Design and Exploration of Braiding Swarms in VR

Daniel Wagner
University of Würzburg

Christian Hofmann
FH Würzburg-Schweinfurt

Heiko Hamann
University of Lübeck

Sebastian von Mammen
University of Würzburg

ABSTRACT

Swarm-based braiding of structures represents a novel research direction in the domain of building architecture. The idea is that autonomous agents, for instance robots that unroll threads or plants that grow, are programmed or influenced to braid. It is an aspect of biohybrid systems where organisms and robots join forces. In order to harness this idea, we have developed a swarm-based model that allows architects to explore the resulting design spaces in virtual reality. In this paper, we present (1) the model of our swarm-based simulation that aims at growing braided structures, (2) the design elements to guide the otherwise self-organising virtual agents, and (3) the user interface that allows the user to configure, place and grow the swarms of braiding agents. We also present results of a first user study with students and faculty from architecture, in which we tried to capture the usability of our first prototype based on a survey and an analysis of the built results.

CCS CONCEPTS

• **Computing methodologies** → **Virtual reality**; *Model development and analysis*; • **Applied computing** → *Computer-aided design*;

KEYWORDS

virtual reality;braiding;agent-based modeling;architecture

ACM Reference format:
Daniel Wagner, Christian Hofmann, Heiko Hamann, and Sebastian von Mammen. 2017. Design and Exploration of Braiding Swarms in VR. In *Proceedings of VRST '17: 23rd ACM Symposium on Virtual Reality Software and Technology, Gothenburg, Sweden, November 8–10, 2017 (VRST '17),* 4 pages.
https://doi.org/10.1145/3139131.3139169

1 INTRODUCTION

Having large numbers of powerful robots available at very low costs opens up new dimensions of interacting with and shaping our environment. In the EU-funded research project Flora Robotica, robots engage with biological organisms to adaptively create and maintain three-dimensional structures in new ways [Hamann et al. 2015; Heinrich et al. 2016]. As part of these efforts, architects envisioned a novel means to create architecture by braiding or weaving structures. Braided structures are lightweight yet robust. Accordingly, they have been having tremendous impact on our economy and economic toolchains–from producing baskets over carpets and straw mats to weaving fabric into textiles [Adanur 2000; Branscomb et al. 2013]. While Flora Robotica researchers have been looking for technological concepts to implement biohybrid systems, we have taken an interest in thinking ahead, asking ourselves how an according biohybrid system could be explored, and how an appropriate simulation for planning and an interface for control and inspection could look like. In order to overcome the limitations of industrial weaving or 3D printers and to harness the potential of biohybrid systems, where robots and plants mingle to create architecture over possibly longer periods of time, we decided to explore braiding structures in a self-organized fashion. First, we created a self-organized, agent-based braiding algorithm. We tested its functionality in a virtual agent world. Next, in order to effectively explore the creative potential of our algorithm and model, we implemented a virtual reality (VR) interface to make our simulation accessible to external testers.

Section 2 outlines preceding and otherwise related works that influenced our concept. Section 3 describes the control algorithm for the virtual braiding agents the user is empowered to deploy in VR, the respective user interface as well as our approach to evaluating our first prototype. Next, in Section 4 and following, we present and discuss our results and motivate future research directions.

2 RELATED WORK

The motivation of the presented research originated from novel approaches that explore the possibilities of plant-robot biohybrid systems to grow novel architectural designs [Hamann et al. 2015]. As part of this endeavor, biohybrid systems have yielded real structures harnessing the growth and movement of plants [Prescott et al. 2014]. Previously, an augmented reality (AR) prototype was presented by [von Mammen et al. 2016], which allowed the user to place and configure robots that in turn guide the growth of plants. It relied on a stereo-vision enhanced, wired HMD, gaze-based selection and gamepad control. The key point of that publication was that (a) the simulation of biohybrid systems is possible and (b) that effective AR interfaces can be designed. However, there were numerous shortcomings of this AR setup, including the wired hardware, the complexity of the interaction interface and the challenges of context-dependent AR rendering. Due to these challenges and due to our goal of exploring a novel approach to biohybrid design, we decided to focus on a VR implementation instead. In order to realize our concept, we built on existing computational models of growth and development that are capable of realtime computation.

VRST '17, November 8–10, 2017, Gothenburg, Sweden

ACM ISBN 978-1-4503-5548-3/17/11…$15.00
https://doi.org/10.1145/3139131.3139169

This implied that (a) their simulation needs to run at interactive speeds (at 90Hz for VR) and that (b) the growth was continuously traceable. Accordingly, we followed the broadly adopted L-system approach to simulating growth [Prusinkiewicz and Hanan 1989]. In particular, we implemented its swarm grammar (SG) extension [Jacob and von Mammen 2007], where the tip of the growing structure is realised as a sentient agent that proliferates and the remainder of the plant-like structure remains fixed.

With respect to the user interface, we relied on teleportation in VR supported by a Bézier curve-projection to the destination which has also been seen in many other simulations or games such as "The Lab" [Steam 2017]. It offers the possibility to reach places of greater altitude by increasing the jaw of the controller. We implemented a controller menu for selecting various options which was inspired by games such as "Crysis" [Crysis 2017]. Here, the menu items are aligned around a reference point and can be accessed by moving the controller. Skimming through the different options by swiping across the controller's touch pad can be found as an example of the VRTK asset package [Unity 2017]. In section 3.2 we detail this approach.

3 MODEL

In this section, we detail (1) the model of the braiding agents, (2) the user interface to configure and guide them, and (3) the design of the tutorials that introduce the user to the application.

3.1 The Self-Organized Braiding Model

Considering the high degrees of autonomy upheld by robots and especially plants, we set out to develop a novel approach to braiding: Instead of a centralized instance that controls the crossings of individual threads, the threads themselves need to concert their movements. We deploy virtual swarm grammar agents (see Section 2) that move about, leave trails and react to their local environments. In order to arrive at an effective behavioral model, we first implemented the usual, centralized approach and let the agents retrace a strict, pre-determined trajectory (Figure 1). This approach, of course, is neither self-organized nor adaptive. As a consequence, every detail has to be fully specified, including for instance the starting locations of the agents. In contrast, a self-organized model works on local knowledge perceived by each agent individually and assumes that each agent acts autonomously. This freedom not only allows the agents to adapt to the local conditions of the built environment but also to coordinate with a possibly fluctuating crowd of interaction partners. The field of view of an agent is determined by two spheres centered around the agent: If another agent collides with the wider sphere, it is approached. If the other agent also collides with the smaller sphere, both agents switch into braiding mode: They average their flight direction and circulate around their geometric centers. If there is no peer to follow or to braid with, the agent changes its direction of movement after a small random number of steps. As seen in Figure 1(b), the agents avoid obstacles on their paths. They detect obstacles by means of scanning for intersections with three rays cast in flight direction and consequently change their acceleration, i.e. obstacle to the left, accelerate to the right, etc. These sensors are also used to ensure that agents are kept within so-called braiding volumes (BVs), as seen in Figure 1(c). BVs are visualized as framed blocks to display clear boundaries and allow the user to observe the agents inside. In general, all the agents act at each simulation step as soon as they have been placed inside the virtual environment. Pausing the simulation as a whole also pauses all the enclosed agents. The agents' step routine first checks against collisions with BVs, then with other obstacles, and finally runs the movement routine as outlined above.

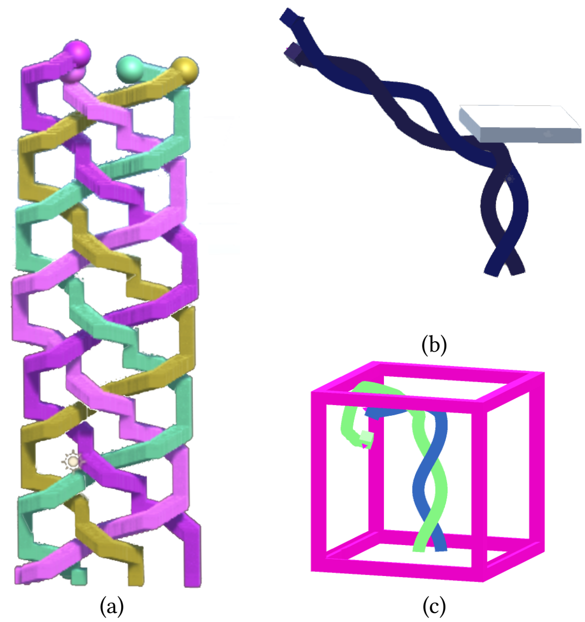

Figure 1: Screenshots of (a) the centralized approach and (b) two self-organising, deflecting agents

3.2 VR Interface

Working with the HTC Vive hardware, we provide the user with two fully tracked 3D controllers with trigger buttons (for the index finger), touchpad (thumb), grip buttons (palm and middle finger). We distinguish between the two controllers to support different kinds of interactions. We project an arrow symbol on one controller used for teleportation and a plant symbol on the other one used for seeding braiding agents, placing and configuring obstacles or BVs. Navigation is realized by Bezier-curve teleportation (see Section 2), with the touchpad pressed for aiming and teleport when released. In addition, the user can take a screenshot by pressing a grip button on the side of the navigation controller. As the plant controller offers numerous options of selecting and configuring agents and objects, we provide a menu (Figure 2). It displays different options as cubic items floating around the controller as soon as the user presses the touchpad. Swipe motions across the touchpad are translated to the movement of the items: The center-top item can be selected pressing the trigger button, the other items are transparent. The menu is designed in a hierarchical fashion such that submenus may unfold when further specifications are required. Unfolding and closing a (sub-)menu is animated to maintain the hierarchical context. The generic implementation of the menu offers fast and modular extensibility for new types of agents, obstacles or other functionalities.

(a) (b)

Figure 2: Screenshots of the (a) menu and (b) edit mode selection panel

In our first prototype, at the top-level menu, the user can choose between agent types, obstacle types and BV types. Obstacles and BVs can be configured before being placed. To this end, an additional panel will be displayed beside the plant controller which allows to toggle the editing mode, e.g. full scaling or movement along a specific axis (Figure **??**). Switching into one of these editing modes (up/down on the touchpad), the user can increase/decrease the respective variable (left/right on the touchpad). In this way, we provide the basic transformations: Scaling, rotating, and translating. The latter has proved itself in iterative test sessions to improve placement precision, especially when working with large numbers or objects. Once the object is configured, it can be placed via the trigger button. Additional options that we offer as part of the menu include pausing/playing/exiting the simulation, adjusting the simulation speed and deletion of objects.

3.3 Interaction Scenario Examples

A first tutorial explains how to move, to interact with the menu and to place agents. A second tutorial introduces to the configuration and placement of obstacles and how they influence the movement of the agents. A third tutorial illustrates the effects of BVs and some advanced features like simulation control and taking screenshots. Each tutorial guides the user by means of simple instructional texts, that disappear when the user presses the grip button, or when he successfully follows the instructions. To ease the learning process, we highlight the controller buttons that need to be pressed. We also offer three competitive levels to test and improve one's skills. They ask the user to place agents in reach specific target zones or to guide them around obstacles. Successful achievements within short periods of time yield greater scores.

4 EXPERIMENTS

In order to evaluate our prototype, we conducted a study with 16 persons between 20 to 35 years of age, with mixed backgrounds, from pedagogy over engineering to architecture, and with varying levels of prior VR exposure. The controls of the simulation were learned either by stepping through the tutorials - if time permitted - or by short oral introductions. Next, the testers were encouraged to spend time in levels or an open-world, sandbox scenario. We asked the users 14 questions with answers on a 5-point Likert scale [Allen and Seaman 2007] and conducted 5-minute interviews. We asked about the users' backgrounds, the effectiveness of the (respective) introduction, the specific scenarios they had played, particular experiences they encountered, the usability of the user interface and about impressions concerning the simulation models, including the behaviors of the agents.

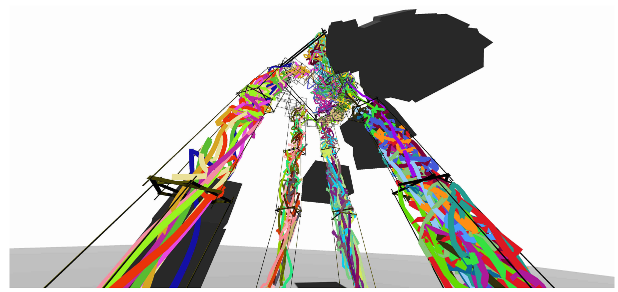

Figure 3: An elevated structures built using obstacles as steps

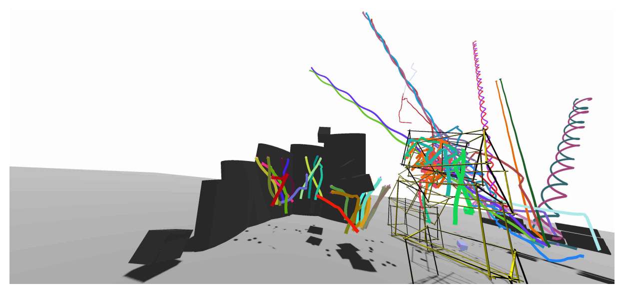

Figure 4: Combination of obstacles and BVs to influence the agents

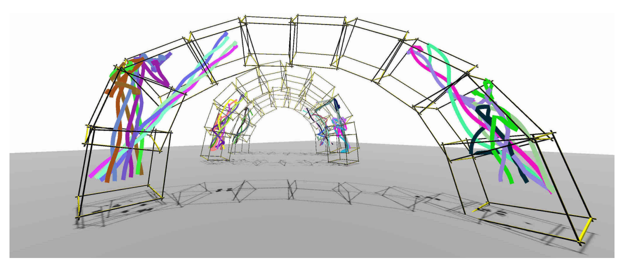

Figure 5: Building archways by combining multiple BVs

Most participants (12 out of 16) had very little or no prior VR experience. Oral instructions were rated more effective than the tutorials. The main reason mentioned was the lack of contextual information. The user interface achieved high acceptance and good

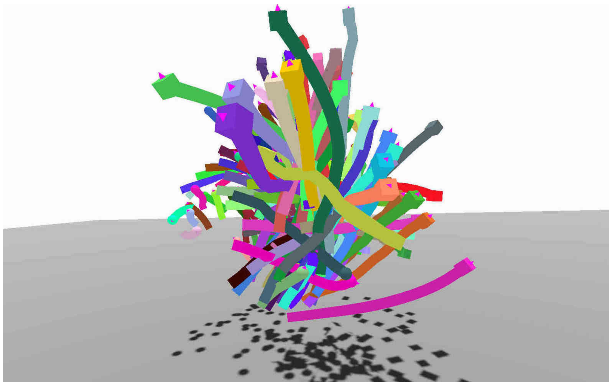

Figure 6: Agents start braiding from the same origin

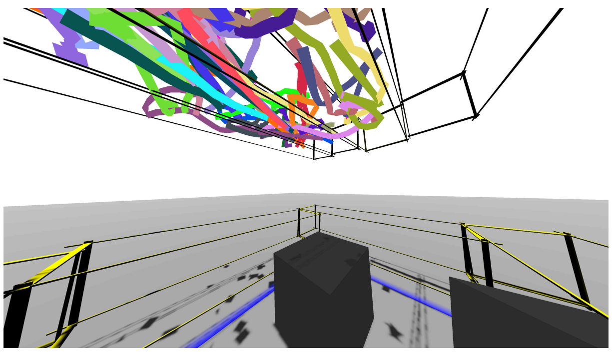

Figure 7: BVs for a roof like structure

ratings despite little VR expertise. Capturing the context, understanding and effectively working with the user interface are crucial, yet, even more important is whether the users could tap into the creative design potential offered by the simulation model. While many of the participants were surprised by the motions of the agents, the emergence of structures and the influence of obstacles and BVs towards the agents were well understood. We assume the element of surprise arose from the default random movement of the agents and, possibly, from those rare events that agents lost the connection to diverted peers. Four architecture students who participated in the study had prior VR experience. In their eyes, the user interface scored a bit lower. They also missed functionality they would have needed for crafting larger-scale designs. Also, they tried to place large numbers of objects and agents quickly, and to use structures as big as possible. Figures 3 to 7 show some of the innovative designs that resulted during these experiments. In this rather professional usage scenario, the interface revealed its greatest shortcoming, namely its efficiency. We believe that (1) increasing the speed of the unfolding/closing animations of the menus, (2) honing the sensitivity of the touchpad for skimming through items, and (3) a better layout with frequently used items at the highest menu level, could amend this issue. The architects' construction efforts challenged the simulation's computational capacity, with frame rates dropping proportionally to the number of built objects and, especially, seeded agents.

In general, the interviews of the architects revealed considerable insights not only considering challenges of the current prototype but also considering future steps. For instance, they expressed the wish to combine the simulation with the tools they are familiar with, like Autodesk FormIt 360 [Autodesk, Inc. 2017]. The architects also tried to integrate the braiding agents into their designs as much as possible and in numerous different ways. As a result, they experienced more situations in which our implementation did not fulfill their expectations, like when the agents moved unexpectedly or did not create the desired structures.

5 FUTURE WORK

Based on the feedback, especially from testers with professional architecture backgrounds, we have identified the following directions to be of great importance: (1) The efficiency of the simulation needs to improve. This can be achieved by using better hardware or generic acceleration algorithms. Alternatively, we could improve on the algorithmic performance of the braiding agents by reducing the full calculation of the field of view to a bare minimum, e.g. reduce the number of frames in which the agent perceives at all or only activate it when moving randomly. (2) In order to use our simulation for planning biohybrid systems and to inform architectural design studies, one could offer high-level APIs and SDKs to integrate the simulation in other contexts. Alternatively, one could extract the generated paths, configurations, and geometries from the simulation and make them accessible to other systems, e.g. using COLLADA. (3) Regarding the user interface, the menu should be revised to work faster, especially for fast, professional use cases. Some users suggested that sound effects for placing objects or agents would further improve the user experience.

REFERENCES

S. Adanur. 2000. *Handbook of Weaving*. CRC Press.

E. Allen and C. Seaman. 2007. *Likert Scales and Data Analyses*. Quality Progress. http://asq.org/quality-progress/2007/07/statistics/likert-scales-and-data-analyses.html

Autodesk, Inc. 2017. FormIt. https://formit.autodesk.com/. (July 2017).

D. Branscomb, D. Beale, and R. Broughton. 2013. New directions in braiding. *J. Eng. Fibers Fabr* 8, 2 (2013), 11–24.

Crysis. 2017. Website. http://www.crysis.com/de. (July 2017).

H. Hamann, M. Wahby, T. Schmickl, P. Zahadat, D. Hofstadler, K. Stoy, S. Risi, A. Faina, F. Veenstra, S. Kernbach, et al. 2015. Flora robotica-mixed societies of symbiotic robot-plant bio-hybrids. In *Computational Intelligence, 2015 IEEE Symposium Series on*. IEEE, 1102–1109.

M. K. Heinrich, M. Wahby, M. D. Soorati, D. N. Hofstadler, P. Zahadat, P. Ayres, K. Stoy, and H. Hamann. 2016. Self-Organized Construction with Continuous Building Material: Higher Flexibility Based on Braided Structures. In *2016 IEEE 1st International Workshops on Foundations and Applications of Self* Systems (FAS*W)*. 154–159. https://doi.org/10.1109/FAS-W.2016.43

C. Jacob and S. von Mammen. 2007. Swarm grammars: growing dynamic structures in 3D agent spaces. *Digital Creativity, Vol. 1(18)* 1 (2007).

T. J. Prescott, N. Lepora, and P. F. M. J. Vershure. 2014. A future of living machines? International trends and prospects in biomimetic and biohybrid systems. *Proc. SPIE* 9055 (mar 2014), 905502–905502–12. https://doi.org/10.1117/12.2046305

P. Prusinkiewicz and J. Hanan. 1989. Lindenmayer Systems, Fractals, and Plants. *ser. Lecture Notes in Biomathematics. New York: Springer* 79 (1989).

Steam. 2017. The Lab on Steam. http://store.steampowered.com/app/450390/The_Lab. (July 2017).

Unity. 2017. VRTK - Virtual Reality Toolkit. https://www.assetstore.unity3d.com/en/#!/content/64131. (July 2017).

S. von Mammen, H. Hamann, and M. Heider. 2016. Robot gardens: an augmented reality prototype for plant-robot biohybrid systems. In *Proceedings of the 22nd ACM Conference on Virtual Reality Software and Technology*. ACM, 139–142.