Gravity points in potential-field approaches to self organisation

Aaron Sampson
*School of Information Technology and
Electrical Engineering*
The University of Queensland
Australia
Email: aaronsampson@yahoo.com.au

Graeme Smith
*School of Information Technology and
Electrical Engineering*
The University of Queensland
Australia
Email: smith@itee.uq.edu.au

Abstract—**One approach to the coordination of the movement of autonomous agents in self-organising systems is to use virtual potential fields, analogous to potential fields found in physics. Agents act as charged particles generating fields which other agents react to by ascending or descending the gradients of these fields, or by moving along equipotential lines. This approach has been suggested for free-flight air-traffic control, urban traffic management and modular robotics, among other applications. In this paper, we extend this approach by the addition of *gravity points* which overlay the agent-centric potential fields with additional fields for attracting or repelling agents. We demonstrate how the approach can be used to improve the efficiency of an algorithm for the self configuration of modular robots.**

Keywords-**self organisation, potential-field models, modular robotics**

I. Introduction

Central to the design and implementation of self-organising systems are the patterns of interaction used by the agents to obtain the overall system goals. In their overview of the engineering of multi-agent systems, Zambonelli and Omicini [12] place such interaction patterns at a *meso* level of observation between the *macro* level of global system behaviour and the *micro* level of the behaviour of individual agents. A number of such meso-level patterns have been presented in the literature. These include patterns inspired by mechanisms present in nature, such as the foraging of ants [2], and society, such as gossip [1].

An important physics-inspired interaction pattern is that of virtual potential fields [5], [4]. In this pattern, agents act as charged particles generating fields which other agents react to by ascending or descending the gradients of these fields, or by moving along equipotential lines. This approach has been suggested for various applications including free-flight air-traffic control by Eby [3], urban traffic management and swarm systems by Mamei *et al.* [5], [4], and self-configuration of modular robots by Støy and Nagpal [9].

In this paper, we extend this pattern by a notion of *gravity points*. These produce additional fields which similarly affect the motion of agents but are not centered on an agent. Specifically, we show how such fields can be used to improve the efficiency of a self-configuration algorithm based on that of

Støy and Nagpal [9]. In order to implement gravity-point fields we use vector-based gradients which provide an agent with its relative position from the gravity point at the centre of the field.

In Section II we provide a brief description of the self-configuration algorithm before introducing our notion of vector-based gradients in Section III. In Section IV we introduce the notion of gravity points and how they can be incorporated with the algorithm. In Section V we provide simulation results showing the effectiveness of gravity points on two examples. We conclude in Section VI.

II. Self configuration using potential fields

In this section, we give a brief description of an algorithm for self configuration of modular robots. The robots, referred to hereafter as 'atoms', autonomously form 3-dimensional shapes through local interaction. The description is based on a family of solutions to this problem devised by Støy and Nagpal [9], [10], [8], [7] and has been formally modelled and verified by Smith and Sanders [6].

A. Atoms

The system comprises a connected mass of atoms: it is initially connected and remains so by establishing a *connection gradient* which is used to restrict the movement of atoms (see [9] for details). Atoms in the system are capable only of local communications with immediate neighbours, with which they have physical contact. They are also capable of movement over or around their neighbours, and of storing information including:

- a representation of the shape to be formed (referred to as the *target*);
- their position (coordinates) within the target, should they reach it;
- their distance from the nearest atom already in the target which requires a neighbour.

Initially, only a single atom, called the *seed* atom, stores the target and its position within it. It uses this information to decide at which of its neighbouring positions it needs atoms, and by creating a (virtual) potential field (as described below) attracts other atoms to these positions. The atoms
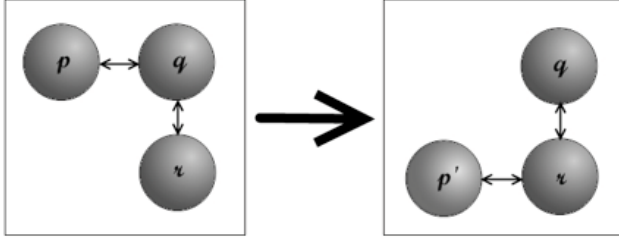
Figure 1. Following the gradient.

Figure 2. Vector-based recruitment gradient values.

which first reach the seed atom are given the target and their position in it by the seed atom. They remain fixed at these positions and themselves create potential fields to attract the neighbours they require. This process continues until all atoms are configured in the target shape.

B. Potential fields

An atom creates a potential field by setting its value for the distance to the nearest atom requiring neighbours to zero, and sending this value to each of its neighbours. When an atom receives such a value, if it does not have a value for the distance to the nearest atom requiring neighbours, or has a value greater than that received, it sets its value to one more than the value received. It then sends this new value to all of its neighbours. If, on the other hand, it already has a value less than or equal to that received, it ignores the received value.

This results in atoms storing a value representing the *shortest* distance to an atom requiring neighbours. If they are not already in a position within the target, they follow the gradient of the potential field to successively smaller values. In this way, they can reach the source of the potential field and join the target shape.

Figure 1 illustrates the movement of an atom along a gradient. Each atom in the figure is labelled with the gradient strength. Such a move is possible only when r is less than q.

When a gradient to a particular atom is no longer required (since the atom has all of its neighbours), the gradient must be removed from the system. This is necessary to enable other gradients to be followed. Hence, the potential fields dissipate after a given time unless explicitly renewed by the atom that created them.

III. VECTOR-BASED GRADIENTS

To enable the creation of potential fields which do not emanate from an atom, the simple integer-based gradients of the previous section do not suffice. If we want an atom to create such a field, it needs to indicate to its neighbours both the distance and direction to the source. Hence to enable the implementation of gravity-point potential fields, we first introduce vector-based gradients to the basic algorithm.
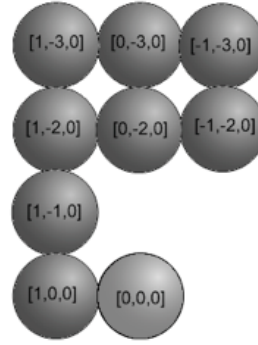
It should be noted that vector-based gradients have been considered by Støy for improving the efficiency of the basic algorithm [7]. The integer-based gradient is extended with a unit vector indicating the local direction of the gradient. This vector is calculated by taking the vector from the neighbour which is closest to the source of the field. However, the local direction of a gradient alone is not sufficient for describing the source of a potential field — more information is required.

Our approach is to represent the gradient by its relative position in 3-dimensional space from the source. Figure 2 shows an atom which has set up a potential field for which it is the source. Its vector-based gradient value is $[0, 0, 0]$. These digits represent the moves required in the *x*, *y* and *z* directions to reach the field's source.

As with the algorithm described in Section II, once an atom has established that it requires neighbouring atoms it will broadcast its own gradient value $[0, 0, 0]$ to all of its neighbours. A neighbour atom will first determine if the gradient value is less than its current gradient value, if any. If the distance is greater than or equal to its own it will disregard the received value, otherwise it will set its value to the received value updated according to the direction of the broadcasting neighbour. For example, the neighbour next to the requesting atom shown in Figure 2 has the value $[1, 0, 0]$ which is a result of adding 1 to the *x* value of the gradient value. The value [1,0,0] indicates that the atom has to move 1 step in the positive *x* direction to reach the field's source.

In a similar fashion, an atom could set up a potential field for a point which is, say, 5 steps in the *x* direction and -3 steps in the *y* direction from its position by setting its gradient value as $[5, -3, 0]$ rather than $[0, 0, 0]$ and broadcasting to its neighbours as above. In the next section, we show how this idea can be used to provide a more efficient approach to self-configuration.

IV. GRAVITY AND ANTI-GRAVITY POINTS

The self-reconfiguration algorithm described in Section II is based on growing a target shape from an initial seed atom. The seed atom creates a potential field which other atoms follow to become its neighbours. These atoms in turn create potential fields to attract other atoms. The result is that all atoms end up moving towards the position of the seed, many not becoming neighbours of the atoms at the centre of the potential field whose gradient they are following.

As well as causing unwanted movement of atoms, this also potentially moves atoms away from spaces where they will be eventually needed. To make the self-configuration process more efficient, we would like to move atoms not only towards the seed atom and its neighbours which are already placed in the target, but towards other positions where we know atoms will be required.

Since the seed knows the target it can decide on these other positions and set up additional potential fields (using the vector-based approach of the previous section) to attract atoms towards them. We call such positions *gravity points* due to their 'gravitational pull' on atoms. Atoms are attracted to a particular space by the use of gravity points, in a similar fashion to the way they are attracted to atoms requiring neighbours.

Figure 3 displays a series of images from a simulation showing a the movement of atoms from an initial configuration to a gravity point that is twenty positions to the right of the seed atom. In this simulation, there was no potential field centred on the seed. For configuring to a target shape, however, the gravity point fields would overlay that of the seed atom. Since, as described in Sections II and III, an atom follows the gradient of the potential field whose centre it is nearest, this would result in those atoms nearest the target moving towards it, and those closer to gravity points moving towards them.

A. Gravity point gradients

Gravity points are generated by the seed atom before the self-configuration process begins. Although an algorithm could be developed to optimise their placement in the most densely populated parts of the target, here we take a simpler approach. The seed atom generates the gravity points by randomly selecting a set of points from the cube that encapsulate the boundaries of the target shape.

If a gravity point is initially created in a location that is heavily populated by target positions then we would like its gravitational pull to be stronger then a gravity point located in a sparsely populated position of the target. Once the gravity points have been selected, the seed determines their strength from the surrounding target positions.

To realise these strengths within the self-configuration algorithm we propose two approaches. Firstly, we could associate an integer strength with gradient values. An atom receiving a value would need to take into account both the
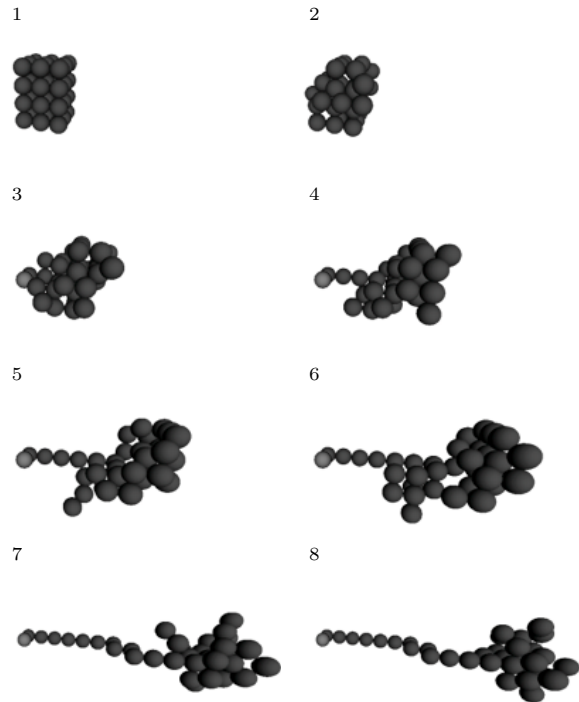

Figure 3. Atoms swarming around a gravity point.

strength and the distance to the centre of the potential field when deciding whether to replace its current gradient value with the received one, or to ignore it.

A second approach which changes the standard algorithm less, would be for the seed to use the calculated strength of the gravity point to determine how many times to reestablish the gravity point's potential field. Gravity-point potential fields dissipate after a given time in the same fashion as those centred on atoms. The difference is, however, that the atom-centric fields are periodically reestablished until all neighbouring positions are filled — there is a fixed number of atoms required. With gravity-points fields there is no fixed number of atoms required, nor is there any way of determining how many atoms have been attracted by the field. Hence, the seed atom needs a strategy for determining how many times to reestablish the field.

If strengths were associated with the gradient values of fields as proposed above, each gravity-point field would be reestablished the same number of times. Instead of this approach, however, we could simply reestablish the gravity-point fields a number of times proportional to the field's strength. Hence, a gravity point in a densely populated area of the target would have its field reestablished more times than one in a sparsely populated area of the target. This would provide it with the opportunity to attract more atoms as desired.

Figure 4 shows an initial configuration of atoms which are required to form the shape of a chair. The figure shows the
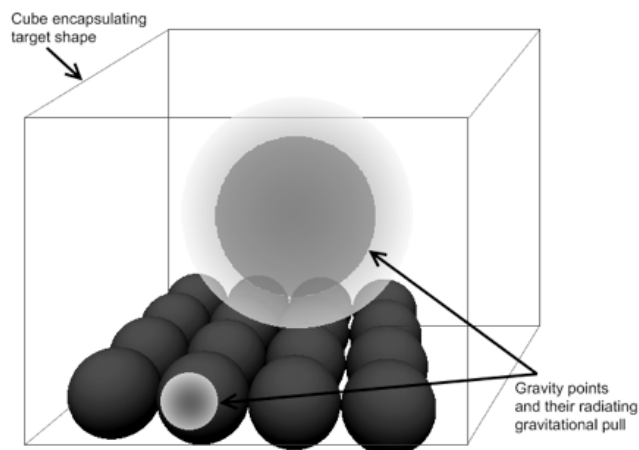
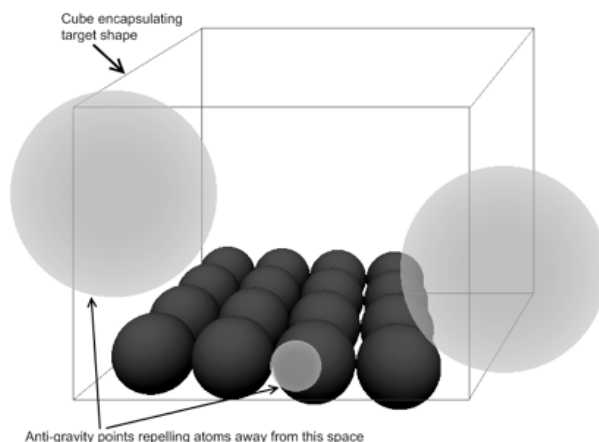Figure 4. Gravity points created from initial atom configuration.

Figure 6. Anti-gravity points created from seed atom repel atoms away.
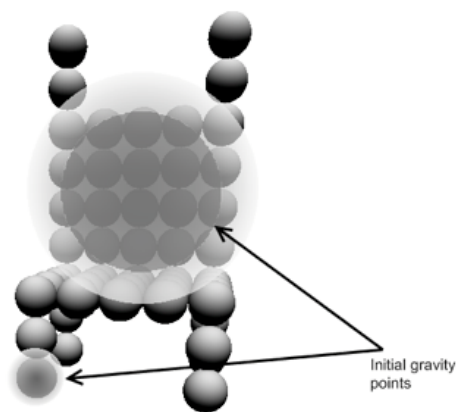

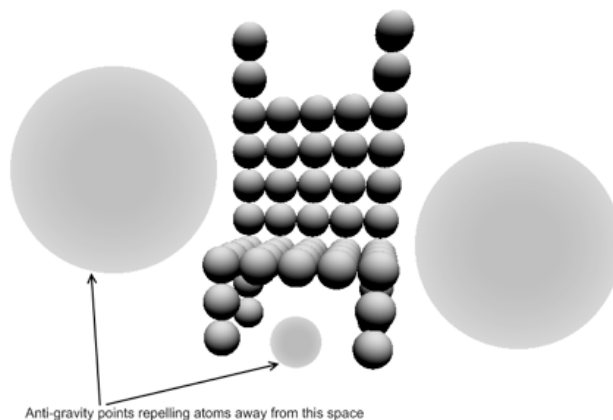Figure 5. The final chair configuration still showing initial gravity points.

Figure 7. The final chair configuration with initial anti-gravity points.

cube from which random points are selected and depicts two (randomly chosen) gravity points as spheres whose diameters are directly proportional to their gravitational pull. The larger sphere pulls more atoms in its direction because a larger number of target positions reside there. In Figure 5 the final target configuration and the initial gravity points that were created by the seed atom are shown. The result of establishing the larger gravity point is that atoms can move to the space where they are likely to be configured before the target has grown to include atoms in that space.

B. Anti-gravity point gradients

In addition to gravity points, we could also consider *anti-gravity points*. These would repel atoms rather than attracting them like gravity points. To establish anti-gravity

points the seed would need to be able to locate positions that have little or no surrounding target positions. Gravity points and anti-gravity points could then be used in conjunction to move atoms closer to target positions.

Anti-gravity points would need to prevent atoms from moving closer to their centre and, like gravity points, would have a strength related to their repulsive force. If, for example, we wish to configure to the target shape used in Figure 5, possible anti-gravity points are shown in Figure 7. Figure 8 shows the final chair configuration with the initial anti-gravity points.

To realise the strength of anti-gravity points, the first proposal for doing so for gravity points could be adopted. That is, we could associate a strength with every gradient value. In the case of anti-gravity points, this strength would

be negative so that atoms would ascend the gradient away from the anti-gravity points, rather than descend it towards them. Other approaches to realising anti-gravity points are targeted as an area of future work.

C. Coarse and fine-mode configuration

The use of gravity and anti-gravity points splits the movement of atoms into two modes. *Coarse-mode configuration* results from atoms responding to the potential fields created by gravity and anti-gravity points. Its purpose is to move atoms closer to where the majority of positions in the target lay. It would only be enabled for the first part of the configuration process since, even for points with a strong attractive or repulsive force, the seed will eventually stop reestablishing the potential field.

Fine-mode configuration is the standard mode of configuration where atoms attract neighbours using potential fields. It is enabled throughout the entire configuration process, terminating when all atoms are in the target shape. Initially, when both coarse-mode and fine-mode configuration are possible, atoms follow the gradient which leads them to the nearest centre of a potential field.

Initial experiments using a custom-built simulator have validated the expectation that the number of moves required to get from one configuration to another is decreased when using this combination of coarse and fine-mode configuration. These experiments used the simple approach of choosing random points in the cube enclosing the target shape for the gravity points. Obviously better algorithms could be used to increase the improvement in efficiency further. The results of the experiments are detailed in the following section.

V. SIMULATION RESULTS

To verify the effectiveness of the gravity-points approach, we performed some experiments using a custom-built simulator. Experiments were run on two self-configuration tasks using the standard algorithm (without gravity points) and then repeated using the extended algorithm with gravity points. Anti-gravity points were not used in these experiments.

The number of atoms in each task were limited by the need for the target shapes to be encoded manually in the simulator. However, the experiments were still able to validate the expected improvement in efficiency when using gravity points. Running further experiments with larger numbers of atoms is left for future work.

The target for the first task was a shape comprising 52 atoms representing a chair. For the second task, the target was a shape comprising 42 atoms resembling an airplane. The tasks were repeated ten times using different randomly generated starting configurations. In all experiments the simulator was able to converge to the desired configuration as expected.

Iteration	Task 1 Time steps	Task 1 Moves	Task 2 Time Steps	Task2 Moves
1	4400	806	1207	386
2	3566	667	730	301
3	4579	731	999	300
4	3408	631	430	178
5	4013	701	426	151
6	3401	596	492	156
7	3381	594	768	326
8	2990	491	391	134
9	2806	540	520	250
10	3300	640	1101	356
Average	3584.4	639.7	706.4	253.8

Table I
RESULTS FOR THE STANDARD ALGORITHM SHOWING THE TIME STEPS AND NUMBER OF MOVES TO ACHIEVE A CONFIGURATION.

Iteration	Task 1 Time steps	Task 1 Moves	Task 2 Time Steps	Task2 Moves
1	3013	510	906	304
2	2670	425	660	251
3	3181	594	575	250
4	3001	496	413	128
5	2806	440	426	132
6	2900	540	297	115
7	3099	606	553	279
8	2966	567	391	119
9	3279	631	520	200
10	2999	505	899	236
Average	2991.4	531.4	564	201.4

Table II
RESULTS FOR THE GRAVITY-POINTS ALGORITHM SHOWING AN IMPROVEMENT IN THE TIME STEPS AND NUMBER OF MOVES TO ACHIEVE A CONFIGURATION.

For each experiment, the time steps taken and the total number of moves performed by atoms were recorded. The time steps needed to reconfigure a target include steps for broadcasting and dissipating gradients as well as for moving atoms.

Table I shows the results for the standard algorithm without gravity points. The average time steps and average number of moves for the first task were 3584.4 and 639.7 respectively. For the second task, they were 706.4 and 253.8 respectively. The time steps needed to complete a task grow approximately linearly with the number of atoms which is accordance with the results of Yim *et al.* [11].

Table II lists the results for the extended algorithm using gravity points. The average time steps and average number of moves for the first task were 2991.4 and 531.4 respectively. For the second task, they were 564 and 201.4 respectively.

The results from these experiments show an improvement in the number of time steps taken and moves to complete a target configuration. A decrease of more than 100 moves is seen for the simulation of the chair configuration and a decrease of about 50 moves for the simulation of the airplane configuration. An improvement of around 16% in both cases.

The time steps have also reduced when compared with the standard algorithm. A decrease of almost 600 time steps is seen for the the simulation of the chair configuration and around 140 time steps for the simulation of the airplane configuration. An improvement of around 20% in both cases.

VI. Conclusion

Virtual potential fields have been proposed to coordinate the movement of autonomous agents in a number of application areas. These fields generally emanate from the agent which creates them, and other agents use them to locate or avoid this agent.

In some applications, it might be useful to locate or avoid other spatial positions. To enable this, we have proposed the notions of gravity and anti-gravity points. Fields emanating from these points can be used by agents in the same way as fields emanating from other agents.

We have applied the approach to an algorithm for the self configuration of modular robots, and validated, through simulation, that it leads to a more efficient process. To enable an agent to set up a gravity-point field, we extended the agent's awareness of a field from a single integer denoting the distance to the field's centre to a vector denoting the relative 3-dimensional position of the agent from the field's centre. We used a simple scheme for determining the location of gravity points and further improvements could be gained by a more sophisticated algorithm. Developing such an algorithm is one area of future work. Another is the application of the gravity-points approach to other self-organising systems.

References

[1] O. Babaoglu and M. Jelasity. Self-* properties through gossiping. *Philosophical Transactions of the Royal Society A*, 366(1881):3747–3757, 2008.

[2] O. Babaoglu, H. Meling, and A. Barabasi. Anthill: A framework for the development of agent-based peer-to-peer systems. In *International Conference on Distributed Computing Systems (ICDCS 2002)*, pages 15–22. IEEE Computer Society Press, 2002.

[3] M.S. Eby and W.E. Kelly III. Free flight separation assurance using distributed algorithms. In *Proceedings of Aerospace Conference*, 1999.

[4] M. Mamei, F. Zambonelli, and L. Leonardi. Co-fields: Towards a unifying approach to the engineering of swarm intelligent systems. In *Engineering Societies in the Agents World III*, volume 2577 of *Lecture Notes in Computer Science*, pages 68–81. Springer-Verlag, 2003.

[5] M. Mamei, F. Zambonelli, and L. Leonardi. Distributed motion coordination with co-fields: A case study in urban traffic management. In *International Symposium on Autonomous Decentralized Systems (ISADS 2003)*, pages 63–70. IEEE Computer Society Press, 2003.

[6] G. Smith and J. W. Sanders. Formal development of self-organising systems. In *Autonomic and Trusted Computing (ATC'09)*, volume 5586 of *Lecture Notes in Computer Science*, pages 90–104. Springer-Verlag, 2009.

[7] K. Støy. Controlling self-configuration using cellular automata and gradients. In *8th International Conference on Intelligent Autonomous Systems (IAS-8)*, 2006.

[8] K. Støy. Using cellular automata and gradients to control self-reconfiguration. *Robotics and Autonomous Systems*, 54:135–141, 2006.

[9] K. Støy and R. Nagpal. Self-reconfiguration using directed growth. In *7th International Symposium on Distributed Autonomous Robotic Systems (DARS)*, pages 149–160. ACM Press, 2004.

[10] K. Støy and R. Nagpal. Self-repair through scale independent self-reconguration. In *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2062–2067. IEEE Press, 2004.

[11] M. Yim et al. Distributed control for 3D metamorphosis. *Autonomous Robots*, 10(1):41–56, 2001.

[12] F. Zambonelli and A. Omicini. Challenges and research directions in agent-oriented software engineering. *Autonomous Agents and Multi-Agent Systems*, 9(3):253–283, 2004.