them be the particle id).
3. Each particles is provided with a short-range wireless communication mechanism enabling it to broadcast messages in its neighborhood and to receive messages sent by other particles. Apart from perceiving and propagating morphogen gradients, this also enables each particles to know how many other particles are in its neighborhood (e.g. each periodically broadcasts "I am here" messages) to estimate the local density of particles

Particles do not have other capabilities other than the ones listed before. In particular:

4. They do not perceive the location (neither direction nor distance) of other particles, they do not have any kind of long range communication

mechanism, nor a global accessible data space. In other words, although a particle can perceive how many particles are in the neighborhood, it can neither perceive in which direction and at which distance a specific particles is nor in which direction a perceived morphogen gradient decreases. To follow downhill a perceived gradient, a particle has to randomly wander until it perceives it is going in the correct direction (i.e., the gradient it was following downhill is now perceived with decreased value).

5. They do not have any notion of time and cannot rely on any global synchronization mechanism. Differentiation of activities over time can occur only on the basis of changes in the locally perceived morphogen gradients.

From a methodological viewpoint, particles exploit morphogen gradients to self-organize their respective positions in space. In particular, starting from any spatial configuration of particles: *(i)* particles start diffusing specific types of morphogen gradients; *(ii)* particles react to locally perceived gradients by trying to follow gradients downhill/uphill, or by changing their activity state possibly depending on the perceived value of the morphogen gradient; *(iii)* changes in the activity state of particles can lead to inhibiting the propagation of some gradient and/or to the diffusion of new types of gradients in the system, leading back to point *(i)*. One can then apply this process several times, with new types of morphogen gradients being propagated in different phases (and exploited in some of the ways discussed in Section 2), so as to incrementally have particles self-organize into the required shape.

Readers can have an early look at the examples in Figures 2, 3, 4 and 5, to get a clue of some shapes we have been able to obtain. However, before detailing such examples, we prefer to discuss related works in the area.

Related approaches

In the last few years several approaches targeting control algorithms for multi-robot systems have been proposed, which address goals similar to ours (a good though not very recent survey can be found in (Cao 1997)).

Several proposals in the area defines distributed algorithms for pattern formation in robots exploiting the strong assumption that each robot, via visual observation, can determine the positions and movements of all other robots (Sugihara 1990; Cao 1997; Gordon 2003). Although this hypothesis makes it rather trivial to promote the formation of a variety of global patterns, the approach is hardly applicable to micro- or nano-robots/particles. Issues of scalability, battery consumption, line-of-sight problems, cost of global localization, etc. all call for a strictly local perception of the environment.

Other approaches have been proposed requiring robots only local perception, but still requiring them the capability to detect the distance and the direction of neighbor robots (Bay 1994; Fredslund 2002). The key idea is that robots, by

positioning themselves at specific distances and directions from other robots, can self-organize in a variety of regular shapes. Little is said about the possibility of making more complex shapes emerge, e.g., by making information flow from robot to robot (as in the case of morphogen gradient).

A possible alternative to promote the formation of spatial patterns in the absence of distance and direction information is to get inspiration from the way chemicals and crystals grows into self-organized regular structures. Approaches of this kinds are explored, for instance, in (Spears 1999; Jones 2002; SBIR). The general idea (with specific differences characterizing different proposals) is to exploit stateful particles capable only of sensing the internal state and the presence of other particles (either via of proximity sensing or via direct contact). Particles are deployed together in an environment and there start randomly moving. When particles keep in touch with other particles, they apply internal transition rules (based on the analysis of their own state and of that of close particles) to decide whether to "stick" to that position or continue moving. Unfortunately, the approach enables the direct programming of transition rules leading to very simple and regular patter only. More complex patterns requires very complex search heuristics to determine the appropriate set of transition rules leading to the desired global pattern. In any case, the process leads to the formation of static crystal-like and non-adaptive patterns.

Algorithms for the control of shape an motion in modular robot have been proposed exploiting an approach strictly related to ours (Bojinov 2000; Shen 2002). There, "hormones" (assimilable to morphogen gradients) are created and propagated through the modules of the robot. Robots' modules decide how to bend their actuators depending on the locally perceived hormones. The result is to have the robot modules self-organize into globally coherent shapes or into globally coherent motion patterns (i.e. gait). Still, this approach (and the hormones being propagated) is strictly coupled with the hardware characteristics of the robots and of its actuators, thus missing in generality.

The approach proposed in the Amorphous Computing project (Nagpal 2002) for the formation of origami shapes in a amorphous networks of particles is the one that most directly relates to our work. Particles, by communicating only with their local neighborhood, can self-organize into various patterns of activity by propagating morphogen gradients and changing their state according to the perceived morphogen gradients. An origami shape is specified with the primitives of a particular language, the Origami Shape Language, with which to specify how a sheet of particles must bend to achieve a desired shape. The main difference from our work is that in the Amorphous Computing approach particles cannot move altering their respective topologies. Furthermore, some particles must be in a special initial state, thus requiring an a priori differentiation among particles.

A possible objection

Algorithms to create a shared coordinate systems on the basis of mere network connectivity (i.e., requiring computational components the only capability of recognizing sending/receiving messages in the neighborhood) have been recently proposed (Nagpal 2003; Shang 2003). Since these algorithms could be executed also by particles with minimal capabilities, why don't build such a coordinate system and use it to direct robots/particles towards specific positions in the coordinate systems? Something similar, has been proposed in (Kondacs 2003; Gordon 2003) and it works as follow:

1. Particles are provided with a *f(x,y)*-like description of the 2D shape to form;
2. They set up a shared coordinate system;
3. They start moving to the closest *(x,y)* within the shape by trying to stay – within the shape - as far as possible from each other, so as to gracefully fill the shape

Although this approach theoretically allows the building of any kind of shape with accuracy, we do not commit to it for the following reasons:

1. Upon nodes' movement the coordinate system should be rebuilt every time. This can lead to a big burden and is likely to saturate the particles limited resources.
2. Building a coordinate system from mere connectivity requires very dense network of nodes (Nagpal 2003). Such a density cannot be guaranteed in every area of the network and can cause an exponential loss of precision in the coordinate system and thus in the shape to be formed.
3. Coding a complex shape in a *f(x,y)*-like representation can turn out to be really difficult and would make in any case the resulting formation of particles not robust, as in the case of particles deployed in a constrained environment where the shape would not fit.

Given that, we decided to remain with out constructive approach, even if this means finding some less direct ways to achieves specific shapes with the use of morphogen gradients.

EXPERIMENTS

Here we describe a set of experiments discussing a variety of patterns we have been able to achieve with the morphogen gradients approach. All the results have been tested on a simulator for large-scale mobile ad-hoc networks developed within our research group (Mamei and Zambonelli 2004a). In the experiments, we assumed that particles have no physical size or (which is the

same) that they can pass through each other. Simple motion mechanisms can be conceived for particles to

Barycenter

In this example, starting from a random distribution of particles, a sort of distributed leader election algorithm is executed to identify the particle closest one to the barycenter, i.e., the "center of gravity", of the whole system. Specifically, given *n* particles in space, the barycenter is that particle that minimizes the sum of the distances to all the *n* particles.

Detecting the barycenter of the system is very important for pattern formation, in that it identifies a reasonable point to which to refer to start subsequent shape formation activities. Morphogen gradients, expressing the approximate distance from their respective sources, enables the definition of a simple and intuitive algorithm for the identification of the barycenter.

The algorithm: Each and every particle propagates a BARYCENTER gradient whose value increases by one at each step. Each particle senses BARYCENTER gradients propagated by all the other particles as they arrive, and adds their values together, call the resulting value *totGradients*. *totGradients* is the sum of distances to all the other particles. Therefore, the particle having the minimum *totGradients* is the barycenter. Since *totGradients* decreases monotonically to the barycenter, each particle can understand whether it has *totGradients* minimum or not, by simply comparing its value with the neighbors' ones. If no neighbors has a lower value of *totGradients*, the particle is the barycenter.

The commented pseudo code of the algorithm is in Figure 2. We emphasize the algorithm does have a well-defined termination point. Simply, each particle keeps on waiting for the income of new BARYCENTER gradients, to evaluate over and over whether it is the barycenter of not. Eventually, the algorithm converge, and particles will no longer receive any new gradient. The evolution of a sample simulation of the barycenter election algorithms is reported in Figure 3. In this figure (as well as in following simulation figures). It can be noted that, during the process, some particles may temporarily recognize themselves as barycenter. However, eventually, a single barycenter remains.

Slight modifications of the algorithm can be defined to elect two particles, aligned around the barycenter at a specific distance from each other, as well as to identify particles on the border of the structure (and have them only to propagate the BARYCENTER gradient, to improve performances).

```
barycenter = false // I am not the barycenter
totGradients = 0 // sum of values received
// propagate the BARYCENTER gradient
injectGradient(BARYCENTER, uniqueNumber)
// infinite cycle
while (1)
// blocks waiting for incoming gradients
Gradients[] = getGradients(BARYCENTER);
// sum the value of all gradients
forall Gradients
totGradients+=Gradients[i].value;
   end forall
// analyse totGradients value of neighbors
count = 0 // a simple counter
forall neighbor particles
if neighbor[i].totGradients > totGradients
count++
end if
end forall
// if all neighbours have a greater value
if numNeighbors = count
   barycenter = true // I am the barycenter
   else
   // otherwise I am not the barycenter
   barycenter = false
   end if
   end while
```

FIGURE 2. Pseudo-code of the barycenter algorithm.

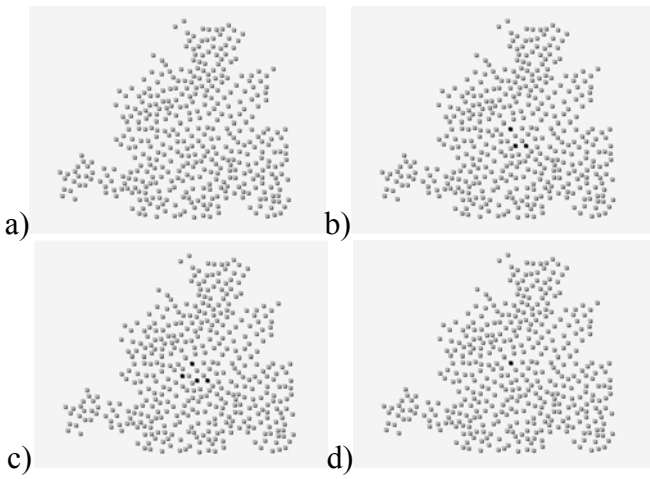

FIGURE 3. Different stages of the establishment of the barycenter in a cloud of randomly distributed particles. As the system evolves, some particles (in black) may temporarily consider themselves the barycenter. Eventually, a single barycenter is left.

Circle

In this example particles run a distributed algorithm to cooperatively assume a circle shape.

The Algorithm: Each particle runs the barycenter algorithm described in the previous section. The resulting barycenter particle will serve as a the circle center. This particle propagates a CIRCLE gradient which increases its value by one at each step. All the other particles sense the gradient. If they sense a value greater than R (intended circle radius) they move following downhill the CIRCLE gradient. Eventually, all particles outside the intended circle radius will collapse toward it.

```
if particle == BARYCENTER
        injectGradient(CIRCLE)
end if
while(1)
  if getGradient(CIRCLE).value > R
        moveDownhill(CIRCLE)
    end if
end while
```

FIGURE 4. Pseudo-code of the circle algorithm

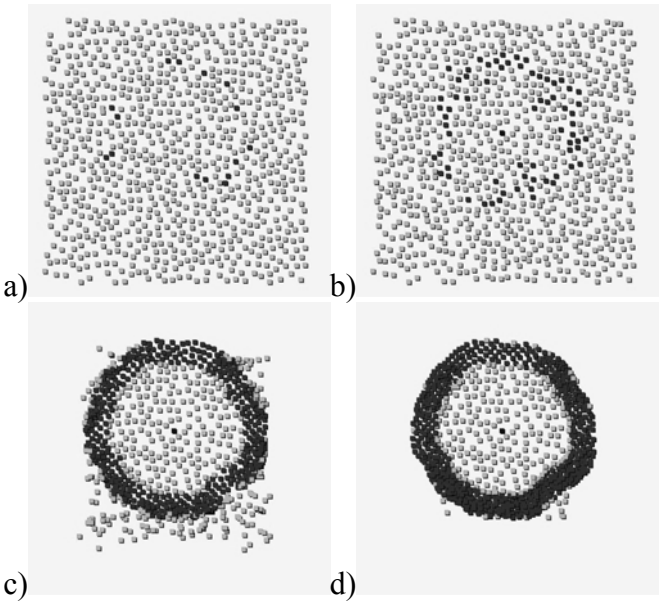

FIGURE 5. Different stages of the circle formation. As the barycenter start propagating the circle gradient, some particles (in black) already recognize themselves as being at the correct distance from the center and do not move; the other particles gradually collapse toward the circle circumference.

The pseudo code of the ring algorithm is in Figure 4. Also in this case, the algorithm does not end: simply, the particles that found themselves inside the circle will stop keep on moving. The result of a sample simulation is in Figure 5.

It is important to note that, despite the fact that during the execution of the barycenter algorithm some particles may temporarily consider themselves the barycenter, this causes not harm to the circle algorithm. Simply, these particles will temporarily diffuse a spurious CIRCLE gradient.

Another remark relates to the fact that, as stated in Section 3.1, our particles cannot sense in which directions a gradient decreases. Therefore, a particle have to randomly chose a direction to move and, if the particle senses that the gradient of interest does not decrease – wrong guess – it can simple invert its direction. The key drawback of this technique (in addition to the introduction of a small overhead, as discussed in Section 5) is that it makes it possible for some particles to get lost, i.e., get disconnected from the network without any further information about where the rest of the particles are. However, since these unlucky events are extremely rare, and since individual particles are not important, this causes no harm.

As an additional note, we emphasize that the execution of the circle algorithm in the presence of two barycenter enables the forming of elliptic shapes.

Ring

In this example, particles run a distributed algorithm, simply extending the circle one, to cooperatively assume a ring shape.

The Algorithm. Once the barycenter algorithm has run, and the circle has been formed (better: once particles on the circumference of the forming circle recognize their being in the current position), the particles on the circumference start propagating a RING gradient which increases its value by one at each step. This RING gradient, which also propagates to the inner parts of the circles, attracts particles towards the circumference, thus emptying the inside of the ring. The thickness of the ring can be tuned by having particles stop following the RING gradient when it reach a value of T, where T will consequently be the thickness of the ring.

The pseudo code of the ring algorithm is in Figure 6. The result of a sample simulation is in Figure 7.

As for the case of the circle, we outline that by executing the ring algorithm in the presence of two barycenter “8”-like shapes can be obtained.

```
  if particle == CENTER
        injectGradient(CIRCLE)
  end if
if getGradient(CIRCLE).value == R
        injectGradient(RING)
end if
if getGradient(CIRCLE).value != R
   while getGradient(RING).value >= T
         moveDownhill(RING)
   end while
end if
```

FIGURE 6. Pseudo-code of the ring algorithm.

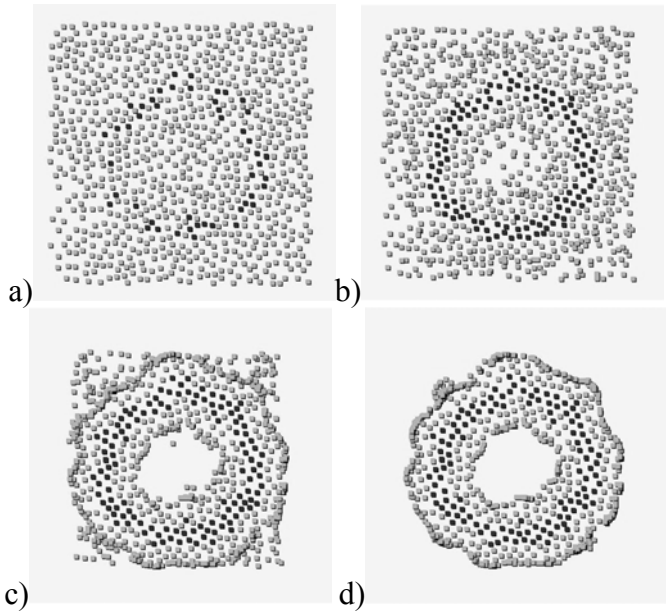

FIGURE 7. Different stages of the ring formation. As the particles at the correct distance from the barycenter self-recognize their being there, they start injecting a gradient that attracts toward them inner particles.

Making Lobes

In this experiment, we tried to break the circular symmetry of previous experiments and let lobes emerge in the global shape. The overall idea is to

exploit particles density as the source to break the symmetry. For instance, when forming a circle, particles start to collapse toward the circle itself. If the number of particles compared to the circle size is very high, then the perimeter of the circle will be very crowded. Our idea is to force particles in very crowded areas to rearrange their positions so as to stay more separate form each other (remember that our particles can sense how many other particles are in the neighborhood). This process ends up in a slight deformation of the circle (i.e., in the emergence of small "lobes") in those part of its circumference where an excess of particles are accumulating. This small emergent lobes can be amplified via an additional mechanism of morphogen gradient inhibition that, in turn, makes larger lobes emerge.

To this end, and with reference to the circle, it is worth noting that the emergence of the circle shape directly derives from having the CIRCLE gradient spreads in every direction uniformly. In this way, all the particles sensing the value R of the CIRCLE gradient ends up in being almost equidistant form the center or, when the density is taken into account, in the circumference of a circle with small lobes However, if one make the CIRCLE gradient increase its value slower in zones of high density then, in these zones, the gradient would reach the value R farther from the source. Particles, following that gradient, would not dispose on a circle, but on a circle with a lobe, where the lobe would correspond to the place in which the gradient reaches value R farther. This leads to the following simple algorithm.

The Algorithm. Particles runs the CIRCLE algorithm and, upon receiving the CIRCLE gradient, have to re-propagate it. However, before doing that, particles sense the number of other particles in their neighborhood. If the number of particles in the neighborhood exceeds a specified threshold (*criticalDensity1*), the particle sets the rate at which the field increases to 0. This increasing rate will be reset to the default value of one when the density falls below another specified threshold (*criticalDensity2*).

The pseudo code of this algorithms is in Figure 8. The result of a sample experiment is in Figure 9. It is worth outlining that the algorithm does not enable to predict where in the circle lobes will form, and how may lobs will eventually form. This is an emergent characteristic of the systems, that critically depends on two non controllable factors: the initial disposition of particles and the outcome of the random movement of particles towards the center. These two factors will affect the way particles will accumulate around the circle and, thus, the density of particles.

```
  if particle == CENTER
        injectGradient(CIRCLE)
end if
if getGradient(CIRCLE).value > R
        moveDownhill(CIRCLE)
else if getGradient(CIRCLE).value == R
/* making particles escape from crowd make
   small lobes emerge */
        moveAwayFromCrowd()
     end if
end if
// the following, makes even large lobes emerge
if numNeighbors > criticalDensity1
     // set the gradient increasing rate to 0
        setGradientAddValue(CIRCLE, 0)
end if
if numNeighbors < criticalDensity2
     // restore default increasing rate
        setGradientAddValue(CIRCLE, 1)
end if
```

FIGURE 8. Pseudo-code of the lobes algorithm

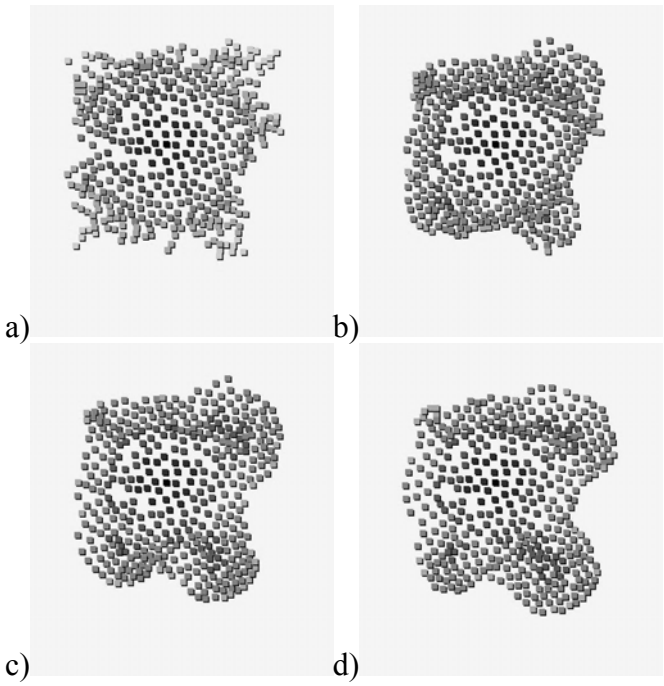

FIGURE 9. Different stages of the circle with lobes formation. The particles in the circle that detect a high density of particles, inhibit the propagation of the circle gradient, thus leading to the formation of lobes.

Polygons

The emergent phenomena of lobes in the previous section is interesting. Still, it would be important to have a way of controlling such emergent behavior. In a further set of experiments we tried to enrich the algorithm to control the number of lobes to be created so as to obtain regular polygon shapes (e.g. triangles exagons, etc.).

```
if particle == CENTER
   injectGradient(CIRCLE)
end if
if getGradient(CIRCLE).value == R
   while not hasGradient(ELECT)
     if getGradient(ELECT).value > L
       iAmLeader();
       injectGradient(ELECT)
       setGradientAddValue(CIRCLE, 0)
     end if
     else if nextRandom() > T
       iAmLeader();
       injectGradient(ELECT)
        setGradientAddValue(CIRCLE,0)
     end if
   end while
end if
else if getGradient(CIRCLE).value > R
        if getGradient(ELECT).value == 1
          setGradientAddValue(CIRCLE, 0)
        end if
       end if
end if
while(1)
 if getGradient(CIRCLE).value > R
   moveDownhill(CIRCLE)
 else if getGradient(CIRCLE).value == R
           moveAwatFromCrowd ()
 end if
end while
```

FIGURE 10. Pseudo-code of the polygons algorithm.

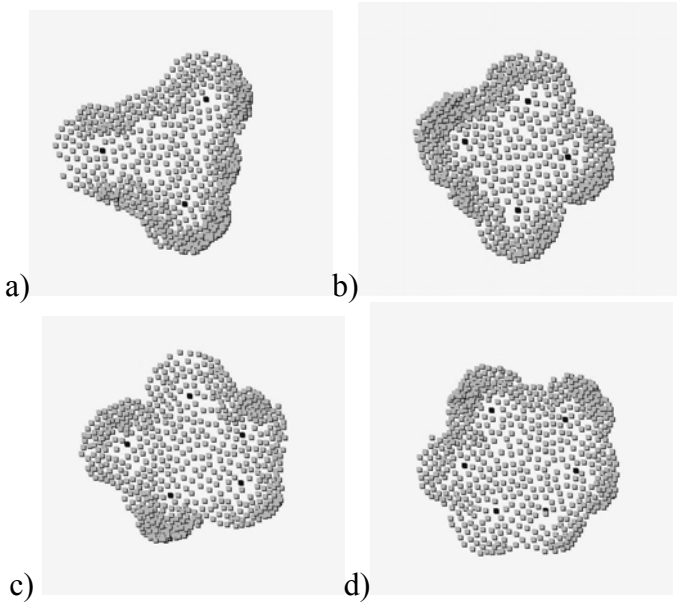

FIGURE 11. Different polygon shapes obtained by multiple lobes: from triangle to hexagon.

The idea to control the number of lobes is again rooted in a leader election mechanism. We want to design an algorithm to elect *n* leaders on the circumference of the circle. These leaders must be and equidistant from one another. Once this has been accomplished, the leaders will be in charge of adopting the trick described in the previous section, i.e., avoiding to increase the value of the CIRCLE gradient being re-propagate, so as to force the emergence of *n* lobes equidistant from one another and, consequently, leading to a nearly regular *n*-polygon shape.

The Algorithm. (i) each node runs the circle algorithm, *(ii)* once a particle on the circumference of the forming circle recognize their being in the current position, it start casting random numbers; *(iii)* each node casting a number greater than a specified threshold becomes a leader – the threshold (T) is chosen so that it is very unlikely that two nodes becomes leaders shortly one after another; *(iv)* the leader starts propagating an ELECT gradient, that propagates only in the circle perimeter region (i.e., particles that are not on the RING inhibits its propagation); *(v)* nodes receiving the ELECT gradient, stop casting random numbers and if the received gradient value overcomes another specified threshold *L,* they become leaders on their turn; *(vi)* each leader sets the ELECT gradient value to 0 and continues its propagation; *(vii)* once that ELECT gradient is fully propagated there should be almost *(circle-length)/L* equidistant leaders on the circle. Thus *L* is a parameter controlling which polygon will emerge.

The pseudo code of the algorithm is in Figure 10. The results of some sample experiments, showing that the algorithm makes possible to control the emergence of regular polygon shapes are in Figure 11.

PERFORMANCE EVALUATION

Validating our approach in terms of performances basically amounts at verifying *(i)* that it is reasonably scalable and *(ii)* that the assumption of minimal capabilities is not too much penalizing. The results are presented in Figure 12 and refer to an abstract, virtual time. Referring to an actual time requires assumptions on hardware and motion speed of particles that are not relevant to our purposes here.

With regard to point *(i)*, we have verified that our approach scales linearly with the number of particles. That is, the time for a group of randomly placed particles to reach a stable configuration increases linearly with the number of particles.

With regard to point *(ii)* we have limited our attention at verifying that the impact of the assumption of non-directional sensing is not too penalizing. We have compared the time required by particles to self-organize into specific shapes with and without the capability to perceive the direction to which a gradient is decreasing. The result is that the overhead due to have particles randomly wander to properly detect in which direction to go is very limited independently of the specific shape to be obtained and independently of the overall number of particles in the system.

Comparing our approach with other approaches based on particles/robots with much powerful capabilities (i.e., global sensing and *a-priori* knowledge of the geometrical shape to obtain) is simply meaningless. In fact, the notably better performances that these approaches would obviously exhibit would be obtained at the price of notably increasing the complexity and thus price of particles.

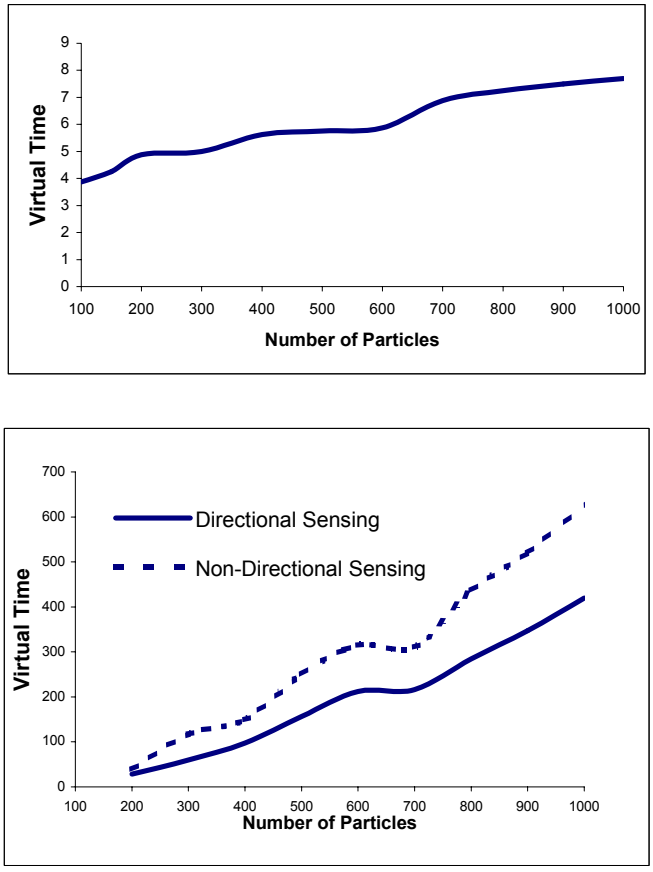

FIGURE 12. Performance evaluation. (top) Identification of the barycenter. The time required to elect a barycenter grows linearly with the number of particles. (bottom) Ring formation. The time required for the shaping of particles into a ring grows linearly with the number of particles. Furthermore, the time overhead of directional sensing is very limited if compared with the time required to form a ring in the case of particles capable of directional sensing.

CONCLUSIONS AND OPEN DIRECTIONS

The morphogen gradient approach enables the effective formation of a variety of complex shapes in computational particles with minimal capabilities. In particular, particles can self-organize in spatial patterns that – up to now – were thought to require much more powerful capabilities.

Despite these promising results, a number of open directions are still to be investigated. Those current of interest to our research group include:

experiencing differentiation of activities and global coordination based on cellular automata inspired approach; trying to define a simple and modular programming model, enabling designer and programmers to enforce in a modular and compositional way a variety of complex pattern; trying to achieve – other than the formation of static patterns – coherent motion gaits in particles; building some hardware prototype for particles and validating our approach in the real world.

REFERENCES

Bay, J. S., and C. Unsal. 1994. Spatial Self-Organization in Large Populations of Mobile Robots. *IEEE International Symposium on Intelligent Control,* Columbus (OH).

Bojinov, H., A. Casal, and T. Hogg. 2000. Emergent Structures in Modular Self-Reconfigurable Robots. *IEEE Intl. Conf. on Robotics and Automation,* San Francisco (CA).

Bonabeau, E. 1997. From Classical Models of Morphogenesis to Agent-based Models of Pattern Formation, *Artificial Life*, 3:191-211.

Cao, Y. U., A. S. Fukunaga, A. B. Kahng, and F. Meng. 1995. Cooperative Mobile Robotics: Antecedents and Directions. *IEEE Int. Conf. on Intelligent Robots and Systems*, Yokoama (J).

Coore, D. 1999. Botanical Computing: A Developmental Approach to Generating Interconnect Topologies on an Amorphous Computer. PhD Thesis, MIT.

Day, S. J., and P. A. Lawrence. 2000. Morphogens: Measuring Dimensions: the Regulation of Size and Shape. *Development* 127:2977-2987.

Estrin, D., D. Culler, K. Pister, and G. Sukjatme. 2002. Connecting the Physical World with Pervasive Networks. *IEEE Pervasive Computing*, 1(1):59-69.

Fredslund, J., and M. J. Mataric. 2002. A General Algorithm for Robot Formations Using Local Sensing and Minimal Communication. *IEEE Transactions on Robotics and Automation*, 18(5):837-846.

Gordon, N., I. A. Wagner, and A. M. Bruckstein. 2003. Discrete Bee Dance Algorithm for Pattern Formation on a Grid. *IEEE International Conference on Intelligents Agents Technologies*, Toronto (CA).

Guo, Y., G. Poulton, P. Valencia, and G. James. 2004. Designing Self-Assembly for 2-Dimensional Building Blocks. *Engineering Self-Organizing Applications*, LNCS No. 1977, Springer Verlag.

Jones, C., and M. J. Mataric. 2003. From Local to Global Behavior in Intelligent Self-Assembly. *Submitted to the IEEE Conference on Robotics and Automation*.

Kondacs, A., 2003. Biologically-inspired Self-Assembly of 2D Shapes Using Global-to-local Compilation. *International Joint Conference on Artificial Intelligence (IJCAI)*, Acapulco (MX).

Lawrence, P.A. 1992. The Making of a Fly: the Genetics of Animal Design. Blackwell Science U.K.

Mamei, M., and F. Zambonelli. 2004. Programming Mobile and Pervasive Computing Applications with the TOTA Middleware. *IEEE Conference on Pervasive Computing and Communications*, Orlando (FL).

Mamei, M., and F. Zambonelli. 2004. Co-Fields: a Physically Inspired Approach to Distributed Motion Coordination. *IEEE Pervasive Computing*, to appear.

Nagpal, R. 2002. Programmable Self-Assembly Using Biologically-Inspired Multirobot Control. *ACM Joint Conference on Autonomous Agents and Multiagent Systems*, Bologna (I).

Nagpal, R., H. Shrobe, and J. Bachrach. 2003. Organizing a Global Coordinate System from Local Information on an Ad Hoc Sensor Network. *Information Processing in Sensor Networks*, LNCS No. 2643, Springer Verlag.

Poor, R. 2001. Embedded Networks: Pervasive, Low-Power, Wireless Connectivity. PhD Thesis, MIT.

Pister, K. 2003. Invited Plenary Talk. *The 23rd International Conference on Distributed Computing Systems*, Providence (RI).

SBIR Project. A Cooperative Multi-Robot Control Architecture. Report No. NAS8-01168, Dynamic Concepts Inc.

Shang, M., W. Ruml, and Y. Zhang. 2003. Localization from Mere Connectivity, *ACM Conference on Mobile ad-hoc Computing and Networking*, Annapolis (MD).

Shen, W. M., B. Salemi, and P. Will. 2002. Hormone-Inspired Adaptive Communication and Distributed Control for CONRO Self-Reconfigurable Robots. *IEEE Transactions on Robotics and Automation*, 18(5):1-12.

Spears, W. M., and D.F. Gordon. 1999. Using Artificial Physics to Control Robots. *IEEE International Conference on Information, Intelligence and Systems*.

Sugihara, K., and I. Suzuki. 1990. Distributed Motion Coordination of Multiple Mobile Robots. *IEEE Int'l Symp. on Intelligent Control*, Philadelphia (PN).
Wolpert, L. 1998. Principles of Development. Oxford University Press (UK).