Multiagent Based Construction for Human-like Architecture

Yifeng ZENG
Machine Intelligence Group
Department of Computer Science
Aalborg University
Fredrik Bajers Vej 7E, 9220
Aalborg, Denmark
yfzeng@cs.aau.dk

Dennis Plougman Buus
Machine Intelligence Group
Department of Computer Science
Aalborg University
Fredrik Bajers Vej 7E, 9220
Aalborg, Denmark
dbuus@cs.aau.dk

Jorge Cordero H.
Machine Intelligence Group
Department of Computer Science
Aalborg University
Fredrik Bajers Vej 7E, 9220
Aalborg, Denmark
corderoh@cs.aau.dk

ABSTRACT

Collaborative construction is a main application in the field of autonomous systems. An interesting subject in the area is the construction of realistic human-like architecture. However, the task of building a human-like architecture is non-trivial since the construction is a real time process without human supervision. In this paper, we present a collective building algorithm based on stigmergy. A swarm of virtual agents construct edifications which resemble basic features in human-like architecture. The algorithm maps sensory information to appropriate building actions.

Categories and Subject Descriptors

I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence

General Terms

Algorithms, Design, Experimentation

Keywords

Swarm Intelligence

1. INTRODUCTION AND BACKGROUND

Collaborative construction on hazardous and remote places may require the use of autonomous agents prior to human arrival. Therefore, it is important to design an algorithm to direct those agents in the building process. Swarm intelligence algorithms have been used for physical and virtual construction [4, 5]. In this paper, we present the foundation of a collaborative construction algorithm (named the CCA) for human-like architecture. Construction in natural systems rely heavily upon two important concepts, self-organization and stigmergy. Self-organization is based on four important mechanisms: Positive/negative feedback, randomness and multiple interactions. Stigmergy is an indirect mechanism of communication between agents [3]. In swarm intelligence, we often talk of two kinds of stigmergy: Quantitative and qualitative. Pheromone trails are an example of quantitative stigmergy [4]. Qualitative stigmergy involves coupling specific stimulus with specific actions, such as nest construction by wasps.

This paper is structured as follows: Section 2 presents the CCA algorithm. Section 3 introduces a genetic algorithm for the evolutive construction of templates. Section 4 describes some experimental results. Finally, Section 5 highlights the major achievements of this study and suggests future work.

AAMAS'07, May 14–18, 2007, Honolulu, Hawai'i, USA.

2. THE CCA ALGORITHM

The CCA algorithm controls a number of agents that move about in a discrete 3D lattice. The agents deposit building materials according to a set of stimulus-response rules. The lattice is an array of cells which contain information about the presence of building blocks and pheromone intensity. Whenever a building block is placed, a certain amount of pheromone is deposited along with it. Agents move by selecting their direction stochastically following the pheromone intensities. Assuming that each cell in the 3D lattice diffuses pheromone to its eight neighbors, the amount of pheromone $\Delta\tau_{c_i}$ that each neighbor receives from cell c_i can be expressed as:

$$\Delta\tau_{c_i} = \frac{\tau_{c_i} \cdot d}{8}, \tag{1}$$

where $0 < d < 1$ is the *diffusion coefficient* which regulates the percentage of pheromone moved from a cell to its neighbors. τ_{c_i} is the amount of pheromone in the cell c_i. Once an area has been built up to the point where rules are no longer matched, agents must no longer continue to be recruited to that area. In order to ensure this, the pheromone is evaporated at a steady rate in a cell c_i as shown below:

$$\tau_{c_i} \leftarrow (1 - \rho) \cdot \tau_{c_i}, \tag{2}$$

where τ_{c_i} is the amount of pheromone in cell c_i and $0 < \rho < 1$ is a coefficient dictating the speed of evaporation.

The agents have a direct perception range which is limited to a cube of $3 \times 3 \times 3$ cells surrounding their position. Additionally, the agents are able to sense the concentra-

978-81-904262-7-5 (RPS)

tions of pheromone in a small area in each of the directions they could travel (2 squares ahead, behind, left and right). As building progresses, the CCA algorithm maintains the front, side and top views of the overall density of the building blocks, which we call *density maps*. The *density maps* form gray-scale image from the environment.

The agents do not move diagonally and have the ability to climb the existing architecture. We utilized an adapted form of the movement selection equation from the work by Deneubourg et al. [1]. Let C be the set of all allowable target cells. η_{c_i} is the desirability of the target cell $c_i \in C$. The probability p_{c_i} that an agent will move to cell $c_i \in C$ is given by:

$$p_{c_i} = \frac{(r + \eta_{c_i})^\alpha}{\Sigma_{c_j \in C}(r + \eta_{c_j})^\alpha}. \quad (3)$$

The parameter α controls the linearity of the function. The parameter r adjusts the tendency of the agent to choose its direction randomly. The branching rules in the CCA algorithm map a triggering configuration to the placement of a block in a specific location (action). Let A be the set of all possible actions that an agent has to choose from when encountering a specific triggering configuration. The probability p_{a_i} that an agent chooses build action $a_i \in A$ is given by:

$$p_{a_i} = \frac{\eta_{Fi} + \eta_{Si} + \eta_{Ti}}{\sum_{a_j \in A} \eta_{Fj} + \eta_{Sj} + \eta_{Tj}}, \quad (4)$$

whereas η_{Fi}, η_{Si}, and η_{Ti} are the front, side, and top desirability values for the cell being considered for the block placement by action a_i. The agents make their decision on which building action to take depending on the density values read from the density maps. In order to do so, we calculate the desirabilities using an adaptation of the basic model of clustering behavior from Deneubourg et al. [2] as follows: Let $M = \{F, S, T\}$ be the set of density maps; front, side and top. The desirability value η_{mi} for action a_i and map $m \in M$ is provided by:

$$\eta_{mi} = \begin{cases} \left(\frac{D_{mi}}{\delta_{mi} + D_{mi}}\right)^2, & \text{if } \delta_{mi} > 0 \\ \left(\frac{|\delta_{mi}|}{|\delta_{mi}| + D_{mi}}\right)^2, & \text{if } \delta_{mi} < 0 \\ 0, & \text{if } \delta_{mi} = 0 \end{cases}, \quad (5)$$

having δ_{mi} as the density threshold and D_{mi} as the density value read from the density map m for the cell being considered for block placement by action a_i. Algorithm 1 provides a high-level description of the CCA algorithm. First, the 3D lattice and the data structures are initialized, a rule set is taken as input and a number k of agents are randomly distributed across the lattice. For each iteration, the agent senses the configuration of the environment and looks up this configuration in the rule set. If the configuration matches a branching rule, then the agent decides which action to take by using equation 4. Then, the agent places a building block and building pheromone at ground level. The agent must then decide which direction to move. It examines each possible cell it can move to and it calculates the probability of moving there by using equation 3. A roulette wheel selection mechanism is used for the agent to decide upon a cell and then it moves there. At the end of an iteration, the pheromone concentrations are diffused and evaporated.

Algorithm 1 The CCA Algorithm

```
/* Initialization */
Input: A set of rules, and the simulation parameters including max_iterations and k
Initialize the 3D lattice
Construct the initial density maps
for each agent k do
  set random (x_k, y_k, z_k)
end for
/* Main loop */
for 0 to max_iterations do
  /* Agent loop */
  for each agent k do
    Construct the sensory information for (x_k, y_k, z_k)
    for each configuration do
      if (sensory information matches rule) then
        for each action a_i ∈ A do
          Calculate p_{a_i} according to equation 4
        end for
        Place a building block according to the rule with the highest p_{a_i}
        Deposit pheromone in the appropriate floor cell beneath the newly placed building block
      end if
    end for
    for all allowable target cells c_i ∈ C do
      Calculate p_{c_i} according to equation 3
    end for
    Select the target cell with Roulette Wheel selection
    Move the agent to the chosen cell
  end for
  for each cell c_i in world do
    Evaporate pheromone according to equation 2
  end for
  for each cell c_i in world do
    Move an amount of pheromone from c_i according to equation 1 to each neighboring cell
  end for
end for
```

3. RULE EVOLUTION

We used a genetic algorithm (GA) to design new construction templates. The implementation of a simple GA requires: A population of solutions (individuals), a method for determining the relative *fitness* of each individual, a strategy for selecting individuals for reproduction and evolutionary variation methods. The fitness function for the CCA algorithm makes use of the density maps which influence the decisions for each agent. Since the maps are equivalent to gray-scale images, we used an image comparison algorithm [6] to calculate the objective measure for the similarity between two structures. In order to compare two images, the root-mean-squared error has to be calculated. Let f and g be two gray-scale images, then the root-mean-squared error is given by:

$$RS(f, g) = \sqrt{\frac{1}{n(X)} \sum_{x \in X} (f(x) - g(x))^2}, \quad (6)$$

where $n(X)$ is the number of pixels in an image X and $f(x)$ is a single pixel in the image. This gives us an objective measure of the dissimilarity between the two pieces of

architecture, which we use as the base for the fitness score (F) of a rule set as follows:

$$\mathrm{F} = \left(\frac{RS(f_F, g_F) + RS(f_S, g_S) + RS(f_T, g_T)}{3} + 1 \right)^{-1}, \tag{7}$$

where f_F, f_S and f_T are the front, side and top density maps of the generated architecture and g_F, g_S and g_T are the front, side and top density maps of the hand crafted architecture.

Algorithm 2 describes the GA used for the CCA algorithm. Initially, a population of random rule sets is generated. For each generation, the algorithm looks at every individual in turn. A a rule set is executed to the CCA algorithm which is allowed to run for a specified number of iterations. Once the CCA algorithm was executed, it generates a set of density maps which are used in the fitness evaluation of the rule set. The GA calculates the fitness function described in equation 7 to determine the fitness of the rule set. Finally, once every rule set has obtained its fitness, the two highest scoring rule sets are chosen through elitist selection. The remainder of the parent pairs are chosen through roulette wheel selection. At the end, crossover and mutation produce two offsprings for the next generation of the parent pair.

Algorithm 2 GA for the CCA algorithm

```
/* Initialization */ Generate population of random rule sets
/* Main loop */
for 0 to max_generations do
  /* Simulation loop */
  for each rule set R in population do
    Run CCA for max_iterations with Rule Set R
    Determine fitness F from resulting density maps using equation 7
  end for
  /* Selection and variation stage */
  Select the parent pair (R1, R2) of rules with highest fitness
  for 0 to max_population/2 − 2 do
    Select the parent pair (R1, R2) of rules with Roulette Wheel selection based on fitness.
  end for
  for each pair (R1, R2) do
    Apply crossover and mutation.
    Add the new children to the population
  end for
end for
```

3.1 Experimental Results

Figure 1 depicts the progress of the construction of a building with one door using the CCA algorithm. The red marks on the ground represent repulsion pheromone and green marks represent attraction pheromone. The simulation used 47 rules with 4 branching rules; 3 to build the corners, 1 to extend the walls upwards until certain height was reached. The rest of the rules were 1:1 mappings between triggering configurations and single actions. Agents are shown as yellow cubes.

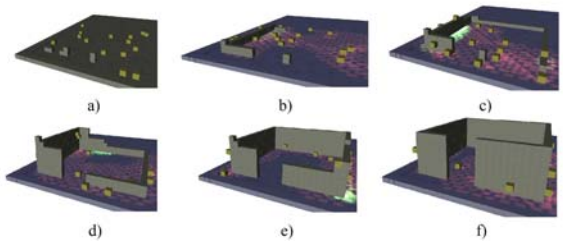

Figure 1: a) The seed structure. b) Agents started foraging. c) Agents constructing an square structure. d) Contiguous walls started to be constructed. e) The construction was finished. f) The agents started foraging again (branching rules were no longer matched).

4. CONCLUSION

Basic shapes of human-like architecture were presented as the final result by using a swarm of building agents. The combination of short range perception and large scale indirect perception in the form of density maps is a novel swarm construction approach. We were able to match a specific local configuration to specific building actions. These building actions depended on the quantitative influence of global building densities. In the case of the GA, the method of comparing two structures based on their density maps provided us with an objective measure of the dissimilarity between them. Future work is to implement a robust GA. Besides that, we are currently working on an multiagent visualization environment for the CCA algorithm and it will be announced in a posterior paper.

5. REFERENCES

[1] J. L. Deneubourg, S. Aron, S. Goss, and J. M. Pasteels. The self-organizing exploratory pattern of the argentine ant. *Journal of Insect Behavior*, 3:159–168, 1990.

[2] J.-L. Deneubourg, S. Goss, N. Franks, A. Sendova-Franks, C. Detrain, and L. Chretien. The dynamics of collective sorting: Robot-like ant and ant-like robot. In *In Proceedings of the First Conference on Simulation of Adaptive Behavior: From Animals to Animats*, pages 356–365, 1991.

[3] M. Dorigo and L. M. Gambardella. Ant colony system: A cooperative learning approach to the traveling salesman problem. *IEEE Transactions on Evolutionary Computation*, 1(1):53–66, April 1997.

[4] D. Ladley and S. Bullock. Logistic constraints on 3d termite construction. In *ANTS Workshop*, pages 178–189, 2004.

[5] J. Nembrini, N. Reeves, E. Poncet, A. Martinoli, and A. Winfield. Mascarillons: flying swarm intelligence for architectural research. In *Swarm Intelligence Symposium, 2005. SIS 2005. Proceedings 2005 IEEE*, pages 7–14, 2005.

[6] D. L. Wilson, A. J. Baddeley, and R. A. Owens. A new metric for grey-scale image comparison. *Intern. J. Computer Vision*, 24:5–17, 1997.