Painting with Outliners and Fillers

Marco Celesti and Mario Verdicchio
Department of Information Technology and Mathematical Methods
University of Bergamo, Italy
www.unibg.it
e-mail: mario.verdicchio@unibg.it

Abstract

Ant-based artificial art has reached significant levels of visual appeal and aesthetics. Nevertheless, the artworks are still clearly recognizable as computer-generated, and present a diffused colouring effect common to most of the ant-based art.
In this work we propose to introduce two different roles, namely outliner and filler, that ants or ant-like agents can play on a digital canvas, aiming at producing artworks that, although sketchy at this stage of our research, look promising in the pursue of generative art that resembles human-made drawings.

1. Introduction

Our proposal stems from what can be considered as the most complete ant-based artificial art research work to date [1]. Ants can be seen as a special type of software agents, and their interaction in a closed system that eventually generates an output viewed as artwork by the audience can be considered as a very peculiar kind of Multi-Agent System (MAS) [2]. The ant metaphor is introduced whenever the agents' behavior reminds of some aspects of ants seeking a path between their colony and a source of food [3]. A typical characteristic of ant-like agents is that they all perform the same set of actions, including moving forward in different directions, modifying the environment by metaphorically leaving pheromone trails behind, and perceiving the trails left by others. We can then see ants as a MAS where all the agents are following a single behavior. Although very effective in solving specific optimization problems, such behavioral uniformity might be a limiting factor, especially in the context of generative art, where researchers, designers, and artists are constantly looking for the emergence of innovative, original patterns and appealing, eye-catching, provocative results without much care for the efficiency of the computational process. We propose to depart from the classical ant metaphor by introducing different behaviors in the system, often referred to in the literature as 'roles' [4]. In particular, we will differentiate our agents into two distinct sets: *outliners* trace contours on the canvas, and *fillers* provide color. From now on, although we took our inspiration from ant-based proposals, we will use the more general term 'agent' instead of 'ant' because we think their behavior is too different from what can be traditionally ascribed to these insects, especially with respect to the trails they leave behind: outliners are not affected by other agents' trails, whereas their trails work as insurmountable borders for fillers, which are named after the fact that, when trapped in a closed loop traced by an outliner, they fill up the area with color.

2. The algorithm behind the artwork

If generative art is supposed to be based on programs that automatically generate the output, a very interesting if not critical issue rises when considering the boundaries between what the software designers directly code and what is left to the random evolution of the running program. This topic points to the very concept of autonomy, which has been long debated ever since Artificial Intelligence researchers started hinting at the possibility of creating software and robotic systems that could 'autonomously' reach a solution once given the specifications of a problem. This work is obviously not meant to tackle such a complex topic, but nonetheless every attempt to create software whose output is not easily predictable by the designers inevitably begs the question on how much of the result of a run is a direct consequence of specific design choices, so that it can be retrieved in all the other experiments, or it is the consequence of a fortuitous combination of randomly set conditions that are unique for the run. To make a formal analysis even more complicated, we do not have a metric for the success of an experiment, as it clearly depends on aesthetic criteria that are personal, social, cultural, or, in other words, far from being pinned down by any simple or complex computational model. Any attempt at generative art lies thus in front of at least three concepts: autonomy, randomness, and aesthetics, which are already hard to define in general, let alone from a computational perspective. Moreover, in our case, we have to simulate the interaction of a significant number of agents (outliners and fillers) in a grid-like environment (the canvas), so we have to add also performance limits of the available hardware on top of all the above-mentioned issues.

Here follows a rough description of the behavior of the different agents in our system.

2.1 Outliners

A specific number of outliners are generated on random spots on the canvas. Each outliner is provided with a DNA that establishes in what direction (left or right) the outliner will turn during its lifespan. Making an outliner turn always in the same way increases its chances to create a closed loop entrapping fillers and thus creating color blobs in the painting. Even if they work as boundaries for fillers, the contours do not affect the outliners' movements. Like ants, our agents only move forward, but the set of possible directions is even more limited for outliners with the aim to maximizing the chances of having loops. The canvas is based on a spherical model, so that when an outliner reaches the left end, it goes on moving and gets to the right end of the painting. The contour is black to be as neutral as possible, as opposed to the varied colors left on the canvas by the fillers. White contours have been scrapped, as we wanted to depart from the typical random color blob appearance of traditional ant-based artificial art. The settings of the run include the length of the outliners' life span. After a specific interval, all outliners 'die', and the contour tracing activity ceases all over the canvas. On average, the contours include a significant number of loops, but also the final strokes of the outliners before their 'death', which are very likely to be comprised of an open segment. A canvas cleaning function then intervenes, by iteratively determining these dead ends of contours (i.e.: black points on the canvas which have only one adjacent black point) and erasing them. Obviously the cleaning process will end as soon as all dead ends are erased and only loops are left on the canvas, so that each black point has at least two neighbouring black points (i.e. they are part of a loop or of a segment linking two loops).

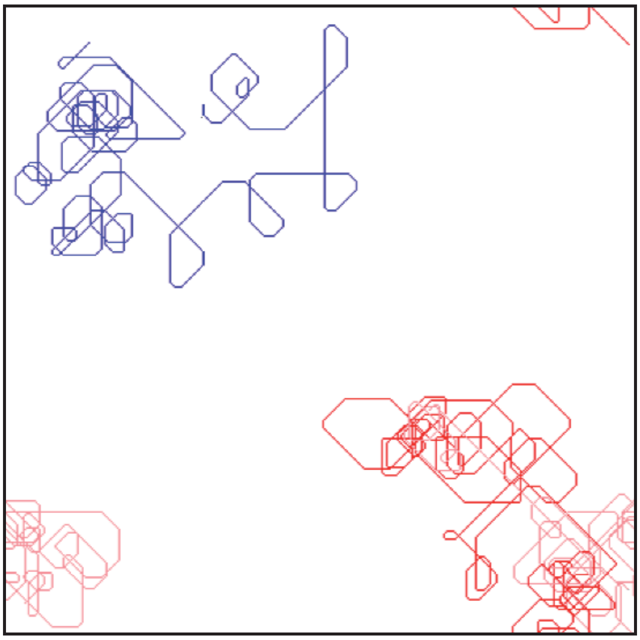

Figure 47: Outliners at work. The colors only help distinguish different outliners, which produce black contours in the final version of the program.

2.2 Fillers

The number of fillers is dynamically established during the run of the program: at every pre-specified amount of time, two fillers are spawned from the piece of contour just traced by each outliner. One filler is spawned on the left hand side, one on the right hand side of the contour. As our main aim is to have fillers color the inside of the loops, it would be a reasonable choice to spawn a filler only on the side the relevant outliner is programmed to turn to. Nevertheless, the filler on the opposite side can possibly be trapped in a bigger loop of the relevant outliner, or even in a loop of another outliner. A filler's DNA establishes which color the agent will leave on the canvas at each occupied position. The color can be randomly picked from the RGB or the HSV space, with different visual outcomes. When two or more fillers are entrapped in the same loop, then the different colors they deposit tend to mix, as a weighted average between the DNA color and the one already present in the environment is calculated and depicted once a filler deposits its color onto a spot which was previously visited by the other filler in the loop. Given enough time, the loop should become uniformly filled with the average of the colors of the agents it has entrapped. Fillers' life span is also computed at runtime: they have a specific quantity of energy at birth, which diminishes at every movement, and which gets replenished every time a filler bumps into a contour. As the energy level goes down, so does the saturation of the color deposited by the filler. Once the color becomes white, the filler is considered 'dead', and removed from the system. By making the fillers' life (and thus their coloring activity) depend on their proximity to the contours, we intend to minimize the effects of the fillers that end up lying on the outside of all loops. The disadvantage of this approach is that big loops become harder to fill up with color, but the alternative solution of calculating the curvature of the contour at the birth of a pair of fillers and then deciding which one to keep turns out to be computationally demanding.

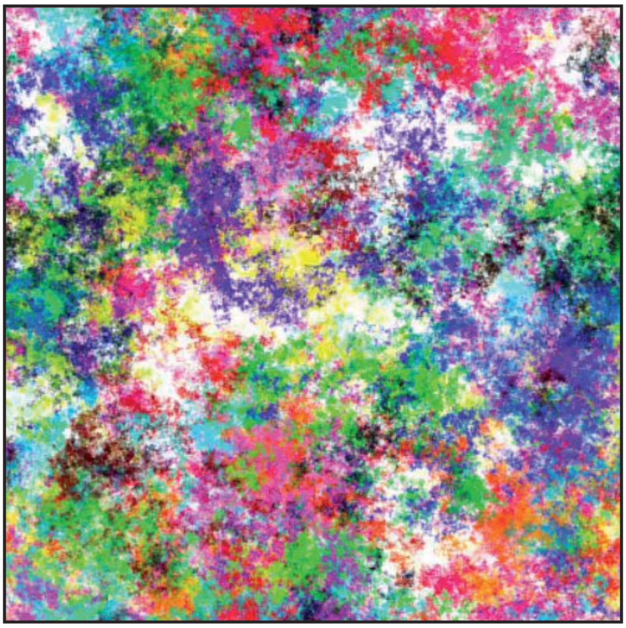

Figure 48: Fillers at work without any entrapping contour.

3. Conclusions and future work

By combining on a white canvas the contour tracing activity of outliners and the coloring activity of fillers, we have implemented a multi-agent system capable of providing a satisfying aesthetic experience in the form of randomly generated paintings. Not surprisingly, the questions on the nature of generative programs, and on their relation with aesthetic criteria have only been skimmed over. Among the three fundamental concepts of autonomy, randomness, and aesthetics, as software designers, we consider the first one to be the easiest to tackle in our future efforts: with a valid help from the Evolutionary Computation research, we would like to increase the portion of our system's behavior which is determined by the evolution of the DNA of our agents, as opposed to design decisions taken by the programmers on the basis of hardware limitations, or common sense criteria. We see the spontaneous emergence of the two different roles of outliners and fillers from the evolution of the DNA of generic agents as the next milestone in our research path.

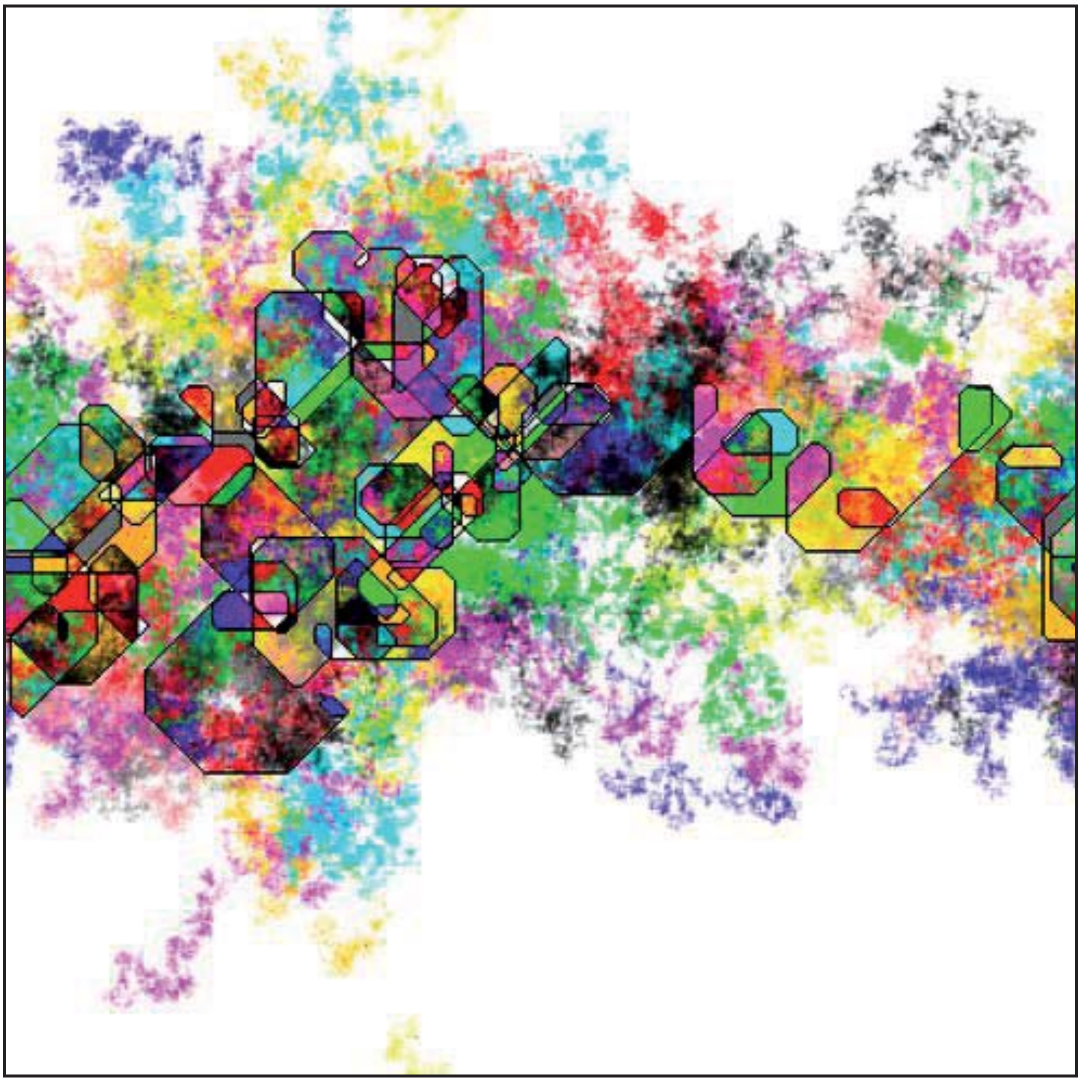

Figure 49: Outliners and fillers at work

References

[1] N. Monmarché et al., “*Artificial Art made by Artificial Ants*”, in The Art of Artificial Evolution, Springer, 2008

[2] M. d'Inverno and M. Luck, “*Understanding Agent Systems*”, Springer, 2004

[3] M. Dorigo and T. Stuetzle, “*Ant Colony Optimization*”, MIT Press, 2004

[4] G. Boella, L. van der Torre, H. Verhagen, "*Roles, an Interdisciplinary Perspective*", Journal of Applied Ontology 2(2), IOS Press, 2007