

# Generating Trace Renditions Using Human Swarm Principles

Aryan Tamrakar<sup>1</sup> and Daniel Palmer<sup>2</sup>

Mathematical and Computational Sciences Department, College of Wooster, Ohio, USA

<sup>1</sup>atamrakar24@wooster.edu, <sup>2</sup>danpalmer@wooster.edu

## Abstract

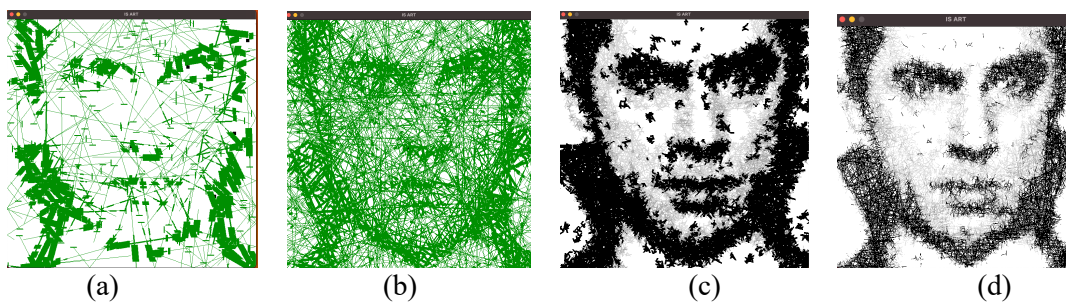
We have developed a swarm-based approach for constructing non-photorealistic renditions of target images. Our approach uses a multi-agent simulation, leaving trails of various widths and colors as agents wander across a pixelated image. We specifically compare the techniques and results of a swarm-based approach (modeled after human interactions within a swarm) against a simple agent-based approach that has no agent interaction.

## Agent-Based Trace Renditions of Target Images

Simple agent-based trace renditions can “recreate” target images with a wide variety of effects. The fundamental characteristic of agent-based trace rendering algorithms is that no interaction takes place between the agents. Each agent wanders, reading the pixels from the target image and then reacting to them – with darker pixels producing thicker trails or deeper colors. Like any set of *independent* tasks, applying more agents proportionally reduces the amount of time required to complete the task. A single agent can produce any of the renditions in Figure 1;  $n$  agents can produce those renditions  $n$  times faster.

This rendering algorithm has identical agents moving over an image and use only the value of the image pixels to determine the size and color of the trail left behind to generate the rendition. The exact number of agents does not matter, as similar results will be generated using any number of agents as long as  $nt = k$ , where  $n$  is the number of agents and  $t$  is the time of the rendering. Different  $k$ 's produce different results as seen in Figure 1. Each simulated agent follows the rules shown in the left column of Table 1.

The effects of agent-based renditions are shown in Figure 1. Different renditions of the target image can be produced by altering simulation parameters such as the number of agents or length of the simulation (Figure 1ab), the number of grey levels (Figure 1cd), the width and color of the trace lines (Figure 1ac), as well as the thresholds for leaving a trace (Figure 1bd).



**Figure 1:** Agent-based rendering of an image of Dracula: (a) rendered with a smaller  $k$ , (b) rendered with a larger  $k$ , (c&d) rendered with different pixel thresholds and multiple shades.

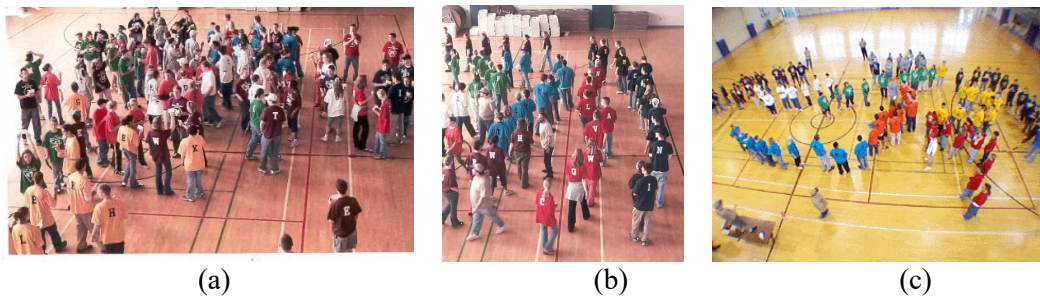
If we keep  $k$  fixed, the equation,  $nt = k$ , specifies an inverse linear relationship between the time we allot to the algorithm and the number of agents performing the algorithm. This relation breaks when we alter the simulation to include interactions between the agents. We are interested in how agent interaction impacts the image rendering, in particular, we use ideas from our previous work in human swarms[3] to develop rendering algorithms that require agent interaction, potentially producing more complex and interesting image renderings.

**Table 1:** *High-Level Rules for Agents Generating Trace Renditions*

Agent-based Trace Rendering Rules	Swarm-based Trace Rendering Rules
Wander, ignoring other agents	Wander, look for agents to follow – no trail while single
Read pixel value of image at current location	When found, link to the agent and make them a leader
Determine trail color and width based on pixel value	Leaders read image pixel value, leaves proportional trail
Leave trail of determined color and width	Followers mimic leader’s trail behavior, reinforcing trail
Repeat until simulation halted	Leaders set pixel threshold based on the length of the chain
	Repeat until simulation halted

### Human Swarm Techniques

Human swarms, consisting of dozens of volunteer “agents”, following high-level commands, can produce non-uniform behavior similar to those found in biological collectives such as food-searching insects, flocking birds, or state-changing slime molds. Observing the actions of the human swarm can lead to insights that are difficult or expensive to obtain using physical robots or by programming simulations. Human swarms rely on ingrained human characteristics and behavior (collision avoidance, long and short range vision sensors, spatial awareness, path-planning) all of which must be explicitly programmed for robots or simulations. However, human swarms require long lead times, extensive planning, incentives for participants, and have a limited effective duration per event (no more than an hour based on our experience.) Therefore, the best use of a human swarm is for gathering insights and discovering directions to investigate [3]. Figure 2(abc) shows a human swarm working on different tasks.

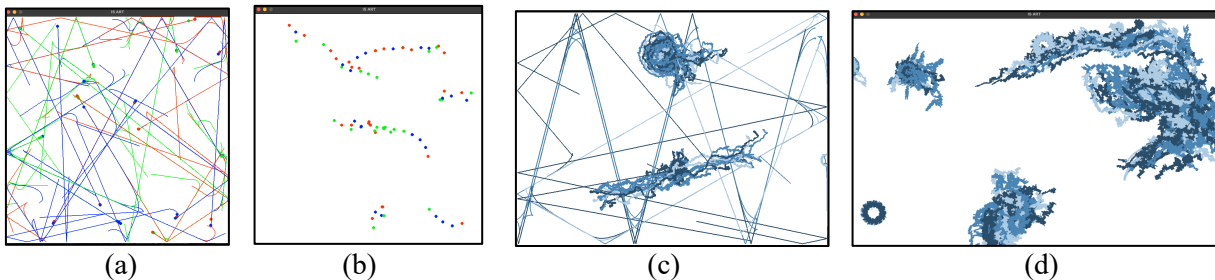


**Figure 2:** *Human Swarms performing tasks: (a) preparing to follow other agents based on cap color, (b) sorting by shirt color, (c) completed task of sorting by shirt color.*

In the caps-shirts experiment (Figure 2(a)), 100 people (agents) wear one of 10 different color T-shirts. Twenty of those agents, randomly selected, are given caps from the same color palette. No agent wears a cap matching their own T-shirt. In this environment, the agents initially move randomly in a manner that can only be described as “milling about”. The agents are then told to look at their shirt color and follow the nearest person who is wearing a cap of that color. Watching the behavior of the swarm over time, the initial chaos gives way to the formation of chains, which can eventually form a closed loop or spiral. The loops further expand into circles. The number and size of the circles and the existence of spirals depends on the starting conditions and the randomness of which agents come into each other’s proximity. The key observation is that from a large number of interacting agents following a simple rule, a higher level organization can arise (i.e. no agent in the human swarm was told to form loops or spirals, yet they did). In this work we harness this organization to produce trace renditions of target images, inspired and informed by the works of Robert Bosch[1] and the swarm-centric approach of Daniel Shiffman[2].

### Agent Interaction Simulations

We developed a software simulation in Python (using PyGame) of the shirts/caps human swarm; the simulation uses three different colors of caps and shirts, but otherwise follows the same rules we described above. Figure 3 shows behaviors from the simulation. Figure 3(a) shows the agents individually wandering, leaving traces, and looking for other agents to follow. Figure 3(b) shows the agents forming chains of followers, when these chains form loops, the agents in the chain can directly impact other agents in the chain, disrupting the standard following behavior. Figure 3(c) shows both individual agent behavior (the regular patterned lines) and the more complex behavior of multiple agents interacting, producing non-regular traces. These disruptions result from agents interacting, and could not be generated by a single agent regardless of the duration of the simulation so this simulation violates the  $nt = k$  inverse-linear relation. Figure 3(d) shows an extreme example of the agent interaction. Note that these dense, overlapping agent trails do not appear in the human swarm, but result from the simulated agents in loops forming very tight interactions, essentially collapsing the loops on themselves in a visually interesting way. We find this unexpected development fascinating and worthy of future investigation.



**Figure 3:** *Swarm of shirt/cap agents following: (a) individual agents leaving traces, (b) agents forming chains, (c&d) examples of non-linear behavior of interacting agents.*

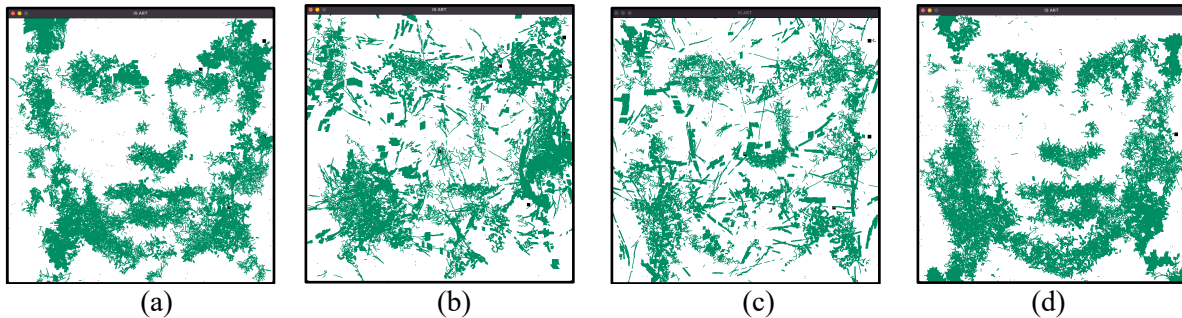
The observed behavior in Figure 3(a) is stable and predictable: agents moving in straight lines and bouncing off the edge boundary. In Figure 3(c), we see the same predictability, but at some point in the simulation, the following agents formed a loop and it collapses, disrupting the stable behavior. In Figure 3(d), the agents formed loops that collapsed almost immediately, showing none of the predictable behavior. It is interesting to note the well-formed circle in the lower right corner of 3(d).

### Swarm-Based Trace Renditions of Target Images

Swarm-based trace renditions differ from agent-based versions by allowing interactions between agents. Initially, agents wander the pixelated image as before, but without reading pixels or leaving trails. The agents only search for other agents to follow. When an agent finds a suitable agent (within their radius of perception) to follow, they form a link that will remain until the simulation ends (generally 60 seconds). This collective then starts leaving a trail. The act of having an agent following, converts the agent in front to a leader-agent. Leader agents read pixel values, and initiate leaving trails. Additional agents can join the chain, increasing its length. The leader only looks for pixels whose darkness is proportional to the length of the chain. In the early parts of the simulation, short chains form and lighter pixels can trigger trails. But as the simulation goes on, only darker pixels initiate trails. Chains of length 2 only reacts to pixels with a threshold of 140 (out of 255). Chains of length 3,4,5 and 6, only perceive pixels with thresholds of 110, 90, 70, and 20 respectively. This guides the simulation to emphasizing the darker regions as the length of the simulation increases. A summary of these rules is provided in the right column of Table 1.

We look at two characteristics of the simulation that could influence resulting image renditions: the radius of perception of the agents, and the distance at which one agent follows another. Radius of perception ( $r$ ) allows agents to “see” other agents to consider following them. Our expectation was that with larger radii, chains would form faster and the resulting images would have fewer details as the darker regions

would garner more attention. However, as can be seen in Figure 4(ad) when we increased the radius by an order of magnitude we saw little difference in the resulting rendition. We found this to be true throughout our experiments. The follow distance ( $d$ ) significantly influences the result rendition as can be seen in Figures 4(abc). In this sequence of renditions, while the radius remains fixed, the follow distance  $d$ , increases from 10 to 200, showing more scattered and loosely cohesive images.



**Figure 4:** Swarm-based rendering of the image altering the Perception Radius: $r$  and the Follow Distance: $d$  (a) radius=10, distance=10 (b)  $r=10$ ,  $d=100$ , (c)  $r=10$ ,  $d=200$ , (d)  $r=200$ ,  $d=10$

The loop collapsing that we found fascinating in generating abstract images, proved problematic in generating image renditions, so once an agent becomes a leader, it no longer searches for another agent to follow, making it impossible for the chains to form loops. All the images in Figure 4 are generated by chains and not loops. Swarm-based rendering is a process involving agents that specifically does *not* follow the inverse relationship between agents and time. Indeed, a single agent using swarm rules cannot generate a rendition at all, and chains of specific lengths are very unlikely to form without a minimum number of agents in the swarm. Sometimes a chain would get stuck in a dark region. To counter this issue, we had the leader agent alternate between being attracted to dark regions and being repelled by the lighter regions. This allowed the chains to get unstuck and potentially find other dark regions.

### Summary and Future Work

We have demonstrated that swarm-based techniques can produce different results in image renditions compared with simple agent-based approaches due to the agent interactions. There are many further avenues of exploration for this approach, including additive pixel values based on the frequency at which a pixel is visited by an agent, using a newsprint greyscale approach to generating image renditions, and revisiting the loop collapsing actions to see if it can be harnessed with a different approach.

### Acknowledgements

We are grateful to Bob Bosch for his inspiration that launched this effort, as well as his insights and suggestions along the way and, his sharing of base images used to drive these swarms. We also thank the reviewers whose comments and questions helped us find a problem in our code and make this a better paper.

### References

- [1] R. Bosch, *Opt Art: From Mathematical Optimization to Visual Design*. Princeton University Press, 2019.
- [2] G. Greenfield and Penousal Machado. “Swarm Art.” *Leonardo*, Volume 47 , Issue 1, 2014, doi:10.1162/LEON\_a\_00695
- [3] D. Palmer, M. Kirschenbaum, et. al. “Using a Collection of Humans as an Execution Testbed for Swarm Algorithms.” *Proceedings of the 2003 IEEE Swarm Intelligence Symposium. SIS '03 Indianapolis, IN, USA , 2003*, pp58-64, doi:10.1109/SIS.2003.1202248