Terrain Leveling by a Swarm of Simple Agents

Daniel Cook
Marine Environmental Research Laboratory for Intelligent Vehicles
Memorial University of Newfoundland
St. John's, Canada
dac456@mun.ca

Andrew Vardy
Department of Computer Science / Department of Electrical and Computer Engineering
Memorial University of Newfoundland
St. John's, Canada
av@mun.ca

ABSTRACT

In this paper we present a new algorithm for leveling a discrete terrain using autonomous agents of minimal complexity. Our primary interest is in developing robust strategies for eventual application in underwater construction. Terrain leveling underwater typically involves the use of large and expensive machinery that may be cumbersome to operate. We propose that a swarm of autonomous robots could be used instead. Each agent is implemented with minimal capabilities for perception and actuation. The goal of each agent is to estimate the average height of the terrain, and move material it finds above that average to locations below the average height. We implement the algorithm in a real-time three-dimensional simulator, as well as an offline simulator for the purpose of performance analysis. From these simulations we find that the algorithm functions correctly, and we provide preliminary performance data for various parameters. We conclude that terrain leveling may be a viable application for a swarm of minimalistic robots.

Keywords

swarm robotics, underwater construction

1. INTRODUCTION

Insects such as bees, ants, wasps, and termites are renowned for their ability to reshape their environments into sophisticated habitats [1]. This ability is remarkable as it is generally assumed that the individual insects do not directly control the behaviour of their peers. If there is anything akin to a master plan in their construction efforts, it lies within their genetic programming. This is very different from human construction activities where a small team develops a blueprint which dictates the efforts of a much larger team. In this paper we do not address the construction task, but rather the issue of site preparation for future construction. In particular, terrain leveling using a set of very simple non-communicating agents.

We are particularly interested in terrain leveling for underwater construction. Leveling can be achieved by running a flat beam, known as a screed, across the surface of the seabed or by dredging [3]. Maintaining both sides of a screed at the same height is challenging and the problem is further exacerbated if the screed is suspended from a surface craft and there is wave activity. This difficulty can be managed by the construction and placement of a seabed frame to support and control the screed. There are various dredging techniques as well which are powerful but may produce undesirable disturbances in the environment. In both cases, the idea is to run a single tool across the seabed, perhaps repeatedly. The tool (screed or dredge) is generally supported by an expensive apparatus and monitored by multiple personnel.

Our vision involves a set of autonomous robots with minimalistic individual capabilities. These robots could be suspended in the water (ROVs or AUVs) or tracked vehicles operating on the seabed. Each robot would have some capacity to gather seabed material, move, then deposit that material. They would be able to sense the height of the seabed at the current location (equivalently, sense their own depth while in contact with the seabed). Our algorithm involves randomized movement, which is a simple and robust strategy for area coverage. Therefore, the robots would need some means to stay within the area of interest. This could be achieved by placing artificial boundaries around the work area or with a simple acoustic beacon that would be perceived by the robots and define the centre of the work area. The overall concept is that the robots sense the height of the terrain and develop an estimate of the overall average height. If the current height is greater than the average, then material would be picked-up. If the height is less than average, then carried material would be deposited. Our objective in this paper is to test whether this simple idea can result in flattened terrain. We have conducted experiments in a relatively abstract simulation environment to confirm the viability of this minimalistic approach.

Using a swarm of minimalistic robots for the terrain leveling task offers several advantages. Current techniques used in industry (such as described above) come with substantial financial cost. Robots of the simplicity we propose should be cheaper to manufacture and operate. Current techniques also impose some safety risk to human operators involved, which can be mitigated by using underwater robots in their place. Using a swarm of simple agents also provides benefits versus using fewer complex, and inherently more expensive agents. Manipulating terrain poses the risk of triggering

.

BICT 2014, December 01-03, Boston, United States

DOI 10.4108/icst.bict.2014.257925

a landslide or other event which may trap or damage an agent. Cheaper, simpler agents are more expendable and reduce risk that would be associated with losing a larger more expensive agent in such an event. A swarm of agents also allows the same operation to be started in multiple locations. This leads to a more parallelized effort than is possible with flattening using a screed or by dredging.

In Section 2 we discuss related work. Section 3 presents a description of our algorithm. Section 4 provides experimental results. Sections 5-7 provide discussion, future work, and conclusions.

2. RELATED WORK

To our knowledge, this work represents the first application of swarm robotic principles to terrain leveling. However, within the context of swarm robotics there have been a number of similar problems studied. The classic work on clustering similar objects together by Deneubourg et al. provides some of the key ideas of our approach [2]. Deneubourg et al. proposed a set of 'ant-like robots' which could cluster objects together by preferentially picking up isolated objects and depositing them in areas with high local object density. In some sense, this is the inverse of terrain leveling where we wish to maximally disperse material throughout the environment. We recently demonstrated enhanced performance in object clustering and sorting as well as providing a significant literature review in this area [6].

Terrain leveling can be seen as a preliminary phase of construction. Parker et al. took inspiration from a species of ant which construct nests by clearing out debris from the flat crevices of rock they inhabit [4]. This approach, known as *blind bulldozing*, involves pushing material forward until sufficient resistance is sensed to trigger a random turn to push material forward in a new direction. The result is to clear the terrain and generate an open clearing surrounded by a circular boundary. Werfel et al.'s recent work on termite-inspired robotic construction translates a given desired shape into a data structure known as a *structpath* [7]. The structpath is provided to a set of robots which all share the same purely local control strategy. The operation of these robots on flat terrain with a constantly regenerating seed brick yields the desired shape. In principle, the work described here could serve as a precursor to level the terrain prior to the application of Werfel et al.'s approach.

3. ALGORITHM DESCRIPTION

In this section we propose the algorithm as executed by each agent in the system and describe the capabilities of the agents, as well as assumptions on the operation of the algorithm.

3.1 Agent Description

Each agent in the system moves over a discrete 2D grid representing the terrain. Each grid cell is assigned a height value, indicating the height of the cell in units of material. Each agent may move to an adjacent cell in a single step, and always sits one unit above the height of the cell in which it is located. Each agent is given limited ability for perception and actuation; an agent may sense only the height of the cell it currently occupies, and may only manipulate material in its current cell. Only one agent may occupy a cell at a time, and agents are aware of each other only as obstacles such that an agent will not move into an already occupied cell.

Each agent cycles between a *sensing* state and an *acting* state; from the acting state it may also non-deterministically enter a *pickup* or *deposit* state. In the sensing state, the agent reads the height value of the cell it is currently located in. The agent's objective is to estimate the overall average height and pickup or deposit material such that the terrain reaches a flattened state. Computing the true average height would require an exhaustive survey, which would necessitate more sophisticated agents. Instead we choose to sample the current height values at randomized intervals. To implement this strategy, the current height value has probability ρ_{sense} of being stored in the agent's memory. The agent's memory is of finite size and is implemented as a queue. As values are stored in memory, the agent calculates an average height α. In the acting state the agent may 'pick-up' one unit of material if it is carrying none already and the height of the current cell is greater than α, or 'deposit' one unit of material if it is carrying material and the height of the current cell is less than α. Both of these actions are executed with probabilities ρ_{pu} and ρ_{de} respectively. Pick-up is represented by decreasing the height value of the cell by one, and deposit is representing by increasing the height value by one. Before leaving the acting state the agent moves on a random walk to an adjacent cell; the agent has a 75% chance to keep moving in its current direction, and a 25% chance to select a new direction at random or not move at all.

3.2 System Description

Algorithm 1 gives a pseudo-code description of the system behaviour. The crux of the system is each agent's movement of material above its sampled average to positions below the average. Each agent in the system will calculate different values for average height, since each will have sampled different height values over the terrain. Collectively, the average of the sampled heights across all agents should tend towards the true average height of the terrain. Thus, despite operating with differing individual information the emergent behaviour will be a tendency towards flattening the terrain about its true average height.

3.3 Choosing Probability Values

Behaviour and efficiency of the algorithm are determined in part by the values chosen for ρ_{pu}, ρ_{de}, and ρ_{sense}. Early work in developing the algorithm relied on constant values for ρ_{pu} and ρ_{de}; this approach may still be useful in implementations where computational resources for each agent are extremely limited. An alternative is to allow the probability to be chosen by the agent based on the height of its cell, either by linear interpolation between the cell height and some known maximum or on a curve as seen in work by Deneubourg et al. [2] and Vardy et al. [6]. Our experimental probability functions are as found in these works; $\rho_{pu} = (\frac{height}{k+height})^2$ and $\rho_{de} = (\frac{k}{k+height})^2$. Values for ρ_{sense} will effectively control the sampling rate of each agent and may be chosen experimentally.

4. EXPERIMENTAL RESULTS

Experiments were conducted on two implementations. The first is a 3D voxel engine, in which each cell of the terrain is represented by a stack of voxels shaded by height. Agents are each represented by an icosahedron. In this implementation agent behaviour can be observed in real time. The second

Algorithm 1 Multi-Agent Terrain Flattening

```
loop
  for all agent do
    if agent sensing then
      if rand() < ρ_sense then
        h ⇐ sensed_height
        enqueue_memory(h)
        α ⇐ update_average()
      end if
      state ⇐ acting
    else if agent acting then
      if agent not carrying material and rand() < ρ_pu and h > α then
        pick up material
      end if
      if agent carrying material and rand() < ρ_de and h < α then
        deposit material
      end if
      move_agent()
      state ⇐ sensing
    end if
  end for
end loop
```

implementation is an 'offline' simulation in which there is no rendered output; this implementation is intended to execute as fast as possible to log performance metrics over various parameter inputs. Figure 1 depicts a random terrain before and after flattening in the voxel implementation. The terrain is generated with a four-dimensional Perlin noise function.

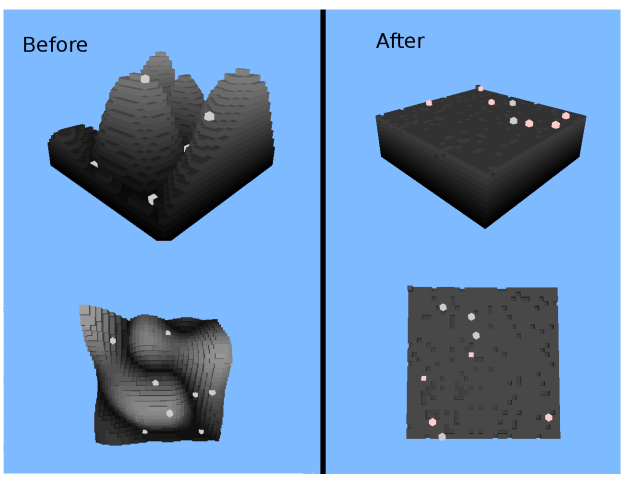

Figure 1: 32x32 terrain before and after flattening by 8 agents. Agents are represented by the light grey objects.

4.1 Performance Evaluation

Performance is evaluated by time to completion, where time is measured as the number of cycles of the algorithm's outer-most loop. A simulation is considered complete when the height of every cell in the grid is less than the maximum height of a pre-computed flat terrain. Experiments were conducted on various constant values for ρ_{pu} and ρ_{de} (see Figure 2), as well as various values for k when the probability curves described previously were used (see Figure 3). In the case of constant values, $\rho_{pu} = \rho_{de}$. All experiments were conducted on a 16x16 grid with 8 agents. For each parameter value tested, the simulation was run 30 times.

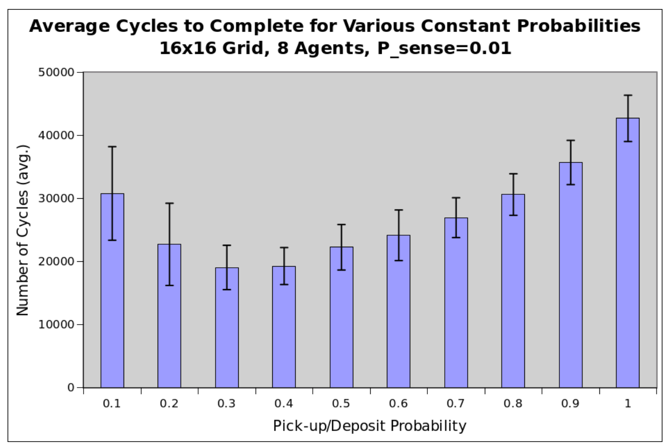

Figure 2: Comparison of various constant values for ρ_{pu} and ρ_{de}. Error bars denote ± 1 standard deviation.

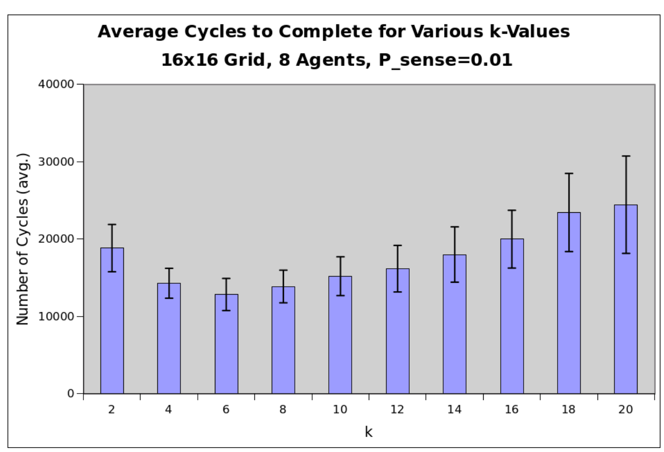

Figure 3: Comparison of various values for k in pick up and deposit probabilities. Error bars denote ± 1 standard deviation.

5. DISCUSSION

In Figure 2 we see a U-shaped function in time to completion with a minimum at 0.3. It is not clear yet what the reason is for this result; however, we feel it is related to the notion that it is more efficient to pick up material from peaks and deposit them in valleys. In the case of pick up, $\rho_{pu} = 1.0$ forces the agent to pick up the first material it finds above its calculated average α. For $\rho_{pu} < 1.0$, the agent is offered a chance to explore further without manipulating the terrain, potentially finding a higher peak later on its random walk. We conjecture that our observed optimal probability of 0.3 is related to the average radius of hills and valleys found on the terrain. This probability may in most cases allow the agent to move forward such that it is generally positioned closer to the maximum before deciding to pick up material. Using symmetric reasoning, the same explanation holds in the case of depositing material. In the case of probability curves a similar U-shape is also observed. The values of k merely shape the curve; it may be that particular values for k shape the curve such that the resulting probability most often corresponds to the constant probability variant of the algorithm. However, the use of the height information in the probability curve results in overall reduced time to completion as seen in Figure 3.

6. FUTURE WORK

The algorithm presented in this paper is rather abstract; it operates with agents which are largely aphysical in a discrete environment. Evaluation of performance in an offline simulated environment does not necessarily translate to high performance in a real environment, and there is no guarantee the algorithm would generate correct results in a real environment in a tractable amount of time. It does however provide a starting point for further development and evaluation of other multi-agent construction algorithms. In terms of performance, we would like to investigate the effect of clustering agents around peaks in the terrain (as opposed to a random walk), similar to clustering in sources of illumination as found in work by Schmickl et al. [5]. Future work will also seek to determine whether the algorithm is practical for real-world implementation, and implement the algorithm in real autonomous robots if so. We will also look to determine whether the principles of the algorithm presented in this paper may be extended to more complex construction and terrain manipulation tasks. In terms of performance and efficiency, future work will aim to find a definitive cause for the experimental results discussed in section 5, and we provide additional analysis of the results and the algorithm itself. In this paper we have explored the algorithm as it operates on a discrete terrain; further work will move the algorithm into a continuous space for further validation of its practicality.

7. CONCLUSIONS

We have presented a new algorithm for leveling a discrete terrain with a swarm of simple agents. We have also demonstrated by way of simulation that the algorithm gives correct results for the parameters and environment tested, and appears viable for futher experimentation and analysis.

The application of swarm robotics for underwater construction remains to be trialed in a real-world scenario. Further development may allow cumbersome and complex machinary to be replaced by relatively inexpensive autonomous robots requiring little, if any, human intervention. Based on the results demonstrated in this paper, it appears that it is a viable option for further exploration and study.

8. ACKNOWLEDGMENTS

The work presented in this paper was funded by the Atlantic Canada Opportunities Agency, through an Atlantic Innovation Fund grant, under the Responsive AUV Localization and Mapping project (REALM), and Research and Development Corporation of Newfoundland and Labrador, Canada.

9. REFERENCES

[1] E. Bonabeau, M. Dorigo, and G. Theraulaz. *Swarm Intelligence: From Natural to Artificial Systems.* Oxford University Press, New York, NY, 1999.

[2] J. L. Deneubourg, S. Goss, N. Franks, A. Sendova-Franks, C. Detrain, and L. Chrétien. The dynamics of collective sorting robot-like ants and ant-like robots. In *First Int. Conf. on the Simulation of Adaptive Behaviour*, pages 356–363, Cambridge, MA, 1990. MIT Press.

[3] B. C. Gerwick. *Construction of Marine and Offshore Structures.* CRC Press, 2007.

[4] C. A. C. Parker, H. Zhang, and C. R. Kube. Blind bulldozing: Multiple robot nest construction. In *IEEE/RSJ International Conference on Robots and Systems (IROS)*, 2003.

[5] T. Schmickl, R. Thenius, C. Moeslinger, G. Radspieler, S. Kernbach, M. Szymanski, and K. Crailsheim. Get in touch: cooperative decision making based on robot-to-robot collisions. *Autonomous Agents and Multi-Agent Systems*, 18(1):133–155, 2009.

[6] A. Vardy, G. Vorobyev, and W. Banzhaf. Cache consensus: Rapid object sorting by a robotic swarm. *Swarm Intelligence*, 8(1):61–87, 2014.

[7] J. Werfel, K. Petersen, and R. Nagpal. Designing collective behavior in a termite-inspired robot construction team. *Science*, 343(6172):754–758, February 2014.